

© 2021 Kuocheng Wang

FEA-BASED SIMULATION OF BREAST DEFORMATION IN REAL-TIME USING
ARTIFICIAL NEURAL NETWORK

BY

KUOCHENG WANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Systems and Entrepreneurial Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor Thenkurussi Kesavadas, Chair and Director of Research
Professor Ramavarapu S Sreenivas
Professor Brad Sutton
Professor Arif Masud

ABSTRACT

Treatment of breast cancer involves two stages: diagnosis and treatment. It is difficult to correlate the imaging results at the two stages because as the patient's posture changes during treatment, the images captured during diagnosis do not represent the tumor location during the treatment. In the absence of real-time imaging during treatment, the visualization of tumor location is challenging for surgeons.

There are many challenges for breast deformation simulation. For example, material properties are very important to simulate the deformation accurately. The simulation speed will decide whether the technology is applicable for clinical use. But because of the limit of hardware, achieving real time simulation is difficult.

This thesis focuses on investigating visualization of breast deformation for different patient's positions. We utilized magnetic resonance imaging (MRI) of a patient collected during diagnosis for this study. This data was preprocessed to form a 3D reconstructed model that was used to run a finite element analysis (FEA) simulation. FEA simulates the deformation of breast tissues for different constraints, such as glandular ratio and gravity angle. However, FEA simulation of such deformation can take a few minutes to as much as 40 minutes to complete using a 8 cores computer. To obtain real-time visualization, we constructed a neural network (NN) model that takes breast gravity angle and glandular / fat ratio (breast material) as input to estimate breast deformation for different patient's positions offline. This NN is used to predict the deformation of the breast and provide visualization in real-

time (5 ms prediction time).

To further validate our result, we carried out MRI of a breast phantom in several angles (to mimic various patient postures). We also implemented an iterative technique to estimate material properties. This data was used to simulate breast deformations at different posture angles. A similar approach was implemented to build an NN model. Our results show that NN has the ability to map the gravity direction to the breast shape and tumor location accurately, while, keeping run time to a minimum.

To my family, friends and teachers

ACKNOWLEDGMENTS

I would like to express my sincerest gratitude to my advisor Dr. Thenkurussi Kesavadas for his invaluable advice, continuous support, and patience during my PhD study. His broad and deep knowledge in different aspects of the research taught me how to be a good researcher. I would also like to thank Dr. Arif Masud, Dr. Ramavarapu S Sreenivas and Dr. Brad Sutton for their suggestions and feedbacks on my research, as well as serving in my committee. Additionally, I feel fortunate to have wonderful colleagues at Health Care Engineering Systems Center - Naveen, Shrey and Shanker for a cherished time spent together. My appreciation also goes out to my family and friends for their encouragement and support all through my studies.

TABLE OF CONTENTS

LIST OF FIGURES	viii
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 LITERATURE REVIEWS	4
2.1 Breast deformation simulation	4
2.2 Breast material property	6
2.3 Real-time deformation simulation	7
CHAPTER 3 SIMULATION OF BREAST DEFORMATION DUE TO GRAV- ITY	10
3.1 Simulation procedure	10
3.2 FEA mesh generation	12
3.3 Material assignment in breast tissue	15
3.4 Introduction to FEA formulation	18
3.4.1 Abaqus “Steps” definition	20
3.4.2 Simulation setup and boundary condition	20
3.5 Neural network	23
3.5.1 Introduction to convolution layer	23
3.5.2 Downsample input with stride	23
3.5.3 Downsample input with pooling	24
3.5.4 Drop out	25
3.6 Neural network formulation	25
3.7 Breast deformation simulator	27
3.8 Neural network error	30
3.8.1 Breast surface displacement	30
3.8.2 Large and small deformation and their error	33
3.8.3 Tumor location prediction	39
3.9 Discussion	41
3.9.1 Neural network	41
3.9.2 Displacement error	42
3.9.3 Breast deformation	43
3.9.4 Tumor location	43

CHAPTER 4	VALIDATION OF REAL-TIME BREAST DEFORMATION	45
4.1	Validation procedure	45
4.2	Data preparation	49
4.2.1	MRI scans collection	49
4.2.2	MRI images preprocessing	51
4.3	Deformation computation	52
4.3.1	Simulation setup and boundary conditions	52
4.3.2	Material estimation	53
4.4	Neural network for breast deformation and tumor location	56
4.5	Results obtained from neural network	57
4.5.1	Neural Network deformation prediction	59
4.5.2	Nodal displacement error	63
4.5.3	Tumor location	64
4.6	Discussion	65
CHAPTER 5	AR DEPLOYMENT	67
5.1	Image tracking	68
5.2	Hardware and software architecture	70
5.3	User interface	71
CHAPTER 6	CONCLUSION AND FUTURE WORK	74
REFERENCES	76

LIST OF FIGURES

1.1	Patient’s postures (Figure from [13])	2
2.1	Breast anatomy (Figure from [65])	7
3.1	Breast quadrant (Figure from [62]).	11
3.2	Overall process from raw breast data to real-time breast simulation model.	11
3.3	Different parts of breast data. (a) Raw data with noise. (b) Data after processing that deleted the noise.	13
3.4	Glandular tetrahedral mesh generated after step1(a) in Fig. 3.2. The green color is the surface of the gland, the purple parts are the tetrahedrons.	15
3.5	Tetrahedral mesh for different parts. Left (glandular tissue), middle (fat), right (tumor).	17
3.6	Material assignment of glandular tissue (left), fat (middle) and tumor (right). Each part is tetrahedralized individually, and compared with the overall mesh.	18
3.7	Boundary condition defined in Abaqus.	21
3.8	Data set generated from Abaqus (80% for training and 20% for validation).	22
3.9	Stride [3].	24
3.10	Simulated breast deformation. Left top: User input panel. Left bottom: Patient’s orientation view panel. Right: Deformation view panel.	28
3.11	Vertex file.	28
3.12	Face file.	29
3.13	Element file.	29
3.14	Transverse plane deformation (mm) at different angles.	32
3.15	Sagittal plane deformation (mm) at different angles.	33
3.16	Large deformation.	35
3.17	Small deformation.	36
3.18	Large deformation. Displacement histogram.	37
3.19	Small deformation. Displacement histogram.	37
3.20	Large deformation. NDE error histogram.	38
3.21	Small deformation. NDE error histogram.	38
3.22	Transverse plane.	40
3.23	Sagittal plane.	40
4.1	Overall process, step 1.	46

4.2	Overall process, step 2.	47
4.3	Overall process, step 3.	47
4.4	Overall process, step 4.	48
4.5	Overall process, step 5.	48
4.6	MRI scan experiment set up.	50
4.7	Tumor.	50
4.8	Breast tetrahedral mesh.	52
4.9	Abaqus boundary condition. The left preserves the shape of the back plane. The right does not preserve the shape.	53
4.10	Material estimation error plot between FEA and NN at 25°. The error is calculated as the closest PTP distance and is calculated between closest points of two breasts without including the back plane (shown in white).	55
4.11	MRI image.	56
4.12	Training, testing and validation set.	57
4.13	Overlap the simulated breast with the ground truth. The solid breast is the ground truth taken from the MRI images at 25° in a left lateral position, the transparent is the breast at upload position (left picture) and after deformation using NN (right picture).	60
4.14	Similar to Fig. 4.13 except the right picture shows the shape after deformation at 50°.	60
4.15	Transverse plane deformation (mm) at different angles.	61
4.16	Sagittal plane deformation (mm) at different angles.	62
4.17	NN error plot at different angles. The error is the PTP distance between NN and FEA simulation.	62
4.18	Error distribution plot (mm) between FEA and NN in the validation set.	63
4.19	Tumor center location at different angles along sagittal plane.	64
4.20	Tumor center location at different angles along transverse plane.	65
5.1	AR app for street view [2].	68
5.2	Tracking breast with a 3D object.	69
5.3	Breast placement in Unity.	70
5.4	AR architecture implementation.	71
5.5	AR app interface.	72
5.6	Tumor in AR.	72
5.7	A user is using the AR app.	73

CHAPTER 1

INTRODUCTION

Breast cancer is the most common cancer among women. The World Health Organization estimated an age-standardized incident rate of 84.9 per 100,000 in 2018 [1]. Breast cancer diagnosis is a crucial step in early-stage detection and treatment. The gold standard for diagnosis is to perform medical imaging, such as a mammogram, followed by a biopsy of suspicious tissue [15]. Invasive treatments for removal of tumor include lumpectomy and mastectomy, which require additional imaging with ultrasound (US), CT, and/or MRI for preoperative, operative, and postoperative visualizations [16]. Conventionally, surgeons use medical imaging to locate and remove suspicious tissues prior to the surgery [17]. They may also draw anatomical landmarks on the patient's breast or use different markers along with medical images (e.g., US/CT, based on availability of facilities) [18]. However, the patient's position inevitably differs significantly between the preoperative and operative stages, and causes a mismatch between the patient on the operating table and the images captured during the preoperative stage. Hence, it is crucial to develop a way to improve breast tissue localization accuracy. Fig. 1.1 shows some standard anatomical postures.

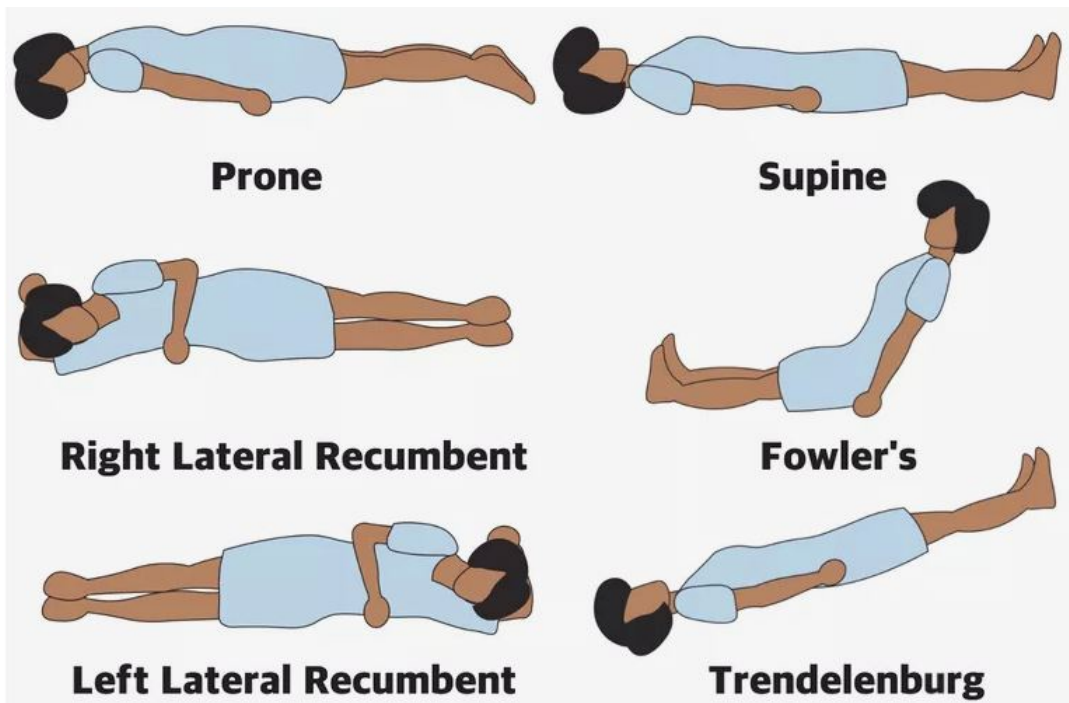


Figure 1.1: Patient's postures (Figure from [13])

Breast cancer diagnosis begins with manual palpation and is usually followed by a mammogram. If a suspicious lump is identified, ultrasound scanning or a biopsy will be performed to extract any suspicious masses [6]. In many developing nations, because of the lack of mammogram facilities, direct biopsy is performed (with ultrasound support when available). As discussed earlier, even when advanced mammogram facilities are available, it is hard for a doctor to understand the image because the patient may lay on the operating room's table in a different position. The 2D image localization is often left to the skill of the person performing the biopsy.

The primary challenge with breast imaging is that breast anatomy is irregular, and with a noisy dataset, it is challenging to reconstruct a 3D model to generate a good mesh model for FEA. Palomar et al. [34] found that even the structure of the breast is complicated; the volume ratio between glandular and fat is the most important factor in breast deformation. Llorens et al. [45] were the first to include skin depth in simulating breast deformation. They

found that including the real depth of skin made simulations much more accurate. Han et al. [68] developed a patient-specific, “open-source graphics processing unit-based, explicit, dynamic, nonlinear finite element (FE) solver.” To determine the breast property, they proposed a model parameter optimization method. Sturgeon et al. [7] created a breast phantom from a real patient’s model and simulated it with FEBio, an FEA based software that can include compression and gravity. For other specific patients, they refined the mesh so that they could map the current model to other patients’. Martínez et al. [8] used the Simulation Open Framework Architecture (SOFA) as their solver for breast deformation simulation. They then used a tree-based approach to train the model and predict the deformation of a breast during plate compression. They claimed the prediction time was less than 0.2 s. For a comprehensive review of deformation simulation in recent surgical simulators, please refer to [9].

The dissertation is organized as follows:

In chapter 2, we introduce the related literature. In chapter 3, we built a surgical simulator to simulate breast deformation in real-time. Details are described in this chapter. In chapter 4, we validated the accuracy of our simulation by collecting data from a breast phantom. In chapter 5, we constructed an augmented reality (AR) simulation in a tablet. Finally, we conclude our research and future work in chapter 6.

CHAPTER 2

LITERATURE REVIEWS

Breast imaging is an effective tool to help doctors locate a tumor. However, the images taken at a preoperative stage will differ from those seen in the operating room. For an MRI scan, a patient can stay in a prone position, but in a biopsy, she is in a supine position. In an ultrasound (US) procedure, the patient lies on a bed, which may be in a variety of possible positions. Currently, a doctor has to transfer the images between the two stages in his/her head. An incorrect diagnosis may occur if the transformation is not understood correctly. Hence, breast deformation simulation is an important topic in biomedical imaging.

2.1 Breast deformation simulation

There are many computation methods in the area of tissue deformation simulation, but only two are usually used in breast tissue; spring damper (SD) and FEA. SD is considered an aesthetic approach for deformation computation. The method uses surface points, and sometimes internal points, associated with masses, velocities, and accelerations to solve a system of dynamic equations. The advantage is that it does not involve any large matrix computation, which makes it fast during real-time simulation. The method does not involve solid mechanics; hence, it is considered an approximation of the actual deformation. To get good simulations, many researchers have worked to incorporate realistic material methods used in FEA into the SD method. Cai et al. [29] recorded the breast motion data when a woman was running on a tread mill. They used a Kelvin-Volgt model with two SD to describe the motion of the breast. The Kelvin-Volgt model is a viscoelastic model which has the advan-

tage of taking time into account. Patete et al. [30] implemented a mass-spring model on a breast tetrahedral mesh. They iteratively minimize the error between the resulting spring force and gravity to find the best spring constant. They achieved a point-to-point (PTP) error of 1.224 ± 2.202 mm, which is quite promising. Roose et al. [31] used an optimization algorithm to solve the dynamic equation that describes breast shape after a breast implantation. They found that the presence of inhomogeneous material or homogeneous material cause very little difference in the accuracy of the simulation (below 2.5 mm). Xu et al. [32] improved the accuracy of spring mass system by adding nonlinearity to the spring to take nonlinear viscoelastic effect of the tissue. Overall, the researchers are trying to improve the SD accuracy by exploring more accurate material modeling methods.

FEA discretizes the deformation domain to solve the continuum mechanics equation. Many researchers separate FEA into linear FEA and non-linear FEA. Generally speaking, linear FEA is a special case of nonlinear FEA; it ignores the nonlinear part, and hence makes simulation faster. When the deformation is small and there is no contact, the result of linear FEA is accurate, but as deformation becomes larger, the result becomes less accurate. The advantage of linear FEA is its simulation speed; hence, it is usually used for surgical training in a virtual reality environment. Most research effort in this area has been finding ways to update the matrix quickly by reserving the shape of the stiffness matrix [26]. Azar et al. [33] simulated breast deformation under plate compression by separating deformation into small steps and simulating each step with small strain theory (i.e., linear FEA). In breast deformation simulation, researchers have usually chosen to use nonlinear FEA because breast deformation is usually large. Del et al. [34] simulated breast deformation in different positions. They combined gland and fat into one material with the help of their volume ratio. They found the simplified model give a good approximation of a real breast's shape. Rajagopal et al. [35] proposed a semi automatic approach to generating a finite element mesh. They formulated a total Lagrangian formulation to solve the equilibrium equation under

gravity. Rajagopal et al. [36] explored the effects of mesh resolution, boundary conditions, and FE solvers on the accuracy of breast deformation simulation. They concluded that a good boundary condition played an important role in the accuracy of simulation. Chuang et al. [37] used a selected subset of segmented breast data to construct an initial FEA mesh. They later smoothed the mesh with a cubic-Hermite basis function. The mesh was then used to simulate breast compression. Their result not only showed good accuracy on the surface of the breast, but also showed that the internal landmarks have a much larger error. Tagliabue et al. [38] simulated probe-breast interaction deformation by using 3 methods: penalty forces, constraint force, and prescribing of the nodes on contact faces. They found that the third method performed the best. Mira et al. [39] included pectoral fascia and suspensory breast ligaments in their simulation against gravity. They obtained a better result when the sliding forces between the pectoral muscle and breast were considered. Inspired by the above research, we chose to use nonlinear FEA for our simulation.

2.2 Breast material property

To use FEA, tissue properties need to be defined. As the material property directly affects the mechanical behavior of a breast, accurate properties are desired. Breast material is considered inhomogeneous, nonlinear, and time-dependent. There are 4 types of tissues as shown in 2.1 in a breast (skin, gland, fat, and sometimes tumor), and the material within any single tissue type is not necessarily homogeneous. There are two types of testing, *in vivo* and *ex vivo*. *In vivo* testing uses an imaging technique, for example, magnetic resonance elastography and sono-elastography [40, 41]. The testing is accurate only for small deformations. Hence, *ex vivo* testing is usually performed to find the material properties. There are many material constitutive models. Among them, hyperelastic material is commonly used [36, 42]. Hyperelastic material is a nonlinear material, but it does not take time into consideration [43]. Some work has started shifting to use of viscoelastic material [44] because

of its ability to consider time. One thing to notice is that material properties reported by different papers vary significantly, for example, by 100 times [34, 45]. Hence, it is challenging to find accurate parameter values. As the material experiment set-up requires much effort, many FEA-based breast simulations are using materials from previous researchers' reports [25, 46]. However, it is well-known that breasts' properties vary a lot between people. A simpler approach suggested by Visentin et al. [43] and Chung et al. [47] is to use an optimization method to find a material property that fits the deformation. Even that method does not test material explicitly, it does not require extra set-up. For a comprehensive review of FEA and material constitution model research on breast deformation and material models, we direct interested readers to [48].

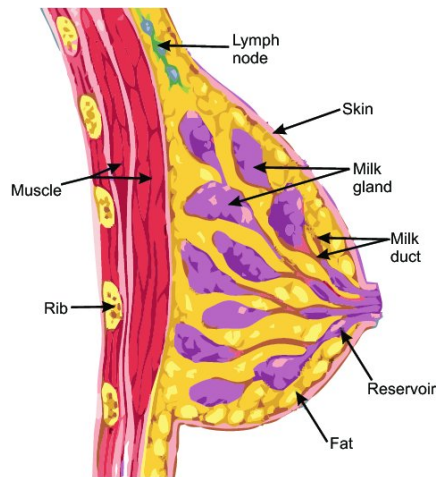


Figure 2.1: Breast anatomy (Figure from [65])

2.3 Real-time deformation simulation

As machine learning (ML) becomes popular recently, a data-driven approach has become popular. The main idea is to combine ML and FEA because the prediction time of ML models is usually fast, while FEA produces accurate results but is computationally expensive. The combination of FEA and ML will produce real-time accurate deformation simulation. Özgören et al. [50] constructed an NN to predict the power and torque values obtained from

a beta-type Stirling engine. Erkan et al. [51] used different learning algorithms to train an NN to predict damage factor of Glass fibre reinforced plastic in the cutting process. Eser et al. [52] improved the mathematical models of surface roughness parameter of AA6061 alloy using an NN and response surface methodology. Several researcher used an NN for bone structure, for example, predict tissue adaptation loads when the input is bone's density distribution [53], predict the stiffness of bones [54] and predicts the displacement of bones when different loads are applied [55]. Taylor et al. [56] implemented an NN for neck strains and fracture loads prediction. Zolfagharnasab et al. [62] used random forests, gradient boosting regression, and multi-output regression to predict the breast shape after a breast-conserving surgery. Random forest performed best in their study. The above research shows that NN can accurately estimate the FEA result.

In terms of real-time simulation, many researchers found good accuracy as well as simulation speed. Several papers have described prediction times between a few milliseconds to 50 ms [57–61]. Deo. et al. [57] simulated a scenario in which the surgical tool touched a stomach. They generated 26 force directions at different touching points, and used a radial basis function to learn the surface point displacement. Their group further developed a simulator that included haptic feedback, which required 1000 Hz [60]. Jahya et al. [58] simulated a transrectal MR-guided biopsy procedure on a prostate phantom. The shape of the phantom was extracted from a set of MR images. They used a needle angle, an insertion depth, and other features to predict the nodal displacements of the phantom. Mendizabal et al. [59] simulated a US probe interaction with a breast. The authors generated FEA scenarios by selecting a point on the surface of the breast, founding all points in a bounding box around that point, and adding displacements to the points. That obviated any need for physical interaction between the breast and US probe during the simulation. After collecting data from different points, they constructed a U-Net to learn the deformation and tumor location. The authors did not consider gravity because as long as the breast stays in the same

position, the geometry already takes gravity into account. Morooka et al. [61] constructed a neural network (NN) for liver deformation prediction when a surgical instrument touches the liver. Their NN input is the forces, and output is the nodal positions of the liver. However, that approach is not physically accurate because rather than giving a Dirichlet boundary condition, the author used a Neumann boundary condition by estimating the force based on how far the surgical tool moves inside the tumor. Lorente et al. [63] used ML to model liver deformation during breathing. They compared linear regression, a decision tree, a random forest, and extremely randomized trees. The best method obtained an average nodal point Euclidean error between FEA and ML of 0.07 mm. Their prediction time was 2.89 s, which was slower than the expected real-time simulation in liver deformation simulation, which was about 0.3 s. Widmer et al. [64] simulated breast palpation with a phantom model. Although they did not use a real breast, their study provided a good indication of what features could be useful during a deformation prediction. Tonutti et al. [46] used an artificial neural network (ANN) and supported vector regression for tumor location prediction in a patient-specific deformation model for brain pathologies. The authors claimed the prediction was instant, but they did not mention the exact prediction time. Motivated by the above research, we decided to combine FEA and NN for real-time breast deformation simulation.

CHAPTER 3

SIMULATION OF BREAST DEFORMATION DUE TO GRAVITY

In this chapter, we will focus on breast simulation with gravity. The breast data was from a real breast data set and we tried to simulate it with FEA and then use an NN to learn the FEA result. We will introduce FEA and NN, and then show how we incorporate these two to simulate the breast deformation.

3.1 Simulation procedure

To develop a simulator for visualizing breast deformation under different patient's positions, it is essential to understand the structure of breast anatomy. The breast anatomy consists of skin, fibrous tissue, fat, ducts, glandular tissue, and sometimes tumors. The breast can be separated into 4 quadrants: the upper-outer quadrant (UOQ), upper-inner quadrant (UIQ), lower-outer quadrant (LOQ) and lower-inner quadrant (LIQ), shown in Fig. 3.1. The patient's tumor can be anywhere in the four quadrants. Medical images provide a density map of the anatomy. Based on the density, the breast tissue is categorized into 4 groups: fatty tissue, scattered glandular tissue, heterogeneous dense breast tissue, and extremely dense tissue [75]. During medical image analysis, those 4 categories are used to interpret a patient's scan to identify suspicious tissues.

The workflow of our methodology is shown in Fig. 3.2. Our methodology consists of 3 steps, data preparation, deformation computation, and simulation. Data processing is performed to improve the 3D segmentation of the breast model through smoothing, clustering,

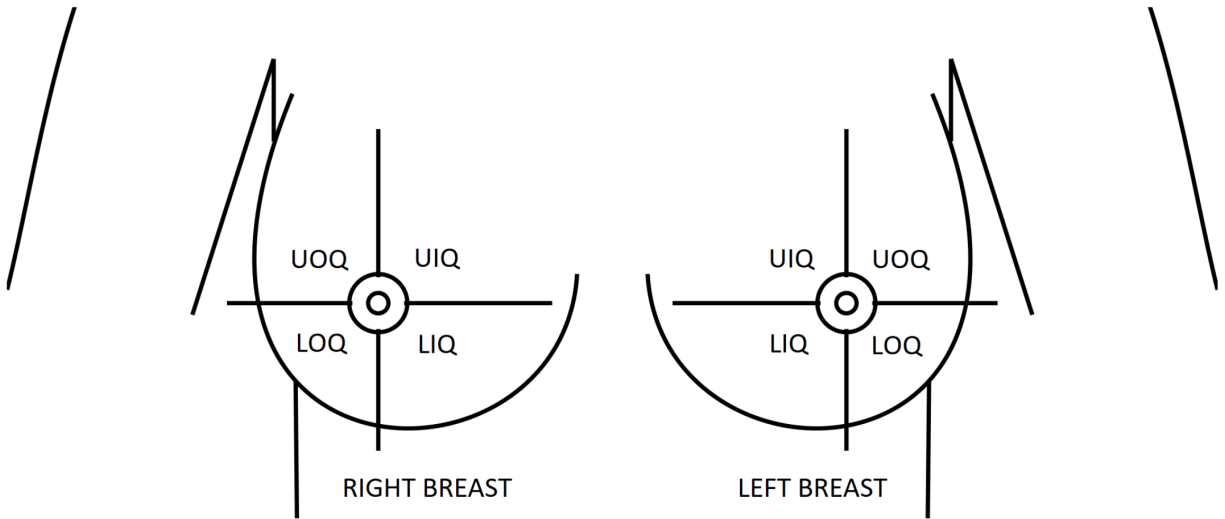


Figure 3.1: Breast quadrant (Figure from [62]).

and self-intersection detection. The processed data are tetrahedralized for FEA using Tetgen [12]. Different regions of the mesh are assigned different material properties based on the corresponding structure of breast tissues. For the deformation computation, the dataset is imported into an FEA solver, Abaqus [5], to compute deformation in different positions. Useful features are extracted from the deformation results to train the NN model. Finally, the trained NN is used for deformation prediction in our simulator.

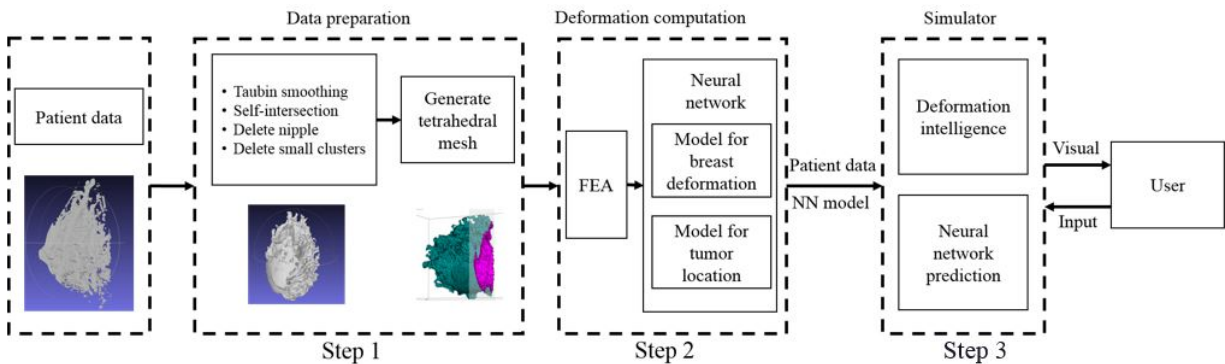


Figure 3.2: Overall process from raw breast data to real-time breast simulation model.

3.2 FEA mesh generation

The segmented breast data we used were from [66]. The authors used 8 breast models of different breast categories; each model contains both left and right breasts. The models contain skin (with a thickness of 2.34 mm), fat, glandular tissues, and tumor. The segmented data from the repository are in a 3D stereolithography (STL) format. The data contain the positions of the point cloud and connections between the points (faces). They need to be processed before tetrahedralization in order to ensure that the generated tetrahedral mesh is good. For example, sample data from our original surface mesh are shown in Fig 3.3(a), in which the gland has many noise clusters. For denoising, we implemented a modified form of breadth first search (BFS) (as shown in Algorithm 1) to get rid of the clusters. The input was the faces from the STL file, and the output was the faces without noise. The idea is that BFS is able to find connected components, and the gland without noise is the largest component. In addition, to remove the jagged edge and reduce the number of points, the data were smoothed with MeshLabTM. Further, we used the Z-painting function to manually smoothen specific areas. Because of the limitations of the tetrahedralization process, the nipple was eliminated from the model. We later imported the smoothed surface mesh into MeshMixerTM to check and repair it. The processed surface mesh is shown in Fig. 3.3(b).

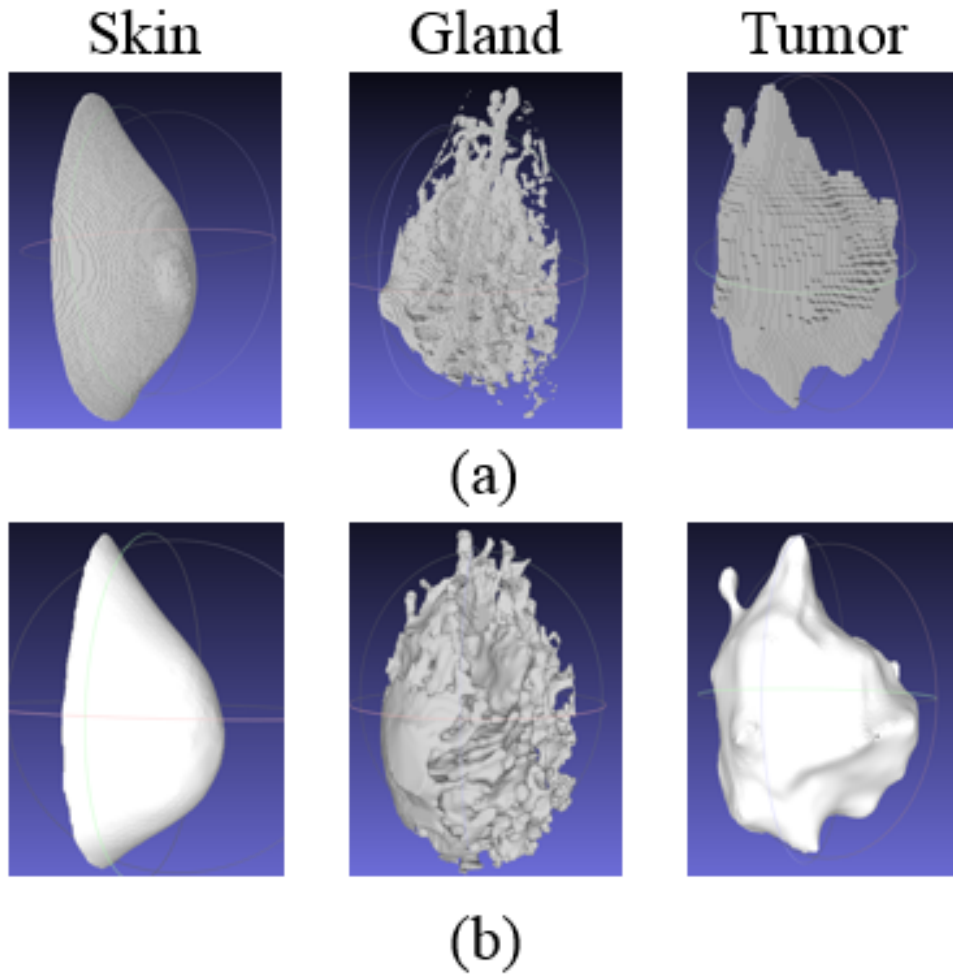


Figure 3.3: Different parts of breast data. (a) Raw data with noise. (b) Data after processing that deleted the noise.

We generated a tetrahedral mesh with Tetgen. The linear tetrahedral element type (C3D4) is preferred for its fast and relatively accurate FEA simulation [67]. Table 3.1 shows details on the generated output of our model. The factors that affect FEA simulation results are the number of tetrahedrons and the quality of the generated tetrahedral mesh. Given a good mesh quality, the total number of tetrahedrons is an indicator of the accuracy of an FEA simulation. In general, a more refined mesh yields more accurate solutions. In our simulation, we used a total of 167,853 tetrahedrons (discussed in detail in Section 2.2). To test whether the chosen mesh density in our FEA simulation was good, we generated

Algorithm 1 Deletion of noise.

Input : faces

Output : part faces

$E \leftarrow$ a list of edges

$S \leftarrow$ a list of list, contains vertex indices of different clusters, initialized as empty

$F \leftarrow$ unique vertex indices

function $MBFS(edges, face) \leftarrow$ Modifies BFS so it returns the connected components of a point as a list

for f in F **do**

if f is not visited **then**

$S \leftarrow S + MBFS(E, f)$

end if

end for

return largest list in S .

Table 3.1: Tetrahedral mesh generated for each tissue type.

	Tissue type	Tetrahedrons	Points
1	Skin	167,853	40,012
2	Fat	57,210	12,040
3	Gland	631,457	170,563
4	Tumor	100,959	23,544

a refined tetrahedral mesh with 317,256 tetrahedrons. The same point was picked in the two simulation results, and the displacement difference was found to be less than 0.1 mm. Hence, our FEA simulation performed well. Fig. 3.4 shows a sample of a tetrahedralized mesh generated using our approach.

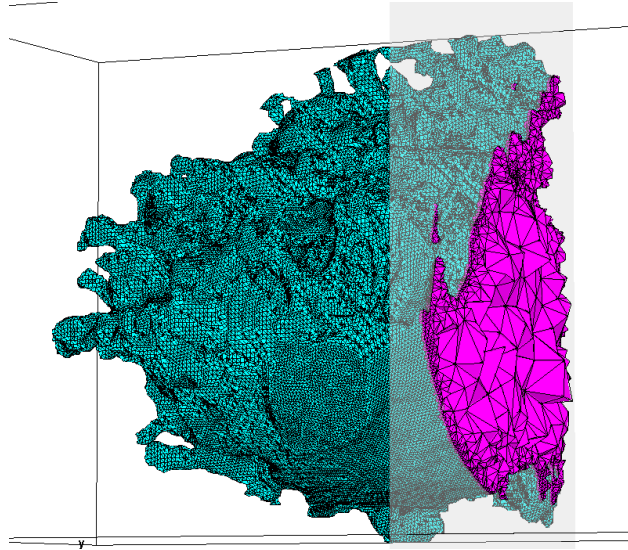


Figure 3.4: Glandular tetrahedral mesh generated after step1(a) in Fig. 3.2. The green color is the surface of the gland, the purple parts are the tetrahedrons.

Here is some more information about MeshLab and MeshMixer. They are open source mesh processing platforms. Many functions can be used to preprocess mesh. We used a Taubin filter and a z-painting in MeshLab to smooth the breast surface. The “Analysis” function is very convenient in MeshMixer. The function checks the defects in the mesh, such as nonmanifold surface and holes. We can use MeshMixer to fix the mesh problems. Tetgen is a mesh generator and it is widely used as the backend for other mesh softwares such as Gmsh. Tetgen generates high quality mesh in a short amount of time. We noticed that even the mesh passes MeshMixer’s quality check, it might still have ”face intersection errors” in Tetgen. In this case, we deleted the faces reported by Tetgen, and regenerate the mesh in MeshMixer. Tetgen was then able to use the surface mesh to generate the tetrahedral mesh.

3.3 Material assignment in breast tissue

Breast tissue is generally recognized as nonlinear, inhomogeneous, and incompressible. As breast tissue is complicated, the mechanical properties of tissue identified by various researchers are different and can vary by 100 times in magnitude ([45] and [34]). For our

research, we selected different properties of the breast across different parts. Homogeneous material is assumed inside each part of the breast. The details are shown in Table 3.2. Skin is defined by the polynomial model, where the strain energy density function is defined by

$$W = \sum_{c_{i,j}=0}^n c_{ij}(I_1 - 3)^i(I_2 - 3)^j. \quad (3.1)$$

Fat and glandular tissue are defined by the Mooney-Rivlin model, where I_1 and I_2 are left Cauchy-Green deformation tensor. The coefficient c_{ij} is determined by the curve fitting between stress-strain curves and the derivative of Eqn. 3.1. The strain energy density function for the Mooney-Rivlin model is

$$W = c_1(\bar{I}_1 - 3) + c_2(\bar{I}_2 - 3). \quad (3.2)$$

Based on [46], we assumed that tumors are twice as dense as glandular tissue. Following [34], we assigned average values of mechanical properties to the materials for glandular tissue and fat. For example, if the glandular volume is 30% of the entire glandular and fat volume,

$$c_1 = 0.3 * 120 + 0.7 * 80 = 96 \text{ PA}. \quad (3.3)$$

Table 3.2: Material property of each tissue type.

Tissue type	Material Type	Density (kg/m^3)	Coefficients (PA)
Skin	Polynomial model	1020 [72]	$c_{10} = 31, c_{01} = 30, c_{11} = 22.5, c_{20} = 50, c_{02} = 60$ [34]
Fat	Mooney-Rivlin	921 [62]	$c_1 = 80, c_2 = 0$ [62]
Gland	Mooney-Rivlin	948.5 [62]	$c_1 = 120, c_2 = 0$ [62]
Tumor	Isotropic heterogeneous	1897	$E=0.7193 * 10^6, \nu= 0.4531$ (dimensionless) [68]

Different parts have different material properties. There are 2 methods for assigning properties to the mesh: (1) one can generate the overall mesh and identify the corresponding part from the generated mesh, or (2) one can generate tetrahedral mesh for each part separately. In the first method, to find the tetrahedrons that belongs to each part, each part is first

tetrahedralized individually, and then the positions are compared between the individual parts and the overall mesh. In the second approach, the parts of the mesh are generated individually and then merged together to form the overall mesh. Even though the geometry of the part in the second approach is well-preserved, the reassembling of the different parts is challenging, as the contact surfaces might not match exactly after tetrahedralization. That problem is completely absent in the first approach, since it is working from one set of mesh. Although the first method does not preserve the geometry of the parts as well as the second method does, we nevertheless chose it for its ease of implementation and reassembly. In the first approach, the mesh of each part is used to find the corresponding tetrahedron in the overall mesh. To speed up the material assignment process, we built a data structure to separate the skin mesh into a 10 x 10 grid. That data structure was then used when we were iterating through the tetrahedral mesh of a part to identify the corresponding tetrahedron in the overall mesh. Fig. 3.5 shows the tetrahedralized mesh found from the breast tetrahedrons with our method. Fig. 3.6 shows the results of the material assignment in Abaqus.

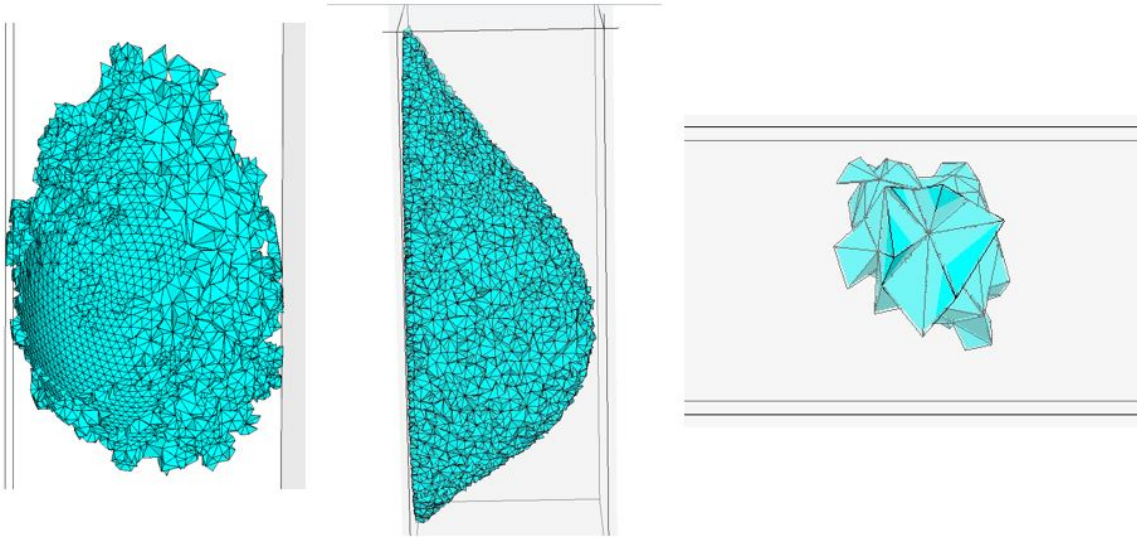


Figure 3.5: Tetrahedral mesh for different parts. Left (glandular tissue), middle (fat), right (tumor).

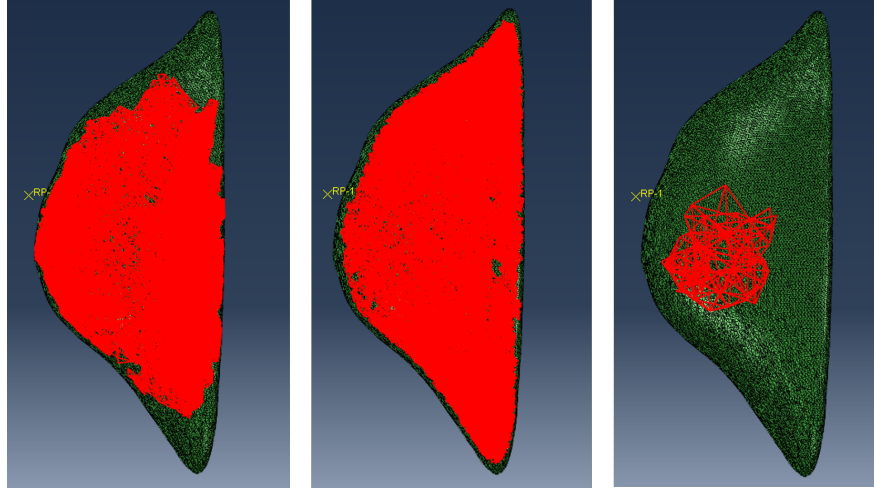


Figure 3.6: Material assignment of glandular tissue (left), fat (middle) and tumor (right). Each part is tetrahedralized individually, and compared with the overall mesh.

3.4 Introduction to FEA formulation

FEA is a numerical way to solve continuum mechanics equations, and different solvers are available. We chose to use Abaqus because it is widely adopted in industry and research society. Abaqus provides user access to define routines, so it's very simple for a user to program and interact with it.

The general form of the nonlinear dynamics equation in FE is

$$M\ddot{u} + N(\dot{u}, u) = F, \quad (3.4)$$

where $N(\dot{u}, u) = K(u)u + C(\dot{u})\dot{u}$. u is the displacement vector, \dot{u} is the velocity vector, and \ddot{u} is the acceleration vector. M is the mass matrix, and N is the nonlinear functions that contain both stiffness and damping. $C(\dot{u})$ is a damping matrix. $K(u)$ is the stiffness matrix, and it has two nonlinear parts: geometric nonlinearity, and material nonlinearity. The geometry nonlinearity comes from the transformation between the material coordinate and the spatial coordinate. The material nonlinearity comes from the stress-strain relationship,

which can be derived from a strain energy density function. In our simulation, we simulated the breast deformation under gravity, so we used a static nonlinear FEA, which meant that the inertia effect was not taken into account. The damping term did not appear in the original FEA derivation because the energy was assumed to be conserved. We added the damping term because it was known that energy dissipates during deformation. However, accurate damping requires very careful testing. In medical simulation, damping is commonly neglected. F is the external force. In order to solve the nonlinear FEA, Newton-Raphson's method is the most fundamental way,

$$K \Delta u^i = F - N(u_{n+1}^i) \quad (3.5)$$

$$\Delta u_{n+1} = u_n + \Delta u \quad (3.6)$$

$$F = \sum_{j=0}^n \Delta F_j, \quad (3.7)$$

where i is the iteration number, and n is the time step. Compared to linear FEA, nonlinear FEA requires many more iterations be solved for each time step, so can be quite time-consuming if the problem is highly nonlinear. Apart from Newton-Raphson's method, other algorithms have also been developed. Abaqus solves the FEA with the Implicit or Explicit method. The advantages of the Implicit method are that it is faster than the Explicit and is unconditionally stable. However, its ability to solve a highly nonlinear problem is limited. The Explicit FEA solver, on the other hand, is stable only when the time increment is small, and hence takes many more steps than Implicit FEM. However, it can solve very highly nonlinear problems. To save computation time, we chose the Implicit solver. To help better understand how to establish the simulation in Abaqus, we will introduce different parameters in details.

3.4.1 Abaqus “Steps” definition

Steps are used in Abaqus to define their analysis. Many options are available such as heat transfer, dynamic explicit/implicit and mass diffusion. Abaqus is capable of solving multiple step problems. In our case, one step is enough, which is “static general”. Dynamics is another option. However, we consider the process to be slow, a static step is appropriate for the simulation. The first step is a default step in Abaqus, and the program does not allow a user to change it. The simulation time was set at 1 s. This value can be adjusted. If the time is too short, the simulation might not converge and result in an error. The simulation precision was set to “double precision” to prevent high bulk modulus. Locking problem is a concern in FEA. The problems will result in a less deformation with the same amount of forces. One way to help solve the problem is use more refined elements. C3D4H is a refined version of C3D4 element, hence it is chosen. As breast deformation is large it involves nonlinear FEA. Abaqus does not enable nonlinear FEA in static step, unless the user set it. We enabled the Nlgeom setting for that purpose.

In nonlinear simulation, Abaqus will generate a lot of results for each step and they are all stored in the result folder. A user is able to visualize the result in Abaqus directly. Several plots such as deformed shape, contour plots for stress or deformation are available to visualize. There are also animation plots available that show the deformation history. The times step is evenly distributed among each time frame.

3.4.2 Simulation setup and boundary condition

We loaded the models into Abaqus for analysis. The patient’s model was influenced by two factors: gravity direction and the degree of firmness it is attached to the chest bone. The patient’s position with respect to gravity, known as the *upload position* influences the breast during a scan. To compensate for the deformation due to the initial gravity, opposing forces

were applied. The breast's back was considered as a pectoral muscle. To simulate the firm attachment of a pectoral muscle to the chest wall, we created constraints by setting the displacement and rotation of points to 0. Fig. 3.7 shows a sample simulation setup on Abaqus, with the arrow indicating the direction of the applied gravity for the unloading condition.

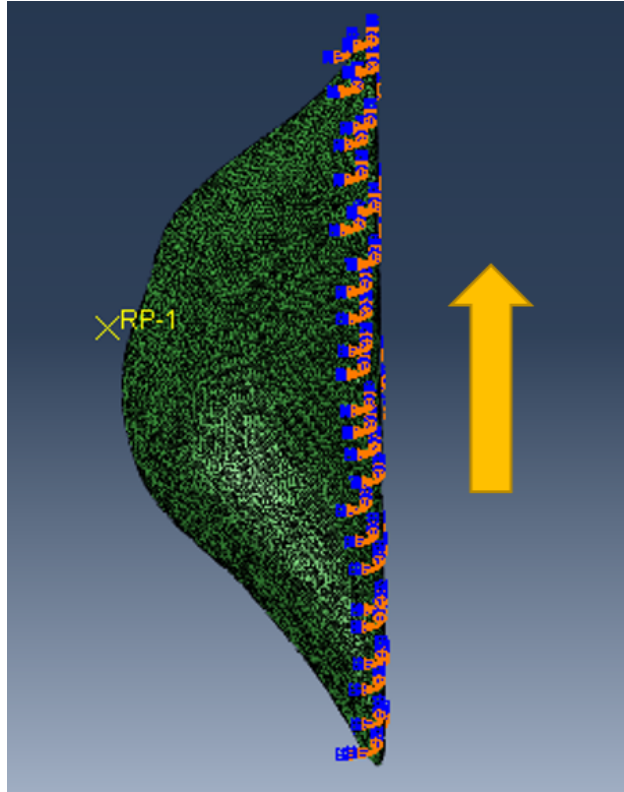


Figure 3.7: Boundary condition defined in Abaqus.

Table 3.3 shows FEA parameters. We chose an element type of C3D4H as a refined mesh. We wrote scripts to automatically generate the scenarios by altering the gravity angle and the glandular/fat ratio. We collected the angle from 0° to 360° with an interval of 5° . The glandular ratio was collected as the ratio of glandular volume over the total volume (including fat and gland), starting from 10% and going to 90%, in 10% increments, as shown in Fig. 3.8. We purchased 8 cores from our school, and the simulation time for each scenario ranged from 2 minutes to 40 minutes.

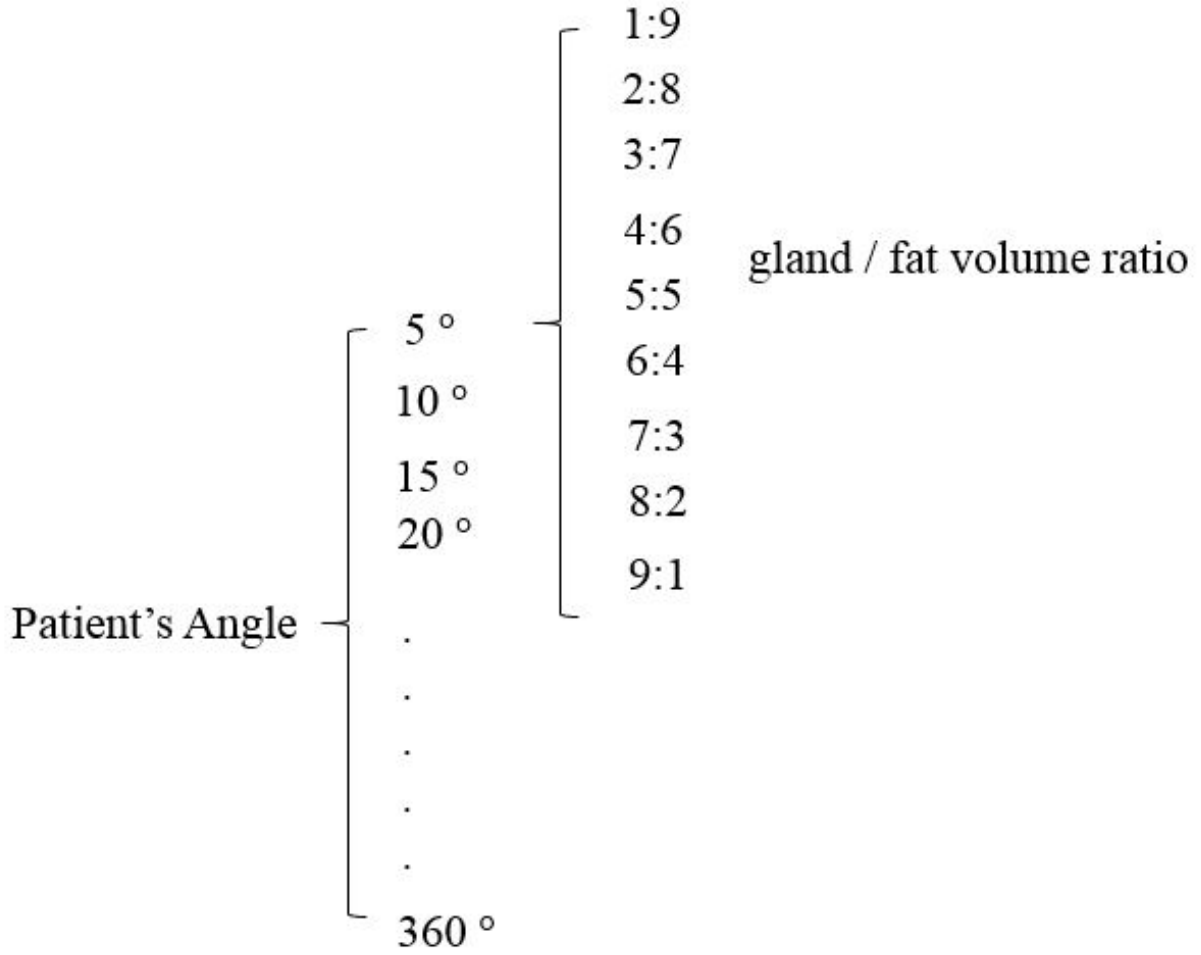


Figure 3.8: Data set generated from Abaqus (80% for training and 20% for validation).

Table 3.3: Abaqus FEA Setup Parameters

FEA parameter	Values
Simulation time	1 s
Simulation precision	Double precision
Element type	C3D4H
Step type	Static general
Solver	Implicit solver

3.5 Neural network

The disadvantage of FEA is its slow processing speed, as it cannot meet real-time needs. We hence improved the deformation prediction by using an ML-based approach. The input is a vector of R^4 , the gravity direction and the glandular volume ratio. The gravity direction is converted to a unit vector. The goal is to predict the tumor displacement and the breast surface nodal displacements. NN has different units. We will introduce some of them in the following sections.

3.5.1 Introduction to convolution layer

A problem associated with NN is overfitting. When the result performs well in the training set but poorly in the validation set, an overfitting problem occurs. A convolution layer is used to reduce the redundancy in the nonlinearity of NN. Suppose $X \in R^{n*n}$, $K \in R^{K_y} * R^{K_x}$. Then

$$(X * K)_{i,j} = \sum_{a=0}^{K_y-1} \sum_{b=0}^{K_x-1} K_{a,b} X_{i+a,j+b}. \quad (3.8)$$

K is acting as a filter in the the above equation. a and b are the dimension of the filter in x and y direction. The filter is applied by using a window function that slides through the input. Some elements in the K matrix are set as 1 and others as 0, and can remove some redundant values in the matrix.

3.5.2 Downsample input with stride

Stride is used to control how far to shift the convolution at a time. Apply a stride s to a convolution layer,

$$H_{i,j} = \sigma \left(\sum_{a=0}^{K_y-1} \sum_{b=0}^{K_x-1} K_{a,b} X_{is+a,js+b} \right). \quad (3.9)$$

Fig. 3.9 shows how stride and convolution work together. s is 1 in this case. In the left figure, red is the filter window before moving with a stride, and green is the one after stride. The dimension of H becomes $(\lfloor (d_y^{l-1} - k_y^l) / s \rfloor + 1) * (\lfloor (d_x^{l-1} - k_x^l) / s \rfloor + 1)$. As s increases, the output shrinks very fast.

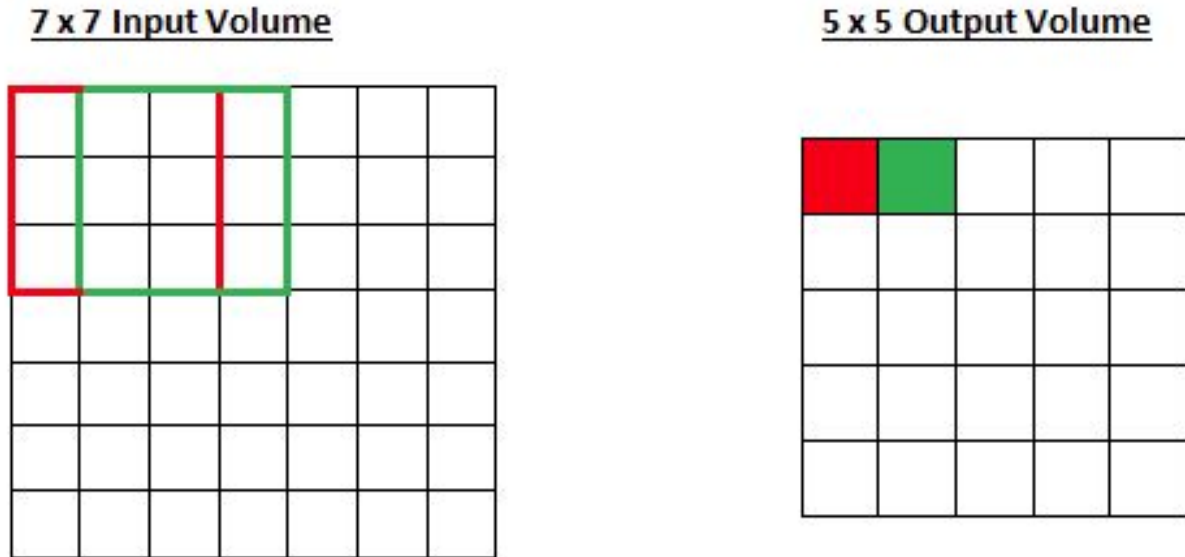


Figure 3.9: Stride [3].

3.5.3 Downsample input with pooling

Pooling is a form of down sampling. It adds some invariance to local translation. Two types of pooling are the most often used: average pooling shown in Eq. 3.10 and max pooling shown Eq. 3.11. In average pooling, the matrix within the sliding window is down sampled to an average value, and max pooling is down sampled to a maximum value. Pooling is also a type of filter, it reduces the input dimension and makes the computation less expensive.

$$v_{i,j} = 1/h^2 \sum_{a=0}^{h-1} \sum_{b=0}^{h-1} H_{is+a,js+b}. \quad (3.10)$$

$$v_{i,j} = \max_{0 \leq a, b \leq h} H_{is+a, js+b}. \quad (3.11)$$

3.5.4 Drop out

Another way to help prevent overfitting problems is drop out. Drop out removes a random set of hidden units (set them as 0). The probability of a unit to be dropped varies from different papers. As it is random dropping, it is only used in training time. There is no scientific proof that drop out can help, nevertheless, it is helpful in practice.

3.6 Neural network formulation

There are two types of problems in NN: classification and regression. If the problem is classification, a softmax function is often used in the last layer because it can map the input to a value between 0 and 1; that value is used to represent the probability that the input belongs to one category. Some examples in classification is image recognition, handwriting detection and fraud detection.

NNs are capable of learning features of functions and use a highly nonlinear function to estimate a good mapping between input X and output $f(X; \Theta)$. An NN forms an optimization problem and learns parameters to minimize the error between the estimated value and the true value. Below is the formulation for a fully connected NN.

$$Z^l = \Theta^{l-1} H^{l-1}, l = 2, \dots, L \quad (3.12)$$

$$H^l = \sigma(Z^l), l = 2, \dots, L \quad (3.13)$$

$$f(X; \Theta) = \Theta^L H^L \quad (3.14)$$

$H^1 = X$ is the input vector to a neural network, where X is $[x_0 \ x_1 \ x_2 \ x_3 \ \dots \ x_{d_l}]$, x_0 is

a bias term, l is the l th layer number, Z^l is the neuron value at the l th layer, and Θ^l is the weighting parameters. H^l is the output at the l th layer. We use d_l to denote the number of hidden units. $X \in R^{d_1+1}$, $H^l \in R^{d_l+1}$, $Z^l \in R^{d_l+1}$ (after adding a bias term), and $\Theta^l \in d_{l+1} \times (d_l + 1)$. σ is a nonlinear activation function. To find the best parameters, Θ , a cost function needs to be defined. The following is a general definition of the cost function:

$$R_{\Theta} = E_{X,Y}[\rho(f(X; \Theta), Y)], \quad (3.15)$$

where $E_{X,Y}$ is the expectation, $\rho(f(X; \Theta))$ is the error function between the NN output and ground truth, and Θ is the parameter to be estimated. $\rho(f(X; \Theta))$ is chosen based on the specific problem. Since our problem is a regression problem, the output is a value or continuous value, without needing an activation function in the last layer. As tumor shape is very irregular, we used the tumor center to represent the tumor, and we predicted only the displacement of the tumor center. Considering that the tumor center is one point, if we put the tumor center displacement and all the surface points' displacements together, the tumor center will be treated as one point, and it will be hard to guarantee the accuracy of this point because the error is the average error of all the points. Hence, two NNs are built in parallel to predict the tumor displacement and breast surface nodal displacements. Take breast surface nodal displacements for example, we use Y to denote the displacements of all points, and $f(X; \Theta)$ to denote predicted displacements. The mean absolute error (MAE) is used in our case:

$$\rho(f(X; \Theta), Y) = 1/(3N) \sum_{i=1}^N |Y_i - f_i(X; \Theta)|, \quad (3.16)$$

where $Y = [u_1^T u_2^T \dots u_N^T]^T$, $f(X; \theta) = [u_1^T u_2^T \dots u_N^T]^T$. $u_i = [u_{i,x} u_{i,y} u_{i,z}]^T$. N is the number of surface points. Tumor displacement has the same error function, with N equal to 1. Now

the optimization problem is formulated as follows:

$$\Theta^* = \arg \min_{\Theta} \rho(X; \Theta) \quad (3.17)$$

To find the best parameter Θ^* , we used back propagation. We chose Adam as the optimizer. The learning rate for both breast surface and tumor location is 0.00001. We created 1,311 training scenarios (3 scenarios failed to generate converged FEA simulation, shown in Fig. 3.8). In the case of breast surface nodal displacements, there are 2 hidden layers, and each layer has 60 neurons. We chose ReLU function as the activation function in the middle layers. The number of neurons in the output layer is the same as the number of output variables. As there are 25,374 surface points, the output is $R^{25374*3}$. The NN structure for tumor position prediction is similar, except it has one hidden layer with 20 neurons.

The data were split into groups of 20% and 80% for validation and training, respectively, and were randomly shuffled at the beginning of each iteration before training. This procedure is usually called as hold-out validation.

3.7 Breast deformation simulator

We built the simulator (Fig. 3.10) using UnityTM with C#. To render the breast model, we need to assign vertices and faces. As we mentioned before, the tetrahedral mesh were generated from Tetgen, which includes a vertex file (Fig. 3.11), a face (Fig. 3.12) file and an element file (Fig. 3.13). The vertex file contains the coordinates of the vertices (second to last column). The face file contains the triangles of the surfaces (second to fourth column). The element file contains the tetrahedrons (second to fifth column). Each number in the face and element file is the index of the vertex. Unity can accept an OBJ file or an FBX file, but can't read Tetgen files directly. The mesh data structure in Unity is defined by an array of vertex vectors and an array of triangles that are described by the vertex indices. We

programmatically assigned the mesh in *Start* function in Unity based on the face file and the vertex file. Hence the mesh is loaded once in the beginning when the simulation starts.

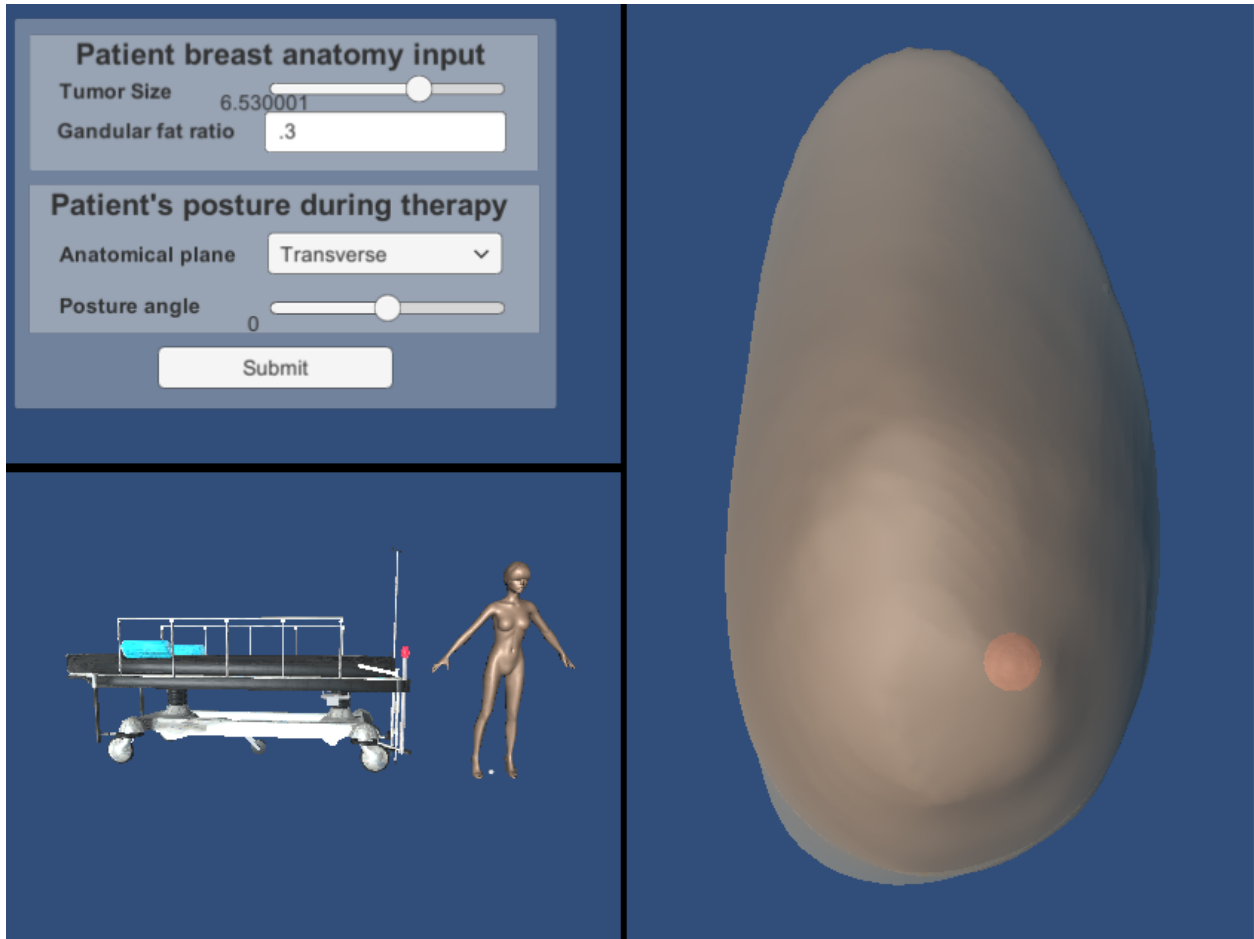


Figure 3.10: Simulated breast deformation. Left top: User input panel. Left bottom: Patient's orientation view panel. Right: Deformation view panel.

```

7  79.662980000000005  14.613390000000001  3.3806639999999999
8  79.671999999999997  14.970980000000001  3.463219
9  79.521270000000001  17.70759  2.8457430000000001
10 79.546449999999993  17.44453  3.2465220000000001
11 79.563320000000004  17.28697  2.7865920000000002
12 79.459890000000001  17.872039999999998  3.183738
13 79.578689999999995  14.52923  2.9740790000000001
14 79.640910000000005  14.957100000000001  3.0805220000000002
15 79.518039999999999  17.68402  2.4995069999999999

```

Figure 3.11: Vertex file.

10	10	11	9	-1
11	9	11	12	-1
12	11	13	12	-1
13	14	12	13	-1
14	14	13	15	-1
15	15	13	16	-1
16	16	17	15	-1
17	17	18	15	-1
18	15	18	19	-1

Figure 3.12: Face file.

2	36274	39963	69481	72096
3	13966	13954	69329	83170
4	56324	54108	54115	67992
5	9710	69355	69613	75812
6	15725	15744	70756	81866
7	72984	11086	74910	78170
8	62159	73917	75247	79688

Figure 3.13: Element file.

The NN models were trained with Python, and socket communication protocols were used to bridge the backend. The simulation software has 3 panels: (1) a user input panel, (2) a patient’s orientation view panel, and (3) a deformation view panel. The user input panel gets input from the user on altering tumor size, glandular / fat ratio, anatomical plane (transverse or sagittal), and position angle. Based on the user settings in this panel, upon submission, the loaded breast model will reconfigure itself to the present user settings. The patient’s orientation view panel will then display the corresponding position for the provided setting. That helps the user plan and visualize patient’s position for treatment. The deformation view panel shows the breast shape and the tumor’s (red ball’s) position corresponding to the patient’s current position. Fig. 3.10 shows a sample scenario. The image shows the patient’s model in a standing position. The simulator currently predicts the deformation with the NN

model in 5 ms. The total run time, including prediction, rendering, and data communication between Python and C#, is in the range of 300 ms to 500 ms. A video was uploaded [77] to show the simulator.

3.8 Neural network error

The neural network training results were as follows: The mean absolute training error for tumor was 0.0152 mm, and the validation error was 0.0168 mm. The breast deformation training error was 0.0043 mm, and the validation error was 0.0048 mm. In our error calculation during training, we averaged the error of x, y and z axes, in order to obtain the average nodal displacement error (NDE), the following inequality relationship can be used:

$$\begin{aligned}
& 1/N \sum_{i=1}^N \sqrt{(u'_{i,x} - u_{i,x})^2 + (u'_{i,y} - u_{i,y})^2 + (u'_{i,z} - u_{i,z})^2} \\
& \leq 1/N \sum_{i=1}^N (|u'_{i,x} - u_{i,x}| + |u'_{i,y} - u_{i,y}| + |u'_{i,z} - u_{i,z}|) \\
& = 3 * \rho(f(X; \Theta), Y).
\end{aligned} \tag{3.18}$$

Hence, for tumor, the training nodal displacement error (NDE) is less than 0.046 mm, the average validation NDE is less than 0.050 mm; for breast, the training NDE is less than 0.013 mm and the validation NDE is less than 0.014 mm. Some other results are discussed in the following subsections.

3.8.1 Breast surface displacement

Figs. 3.14 and 3.15 show color maps of breast displacement magnitude in different positions. The maximum value in the scale corresponds to the maximum displacement of the point. The visualization result was simulated in Unity. The simulation was performed starting from

a lying-down patient's position (as shown in Fig. 3.14(i) and 3.15(i)) and slowly changed along the sagittal or transverse plane. It can be observed that the deformation magnitude increases as the position angle increases (A lying position is shown in Fig. 3.14(i) at 12° , and an upright position is shown in 3.14(v) at 88° .) Usually, the area close to the nipple has larger deformation than the surrounding areas. There are two factors that affect the deformation: glandular ratio and stiffness. Glandular ratio is the ratio of glandular tissue volume over the total volume (which includes fat and gland). As the gland ratio increases from 0.33 to 0.66, the breast becomes heavier, and thus more deformation is expected. However, from the figure, it can be observed that the deformation decreases, which happens because glandular tissue has greater stiffness than fat (Eqn. 3.3). The effect of the glandular/ volume ratio can be seen more clearly in the sagittal plane plot. Another observation we gained from this visualization is that because the breast shape is not symmetrical, the deformation also varies in different quadrants of the breast. For example, in Fig. 3.14 (ratio 0.33), the upper-inner quadrant and lower-inner (left side view) quadrant are darker in color, which indicates that they have more deformation.

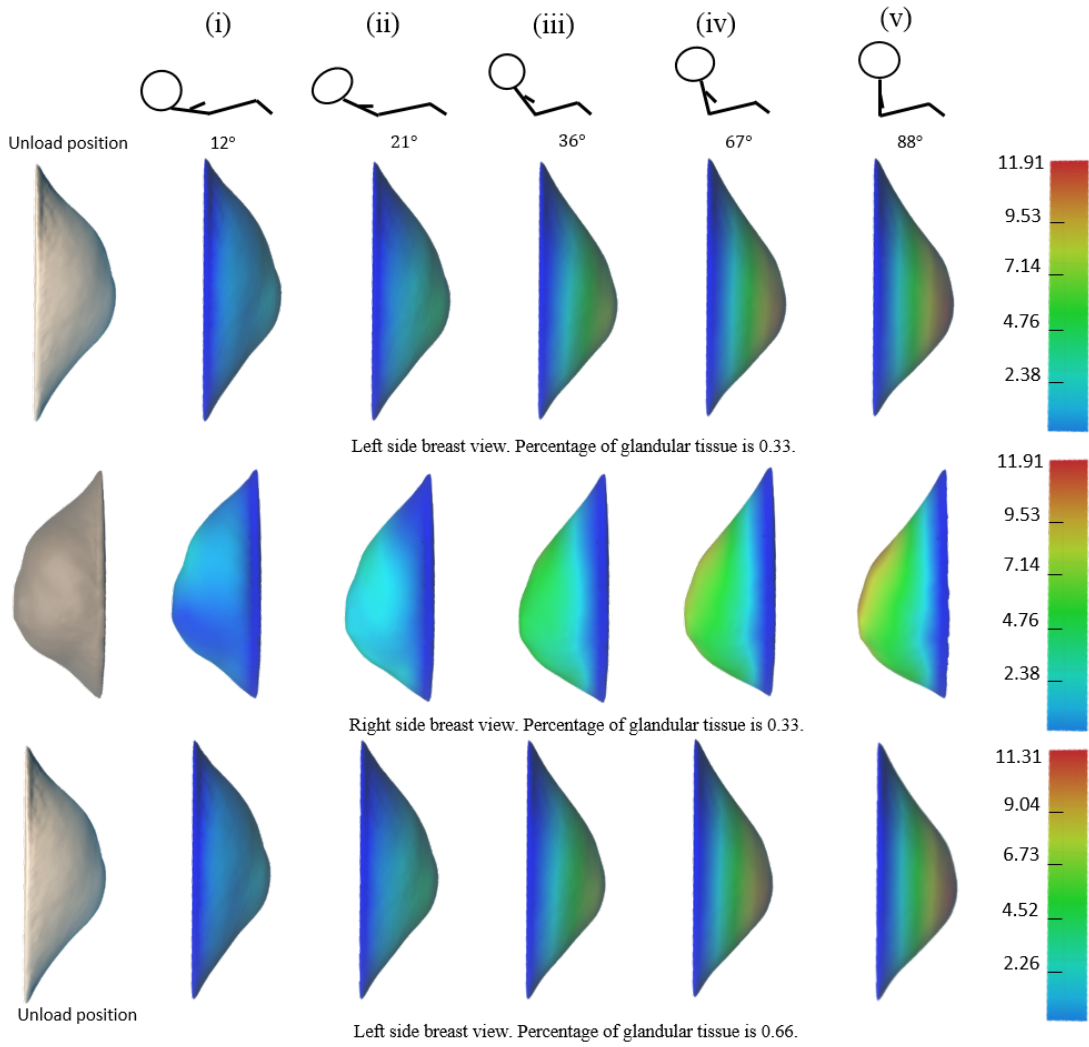


Figure 3.14: Transverse plane deformation (mm) at different angles.

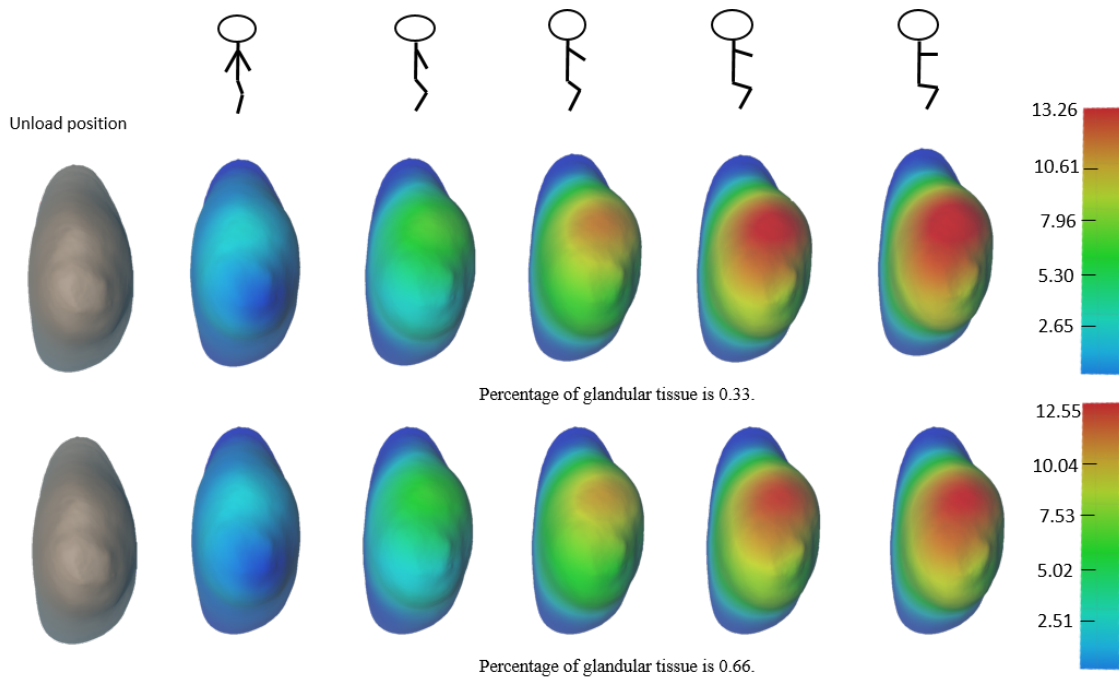


Figure 3.15: Sagittal plane deformation (mm) at different angles.

3.8.2 Large and small deformation and their error

In order to get a better insight into the quality of the simulation, we discuss two scenarios: one with a small deformation and another with a large deformation. We calculated the average NDE error and plotted the distribution of error percentage versus the magnitude of error and the distribution of error percentage versus the displacement magnitude. The reason is that although we used a hold-out validation and found the errors are rather small, different scenarios have different displacements, ranging from small to large. The area that has the largest deformation may not be the area with the largest errors. What we are doing here is similar to the authors of [8]. Rather than using two scenarios, they calculated the average error on their validation set. Fig. 3.16 to 3.21 show some color map and histogram visualization of the displacement and the NDE against the FEA corresponding to breast displacement. Fig. 3.16 and Fig. 3.17 are the surface nodal displacement magnitude. Fig. 3.18 and Fig. 3.19 are the displacement distribution from Fig. 3.16 and Fig. 3.17. Fig. 3.20 and

Fig. 3.21 are the NDE error when compare NN results with FEA simulation. The histograms are separated into 5 bins. Table 3.4 shows the mean and standard deviations of the errors based on the displacement magnitude for each bin. Figs. 3.18 and 3.20 represent the histograms for large deformation, and Figs. 3.19 and 3.21 represent the histograms for small deformation. For the displacement histogram in Fig. 3.18, the bin interval is 2.8 mm with a maximum point displacement of 13.2 mm. Almost 50% of the nodes have a displacement in the range of 0 to 2.8 mm, and the remaining nodes are spatially distributed in the other bins of the displacement range.

Similarly, for the error histogram in Fig. 3.20, the bin interval is 0.003 mm. About 50% of that nodal error is between 0 and 0.003 mm. A total of 90% of that error is below 0.009 mm. The largest error is 0.0158 mm, which is 800 times lower than the largest displacement. In the error color map in Fig. 3.16, the largest errors are located in the side of the breast, which is not the area with the largest displacements. The displacement color map indicates that the displacement in the largest error ranges between 3 mm and 10 mm, which is more than 190 times greater than the largest error.

For the small deformation in Fig. 3.17, the maximum error area is in the area where the largest displacement occurs. The largest displacement is 4.94 mm, which is 2.7 times smaller than that in the large deformation. Based on a comparison between the above two scenarios for small and large deformations, it can be inferred that the maximum displacement and maximum error are in the same region for smaller deformations and not in the same region for larger deformations. The largest error we found for the small deformation was 0.0774 mm, which is almost 5 times larger than the largest error we found for the large deformation. The bin interval for the smaller deformation is larger than the bin interval for the larger deformation with a similar distribution. However, it is unclear whether the error is large when the displacement is large, or the other way around. Table 3.4 shows more detail:

the mean error increases as the displacement magnitude increases for both deformations. For each bin, the mean error of the small deformation is larger. That indicates that the small deformation has a larger error. This conclusion is the same as that in [8].

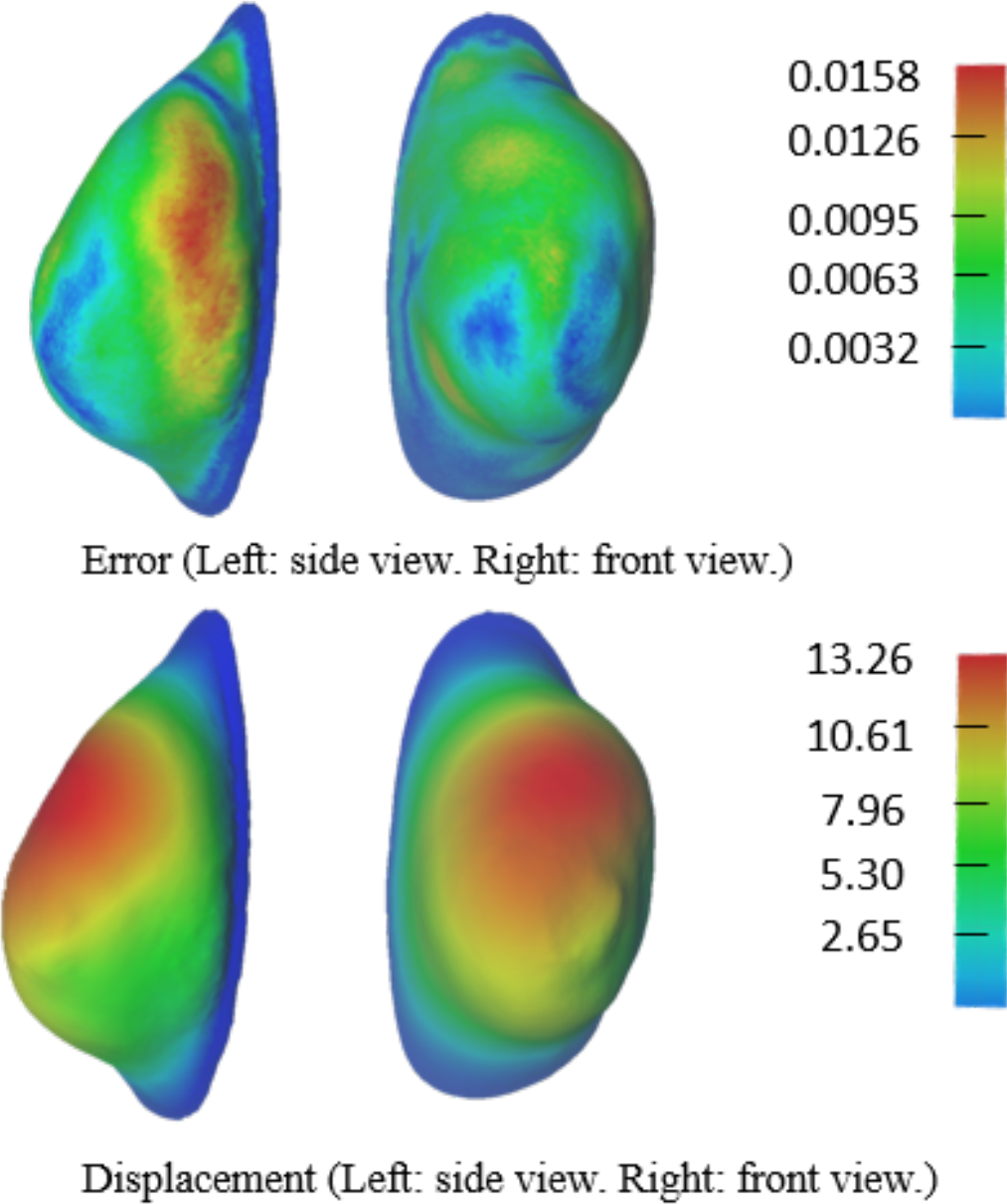
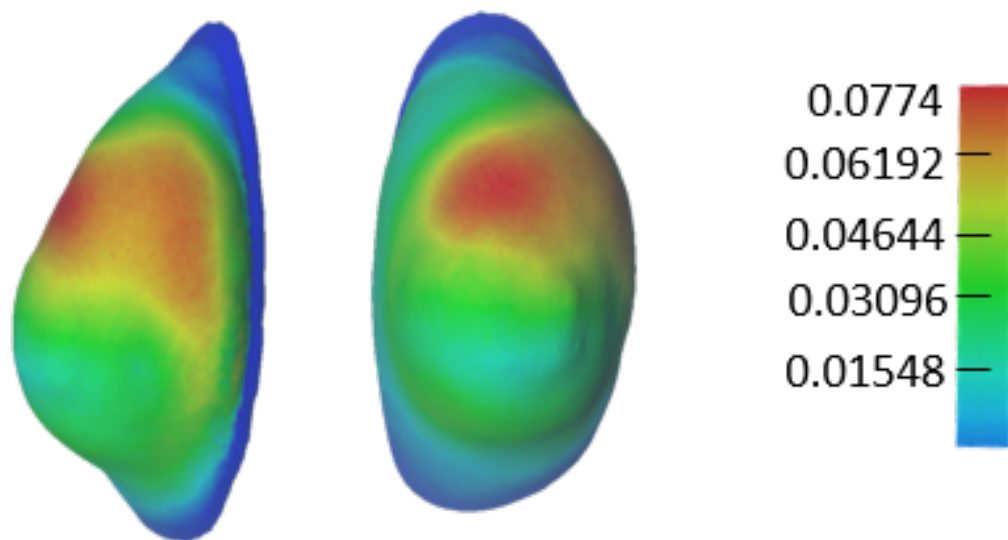
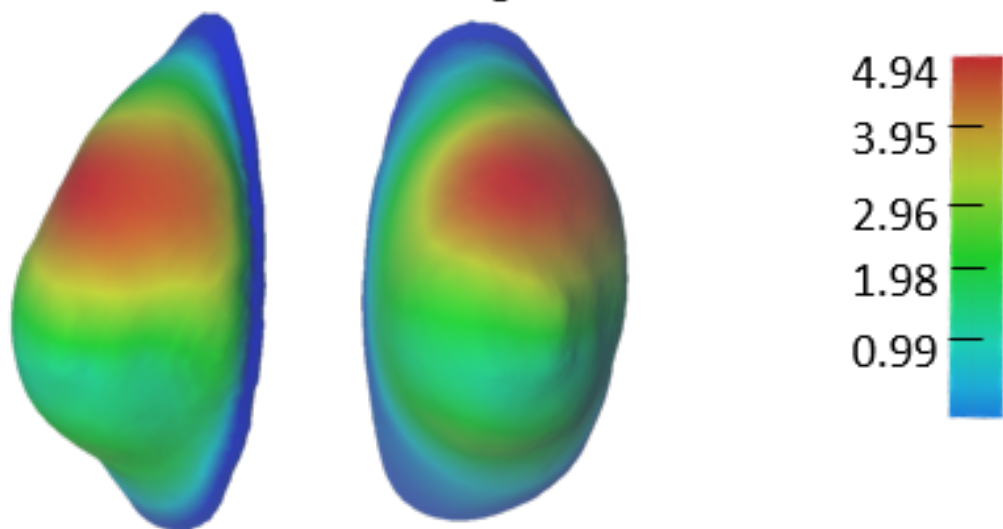


Figure 3.16: Large deformation.



Error (Left: side view. Right: front view.)



Displacement (Left: side view. Right: front view.)

Figure 3.17: Small deformation.

node percentage vs. displacement (mm)

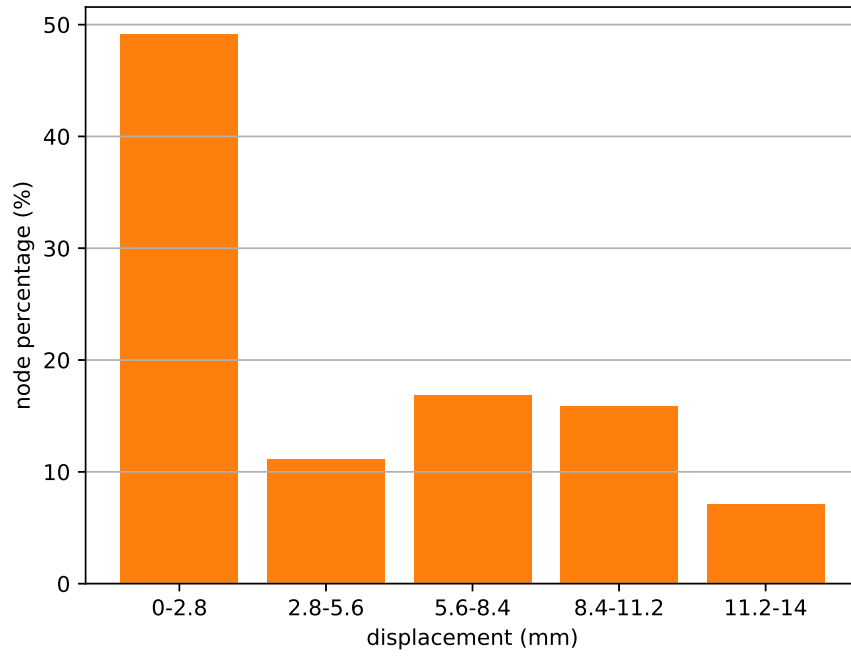


Figure 3.18: Large deformation. Displacement histogram.

node percentage vs. displacement (mm)

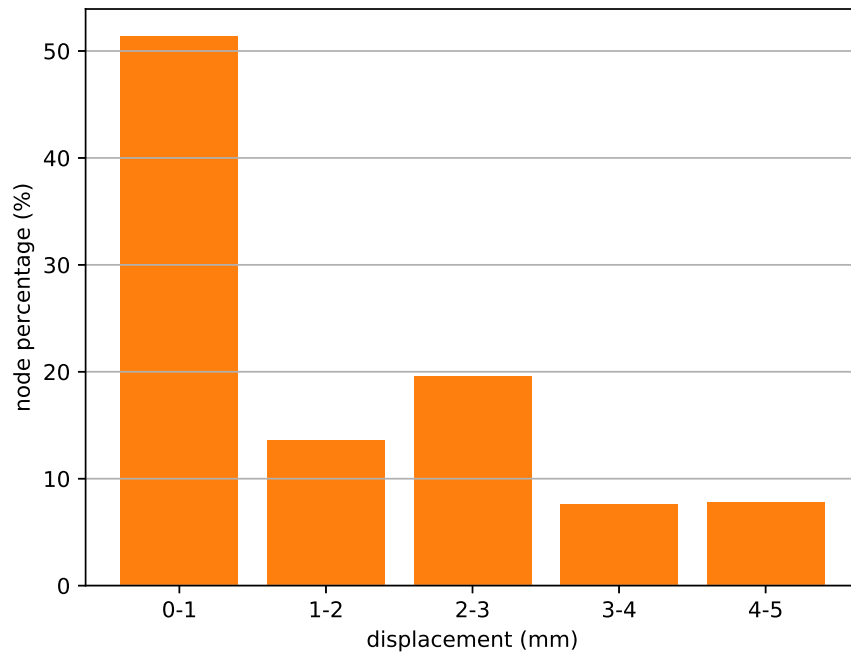


Figure 3.19: Small deformation. Displacement histogram.

node percentage vs. error

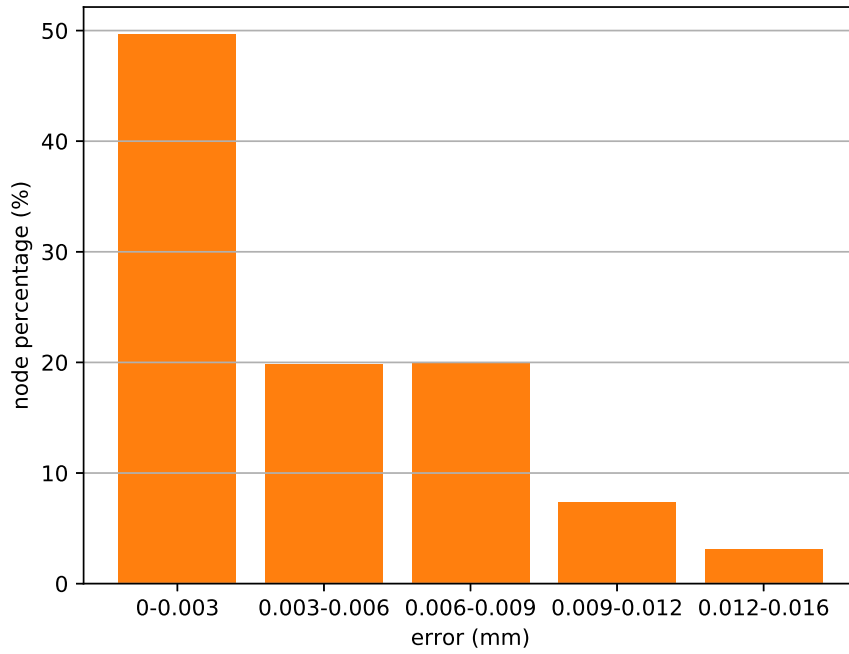


Figure 3.20: Large deformation. NDE error histogram.

node percentage vs. error

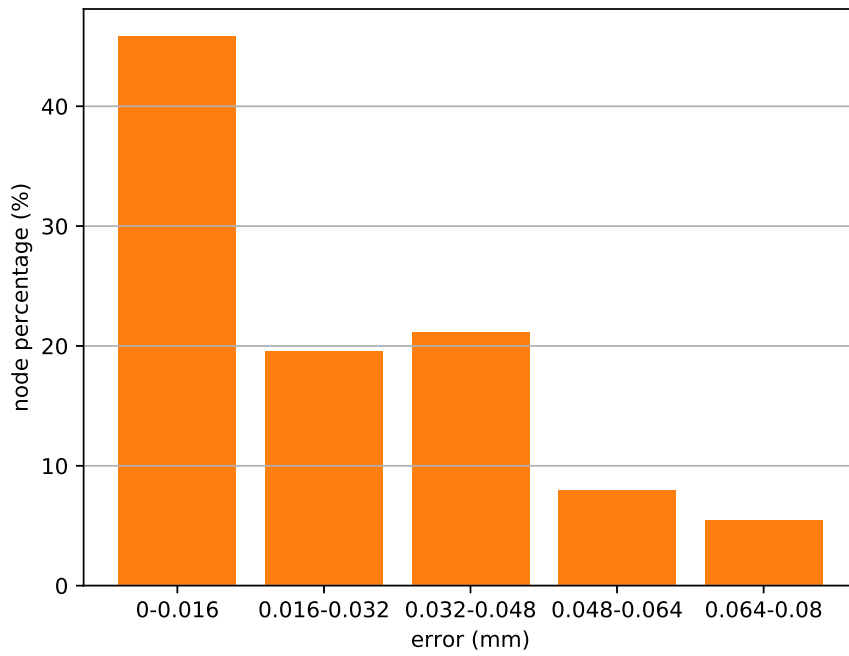


Figure 3.21: Small deformation. NDE error histogram.

Table 3.4: Mean and standard deviations of breast nodal displacements at large and small deformation.

deformation\bin#		1	2	3	4	5
Large (mm)	Mean	0.0013	0.0044	0.0072	0.0069	0.0073
	Std	0.0019	0.0022	0.0030	0.0033	0.0021
Small (mm)	Mean	0.006	0.025	0.0365	0.0507	0.0654
	Std	0.0099	0.0083	0.0065	0.0075	0.0047

3.8.3 Tumor location prediction

For different patient's position angles, the corresponding tumor locations are shown in Fig. 3.22 (transverse position plane) and Fig. 3.23 (sagittal position plane). The position scenarios were repeated for different glandular ratios, which are plotted in the same figure for comparison. In the transverse plane, the tumor starts from the highest position and slowly goes down. There is a little shift to the side (x plane) during deformation: about 0.3 mm. The shift in the z plane is almost 6 mm. In the sagittal plane, the tumor starts from the leftmost position and keeps moving right (x plane), with a little shift in the y plane of about 0.3 mm. The shift in the x plane is about 5 mm. The displacement magnitude decreases as the glandular ratio increases, which is consistent with the movement of the breast.

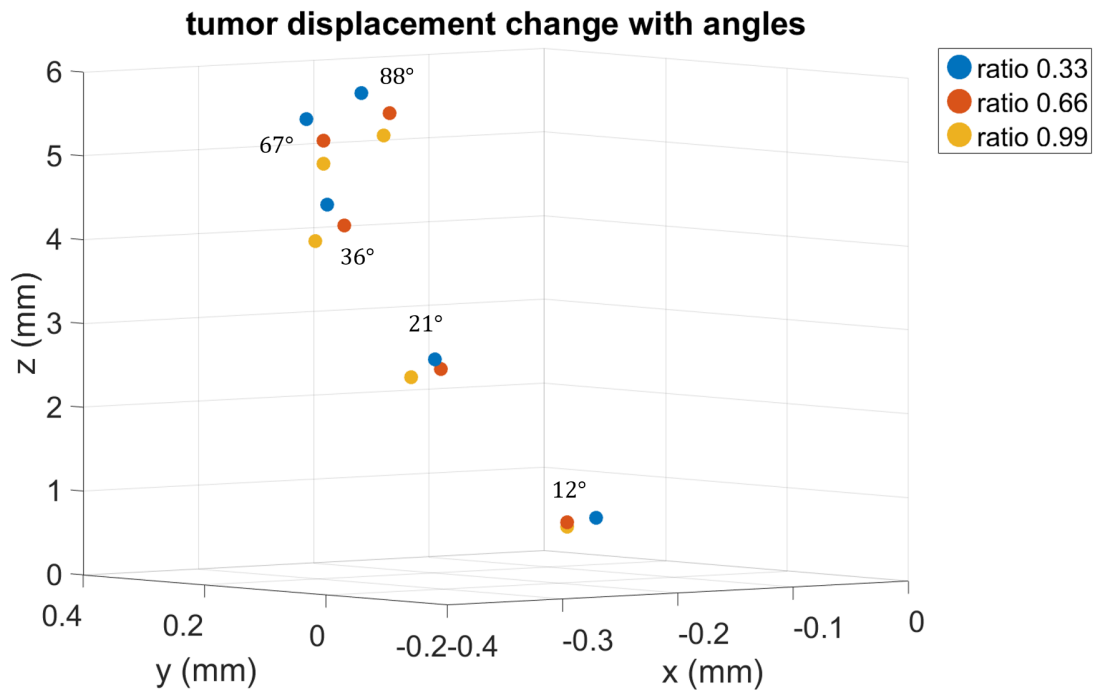


Figure 3.22: Transverse plane.

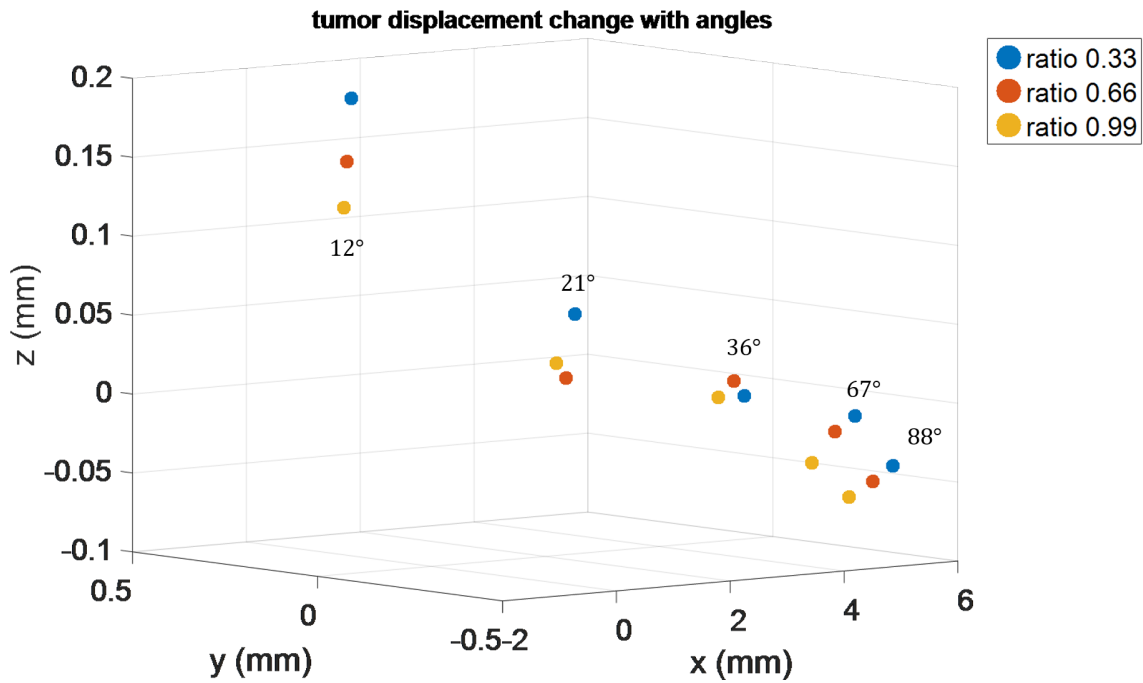


Figure 3.23: Sagittal plane.

3.9 Discussion

The process flow we developed to simulate breast deformation has two parts: (1) simulation of patient-specific data with FEA, and (2) building of a neural network for fast and accurate computation. The process started with the segmented data, followed by the data preprocessing. The raw data that were noisy and irregular were cleaned up and prepared for FEA simulation. When assigning materials to different parts of the breast, one can use either of two methods. The first is to generate a mesh for each part and assign material properties to the meshes, and the second is to generate one set of mesh and assign material based on the element location. The former is more accurate because the shape of each part is preserved, but it requires careful boundary condition setup; hence, we chose the latter. The chosen method for mesh generation can be inaccurate for tumor mesh generation yet accurate for glandular tissue and fat. The reason is that when a part is small and located close to the center of the breast, Tetgen generates a large tetrahedral element without preserving the shape. In our case, the glandular tissue and fat were not an issue because they covered large areas and were close to the skin surface. However, the tumor shapes were not preserved because the tumor was small and relatively close to the center of the breast. In the future, we will explore the accurate preservation of tumor shape.

We chose nonlinear material to ensure realistic simulation (skin, glandular tissue, and fat). The material properties have wide variations that depend on which published data are used, That may affect the accuracy as well. In order to reduce computation time, we constructed an NN to learn the deformation.

3.9.1 Neural network

We built two NNs in parallel for the same input: one model for tumor center displacement, and another for surface points' displacement. The output can be predicted as either an

exact position of each point or a displacement of the point. The problem with the output as an exact position of each point is that the magnitude of the position output is large, and that results in larger errors, which make NN training difficult. Since displacement is small compared to the position value, NN won't learn the displacement well. Our NN model's final training error and validation error are both very low. We created several color plots for different gland ratios with different patient angles (Fig. 3.16 to 3.21).

3.9.2 Displacement error

As presented before, we found that the error for a small displacement is larger than the error for a large displacement. It is an interesting observation because we expected that a large deformation would be harder for the NN to learn. One reason for the result might be that when the deformation is small, most of the points' displacements are close to 0, and can be regarded as noise. Those values do not give much information compared to those in the large deformation case. Based on the standard deviation from Table 3.4, for the first bin, the standard deviation of error is larger than the error itself, and this is true for both deformations. Thus, when the displacement is small, the error varies more from the mean, which indicates that the prediction is noisy around small displacements. Because of the large errors, the smaller deformation also has larger standard deviations. From the above analysis, we can see that the NN fits better for the larger deformation. However, even so, our errors are very small compared to those found in some of the similar research on tissue deformation, [62] for example. The authors use ML to predict breast shape after breast-conserving surgery. Their error was around 1 mm. Our observed validation error, 0.0048 mm, is much lower.

3.9.3 Breast deformation

From Figs. 3.14 and 3.15, we can observe that when the glandular ratio increases, the deformation magnitude decreases. That indicates that the stiffness affects the deformation more than the density does. In addition, the breast deformations in the inner and outer quadrants are not uniform because the breast shape is not symmetric. To better understand our result, we performed two statistical experiments on large and small deformations. We found that the trained NN works better with the large deformation scenario, perhaps because when the deformation is small, the values are more random or noisy than for a large deformation.

3.9.4 Tumor location

As mentioned before in Section 3, the geometry of the tumor's shape was not well preserved in our research. Like Pathmanathan [73], we did not perform tumor shape corrections to present an accurate tumor shape. In fact, earlier researchers used a ball shape or a single element [73] [46] [76] to represent a tumor, without retaining the actual tumor shape. One advantage of our work is that we attempted to use an actual tumor model segmented from a patient for FEA simulation. The result shows that the tumor's movement is consistent with the breast's deformation.

We are not aware of any previous work that did the same study as we did, but there have been similar efforts. Tonutti et al. [46] calculated a mean absolute position error for tumor between 0.1 mm and 0.2 mm. Their research focused on the tumor position when different forces were imposed on brain anatomy. They represent the tumor with a sphere and calculate the error based on all the points in the mesh. The authors' physics is more complex than ours because they used other forces in addition to gravity. Also, instead of predicting the center of the tumor, they predicted each point of the tumor. Hence, it is understandable that their error is much larger than ours. Zolfagharnasab et al. [62] achieved

a pairwise average distance of 0.827 mm in predicting breast shape after conserving surgery. While the above two examples had different purposes, they give a good indication of how much error is acceptable for tumor location prediction.

CHAPTER 4

VALIDATION OF REAL-TIME BREAST DEFORMATION

In Chapter 3, we used the data from online data set. We did a further validation of how gravity affects breast deformation in this chapter. As accessing real patient's data is challenging, we performed the experiment on a breast phantom and collected MRI images from different positions. To find the material properties, we performed an optimization approach as in [43,47]. We then built an FEA model and simulated breast deformation against different angles along the anatomical plane. Two NNs were constructed to map the gravity angle to the surface nodal displacements and a tumor center location. We performed a parametric study against different angles along an anatomical plane to learn the gravitational effects, and we studied NNs' ability to handle gravitational effects on breast deformation and tumor location.

4.1 Validation procedure

Fig. 4.1 to Fig. 4.5 show our overall process. The process is an extension to the work presented in [43], which studied the accuracy of FEA prediction on a breast phantom at different patient's position. We extended their process by adding the NN prediction step and validated the result with the MRI images.

We started by capturing MRI scans of the breast from different angles. Then, we registered the breast images to the same coordinates. After preprocessing of the breasts, the tetrahedral mesh was generated. After that, we implemented Visentin et al.'s [43] algorithm to find the

properties of the breast. We validated the result with a different angle. Later, we generated breast deformations under different angles. Two NNs were constructed to find the relation between the gravity direction and breast deformation, and between the gravity direction and the tumor center. There were 19 tumors in the breast. We chose one for our prediction. The same method could be extended to the other tumors. Finally, we compared the predicted breast shape and the MRI images taken at 25° and 50° .

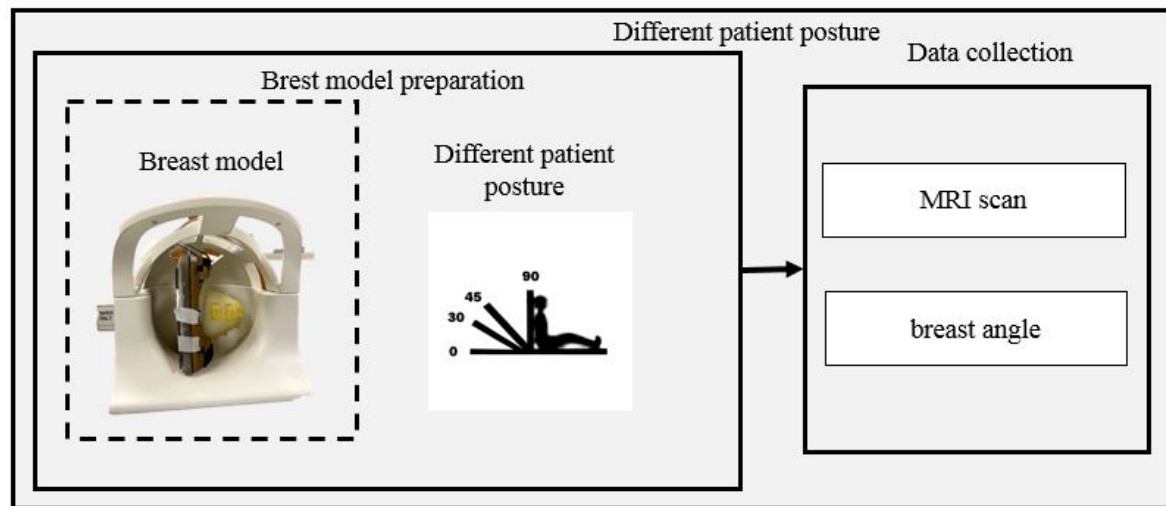


Figure 4.1: Overall process, step 1.

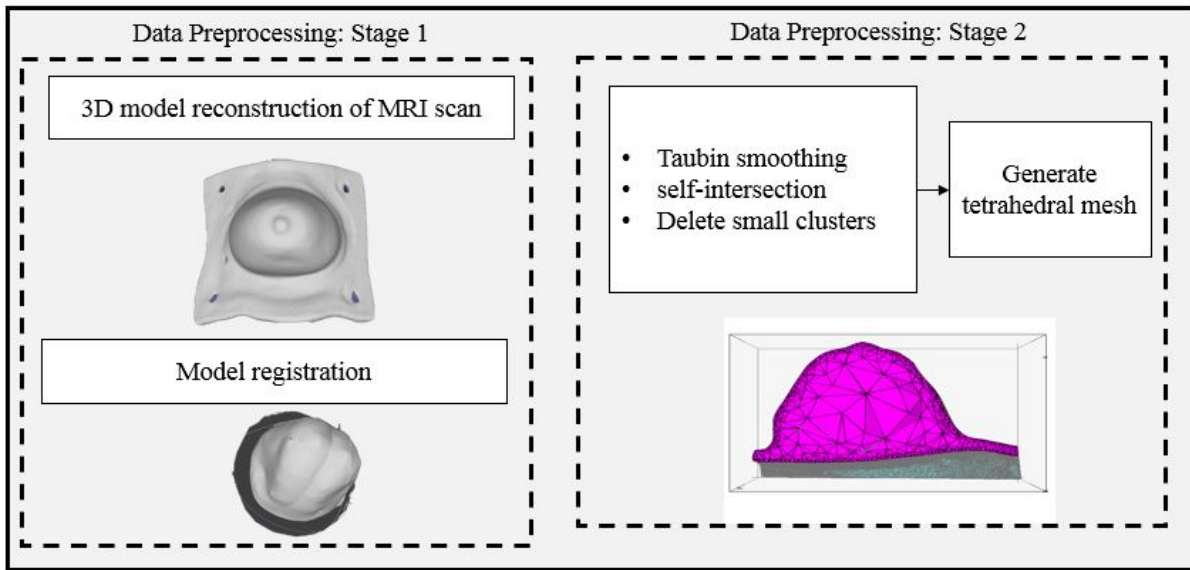


Figure 4.2: Overall process, step 2.

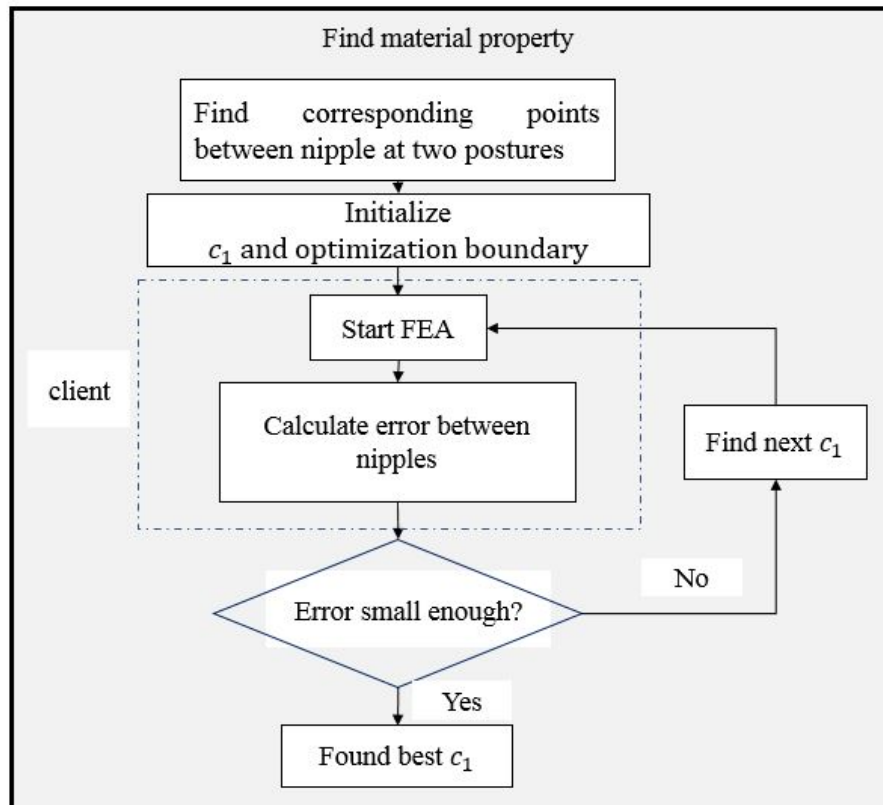


Figure 4.3: Overall process, step 3.

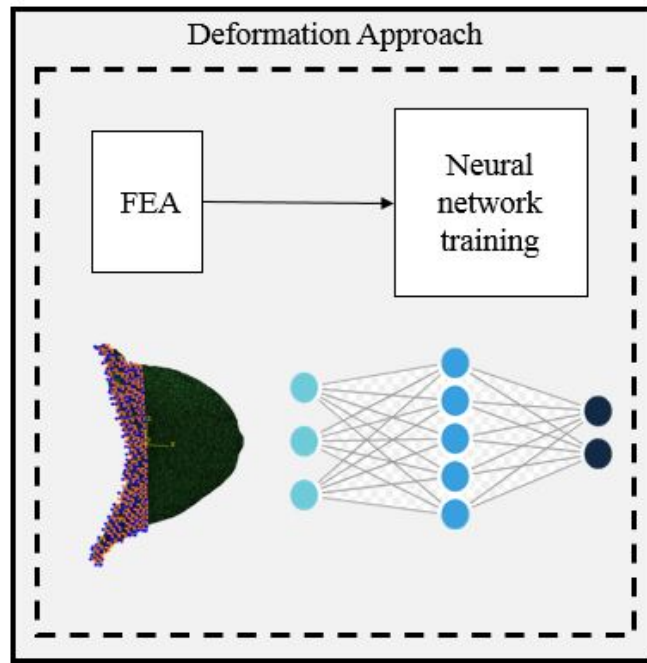


Figure 4.4: Overall process, step 4.

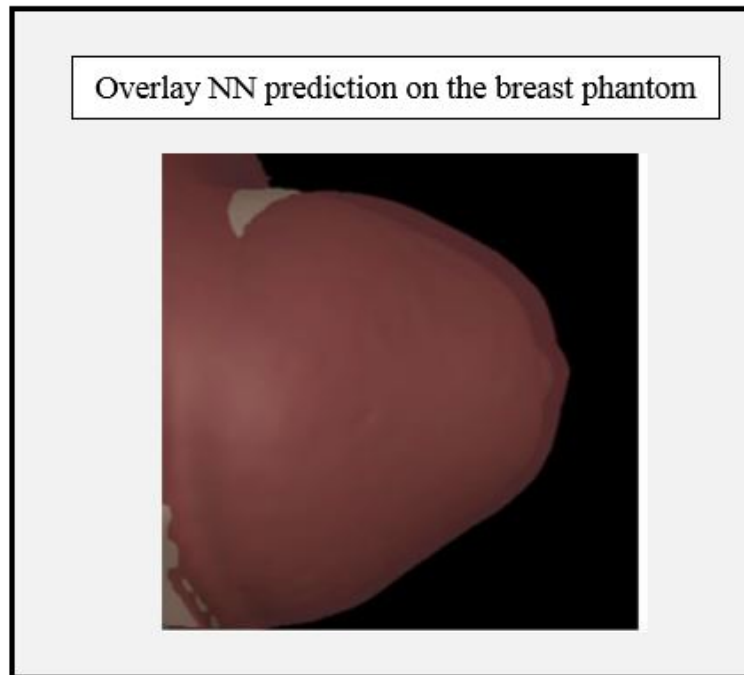


Figure 4.5: Overall process, step 5.

4.2 Data preparation

4.2.1 MRI scans collection

We purchased a CIRS 0731 (CIRS Inc.) breast phantom. The breast shape is irregular, and it mimics the heterogeneous breast tissue accurately. There are 19 internal stiff masses, whose diameters range from 3 mm to about 10 mm. We picked one tumor for our study, as shown in Fig. 4.7. Fig. 4.6 shows the set-up for our MRI image extraction. We used a 3T Siemens MRI scanner, and we chose a T2-weighted sequence for its clarity on tumor imaging. The slice thickness was 0.8 mm, and the resolution was 256 x 256. We don't have a breast coil, so we used a 20-channel head coil instead. To fit the breast into the coil, we removed the cover of the breast base, and replaced it with a paper cover. There was a small piece of Velcro at the top of the breast model, another on top of the 20-channel head coil, and two free Velcro straps to use for anchoring everything in place in a lateral orientation. 3 fiducial markers were placed on the back plane for registration purposes. Scans were captured at the left lateral and right lateral position. We also captured scans at 25° and 50° for further validation.

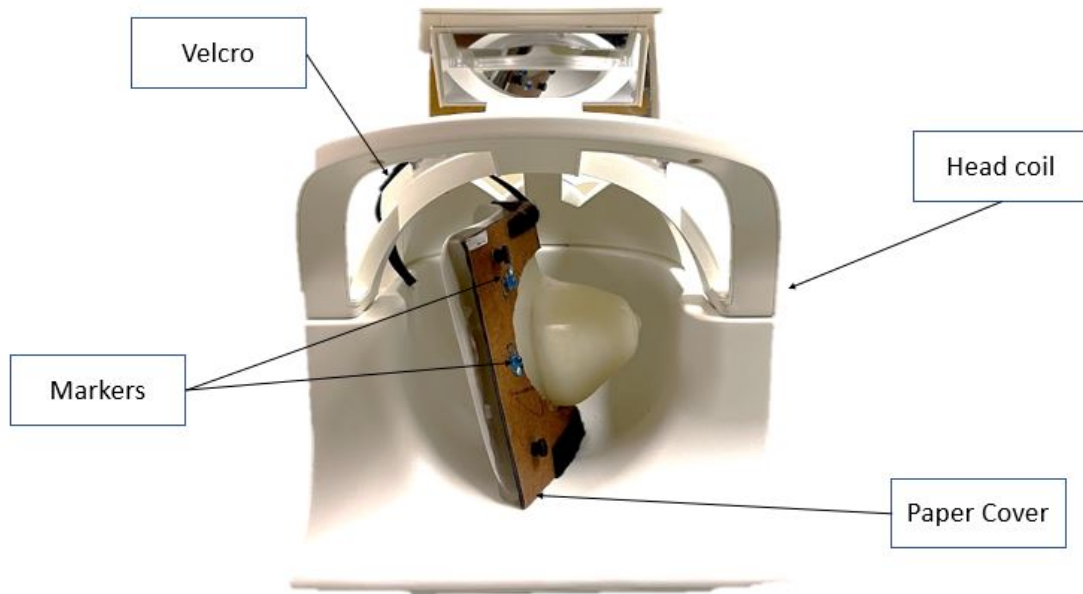


Figure 4.6: MRI scan experiment set up.



Figure 4.7: Tumor.

4.2.2 MRI images preprocessing

The generated MRI images were imported into a software called 3D slicer [11] for segmentation and 3D reconstruction. Arun et al.'s [10] algorithm was implemented to register all breasts to the same coordinate system with the help of the markers. The surface of the breast was decimated and smoothed using Z-painting function in MeshLab. We used TetGen [12] and generated a 4-node tetrahedral mesh for the phantom. When there were intersections on the surface mesh, TetGen failed to generate a tetrahedral mesh. We identified those points and fixed them with MeshMixer. There were 146,911 points and 579,990 tetrahedrons. More simplification of the mesh will cause a failure in TetGen. We observed that more than half of the tetrahedrons belong to the back plane. Fig. 4.8 shows the tetrahedral mesh we generated.

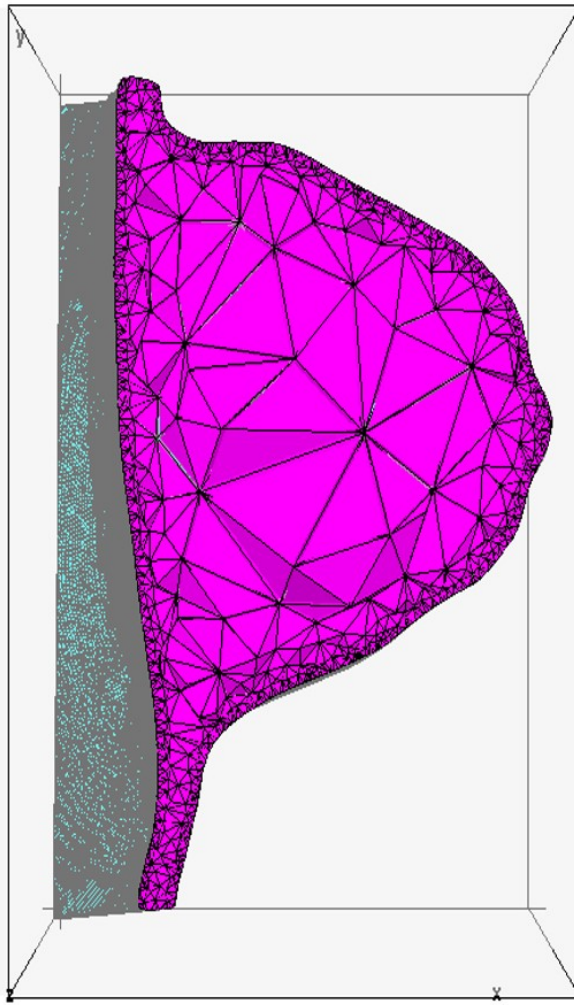


Figure 4.8: Breast tetrahedral mesh.

4.3 Deformation computation

4.3.1 Simulation setup and boundary conditions

To keep the simulation consistent, we performed the FEA simulation using Abaqus. 2 times of gravity forces was added to deform the breast from one lateral position to another. To achieve good results from FEA, the boundary conditions play an important role. Visentin et al. [43] achieved good results by cutting the entire back plane, and fixed the back of the breast. That also made the simulation easier. However, after finding the best material

properties and calculating the average closest point-to-point (PTP) distance of the whole breast surface, the average error was large (more than 8 mms). We had to explore how to improve the boundary conditions. The first thing we came up with was to include the back plane. We performed two experiments on how to include the back plane. Fig. 4.9 shows the boundary conditions. The left identifies the back plane tetrahedrons and preserves the shape of the back plane. The right is similar but cuts out the back of the plane. The left set-up has an error of 5.37 ± 3.77 mm, and the right set-up has a lower error, 4.93 ± 3.29 mm. That is a little surprising because the left one has more accurate boundary conditions. We think that one reason may be that the left set-up is harder for FEA to simulate and hence results in a larger number of numerical errors.

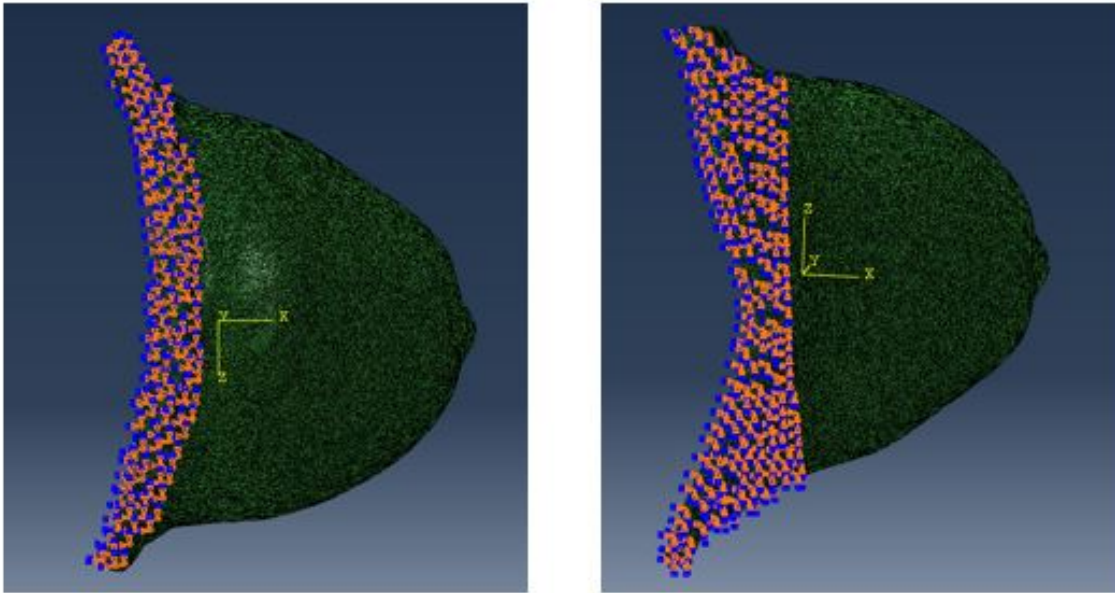


Figure 4.9: Abaqus boundary condition. The left preserves the shape of the back plane. The right does not preserve the shape.

4.3.2 Material estimation

We implemented Visentin et al.'s algorithm [43] for material estimation. Neo Hookean was the material we chose; Eq. 4.1 gives the strain energy density function for incompressible

material,

$$W = c_1(I_1 - 3), \quad (4.1)$$

where c_1 is a material constant. $I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$, and the λ are the eigenvalues of the stretch tensor [4]. Fig. 4.3 shows our detailed implementation. The basic idea is to guess an initial value of the material properties and, based on the FEA simulation, estimate the next value of the material properties. The error was calculated based on the closest PTP distance of the nipple. The iterative closest point (ICP) was used to identify the corresponding points. When c_1 is too small, the breast deformation becomes very large, and hence causes a long running time and even failure to converge to a solution. Hence, the optimization boundary should be set carefully before the optimization process.

Broyden–Fletcher–Goldfarb–Shanno (BFGS) was chosen as the optimization method. The whole process was implemented in Python; the Scipy package was used for the optimization method. As Abaqus uses an old version of Python, we were not able to import both packages into the same script. Hence, we created a socket communication between the two scripts. The client side was responsible for running Abaqus and reading the result, and the rest was on the server side. The best result of c_1 was at 420 PA, and the nipple’s PTP correspondence error was 4.20 mm. We simulated a different scenario using the optimized material at 25°, and compared the FEA simulation with the ground truth. Fig. 4.10 shows the error plot. The average error was 3.63 ± 1.12 mm. The maximum error was 7.3 mm and occurred at the bottom of the breast. That is the part that connects to the boundary plane, which was almost fixed. Most of the area had an error between 3 mm and 4 mm. To calculate the tumor center position, we moved the tumor from the upload position (no gravity) with a displacement equal to the center of the tetrahedral element closest to the tumor. The error of the tumor was 3.647 mm. Many things can cause such errors. In the MRI images, we can see the breast has a skin-like layer, which we did not take into account in the material

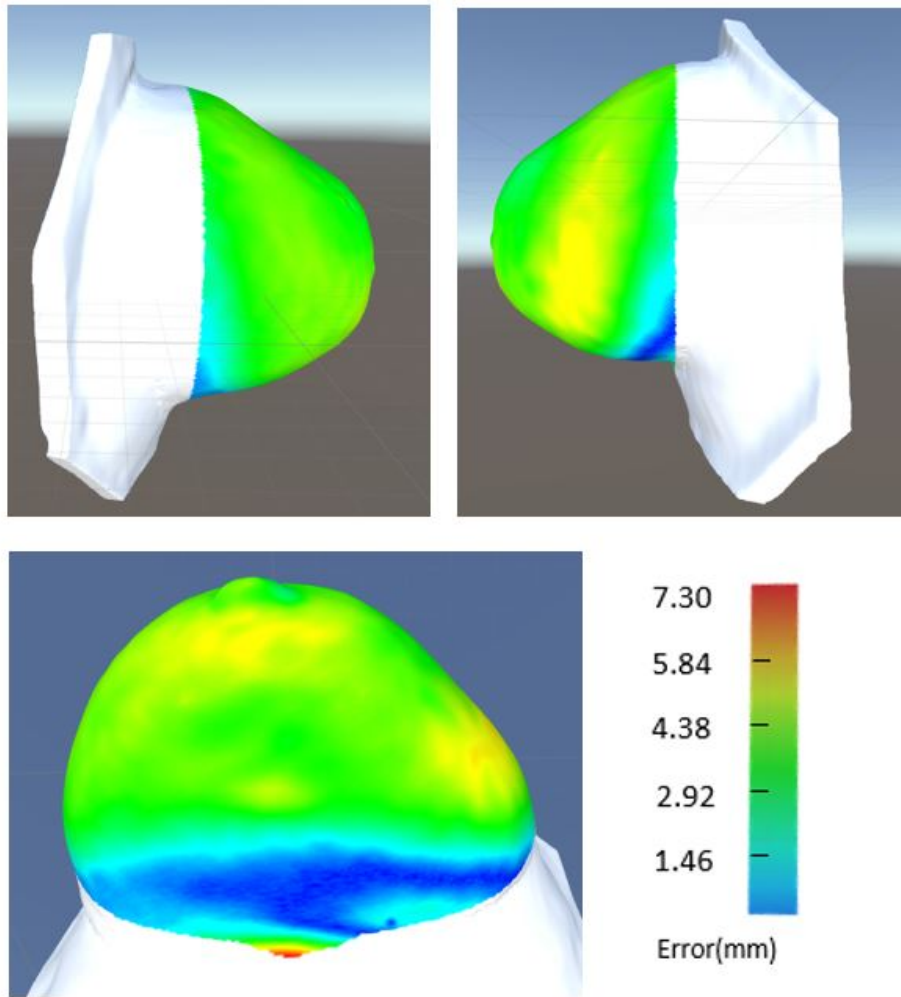


Figure 4.10: Material estimation error plot between FEA and NN at 25° . The error is calculated as the closest PTP distance and is calculated between closest points of two breasts without including the back plane (shown in white).

model (Fig. 4.11). That might be the largest source of error. Another source of error could be the simplification of the material we chose. Finally, we used the nearest PTP distance to calculate the error, but in reality, we don't know the exact mapping between the points.

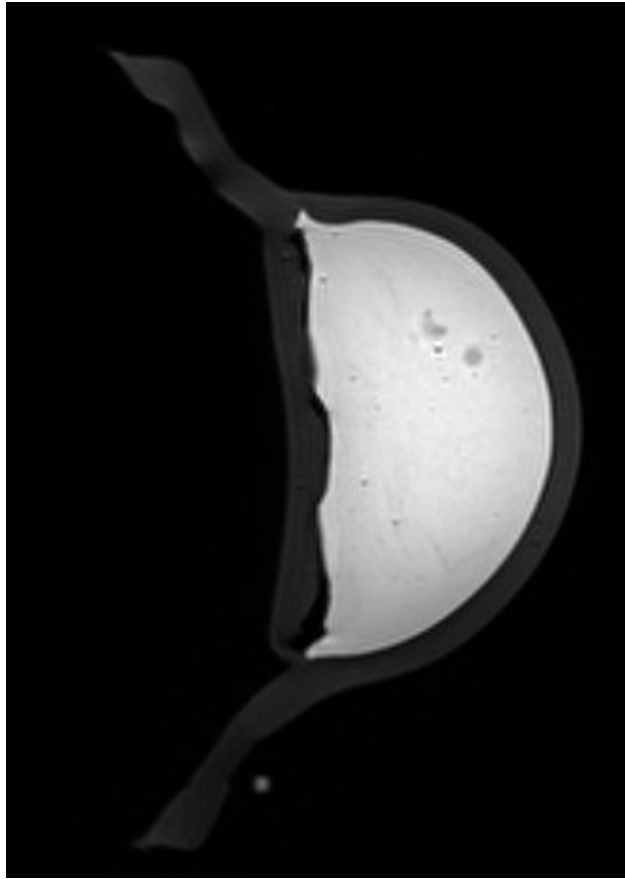


Figure 4.11: MRI image.

4.4 Neural network for breast deformation and tumor location

We implemented an ANN with the same architecture as in previous chapters for surface nodal displacements prediction and tumor center prediction. The only difference is we didn't include gland / fat ratio because we assumed a homogeneous material. We generated ground truth from Abaqus, starting from a gravity angle of 0° and ending at 360° , with an interval of 15° . Adam was chosen as the optimizer, and the learning rate was 0.00001. We used a mean absolute error (MAE) for the two scenarios. We performed a hold out validation during training, and split the data into 80% used for training and 20% used for testing (left side in Fig. 4.12).

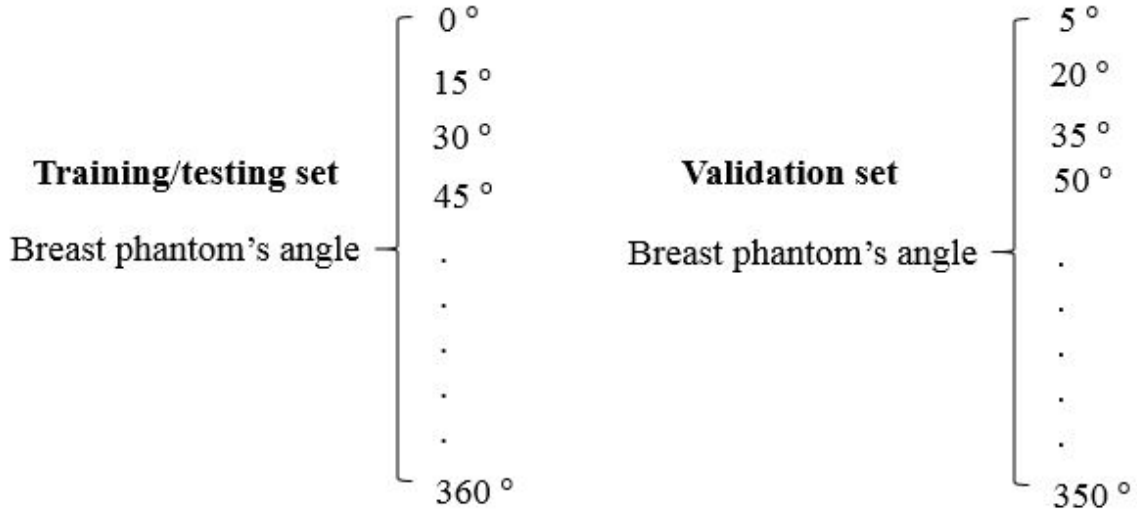


Figure 4.12: Training, testing and validation set.

4.5 Results obtained from neural network

We generated a separate set of data from Abaqus for validation, which are 10° apart from those in the training set (right side in Fig. 4.12). The validation data includes both the transverse and sagittal planes (a total of 48 scenarios). We trained the NN for 30,000 iterations, and Tensorflow was used to construct the NN. In breast deformation prediction, The training error was 0.0358 mm, and the testing error was 0.0415 mm. In tumor center prediction, the training error is 0.0731 mm and the testing error is 0.0896 mm. Since we average the error among x, y and z axes, we need to transfer the error to PTP distance. We can use Eqn. 4.2,

Table 4.1: Errors on tumor and breast deformation. The error of tumor is the center distance. The breast deformation error is the average PTP distance in Training, testing and validation set. In 50°, we used the closest PTP distance.

	Tumor error (mm)	Breast deformation error
Training (compare NN with FEA)	≤ 0.219	≤ 0.107
Testing (compare NN with FEA)	≤ 0.269	≤ 0.124
Validation (compare NN with FEA)	0.176	0.192
50° (compare NN with MRI data)	3.65	2.69

$$\begin{aligned}
& 1/N \sum_{i=1}^N \sqrt{(x'_i - x_i)^2 + (y'_i - y_i)^2 + (z'_i - z_i)^2} \\
& \leq 1/N \sum_{i=1}^N (|x'_i - x_i| + |y'_i - y_i| + |z'_i - z_i|) \\
& = 3 * \rho(f(X; \Theta), Y).
\end{aligned} \tag{4.2}$$

The result is shown in Table 4.1. Training, testing and validation are the comparison between NN and FEA result. In order to directly compare the result with the real breast data, we took MRI images of the breast at 50°. The NN prediction displacement of tumor at 50° is 10.617 mm. However, this does not indicate that our error mainly come from NN. We compared the result with [43], since they are using the same breast model, but they only compared the FEA result with the MRI images. They found the error of the breast surface displacement is between 2.499 mm to 3.813 mm from different angles, and their tumor error is a little under 2 mm. Hence, our result is relatively close to what they achieved. We can also concludes the main error source comes from the FEA part. We will further discuss the result in the following section.

4.5.1 Neural Network deformation prediction

Fig. 4.13 and Fig. 4.14 show the breast deformation predictions obtained using NN. The final shape of the prediction shows a good match with the ground truth. Fig. 4.15 and Fig. 4.16 show the breast deformation at different angles. We normalized the color with respect to the largest and smallest nodal displacement. The displacements were calculated against the upload position. As one would expect, the breast deformation is small when the angle is small, and increases when the angle increases. The breast nipple has the largest nodal displacements; the displacements slowly decrease as the nodal point locations are further away from the nipple.

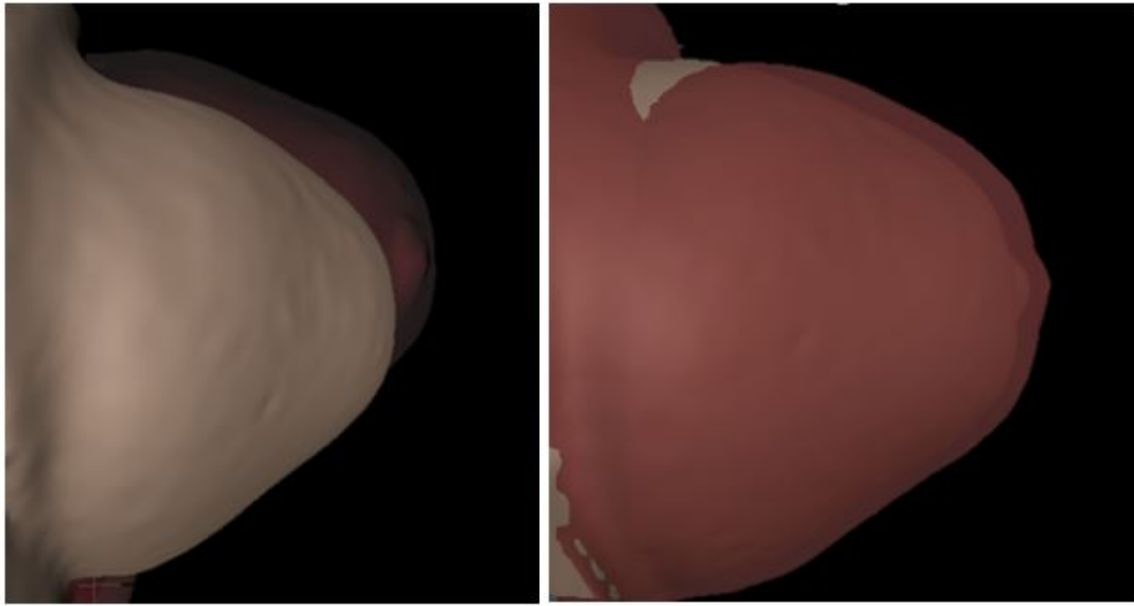


Figure 4.13: Overlap the simulated breast with the ground truth. The solid breast is the ground truth taken from the MRI images at 25° in a left lateral position, the transparent is the breast at upload position (left picture) and after deformation using NN (right picture).

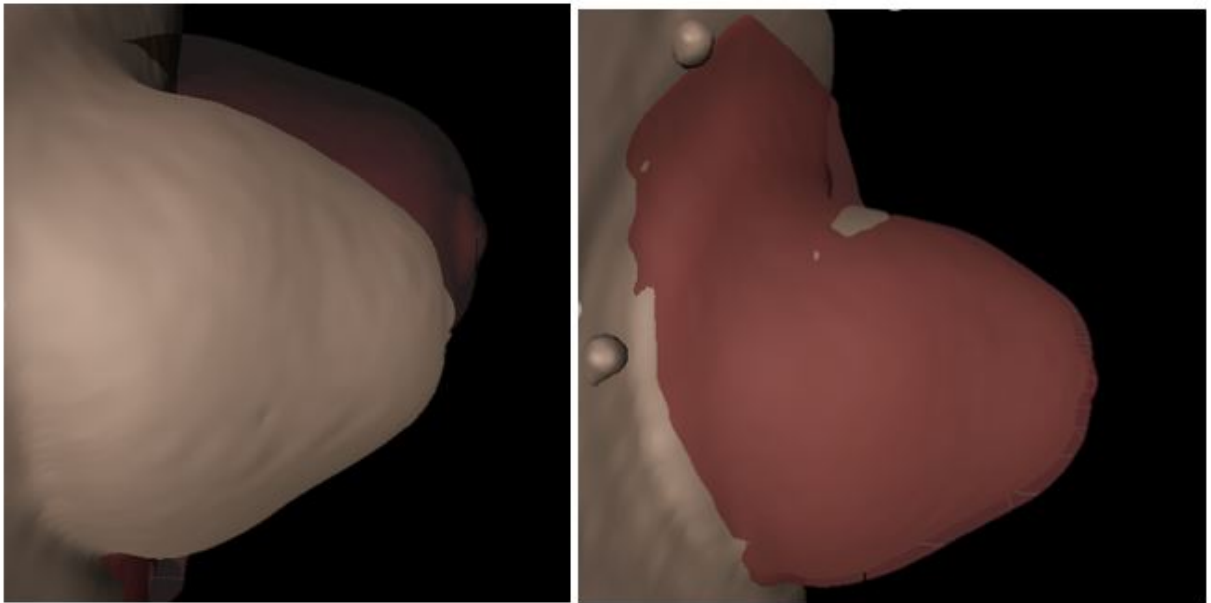


Figure 4.14: Similar to Fig. 4.13 except the right picture shows the shape after deformation at 50° .

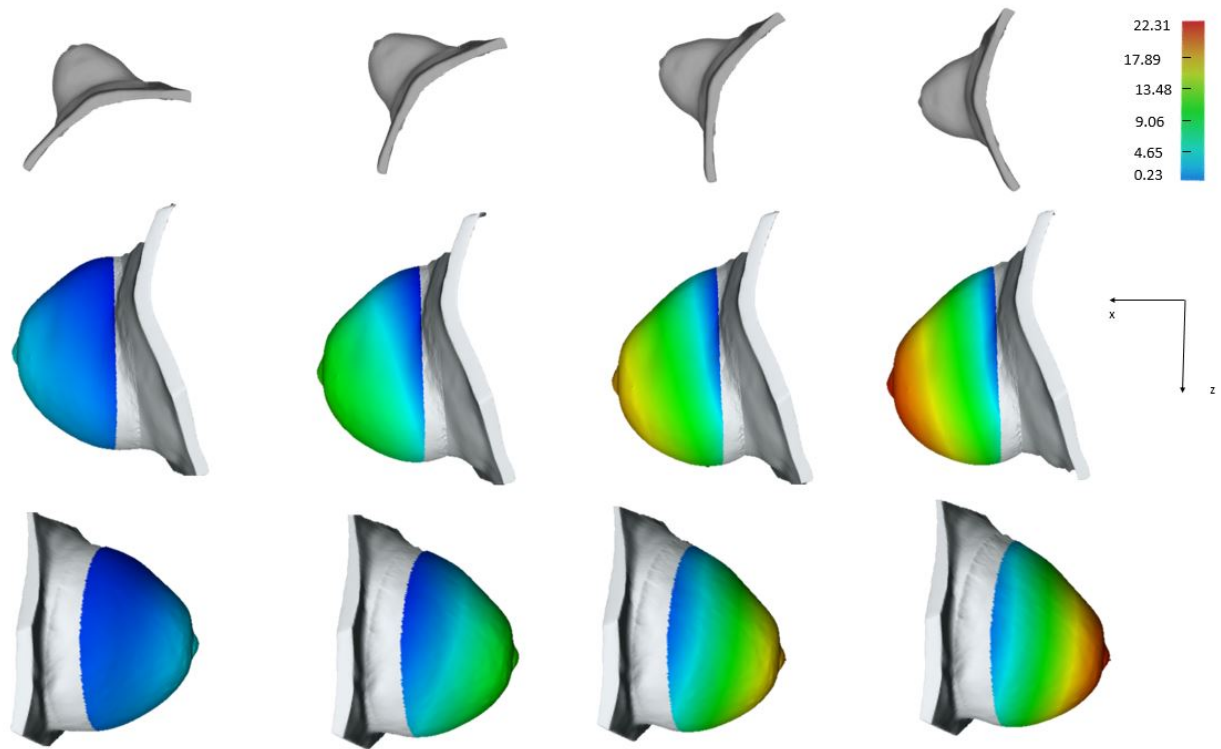


Figure 4.15: Transverse plane deformation (mm) at different angles.

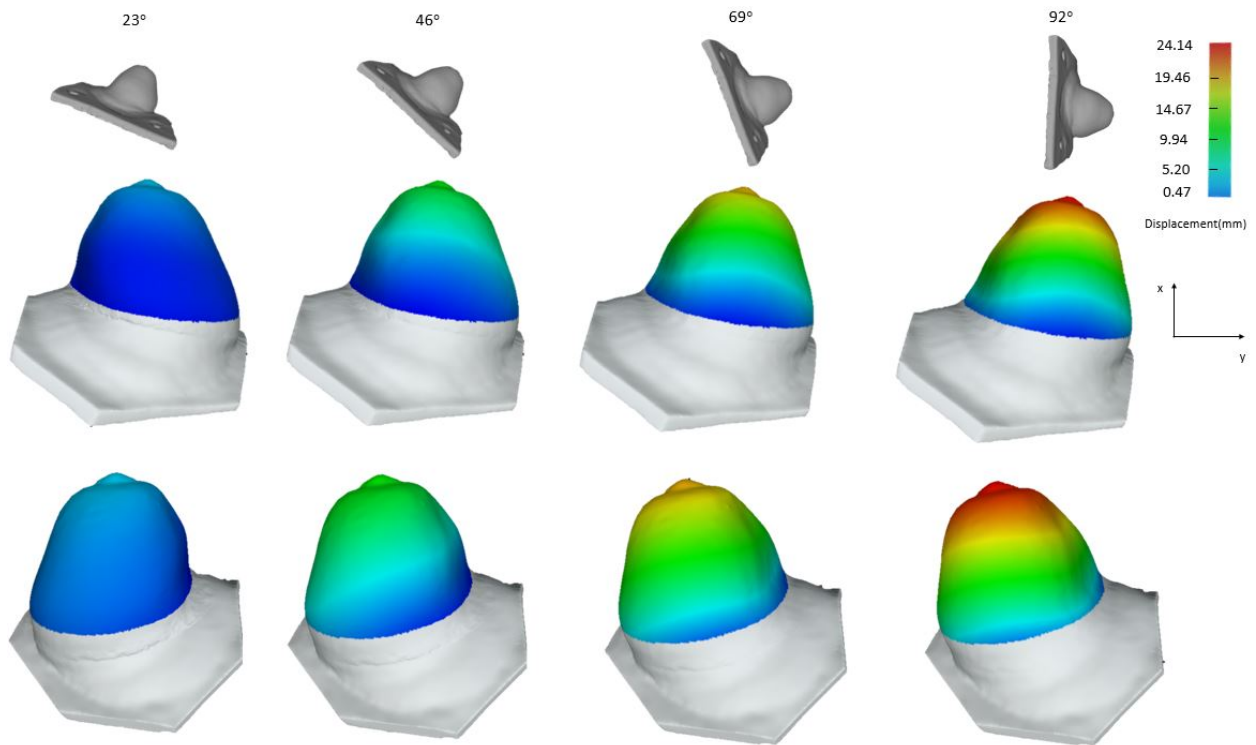


Figure 4.16: Sagittal plane deformation (mm) at different angles.

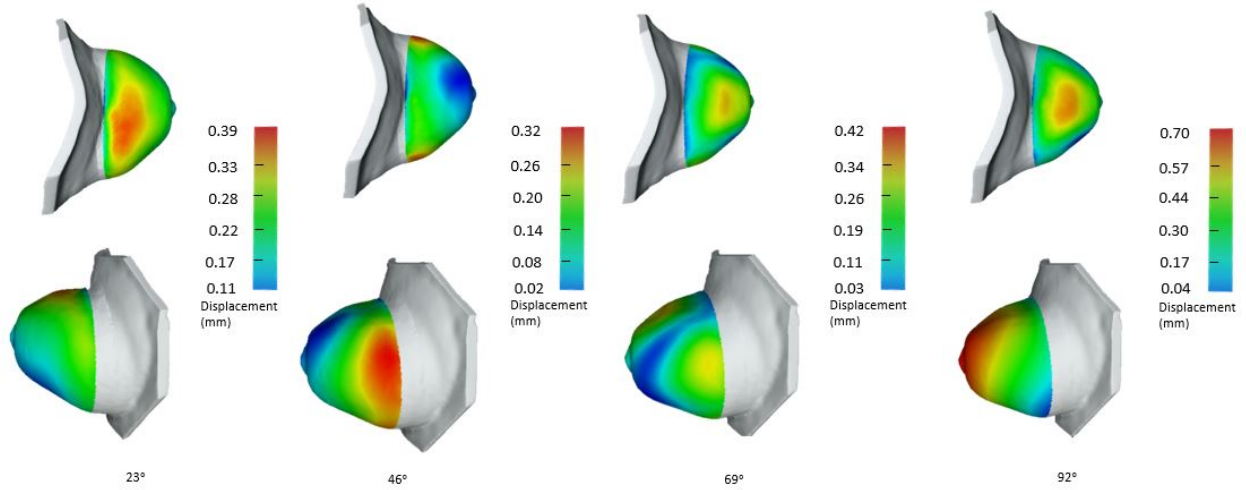


Figure 4.17: NN error plot at different angles. The error is the PTP distance between NN and FEA simulation.

4.5.2 Nodal displacement error

Fig. 4.17 shows an error plot of NN compared to FEA at several angles. When the angle is small, the largest error area (red) is close to the back plane. As the angle gets larger, the largest error slowly shifts to the nipple area. The error gets larger as the angle gets larger because of the increasing displacement. The error distribution changes quite a lot in different angles. Even so, the error is still quite small.

Fig. 4.18 is the error distribution plot. As with the displacement distribution plot, the vertical axis is the percentage of nodal points that fall within the error range. As can be seen, more than 60% of the points fall within 0.19 mm. The number of points with an error more than 0.67 mm is very small. The largest error is 0.96 mm, which is less than 3.8% of the largest displacement.

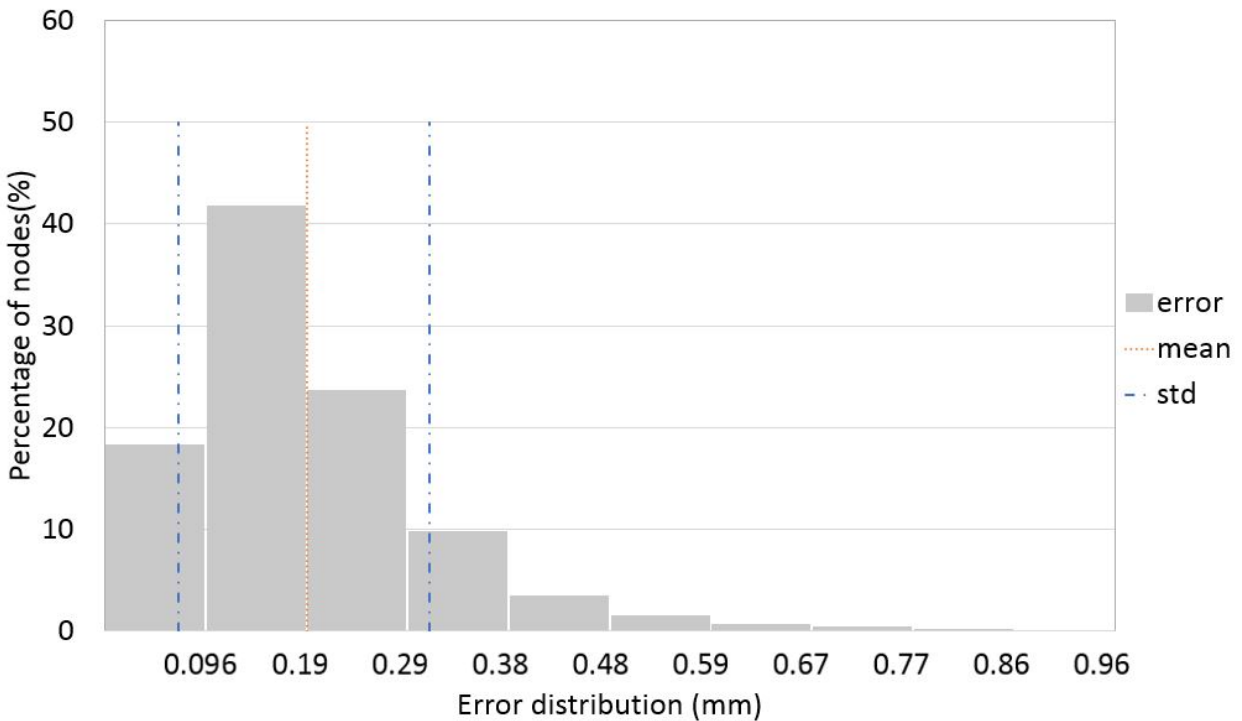


Figure 4.18: Error distribution plot (mm) between FEA and NN in the validation set.

4.5.3 Tumor location

Fig. 4.19 and Fig. 4.20 show the tumor center displacement at different angles. The plots in Fig. 4.19 and Fig. 4.20 are 2D because the tumor displacement along the third axis is very small (≤ 0.75 mm). The axes correspond to the axes in Fig. 4.16. and Fig. 4.15. In both cases, the displacement magnitude along the x-axis decreases as the angle increases. The reason is that the angle alignment between gravity and the x-axis becomes larger, so the effect of gravity is reduced. As the angle of gravity shifts towards y-axis (Fig. 4.19) and z-axis (Fig. 4.20), the tumor displacement increases along the two axes.

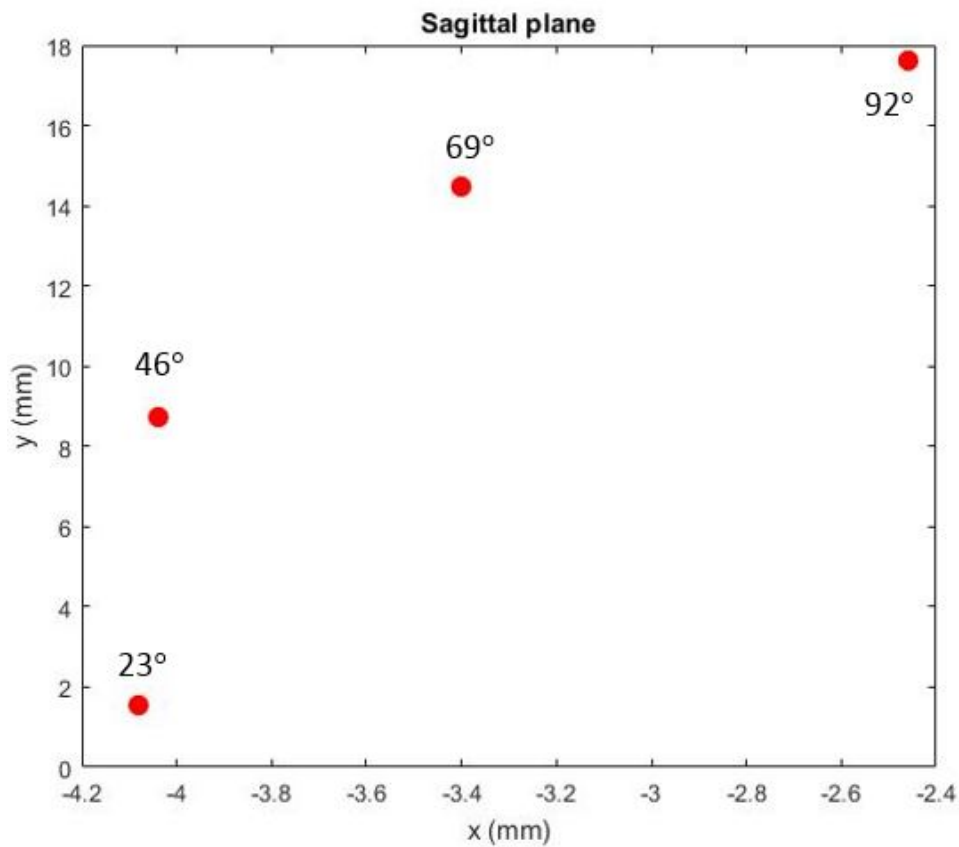


Figure 4.19: Tumor center location at different angles along sagittal plane.

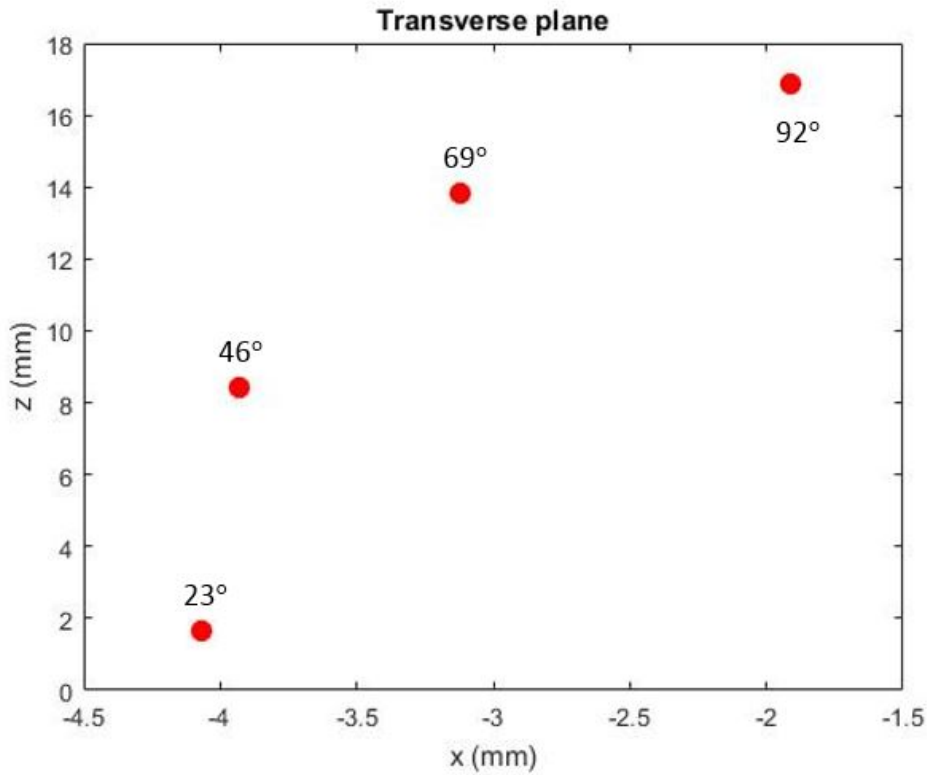


Figure 4.20: Tumor center location at different angles along transverse plane.

4.6 Discussion

We have to make many engineering decisions in this project. During FEA simulation, we chose to use a homogeneous material to describe the entire breast. While it might be more accurate to separate the breast into skin, tumor, and other parts, the results we achieved are good for our simulation. In fact, the authors of [43] and [59], who purchased the same breast phantom as us, also used a homogeneous material for their simulation. The back plane of the breast is a structure with a curved shape. Cutting the back plane will greatly reduce the number of tetrahedrons, but we found the error was large during material optimization. We explored two ways to include the back plane. When choosing the NN structure, we used the shallow NN for the deformation prediction because that approach achieved good accuracy in the literature. We took the average of the sum of

the errors in the x, y, and z directions for training purposes, and converted to PTP distance error. Later, we collected statistics on the PTP displacement in the validation data set. The error turned out to be quite small. We overlapped the NN result onto the data collected from MRI scans, and found good matches.

CHAPTER 5

AR DEPLOYMENT

AR and virtual reality (VR) becomes popular as more technologies are available recently. Both technologies help users interact with the environment. The difference between the two is AR fuses the virtual objects with the real world environment and VR creates a completely separate virtual world. In medical simulation, AR has been explored in endotracheal intubation [74], laparoscopic surgery [69], ventriculostomy [70] and other procedures. According to Frenk et. al [71], AR provides students with rich contextual learning that enhances their training skills. Hence, extending our application in an AR environment can be helpful for medical training or simulation. However, a challenge for us to extend to an AR environment is the integration between AR and ML. The AR platform exists today cannot be integrated with the NN model we trained. We did some exploration in this chapter and found the result promising.

People have been using AR for deployment in mobile devices. Fig. 5.1 is an example that uses AR to show different places' names on a street. Android and iOS phones and tablets are all capable of AR app deployment. Another very popular device is HoloLens, which use eye glasses to render scenes in front of people's field of view. Different platforms are available for AR development. ARKit is the software development kit (SDK) for iOS devices. It is integrated with Xocde natively. ARCore is the SDK for Android and pairs with Unity. Vuforia is a computer vision algorithm library that can be used for maker or makerless based tracking in AR environment. We chose to use Vuforia because it can be integrated with Unity. In terms of hardware, we tested an Android tablet and an iPhone.

In iPhone 12, it took a few minutes to build an app, and it can only stay in iPhone for 7 days. iPhone also requires us to use Mac because we need to use XCode other than Unity. Compared to iPhone, Android tablet doesn't have a time restriction and the deployment takes within 1 minute. The deployment is also simpler since it only requires Unity. Hence, Android tablet was chosen. Specifically, we used Samsung Galaxy Tab S6 Lite as our device. It has a 4GB RAM with an Exynos 9610 processor. Since we are using Unity and Vuforia, the same program we developed can be easily adapted to other devices, including HoloLens and Apple devices.



Figure 5.1: AR app for street view [2].

5.1 Image tracking

In Vuforia, there is marker based and markerless based tracking. In marker based tracking, the user can use a QR code, a 3D object and a 2D image. The marker should have a good number of feature points. In markerless based method, Vuforia uses data from sensors such as gyroscope and accelerometer to place objects in the environment. We tested the tracking

with a 3D object approach. We drew some markers on the surface of the breast and tested with our AR application, shown in Fig 5.2. Unfortunately, Vuforia is not able to detect the breast. The reason might be that our object is too large. Hence, we chose to use a marker based method with a 2D image. We printed the marker on a piece of paper and placed the virtual object on the marker. One can upload the image to Vuforia website, and it will evaluate the quality of the marker. The better the quality, the better it is able to track. Fig. 5.3 is the breast and label in Unity. The breast is a child object of the label. Hence we can place breast relative to the label and move breast in AR environment.

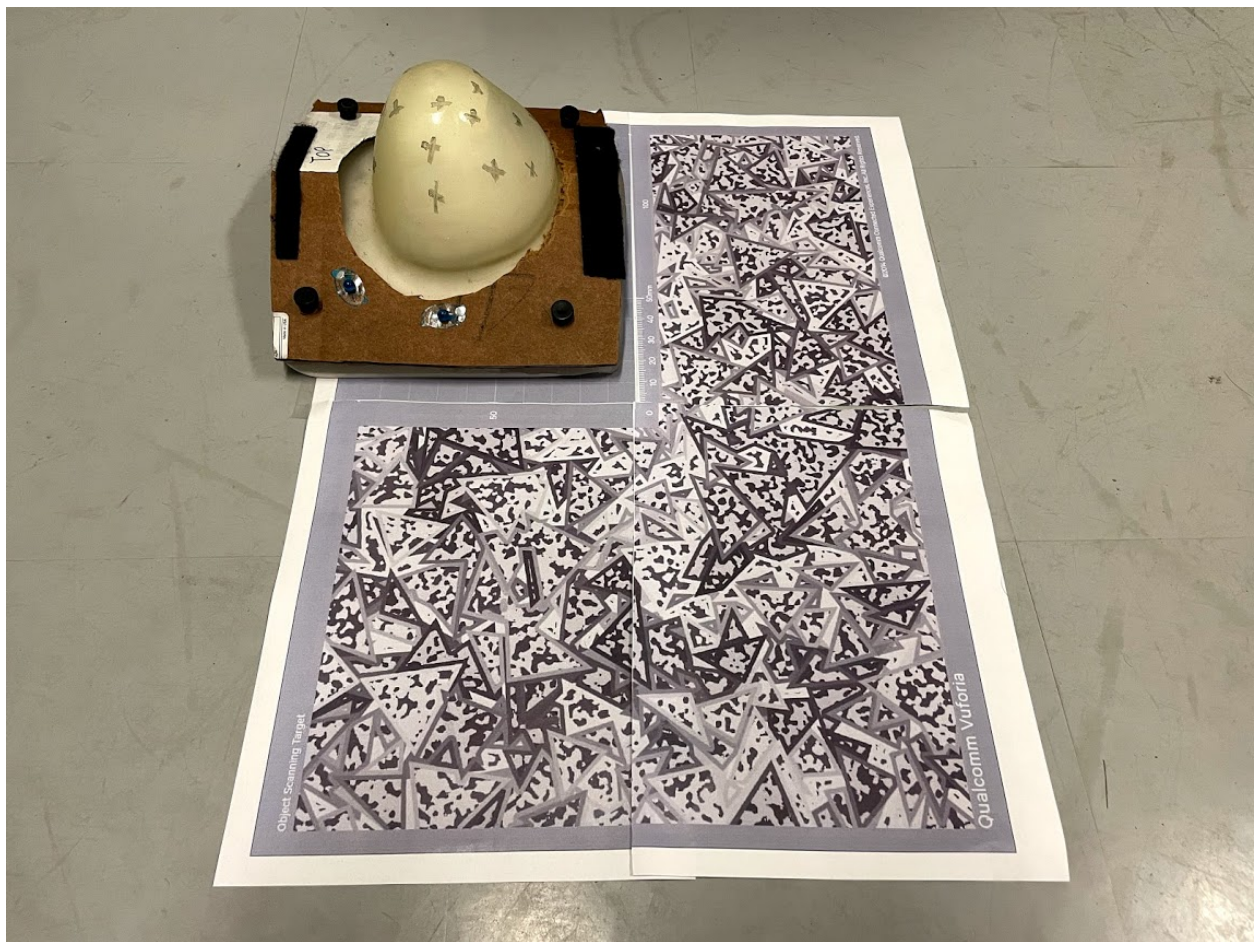


Figure 5.2: Tracking breast with a 3D object.

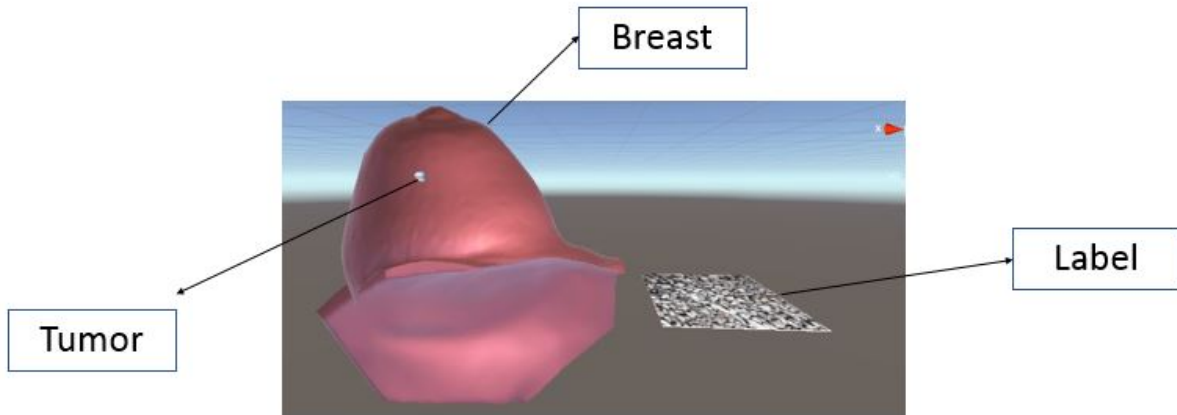


Figure 5.3: Breast placement in Unity.

5.2 Hardware and software architecture

Fig. 5.4 is the architecture for the AR implementation. We kept the NN prediction on a desktop, and the AR was deployed to the Android tablet. The data communication between the two devices is through TCP. The desktop is the server and the tablet is the client. The reason that desktop is the server side is desktop stays in the same place and its IP address does not change. In some AR / VR app, they have a requirement of 60 frames per second (FPS). This is not a requirement for us currently. In fact, one big challenge in AR during packet transfer in networking is the speed. Our goal is not to have the breast deform together with the physical breast in real time. We designed an interface to ask the user to enter gravity direction and the app will deform accordingly.

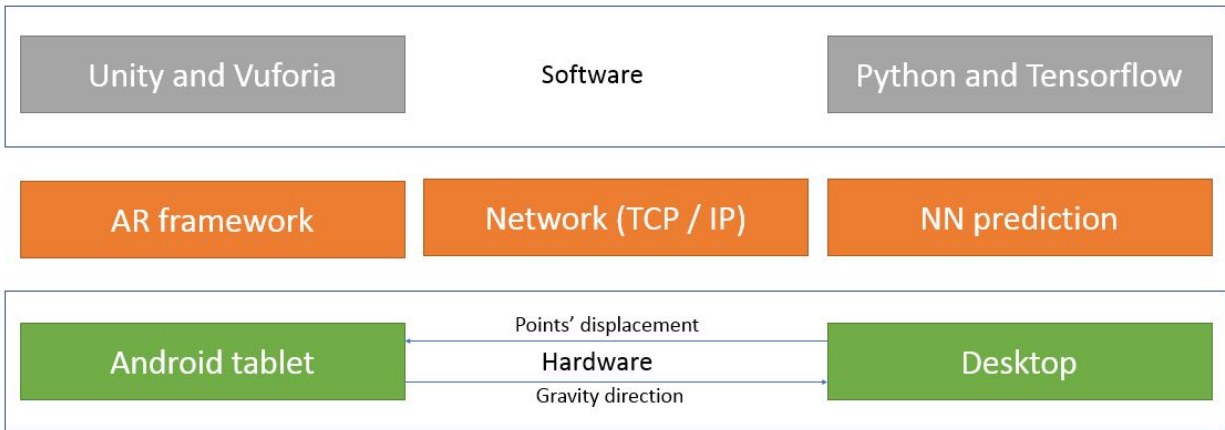


Figure 5.4: AR architecture implementation.

5.3 User interface

Fig. 5.5 is the AR app we developed. The user can change the position of breast using the x, y and z sliders. The “scale value” is used to change the size of the breast. After setting the breast position and scale, the user clicks the “Position Button” to reconfigure the breast. The plane drop down is used to switch between sagittal and lateral plane. There is an angle slider bar that the user can use to set gravity angle, which is measured from vertical direction to current breast position. After selecting an angle, the user can press “Angle button” to submit current breast angle. The breast will be reconfigured with the breast shape after it receives data from the Desktop’s NN model prediction. Fig. 5.6 is the AR app when we include tumor. In order to see the tumor, we changed the transparency of the breast model. Fig. 5.7 is a photo of a user when he is using the AR app. We also uploaded a video to as a demo [78].

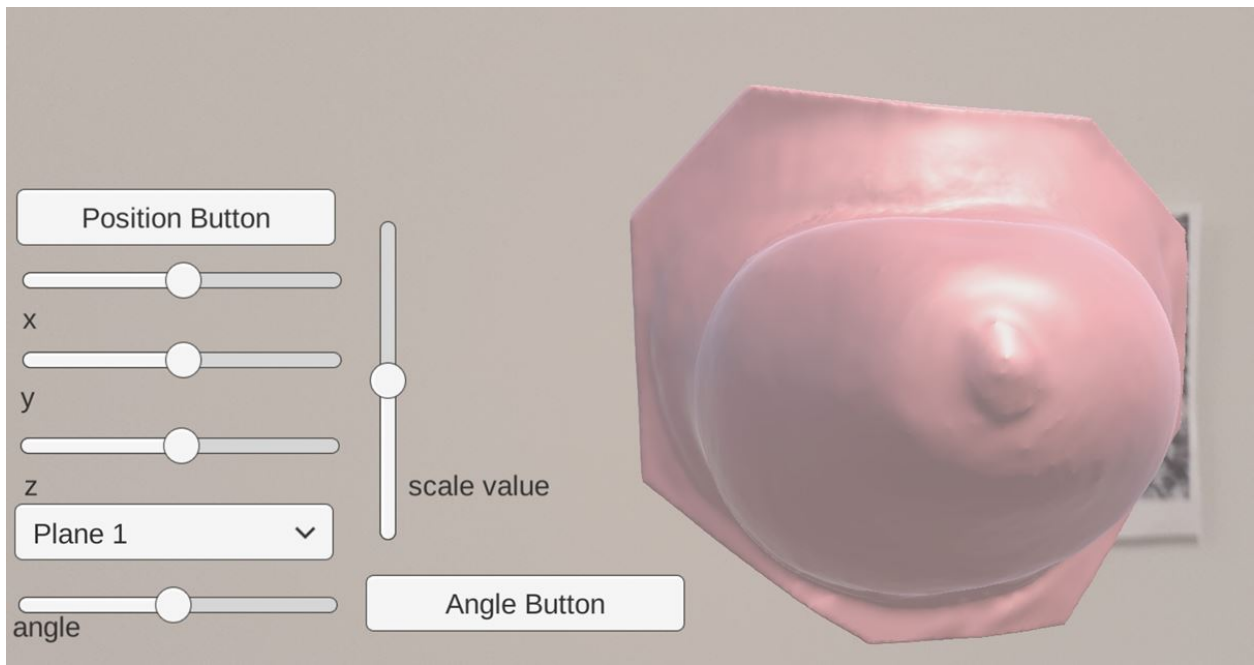


Figure 5.5: AR app interface.

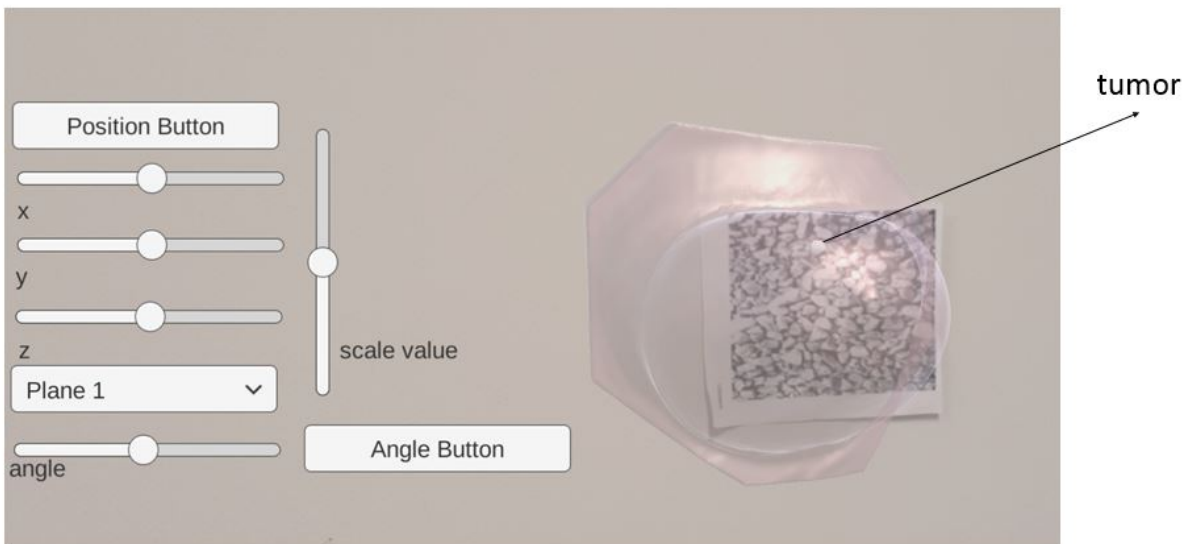


Figure 5.6: Tumor in AR.

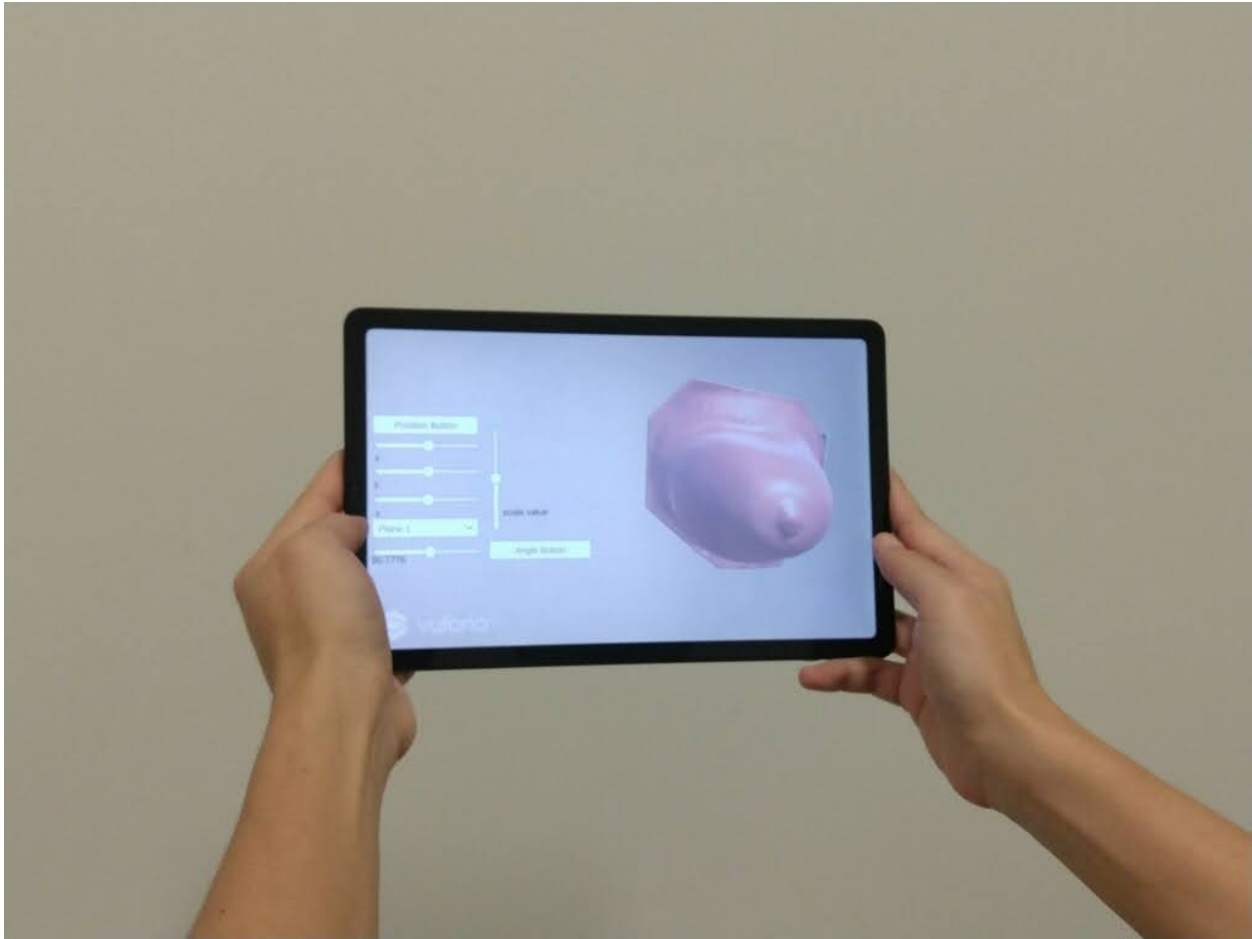


Figure 5.7: A user is using the AR app.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This thesis explored the accuracy of using NN for breast deformation simulation due to gravity. In the first part of the thesis, we used patient-specific breast data and did FEA simulation against different patient's positions and material properties. The FEA simulation results are used as training data to construct two NNs, one to predict breast surface nodal displacement and another to predict the tumor center's displacement. The NN models were used to build a surgical simulator that can predict the breast deformation under different breast material structure and patient's positions. The average point's validation NDE is less than 0.014 mm for breast, and the tumor center NDE is less than 0.05 mm. The errors were smaller compared to some similar research we mentioned before [46] [62]. The prediction time of NN is 5 ms and satisfies real-time requirement. In the second part, we collected data from a breast phantom, which mimics breast deformation accurately. We segmented out the tumor and did a 3D reconstruction. The 3D model was used to generate a tetrahedral mesh, and the mesh was used for FEA simulation. We implemented an iterative optimization approach to find the breast material properties. The average PTP error is 3.63 ± 1.12 mm at 25° . Different scenarios were generated to mimic different patient's positions. We then used the generated scenarios as the ground truth and built two NNs to find the mapping between gravity direction and the nodal displacements or tumor center location. As validation set shows, more than 40% of the points had a displacement of more than 10 mm. That proved the effect of gravity in breast deformation simulation. The NN average PTP error is 0.19 ± 0.12 mm when compared to FEA in the validation set. We overlapped the NN prediction results over the phantom's 3D data points at 25° and 50° and found good matches. Since 25°

was used to calculate error for material property, we used 50° to calculate the error between NN prediction and the 3D MRI constructed model. The error is 2.69 ± 0.7 mm. The tumor center error is 3.65 mm. The error is small and it shows that NN is a promising method for real-time breast deformation prediction in a clinical environment.

There are a few things we plan as the future work of the project. The simulation software need to be improved to perform real-time deformation prediction inside an operating room. We can achieve that by incorporating a motion-capture system to register a patient's movement in real-time and predict breast deformation and tumor location based on sensor data from the patient. What's more, currently, C# does not have a native support for ML, and the communication between Unity and Python programming environments is done through a socket. The I/O bound is the main bottle neck because of the large number of surface points, increasing the total run time to between 300 ms to 500 ms. Other methods of communication should be to be explored to improve the performance. Besides, our approach can be extended to create a generalized NN model that can handle different breast anatomy and shapes.

REFERENCES

- [1] Lam, F., . Cancer today. <http://gco.iarc.fr/today/home>.
- [2] Perdue, Time, . Applications of Augmented Reality. <https://www.lifewire.com/applications-of-augmented-reality-2495561>.
- [3] Adit, D. Deep AI. <https://deepai.org/machine-learning-glossary-and-terms/stride>.
- [4] Holzapfel, A.G., 2000. Nonlinear solid mechanics ii.
- [5] Manual, A.U., 2014. Abaqus theory guide. version 6.14. USA.: Das-sault Systemes Simulia Corp.
- [6] Panieri, E., 2012. Breast cancer screening in developing countries. Best practice & research Clinical obstetrics gynaecology 26, 283–290.
- [7] Sturgeon, G.M., Kiarashi, N., Lo, J.Y., Samei, E., Segars, W., 2016. Finite-element modeling of compression and gravity on a population of breast phantoms for multimodality imaging simulation. Medical physics 43, 2207–2217.
- [8] Martínez-Martínez, F., Rupérez-Moreno, M.J., Martínez-Sober, M., Solves-Llorens, J., Lorente, D., Serrano-López, A., Martínez-Sanchis, S., Monserrat, C., Martín-Guerrero, J.D., 2017. A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time. Computers in biology and medicine 90, 116-124.
- [9] hang, J., Zhong, Y., Gu, C., 2018. Deformable models for surgical simulation: A survey. IEEE reviews in biomedical engineering 11, 143-164.
- [10] Arun, K Somani and Huang, Thomas S and Blostein, Steven D, 1987. Least-squares fitting of two 3-D point sets. IEEE Transactions on pattern analysis and machine intelligence, 698-700
- [11] Pieper, Steve and Halle, Michael and Kikinis, Ron, 2004. 3D Slicer. 2nd IEEE international symposium on biomedical imaging: nano to macro, 632-635
- [12] Si, H., 2015. Tetgen, a delaunay-based quality tetrahedral mesh generator. ACM Transactions on Mathematical Software (TOMS) 41, 1–36.

- [13] Regina Bailey. Anatomical Position: Definitions and Illustrations. <https://www.thoughtco.com/anatomical-position-definitions-illustrations-4175376>
- [14] Council, N.R., et al., 2001. Mammography and beyond: developing technologies for the early detection of breast cancer. National Academies Press.
- [15] Council, N.R., et al., 2001. Mammography and beyond: developing technologies for the early detection of breast cancer. National Academies Press.
- [16] Gittleman, M.A., 2003. Single-step ultrasound localization of breast lesions and lumpectomy procedure. *The American journal of surgery* 186, 386–390.
- [17] Barrett, A., 2015. Preoperative breast MR imaging: its role in surgical planning. *Radiologic technology*, 86(5), pp.499-510.
- [18] Kasemsiri, P., Solares, C.A., Carrau, R.L., Prosser, J.D., Prevedello, D.M., Otto, B.A., Old, M., Kassam, A.B., 2013. Endoscopic endonasal transpterygoid approaches: anatomical landmarks for planning the surgical corridor. *The Laryngoscope* 123, 811–815.
- [19] Eiben, B., Vavourakis, V., Hipwell, J.H., Kabus, S., Buelow, T., Lorenz, C., Mertzaniidou, T., Reis, S., Williams, N.R., Keshtgar, M. and Hawkes, D.J., 2016. Symmetric biomechanically guided prone-to-supine breast image registration. *Annals of biomedical engineering*, 44(1), pp.154-173.
- [20] Zhang, J., Zhong, Y. and Gu, C., 2017. Energy balance method for modelling of soft tissue deformation. *Computer-Aided Design*, 93, pp.15-25.
- [21] Patete, P., Iacono, M.I., Spadea, M.F., Trecate, G., Vergnaghi, D., Mainardi, L.T., Baroni, G., 2013. A multi-tissue mass-spring model for computer assisted breast surgery. *Medical engineering & physics* 35, 47–53.
- [22] Azar, F.S., Metaxas, D.N., Schnall, M.D., 2000. A finite element model of the breast for predicting mechanical deformations during biopsy procedures, in: *Mathematical Methods in Biomedical Image Analysis*, 2000. Proceedings. IEEE Workshop on, IEEE. pp. 38–45.
- [23] Comas, Olivier, et al. "Efficient nonlinear FEM for soft tissue modelling and its GPU implementation within the open source framework SOFA." *International Symposium on Biomedical Simulation*. Springer, Berlin, Heidelberg, 2008.
- [24] Taylor, Z.A., Cheng, M., Ourselin, S., 2008. High-speed nonlinear finite element analysis for surgical simulation using graphics processing units. *IEEE transactions on medical imaging* 27, 650–663.
- [25] Joldes, G.R., Wittek, A., Miller, K., 2010. Real-time nonlinear finite element computations on gpu—application to neurosurgical simulation. *Computer methods in applied mechanics and engineering* 199, 3305–3314.

- [26] Nakao, M., Minato, K., 2010. Physics-based interactive volume manipulation for sharing surgical process. *IEEE Transactions on Information Technology in Biomedicine* 14, 809–816.
- [27] Courtecuissea, H., Kerfridena, P., Duriezb, C., Allardb, J., Bordasa, S.P., Cotinb, S., 2013. Real-time simulation of surgical cutting in biological tissues using a semi-implicit time integration scheme. *Methodology* 12, 17.
- [28] Bro-Nielsen, M., Cotin, S., 1996. Real-time volumetric deformable models for surgery simulation using finite elements and condensation, in: *Computer graphics forum*, Wiley Online Library. pp. 57–66.
- [29] Cai, Y., Chen, L., Yu, W., Zhou, J., Wan, F., Suh, M., Chow, D.H.k., 2018. A piecewise mass-spring-damper model of the human breast. *Journal of biomechanics* 67, 137–143.
- [30] Patete, P., Iacono, M.I., Spadea, M.F., Trecate, G., Vergnaghi, D., Mainardi, L.T., Baroni, G., 2013. A multi-tissue mass-spring model for computer assisted breast surgery. *Medical engineering physics* 35, 47–53.
- [31] Roose, L., De Maerteleire, W., Mollemans, W. and Suetens, P., 2005, May. Validation of different soft tissue simulation methods for breast augmentation. In *International congress series* (Vol. 1281, pp. 485-490). Elsevier.
- [32] Xu, L., Lu, Y., Liu, Q., 2018. Integrating viscoelastic mass spring dampers into position-based dynamics to simulate soft tissue deformation in real time. *Royal Society open science* 5, 171587.
- [33] Azar, F.S., Metaxas, D.N., Schnall, M.D., 2001. A deformable finite element model of the breast for predicting mechanical deformations under external perturbations. *Academic Radiology* 8, 965–975.
- [34] Del Palomar, A.P., Calvo, B., Herrero, J., López, J., Doblaré, M., 2008. A finite element model to accurately predict real deformations of the breast. *Medical engineering physics* 30, 1089–1097.
- [35] Rajagopal, V., Nielsen, P., Nash, M., 2004. Development of a three-dimensional finite element model of breast mechanics, in: *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE. pp. 5080–5083.
- [36] Tanner, C., Schnabel, J.A., Hill, D.L., Hawkes, D.J., Leach, M.O., Hose, D.R., 2006. Factors influencing the accuracy of biomechanical breast models. *Medical physics* 33, 1758–1769.
- [37] Chung, J.H., Rajagopal, V., Nielsen, P.M., Nash, M.P., 2008. Modelling mammographic compression of the breast, in: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer. pp. 758–765.

- [38] Tagliabue, E., Dall’Alba, D., Magnabosco, E., Peterlik, I., Fiorini, P., 2020. Biomechanical modelling of probe to tissue interaction during ultrasound scanning. *International Journal of Computer Assisted Radiology and Surgery* .
- [39] Míra, A., Carton, A.K., Muller, S., Payan, Y., 2018. A biomechanical breast model evaluated with respect to mri data collected in three different positions. *Clinical Biomechanics* 60, 191–199.
- [40] Parker, K.J., Doyley, M.M., Rubens, D.J., 2010. Imaging the elastic properties of tissue: the 20 year perspective. *Physics in medicine & biology* 56, R1.
- [41] Doyley, M.M., 2012. Model-based elastography: a survey of approaches to the inverse elasticity problem. *Physics in Medicine & Biology* 57, R35.
- [42] Bakic, P., 2001. Breast tissue description and modeling in mammography.
- [43] Visentin, F., Groenhuis, V., Maris, B., Dall’Alba, D., Siepel, F., Stramigioli, S., Fiorini, P., 2019. Iterative simulations to estimate the elastic properties from a series of mri images followed by mri-us validation. *Medical biological engineering computing* 57, 913–924.
- [44] Baghban, M., Mojra, A., 2018. Early relaxation time assessment for characterization of breast tissue and diagnosis of breast tumors. *Journal of the mechanical behavior of biomedical materials* 87, 325–335.
- [45] Solves Llorens, J.A., Rupérez, M., Monserrat, C., Feliu, E., García, M., Lloret, M., 2012. Segmentation of the breast skin and its influence in the simulation of the breast compression during an x-ray mammography. *The Scientific World Journal* 2012.
- [46] Tonutti, M., Gras, G., Yang, G.Z., 2017. A machine learning approach for real-time modelling of tissue deformation in image-guided neurosurgery. *Artificial intelligence in medicine* 80, 39–47.
- [47] Chung, J.H., Rajagopal, V., Nielsen, P.M., Nash, M.P., 2005. Computational modeling of the breast during mammography for tumor tracking, in: *Medical Imaging 2005: Physiology, Function, and Structure from Medical Images*, International Society for Optics and Photonics. pp. 817–824.
- [48] Hipwell, J.H., Vavourakis, V., Han, L., Mertzanidou, T., Eiben, B., Hawkes, D.J., 2016. A review of biomechanically informed breast image registration. *Physics in Medicine Biology* 61, R1.
- [49] Zhu, B. and Gu, L., 2012. A hybrid deformable model for real-time surgical simulation. *Computerized Medical Imaging and Graphics*, 36(5), pp.356-365.
- [50] Özgören, Y.Ö., Çetinkaya, S., Sarıdemir, S., Çiçek, A., Kara, F., 2013. Artificial neural network based modelling of performance of a beta-type stirling engine. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering* 227, 166–177.

- [51] Erkan, Ö., Işık, B., Çiçek, A., Kara, F., 2013. Prediction of damage factor in end milling of glass fibre reinforced plastic composites using artificial neural network. *Applied Composite Materials* 20, 517–536.
- [52] Eser, A., Aşkar Ayyıldız, E., Ayyıldız, M., Kara, F., 2021. Artificial intelligence-based surface roughness estimation modelling for milling of aa6061 alloy. *Advances in Materials Science and Engineering* 2021.
- [53] Zadpoor, A.A., Campoli, G., Weinans, H., 2013. Neural network prediction of load from the morphology of trabecular bone. *Applied Mathematical Modelling* 37, 5260–5276.
- [54] Mouloudi, S., Rahmanpanah, H., Burvill, C., Gohari, S., Davies, H.M.S., 2020b. Experimental, regression learner, numerical, and artificial neural network analyses on a complex composite structure subjected to compression loading. *Mechanics of Advanced Materials and Structures* , 1–17.
- [55] Mouloudi, S., Rahmanpanah, H., Burvill, C., Davies, H.M., 2020a. Prediction of displacement in the equine third metacarpal bone using a neural network prediction algorithm. *Biocybernetics and Biomedical Engineering* 40, 849–863.
- [56] Taylor, M., Perilli, E., Martelli, S., 2017. Development of a surrogate model based on patient weight, bone mass and geometry to predict femoral neck strains and fracture loads. *Journal of biomechanics* 55,121–127.
- [57] Deo, D., De, S., 2009. Phyness: A physics-driven neural networks-based surgery simulation system with force feedback, in: *World Haptics 2009-Third Joint Euro Haptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, IEEE. pp. 30–34.
- [58] Jahya, A., Herink, M., Misra, S., 2013. A framework for predicting three-dimensional prostate deformation in real time. *The International Journal of Medical Robotics and Computer Assisted Surgery* 9, e52–e60.
- [59] Mendizabal, A., Tagliabue, E., Brunet, J.N., Dall’Alba, D., Fiorini,P., Cotin, S., 2019. Physics-based deep neural network for real-time lesion tracking in ultrasound-guided breast biopsy, in: *Computational Biomechanics for Medicine*. Springer, pp. 33–45.
- [60] De, S., Deo, D., Sankaranarayanan, G., Arikatla, V.S., 2011. Aphysics-driven neural networks-based simulation system (phyness) for multimodal interactive virtual environments involving nonlinear deformable objects. *Presence: Teleoperators and Virtual Environments* 20, 289–308.
- [61] Morooka, K., Chen, X., Kurazume, R., Uchida, S., Hara, K., Iwashita,Y., Hashizume, M., 2008. Real-time nonlinear fem with neural network for simulating soft organ model deformation, in: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer. pp. 742–749.

- [62] Zolfagharnasab, H., Bessa, S., Oliveira, S., Faria, P., Teixeira, J., Car-doso, J., Oliveira, H., 2018. A regression model for predicting shape deformation after breast conserving surgery. *Sensors* 18, 167.
- [63] Lorente, D., Martínez-Martínez, F., Rupérez, M.J., Lago, M., Martínez-Sober, M., Escandell-Montero, P., Martínez-Martínez, J.M., Martínez-Sanchis, S., Serrano-López, A.J., Monserrat, C., et al., 2017. A framework for modelling the biomechanical behaviour of the human liver during breathing in real time using machine learning. *Expert Systems with Applications* 71, 342–357.
- [64] Widmer, A., Hu, Y., 2010. Statistical comparison between a real-time model and a fem counterpart for visualization of breast phantom deformation during palpation, in: *CCECE 2010*, IEEE. pp. 1–4.
- [65] Figueiredo, A.A.A., do Nascimento, J.G., da Silva Ignacio, L.H., Malheiros, F.C. and Guimarães, G., COBEM-2017-0602 NUMERICAL ANALYSIS OF HEAT TRANSFER IN A BREAST CONSIDERING TWO CANCER TYPES.
- [66] Omer, M., Fear, E., 2018. Anthropomorphic breast model repository for research and development of microwave breast imaging technologies. *Scientific data* 5, 180257.
- [67] Carter, T.J., Tanner, C., Hawkes, D.J., 2015. A comparison of linear and quadratic tetrahedral finite elements for image-guided surgery applications. *Medical Images and Signals IRC* , 47.
- [68] Han, L., Hipwell, J.H., Tanner, C., Taylor, Z., Mertzaniidou, T., Car-doso, J., Ourselin, S., Hawkes, D.J., 2011. Development of patient-specific biomechanical models for predicting large breast deformation. *Physics in Medicine Biology* 57, 455.
- [69] Akladios, C., Gabriele, V., Agnus, V., Martel-Billard, C., Saadeh, R., Garbin, O., Lecointre, L. and Marescaux, J., 2020. Augmented reality in gynecologic laparoscopic surgery: Development, evaluation of accuracy and clinical relevance of a device useful to identify ureters during surgery. *Surgical endoscopy*, 34(3), pp.1077-1087.
- [70] Schneider, M., Kunz, C., Pal'a, A., Wirtz, C.R., Mathis-Ullrich, F. and Hlaváč, M., 2021. Augmented reality–assisted ventriculostomy. *Neurosurgical Focus*, 50(1), p.E16.
- [71] Frenk J, Chen L, Bhutta ZA, Cohen J, Crisp N, Evans T, Fineberg H, Garcia P, Ke Y, Kelley P, Kistnasamy B, Meleis A, Naylor D, Pablos-Mendez A, Reddy S, Scrimshaw S, Sepulveda J, Serwadda D, Zurayk H. 2010. Health professionals for a new: transforming education to strengthen health systems in an interdependent world. *Lancet* 376(9756):1923-1958
- [72] Liang, X., Boppart, S.A., 2009. Biomechanical properties of in vivo human skin from dynamic optical coherence elastography. *IEEE Transactions on Biomedical Engineering* 57, 953–959.

- [73] Pathmanathan, P., Gavaghan, D.J., Whiteley, J.P., Chapman, S.J., Brady, J.M., 2008. Predicting tumor location by modeling the deformation of the breast. *IEEE Transactions on Biomedical Engineering* 55, 2471–2480.
- [74] Hamza-Lup, F.G., Rolland, J.P. and Hughes, C., 2018. A distributed augmented reality system for medical training and simulation. *arXiv preprint arXiv:1811.12815*.
- [75] Breast Type, 2019. Dense breasts: Answers to commonly asked questions. <https://www.cancer.gov/types/breast/breast-changes/dense-breasts>. Accessed: 2019-12-17.
- [76] Naser, M.A., Sayed, A.M., Wahba, A.A., Eldosoky, M.A., 2018. Modeling procedures for breast cancer diagnosis based on clinicaelastography images, in: 2018 13th International Conference on Computer Engineering and Systems (ICCES), IEEE. pp. 671–677.
- [77] Wang.K., 2021. *Breast deformation demo*. 11, January, 2020. Available at: <https://www.youtube.com/watch?v=PrAQMhqZtWM> (Accessed: 22, November.)
- [78] Wang.K.,2021. *Breast deformation against gravity simulation in AR*. 16, June. Available at: <https://youtu.be/tfnijMcjdYQ> (Accessed: 13 October 2021).