

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: nucleaire
technologie

Masterthesis

Realization of a framework for large scale virtual clinical studies in breast imaging

Ferre Vanmuysen

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: nucleaire technologie, afstudeerrichting nucleair en medisch

PROMOTOR :

Prof. dr. Brigitte RENIERS

PROMOTOR :

Prof. ir. Hilde BOSMANS

BEGELEIDER :

ing. Katrien HOUBRECHTS

Gezamenlijke opleiding Universiteit Hasselt en KU Leuven



Universiteit Hasselt | Campus Diepenbeek | Faculteit Industriële Ingenieurswetenschappen | Agoralaan Gebouw H - Gebouw B | BE 3590 Diepenbeek

Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE 3590 Diepenbeek
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE 3500 Hasselt



2022
2023

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: nucleaire
technologie

Masterthesis

Realization of a framework for large scale virtual clinical studies in breast imaging

Ferre Vanmuysen

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: nucleaire technologie,
afstudeerrichting nucleair en medisch

PROMOTOR :

Prof. dr. Brigitte RENIERS

PROMOTOR :

Prof. ir. Hilde BOSMANS

BEGELEIDER :

ing. Katrien HOUBRECHTS



KU LEUVEN

Preface

To bring my research to fruition, I was able to count on the help and support of a number of people. First and foremost, I would like to thank my supervisor Ing. Katrien Houbrechts for every moment of time she has freed up to guide me through this thesis. Her advice and feedback were crucial in making this study a reality. I would like to mention my external promotor Prof. Ir. Hilde Bosmans for letting me do my master's thesis at her research group in UZ Leuven. I would also like to thank Prof. Dr. Brigitte Reniers for being my internal promotor and giving me the opportunity to conduct my master's thesis at the U Hasselt. A special thank you goes to my family and friends for their support and patience during this research process. Last, I would like to thank the jury for reading this thesis.

Table of contents

Preface.....	1
Table of contents.....	3
List of tables.....	5
List of figures.....	7
List of abbreviations.....	9
Abstract.....	11
Abstract in het Nederlands.....	13
1 Introduction.....	15
1.1 Mammography.....	15
1.1.1 Basic principles.....	15
1.2 Virtual clinical trials.....	17
1.2.1 Hybrid simulation framework (UZ Leuven).....	17
1.3 Aim of master's thesis.....	19
2 Materials and methods.....	21
2.1 Converting the MATLAB framework to the Python environment.....	21
2.2 Validation of the Python tool.....	24
2.2.1 Raytracing.....	24
2.2.3 Modulation transfer function.....	24
2.2.3 Aluminum sheet.....	25
2.2.4 Aluminum sphere.....	26
2.3 Application: Clinical cases.....	27
2.3.1 Different breast density types.....	27
2.3.2 Spiculated and non-spiculated lesion.....	28
3 Results.....	31
3.1 Converting the MATLAB framework to the Python environment.....	31
3.2 Validation of the Python tool.....	32
3.2.1 Raytracing.....	32
3.2.2 Modulation transfer function.....	33
3.2.3 Aluminum sheet.....	34
3.2.4 Aluminum sphere.....	38
3.3 Application: clinical cases.....	39

3.3.1 Different breast density types.....	39
3.3.2 Spiculated and non-spiculated lesion.....	41
4 Discussion.....	43
4.1 Converting the MATLAB framework to the Python environment	43
4.2 Validation of the Python tool.....	43
4.3 Application: clinical cases.....	44
4.4 Future development	45
5 Conclusion	47
Bibliography	49

List of tables

Table 1: Comparison of MATLAB tool and Python tool when simulating a 2 mm sphere....	32
Table 2: Error on MTF from validation.....	33
Table 3: Signal, Peak contrast and SDNR ratios of the Python simulation and the real small aluminum sphere	38
Table 4: Signal, Peak contrast and SDNR ratios of the MATLAB and Python simulations of a small aluminum sphere	38

List of figures

Figure 1: digital mammography system. 1 X-ray tube, 2 X-ray beam, 3 compression paddle, 4 breast support, 5 detector, 6 C-arm, 7 monitor for angle, breast thickness, and compression force [6].....	15
Figure 2: Difference between CC (left) and MLO (right) view [12]	16
Figure 3: DBT acquisition - The breast is held compressed against the stationary support table, and a sequence of small exposures is made as the tube gantry moves along an arc [6]	16
Figure 4: Pipeline of the hybrid simulation framework [24].....	18
Figure 5: 3D matrix depiction of a small sphere.....	21
Figure 6: Example of the value matrix of a template	22
Figure 7: Example of a template of a sphere (2 mm) at an angle of 0° (left) and at an angle of 25° (right).....	22
Figure 8: Pixels of an image (left) can be considered as the intersection areas of equally spaced lines (right) [25].....	23
Figure 9: To validate the MTF, the simulated edge is inserted in the real test object (white) vertically (left) and horizontally (right).....	25
Figure 10: ROI location for measuring the mean pixel value of the aluminum sheet.....	26
Figure 11: different breast densities [28].....	27
Figure 12: The central slice of the voxel model of a spiculated mass (left) and a non-spiculated mass (right) used in the simulations.	28
Figure 13: Results of raytracing validation	32
Figure 14: Results of the MTF validation in 2D mode.....	33
Figure 15: Results of the MTF validation in DBT mode.....	34
Figure 16: SDNR and peak contrast ratios of simulated and real aluminum sheet in 2D mode	35
Figure 17: SDNR and peak contrast ratios of simulated and real aluminum sheet in DBT mode	35
Figure 18: Output image of the 2D simulation with a PMMA thickness of 40 mm and a tube voltage of 28 kV	36
Figure 19: SDNR and peak contrast ratios of python and MATLAB simulations of a aluminum sheet in 2D mode.....	36
Figure 20: SDNR and peak contrast ratios of python and MATLAB simulations of a aluminum sheet in DBT mode.....	37
Figure 21: Profile through the aluminum sheet along the x-axis of the MATLAB and Python simulation.....	37
Figure 22: Central projection of the aluminum sheet simulation in DBT mode (40 mm PPMA, 28 kV) of MATLAB (left) and Python (right).....	38
Figure 23: Profile in the z-direction of the Python and MATLAB simulation of a small aluminum sphere	39

Figure 24: Projection of the small aluminum sphere generated from the simulations from the MATLAB tool (left) and the Python tool (right)	39
Figure 25: Reconstruction of the simulation of type A breast with different densities of the lesion	40
Figure 26: Reconstruction of the simulation of type B breast with different densities of the lesion	40
Figure 27: Reconstruction of the simulation of type C breast with different densities of the lesion	41
Figure 28: Reconstruction of the simulation of type D breast with different densities of the lesion	41
Figure 29: Reconstructed image of the clinical case with the spiculated lesion	42
Figure 30: Reconstructed image of the clinical case with the non-spiculated lesion	42

List of abbreviations

DM	Digital mammography
CC	Craniocaudal view
MLO	Mediolateral oblique view
DBT	Digital breast tomosynthesis
VCT	Virtual clinical trials
GUI	Graphical user interface
MTF	Modulation transfer functions
SPR	Scatter-to-primary ratio
PMMA	Polymethyl methacrylate
SDNR	Signal difference to noise ratio
ROI	Region of interest
ASF	Artifact spread function

Abstract

The Medical Physics & Quality Control research group at UZ Leuven has been using a tool created in MATLAB to simulate breast imaging for eleven years. However, this tool is not suitable to be used on a large scale, making the simulation times for large amounts of simulations longer than necessary. In addition, the user interface is not user-friendly. This master thesis aims to develop a new simulation tool with a system in place to easier perform simulations on a larger scale and make it more user-friendly.

The new simulation tool was coded in the Python environment Spyder. Validation of the tool was done by comparing the results with those of the MATLAB tool and real images. This involves comparing various parameters such as *peak contrast*, *SDNR* and *ASF*. The Tool is also tested for its ability to be applied for clinical research by simulating clinical cases.

The resulting code contains the full functionality of the MATLAB tool. The code is also able to more easily perform simulations on a large scale. The validation also shows that Python and MATLAB simulations match well with exception to certain cases and that the tool can be used for simulating clinical cases.

Abstract in het Nederlands

De onderzoeksgroep Medische Fysica & Kwaliteitscontrole in het UZ Leuven gebruikt al elf jaar een tool gemaakt in MATLAB voor het simuleren van borstbeeldvorming. Deze tool is echter niet geschikt om op grote schaal gebruikt te worden waardoor de simulatietijden voor grote hoeveelheden simulaties langer zijn dan nodig. Bovendien is de *user interface* niet gebruiksvriendelijk. Deze masterproef heeft als doel een nieuwe simulatie tool te ontwikkelen en ervoor te zorgen dat het makkelijker is om simulaties op grotere schaal uit te voeren.

De nieuwe simulatietool werd gecodeerd in de Pythonomgeving Spyder. Validatie van de tool werd gedaan door de resultaten te vergelijken met de resultaten van de MATLAB-tool en echte beelden. Hierbij worden verschillende parameters vergeleken zoals *peak contrast*, *SDNR* en *ASF*. De Tool wordt ook getest op zijn vermogen om toegepast te worden voor klinische onderzoek door klinische *cases* te simuleren.

De resulterende code bevat de volledige functionaliteit van de MATLAB-code. De code is ook in staat om makkelijker simulaties uit te voeren op grote schaal. De validatie laat ook blijken dat de simulaties van Python en MATLAB goed overeenkomen met uitzondering op bepaalde gevallen en dat de tool kan gebruikt worden voor het simuleren van klinische *cases*.

1 Introduction

1.1 Mammography

Breast cancer is a common form of cancer that mainly affects women. Early detection is key to improve patients' prognoses and reducing mortality rates. Because of this, a national breast cancer screening program has been implemented in Belgium. As part of this program, women aged 50-69 are invited to receive a free mammogram every two years [1], [2]. Mammography can detect breast cancer in its early stages, when it is most treatable. This can lead to better outcomes for patients and can help reduce mortality rates. However, the use of ionizing radiation in mammography can slightly increase the risk of cancer, although the risk is very small and outweighs the benefits of better treatment outcome associated with early detection [3]–[5]. To ensure that the benefits far outweigh the harms, screening techniques need to be optimized continuously. The quality and effectiveness of the imaging techniques used in the screening process are strictly monitored to ensure that they are as safe and accurate as possible [6], [7]. For women who have symptoms or an elevated risk of breast cancer, the standard screening process may be adapted on the advice of a healthcare provider or in accordance with local guidelines [1], [2].

1.1.1 Basic principles

A mammography unit is a machine specialized in the imaging of breasts. It consists of an X-ray tube for the generation of X-rays and a table where the detector is situated as seen in figure 1. When doing a mammography, the patient's breast is placed on the table. On this table a compression device parallel to the detector is in place to compress the patient's breast. The compression is done to improve image quality. It reduces patient motion during the imaging which otherwise would cause motion blur. Also, it improves the homogeneity of the breast which improves the amount of scattering and geometric unsharpness. Lastly this reduces the overall attenuation of the incident X-ray beam. When the breast is in place the X-ray tube generates a beam that passes through the compressed breast and hits the detector, producing an image [8], [9].

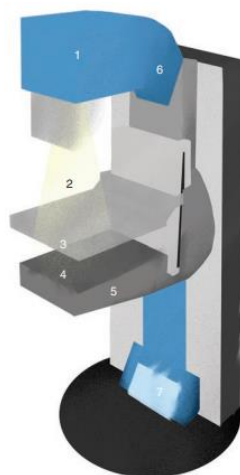


Figure 1: Digital mammography system. 1 X-ray tube, 2 X-ray beam, 3 compression paddle, 4 breast support, 5 detector, 6 C-arm, 7 monitor for angle, breast thickness, and compression force [7]

The conventional way of doing a screening is 2D digital mammography (DM). This technique involves the acquisition of two projection images of each breast on an electronic image detector. The first image is acquired in craniocaudal view (CC) at an angle of 0° , the second one is acquired in mediolateral oblique view (MLO) at an angle of 45° . This is done to increase the detection of abnormalities that are overlapping [10]–[13].

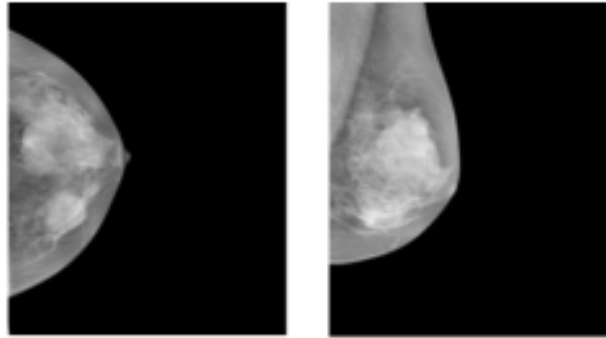


Figure 2: Difference between CC (left) and MLO (right) view [14]

When doing a 2D mammography of a 3D object the outcome is still prone to certain lesions not being detected due to overlapping tissue and can introduce false positives leading to reduced cancer detection, especially in dense breasts. To try to overcome this insensitivity of DM another screening technique can be used: digital breast tomosynthesis (DBT). DBT is a type of mammography that uses digital technology to create pseudo three-dimensional images of breast tissue. The procedure involves the acquisition of multiple low dose 2D X-ray projection images of the breast as the X-ray tube pivots in a limited angular range in a plane aligned with the chest wall. The result of DBT are projections, these projections can be reconstructed with specific software resulting in a cross section of the breast in slices. The impact of overlapping tissue is decreased with this technique as can be seen on Figure 3 [13], [15], [16].

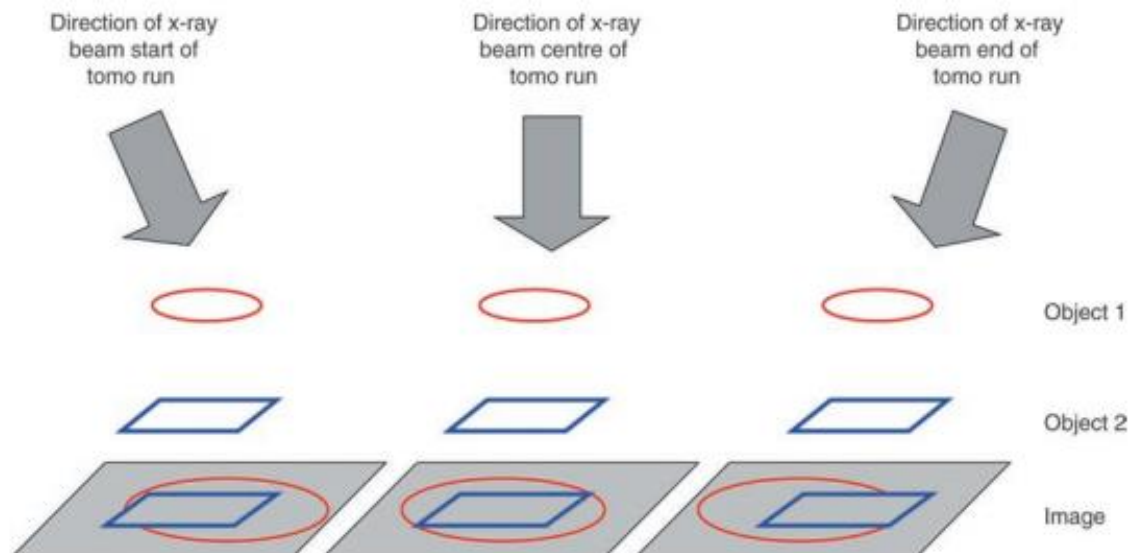


Figure 3: DBT acquisition - The breast is held compressed against the stationary support table, and a sequence of small exposures is made as the tube gantry moves along an arc

[7]

During a mammogram, ionizing radiation is used. Therefore, it is important to ensure the quality of these interventions. An important aspect is the dose administered to the patient. A patient should not receive an unnecessary, but also not too little dose to ensure that the produced images are accurate and reliable in order for the radiologist to make a correct diagnosis. To optimize imaging technique, clinical trials are needed to identify the quality of the image quality and the influence of various scan parameters of the mammography device on the image [17]–[19].

1.2 Virtual clinical trials

Clinical trials involving human subjects are the most common way to evaluate the effectiveness of new technologies or techniques, including diagnostic imaging techniques. However, traditional clinical trials can be time-consuming, often taking several years to complete. Additionally, they require large investments in clinical resources, participant recruitment, and compliance. They may be undesirable in the context of breast cancer screening, where the participants are largely healthy individuals. There are also ethical considerations when looking at the increased exposure to radiation that trial participants may experience. To address these challenges, the field has been moving towards the adoption of virtual clinical trials to complement traditional clinical trials [20]–[22].

Over the past ten years, significant progress has been made in the use of computer simulations to design and improve medical imaging devices. These simulations, which are created using specialized software, allow researchers to virtually test and evaluate the performance of an existing or proposed device. This process, known as *in silico* testing, can be a valuable tool for improving the quality of medical imaging technologies. This is what is called virtual clinical trials (VCT) [23]. When using virtual clinical trials, the step of including patients and administering the physical tests are no longer needed so time can be saved. Also, ethics will not be taken into account as there are no real patient involved [24].

An example of a virtual clinical trial is described in [25]. In this research a framework to simulate mammography called VICTRE is used to evaluate the replacement of DM with DBT for the detection of lesions so that guidelines can be established on the use of which imaging modality in screening. This study was a computational replica of a real clinical study, to illustrate that virtual clinical trials, when implemented in sufficient detail, can lead to the same conclusions as studies with patients.

1.2.1 Hybrid simulation framework (UZ Leuven)

When looking at simulation frameworks for mammography, broadly speaking, we can distinguish 2 approaches. The first approach uses a computational breast phantom where a computation lesion can be inserted. Next, this breast phantom with lesion is imaged taking into account all the physics processes that take place in a mammography unit, to obtain a simulated cancer case.

The second approach is the hybrid framework. This framework simulates an object or lesion and inserts this in an existing breast image instead of using a virtual breast phantom.

In 2011 the research group Medical Physics & Quality Control at the UZ Leuven developed a hybrid simulation framework in MATLAB. This tool allows to simulate a lesion and next, to

insert this in an image of a real patient in 2D or DBT mode in three steps. Figure 4 shows a flowchart of the different steps.

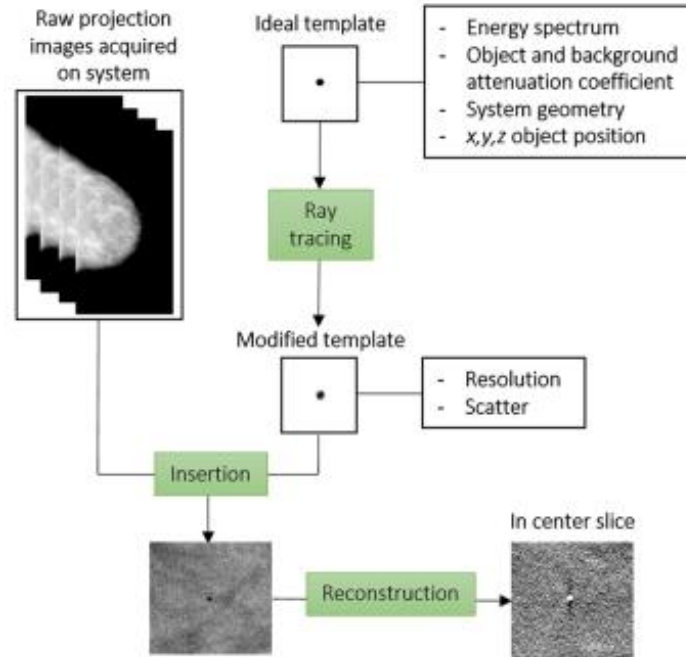


Figure 4: Flowchart of the hybrid simulation framework [26]

The first step is the ideal template generation. In this step a 2D matrix is made in which the values, ranging from 0 to 1, depict how much the X-ray bundle has been attenuated at a specific location in the matrix due to the presence of a 3D object. The user defines which 3D object to simulate. This can be a geometric shape like a sphere or cube with user defined dimensions or a custom voxel model of a lesion [26].

The second step is to modify the ideal template to generate a more realistic projection because the generated template is not corrected for resolution, noise, and scatter. The loss of image quality is unique for each imaging system. These losses are quantified by measurements done for specific systems [26].

The last step is to insert the modified template of the 3D object into the real breast image. This is done by multiplying the template with the pixel values of the image on the spot where the object needs to be inserted. The result is one or multiple projections when we want to simulate DM or DBT, respectively. When creating DBT simulations, these projections need to be reconstructed with another tool to get a sliced view [26].

The tool made in 2011 is still used to this day by the research group Medical Physics & Quality Control at UZ Leuven. However, this framework has a few shortcomings. The first one is the usability of the graphical user interface (GUI). The GUI is not user-friendly because every individual simulation requires clicking multiple menus, even when doing the exact same simulation twice. Moreover, there are no error messages implemented when the tool crashes making it very hard for the user to know what is wrong. The third one is the structure of the code. The different functions are all spread across 32 files.

This in combination with vague and incomplete documentation makes it very difficult to change the code yourself or implement new features.

The fourth shortcoming is the use of MATLAB 2011. Every user of the simulation tool must have a paying MATLAB license. Besides, the 2011 version is not available anymore by normal means making the tool not easily accessible.

The last and most important one is the inability to easily perform multiple simulations. As mentioned before the GUI is very click intensive which makes doing multiple simulations very time consuming. There is also no system in place for doing multiple simulations meaning that in between every simulation there is a need for a user input. All this together means that the tool cannot be used for virtual clinical trials on a large scale.

1.3 Aim of master's thesis

In this work, the aim is to code a new framework that solves the problems mentioned in the previous paragraph meaning that the following functionalities need to be implemented. The tool can perform multiple simulations without the input from the user in between two simulation. It needs to contain the full functionality of the existing MATLAB tool meaning the same simulations can be done in both tools with the same results. Another aim is to validate the tool. This is done by comparing different parameters from the new tool to the MATLAB tool and real images. The goal here is to check if the results of the newly made tool are accurate.

2 Materials and methods

2.1 Converting the MATLAB framework to the Python environment

The first step before converting the code was to become familiar with the MATLAB tool. This involved running multiple simulations to understand how the tool works. These simulations made it clear what functionality the new code should include and highlighted any shortcomings the code has in terms of user interface and simulation time. After gaining an understanding of the tool, it was time to delve deeper into the written code. A flowchart was provided as an overview of the different functions in the MATLAB code. Each part of the code was thoroughly studied and provided with additional notes and documentation.

Next is the conversion of the code to Python. For this the Python environment Spyder is used because it is a free, open-source platform and has long term support. The conversion was done by systematically programming each step of the pipeline and translating this to Python syntax. It is important to maintain the complete functionality of the MATLAB tool when converting the code. With the current functionality, the parameters that can be chosen for a simulation are:

- Dimensions, insertion position and material of the 3D object
- Mammography system
- 2D or DBT mode
- Settings of the x-ray tube (tube voltage, target material and filter material/thickness)
- Thickness, material, and orientation of the background

The next paragraphs will explain each step of the simulation pipeline in more detail and how it is implemented in the Python pipeline.

Lesion generation

The first step in the pipeline is the generation of the 3D object. For this step the user should define which 3D object they want to simulate (cylinder, cube, sphere or custom voxel model) and what the dimensions are. The tool then generates a 3D matrix where the zeros represent outside the object, and the ones represent inside the object as seen in figure 5.

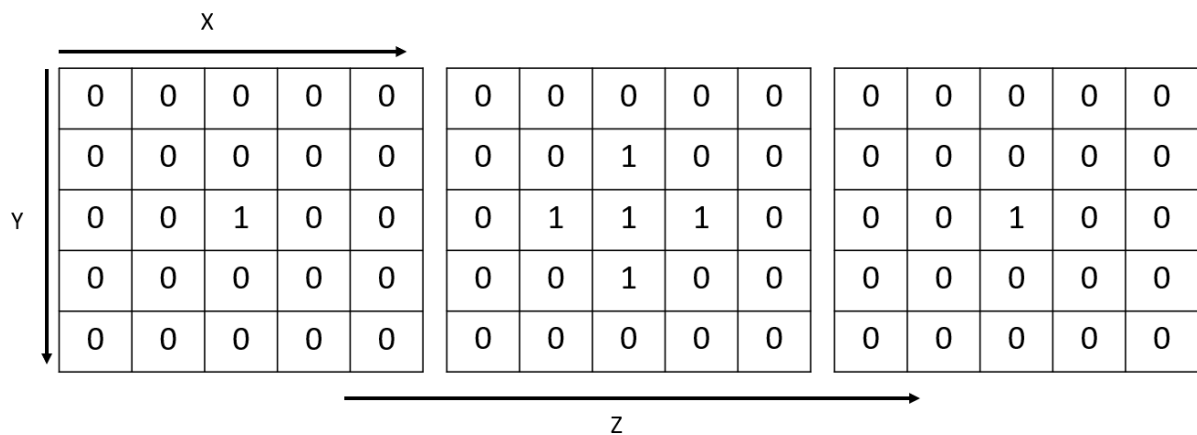


Figure 5: 3D matrix depiction of a small sphere

Every element in the matrix represents a voxel and every voxel has a defined pixel size. For example, when the pixel size is 1 mm/pixel in every direction (x, y and z), every element in the matrix represents a 1mm x 1mm x 1mm cube. This 3D matrix is used as input for the raytracing.

Raytracing

Raytracing means the tracking of each ray from the bundle and following its path to see how much it attenuates in a certain 3D object. In this step, the ideal template will be made. As explained before a template is a 2D matrix that acts as a projection of the 3D object. The values of the matrix indicate how much the beam attenuates at that specific spot by showing which fraction of the beam resides. An example of the 2D template of a simple small sphere can be seen in figure 6. When simulating in DBT mode, multiple templates are made for each angle. Figure 7 shows the templates of a bigger sphere where the pixel values depict the value of the matrix. The left template is at 0° and the right template is at 25°.

1	1	1	1	1
1	1	0,95	1	1
1	0,95	0,8	0,95	1
1	1	0,95	1	1
1	1	1	1	1

Figure 6: Example of the value matrix of a template



Figure 7: Example of a template of a sphere (2 mm) at an angle of 0° (left) and at an angel of 25° (right)

The attenuation is calculated with the exponential formula:

$$attenuation = e^{-\mu * l}$$

With:

l = thickness of the attenuating object

μ = the attenuation coefficient

The thickness of the attenuating object is calculated via the Siddon algorithm as described in [27]. It is a fast algorithm used in medical imaging applications. Instead of being individual elements, voxels are looked at as the intersection volumes of orthogonal sets of equally spaced, parallel planes as illustrated in figure 8 for the two-dimensional case. This concept can be generalized to any dimension without loss of accuracy. Rather than the intersections of the ray with the individual pixels, the intersections with the parallel lines are calculated. This can be calculated easier than calculating the intersection with the pixels because it is only necessary to determine the first intersection and generate all the others by recursion.

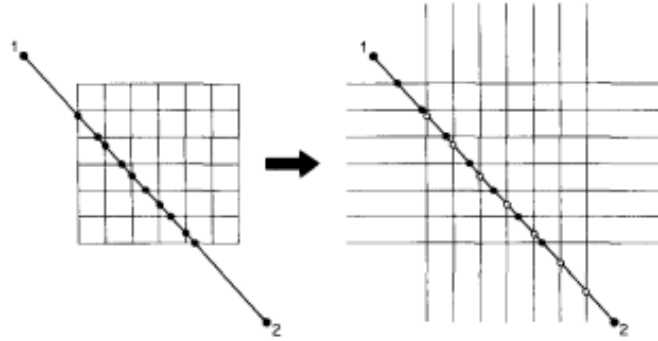


Figure 8: Pixels of an image (left) can be considered as the intersection areas of equally spaced lines (right) [27]

Modulation transfer function

When using a real system there are multiple factors that induce blurring. These factors include the finite focal spot of the X-ray source and the movement of the source during DBT acquisition. When performing the raytracing, the ideal templates do not take into account these factors which would result in an image with unrealistic sharpness. Therefore when simulating a system, the modulation transfer function (MTF) is used as a quantification to correct for this blurring. The MTF is a function that describes the spatial frequency response of a linear, spatially invariant imaging system. In other words, the MTF tells us how accurately the detector is able to transfer the contrast of the original object to the image and is unique for each system [28]. The MTF of a system can be measured from an image with a high density edge rotated at a very small angle of 1-3°. The modulation transfer function is applied to the templates in the frequency domain. For the implementation in the code, this means that the template should first be transformed using the fast Fourier transform. Next, the template is multiplied by the MTF and an inverse fast Fourier transform is performed.

Insertion

The insertion step combines the MTF modified template with the breast image. This is done by simply multiplying the corrected template with the image at the location on the projection. The location on the projection is calculated based on the insertion position and the angle of the tube. Because the incident rays fall in at an angle (except when the mass is directly under the source) the insertion position in the breast is not equal to the position on the projection.

In this step, the correction for scatter is also applied. The scatter is implemented in the code by using the scatter-to-primary ratio (SPR). The SPR is calculated with the following formula:

$$SPR = SPR_{slope} * thickness + SPR_{offset}$$

The thickness used in the formula is the thickness of the background. When simulating a breast this is the thickness of the compressed breast. SPR_{slope} and SPR_{offset} are both constants unique for each mammography system and can be measured on the system using the beam stop method [29]. After the SPR is calculated, the offset due to scatter is calculated and subtracted from the template.

2.2 Validation of the Python tool

The results of the Python code will be validated visually inspecting the simulated images or by comparing them to real projections or MATLAB simulations. Different aspects of the simulation pipeline will be validated with the use of different test objects explained in the next paragraphs. The validation will be performed for the Siemens Mammomat Revelation system. ImageJ will be used to process the images and measure the different parameters.

2.2.1 Raytracing

This validation is a visual test performed in 2D mode. The Python tool will simulate a cylinder with a diameter of 10 mm and a height of 30 mm made from soft tissue at different places inside a Polymethyl methacrylate (PMMA) slab. The goal of this test is to visually inspect the shape of the simulated cylinders and see if they are logical. Because the source is located at the chest wall, the cylinders further away from the source should have a different projection than the cylinders closer to the chest wall. The cylinders right underneath the source should look like a normal circle when projected on the detector while the cylinders further away should look like it has a shadow.

This test will validate if the raytracing is working correctly. When the position of the simulated object changes the angle of the incident rays will change resulting in a different projection of the cylinder on the detector.

2.2.3 Modulation transfer function

To validate the sharpness modification of the object templates a custom voxel model of the edge test object was generated and inserted in the image with the real MTF edge acquired on the system as seen on figure 9. The simulated 3D object was a tungsten edge with a thickness of 80 μ m. The simulation was done in 2D mode with the edge placed on the breast support table and in DBT mode with a varying height above the table of 2, 4 and 6 mm. First the edge was placed vertically to measure the tube-travel MTF (parallel to the side of the table where the detector is situated). Second the edge was placed horizontally to measure the front-back MTF. After the edge was simulated the MTF was measured by using an offline tool for both the horizontal and vertical MTF.

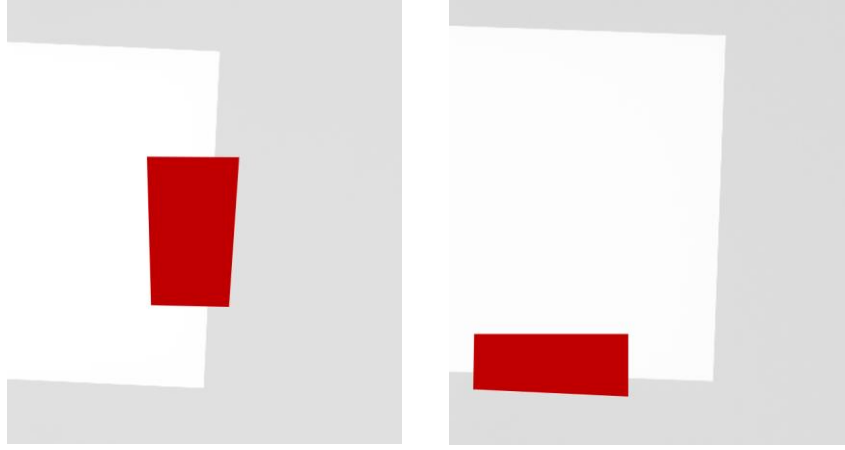


Figure 9: To validate the MTF, the simulated edge is inserted in the real test object (white) vertically (left) and horizontally (right)

2.2.3 Aluminum sheet

In this test, the accuracy of a simulated large area object is verified. This is done by comparing the peak contrast and signal difference to noise ratio (SDNR) between simulated object from the MATLAB tool and the Python tool. The object in this case was a 10 x 10 x 0.2 mm³ aluminum sheet inserted 10 mm above the table within a stack of PMMA plates. two acquisition sets were made for this verification:

- Varying the PMMA stack thickness from 20 to 70 mm scanned with a varying tube voltage from 26 kV to 32 kV for DM mode.
- Varying the PMMA stack thickness from 20 to 70 mm scanned with a varying tube voltage from 26 kV to 32 kV for DBT mode.

Peak contrast is defined as the difference in intensity between the object and the background relative to the background intensity. SDNR is the difference in intensity between the object and the background relative to the standard deviation of the background. These parameters are calculated with the formulas:

$$\text{contrast} = \left| \frac{\overline{PV}_{obj} - \overline{PV}_{back}}{\overline{PV}_{back}} \right|$$

$$\text{SDNR} = \left| \frac{\overline{PV}_{obj} - \overline{PV}_{back}}{SD_{back}} \right|$$

With:

\overline{PV}_{obj} = mean pixel value of the aluminum sheet

\overline{PV}_{back} = mean pixel value of the background

SD_{back} = mean standard deviation of the background

The mean pixel value of the object is measured with ImageJ in an ROI of 8 x 8 mm² placed in the middle of the aluminum sheet. The background is measured by placing the same ROI four

times around the object and calculating the mean of these four measurements as illustrated in figure 10.

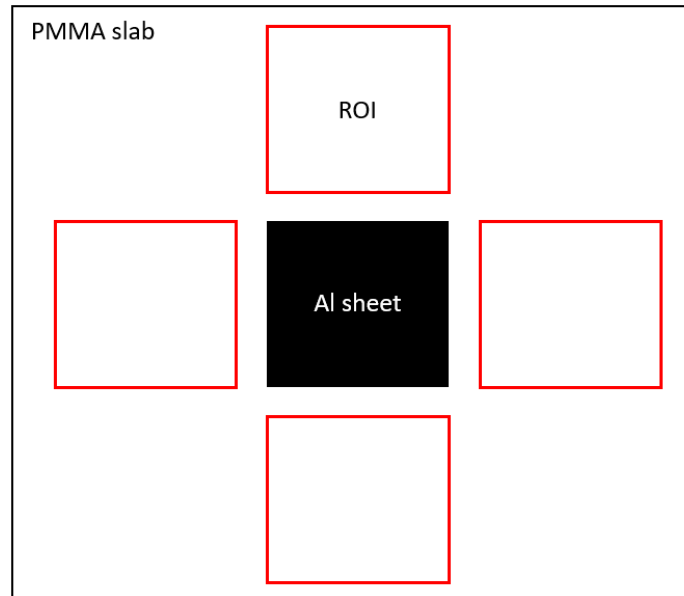


Figure 10: ROI location for measuring the mean pixel value of the aluminum sheet

This measurement was both done for the MATLAB simulation, the Python simulation and a real image. After this, the ratio of SDNR and peak contrast between the two simulation tools was calculated to see the difference.

A last comparison was done between the profiles through the simulated sheet with the MATLAB and Python tool. This is done by making a graph with on the y-axis the pixel value and on the x-axis the distance in the x-direction in mm.

2.2.4 Aluminum sphere

The verification of a small high contrast object was done by simulating a 0.5 mm aluminum sphere into projection images of a PMMA slab and comparing this to the MATLAB simulations with the same aluminum sphere. The acquisition settings are a 4 cm PMMA background and 28 kV. The initial validation was done by comparing the peak contrast and the SDNR of the two simulations in DBT mode. This was both measured in the central projection of the simulation and in the reconstructed image of the projections. The reconstruction was done with an offline reconstruction tool which reconstructs 2D projections into a sliced view in the z-direction.

Another parameter that was calculated was the artifact spread function (ASF). This parameter measures the spread of a signal over the planes in the z-direction. This is done by making a profile in the z-direction. This is measured in both the central projection image and reconstructed image. The error of each individual measuring point is calculated with the following formula:

$$\varepsilon = \left| \frac{\overline{PV}_{Python} - \overline{PV}_{Matlab}}{\overline{PV}_{Matlab}} \right|$$

With:

\overline{PV}_{Python} = mean pixel value of aluminum sphere in Python

\overline{PV}_{Matlab} = mean pixel value of aluminum sphere in Matlab

The mean pixel values were measured by a ROI with size 0.43 mm² placed in the middle of the aluminum sphere.

The mean error was calculated with the following formula:

$$\varepsilon = \frac{1}{N} \sum_{i=0}^{N-1} \left| \frac{\overline{PV}_{Python}(i) - \overline{PV}_{Matlab}(i)}{\overline{PV}_{Matlab}(i)} \right|$$

2.3 Application: Clinical cases

The end goal of the simulation tool is to imitate realistic images as if they were taken directly on a mammographic system. To test this ability, the tool is used to simulate real life clinical cancer cases. The results will be visually checked.

2.3.1 Different breast density types

When looking at a radiological image of a breast, the breast can be categorized into four groups based on density. These group range from group A, which is a non-dense breast containing little glandular tissue and much fatty tissue, to group D which are dense breast containing mostly glandular tissue. Figure 11 shows the radiological effect of different densities of breasts.

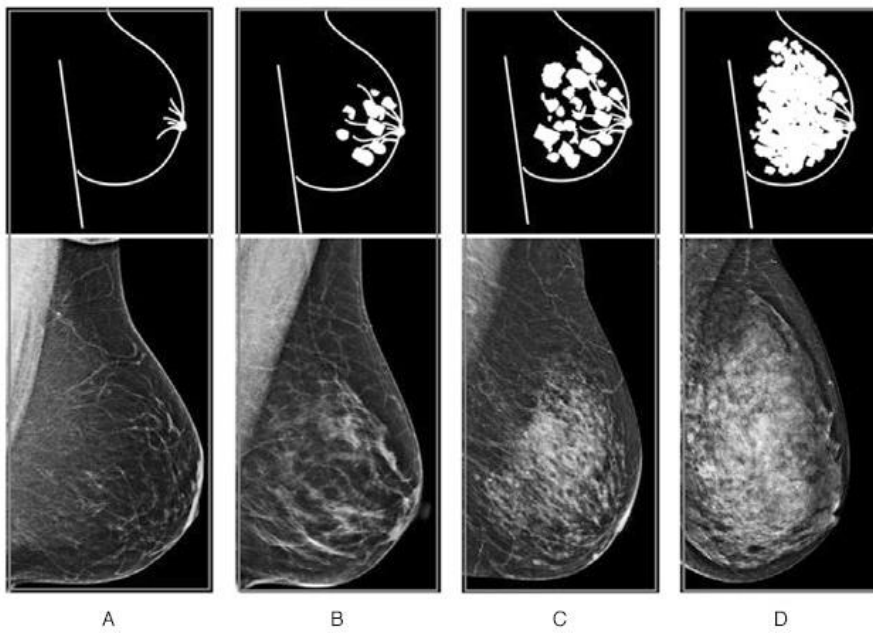


Figure 11: different breast densities [30]

The simulation tool needs to have the ability to simulate lesions in these different breast density types. To test this, the tool will simulate masses in different types of breasts background with the percentage of glandular tissue and fatty tissue of the background adjusted to the respective type. The different types that will be simulated are A, B, C and D. Besides this, the tool must also be able to simulate lesions of different densities, this to make it possible to change the visibility of the masses. To test if this is correctly implemented and if this is easy to adjust, the same mass will be simulated 3 times for each breast density type each time with a different mass density. The attenuation coefficients used for the simulations are:

- The attenuation of glandular tissue: It is generally assumed that a mass consists of glandular tissue.
- 1.04 times the attenuation of glandular tissue: This is the mass material used by another research group as described in [31].
- 1.1 times the attenuation of glandular tissue: This is 10% denser than normal glandular tissue to see the effect of an increase in density.

For breast type A only the two densest lesions will be simulated. The resulting projection will be reconstructed with an offline tool to get the sliced view in the z-direction.

2.3.2 Spiculated and non-spiculated lesion

In oncology we can distinguish different types of lesions with the first major subdivision, microcalcifications and masses. Based on their margins, masses can be classified as spiculated or non-spiculated. The simulation tool needs to be able to generate realistic images of all mass types occurring in clinical practice. In the simulation of these 2 clinical cases, the ability to simulate these two different lesion types will be tested. This is done by simulating a custom voxel model of a spiculated and a non-spiculated lesion in a breast image. The custom voxels used in this verification are seen on figure 12. The insertion position of the lesion is randomly chosen inside the breast, the material of the lesion is glandular tissue. After the simulation is done, the projections will be reconstructed in a sliced view in the z-direction with an offline tool.



Figure 12: The central slice of the voxel model of a spiculated mass (left) and a non-spiculated mass (right) used in the simulations.

When evaluating the simulations for the different breast types, spiculated and non-spiculated lesions were used with types B and C. These simulations are looked at in more detail for

evaluating the ability to simulate spiculated and non-spiculated lesions. The density of lesion used is 1.04 times the density of glandular tissue.

3 Results

3.1 Converting the MATLAB framework to the Python environment

The resulting code is a tool that has the same features as the MATLAB tool meaning it can simulate a custom 3D object into the projection of a real breast. Other features include the changing of the tube voltage, target material, filter thickness and composition of the background. Besides these features that were already available in the MATLAB tool others have been implemented. These new features are:

- User friendly error handling: when using the tool, the Python command prompt can give an error that is not easy to understand, or something goes wrong with the simulation without an error from Python. Error messages have been implemented which makes it easier to understand and instruct the user what can be done to avoid these errors.
- Ability to perform multiple simulations without user input: This is the most important feature. The tool now has a system in place that can do multiple simulations without intermediate input from the user. This is done by filling in an Excel file with the different parameters for every simulation.
- Less click intensive user interface: Instead of opening multiple menus to change one parameter The tool now has a list of all the parameters. To change one parameter, the user only needs to find it in the list and adjust it.

When comparing the simulation times there is also a difference. For example, when simulating a small sphere of 2 mm diameter, it takes 144 seconds to simulate this with the MATLAB code (changing the settings included). When doing the same simulation, it takes 66 seconds with the Python tool. Table 1 gives an overview of features and comparison of the Python and MATLAB tool. Besides the speed, the Python tool only requires one-click from the start of the simulation until the end meaning other things can be done in the meantime. This was not possible with the MATLAB tool as it required user input during the simulation. Another improvement is the structure of the code. The code now consists of 3 python scripts. The first one is the pipeline. This script consists of the code where the template generation, raytracing, MTF correction and insertion are done. The pipeline script is provided with a clear structure for each part of the simulation, a clear documentation has also been implemented by means of comment lines. The second script is the constants, here are all the constants stored needed for the simulation. For example, the geometry and settings of the mammography system you want to simulate. This overview makes it easy to for example add a new system. The last file is the script to run the pipeline. This file contains the means to change the settings of the simulation and allows the user to run the simulation.

Table 1: Comparison of MATLAB tool and Python tool when simulating a 2 mm sphere.

For 1 simulation	MATLAB	Python
Simulation time	144	66
Clicks	34	1
Error handling	x	√
User friendly	√	√
GUI	√	x
Easy to perform multiple simulations	x	√

3.2 Validation of the Python tool

3.2.1 Raytracing

Figure 13 shows the cylinders simulated in a grid. When looking at the different cylinders we see that depending on the place they have a different shape. The source of the system is located on the left edge of the picture, centered in height. When simulating an object further away from the source the incident rays tend to strike the side of the cylinder making it visible on a projection. The opposite can be said about when the cylinder is directly underneath the source meaning the incident rays only hits the top of the cylinder making the projection look like a plain sphere. With this in mind the results show that the geometric aspect of the raytracing is correctly implemented in the simulation tool.

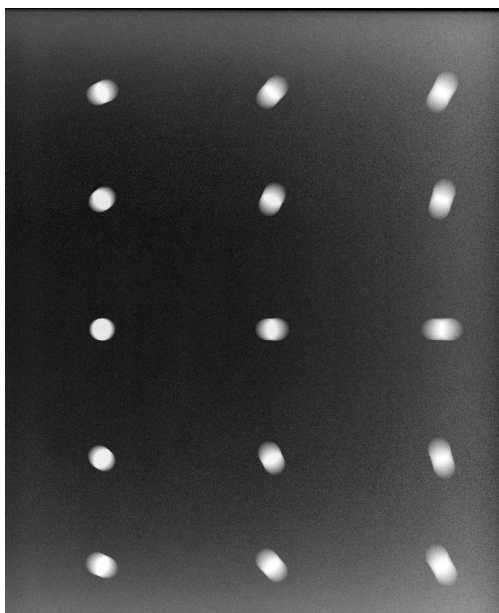


Figure 13: Results of raytracing validation

3.2.2 Modulation transfer function

The error between the 2D MTF measured from a real and a simulated edge is 1.47% for the horizontal MTF and 4.83% for the vertical MTF. The error is maximum 4.83% meaning that the MTF matches well. This is supported by the graph in figure 14. The graph shows that the MTF align well with each other.

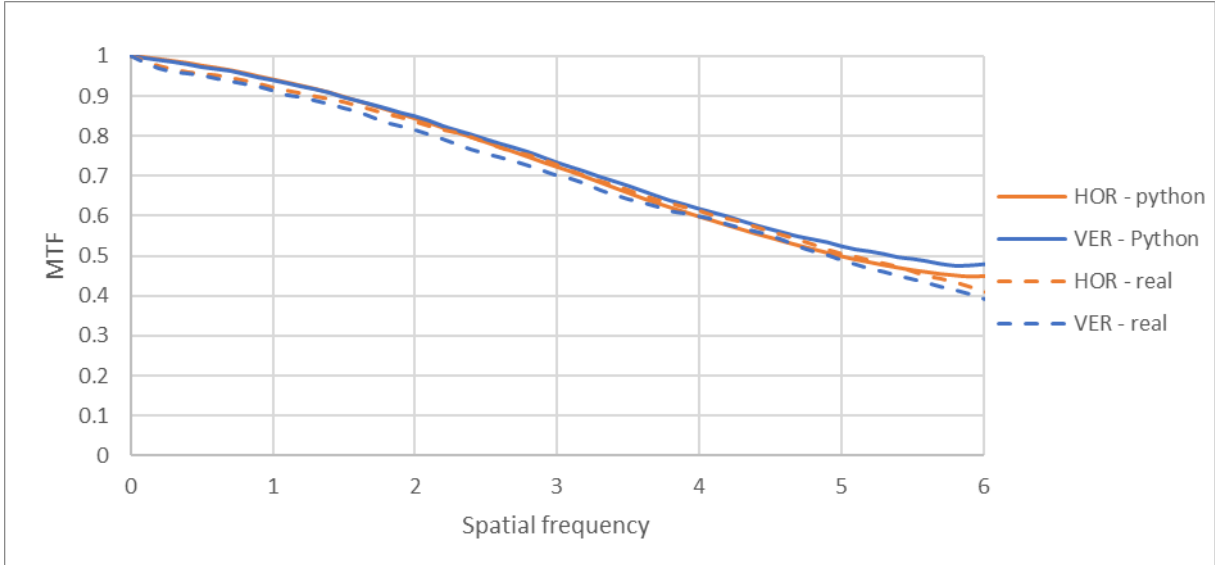


Figure 14: Results of the MTF validation in 2D mode

Table 2 shows the error in percentage from the comparison of the Python tool MTF with the real MTF for different heights above the breast support table. The biggest error made is 9.22% when the edge is located 6 cm above the table which is still acceptable. Figure 15 shows the MTF comparison of the DBT simulations for the different heights above the breast support table. Looking at the vertical MTF there is a slight offset, but the general trend is similar. The same can be said about the horizontal MTF.

Table 2: Error on MTF from validation

	2 cm hor	2 cm ver	4 cm hor	4 cm ver	6 cm hor	6 cm ver
Error	3.85%	8.57%	5.94%	8.14%	9.22%	4.53%

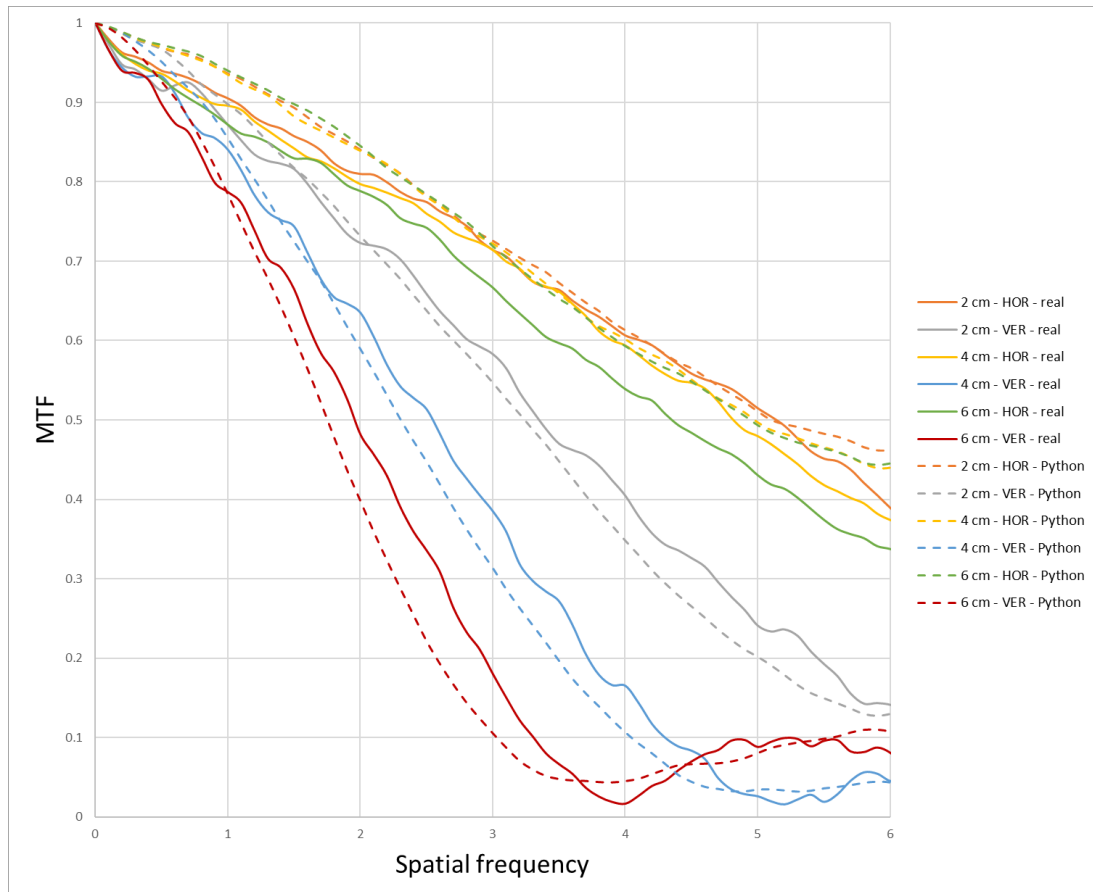


Figure 15: Results of the MTF validation in DBT mode

3.2.3 Aluminum sheet

Figure 16 and figure 17 show the graphs of the SDNR ratio between the real aluminum plate and the Python simulation in 2D mode and DBT mode, respectively. The straight black line is the ideal one-on-one ratio where the SDNR values perfectly match. The orange line is the 10% error line which depicts a difference of 10% between the SDNR values.

Results of the 2D simulation show a maximum error of 14% for the SDNR and peak contrast when comparing it to the real sheet. Conventionally when doing a validation of a simulation framework the aim is to have an error less than 10%. The error is bigger than 10% when the thickness of the PMMA stack is 30, 40, 45 and 50 meaning in these cases the simulation doesn't match well. When looking at the DBT mode the maximum error is 8%. This falls into the 10% error margin indicating good agreement between the real and simulated image.

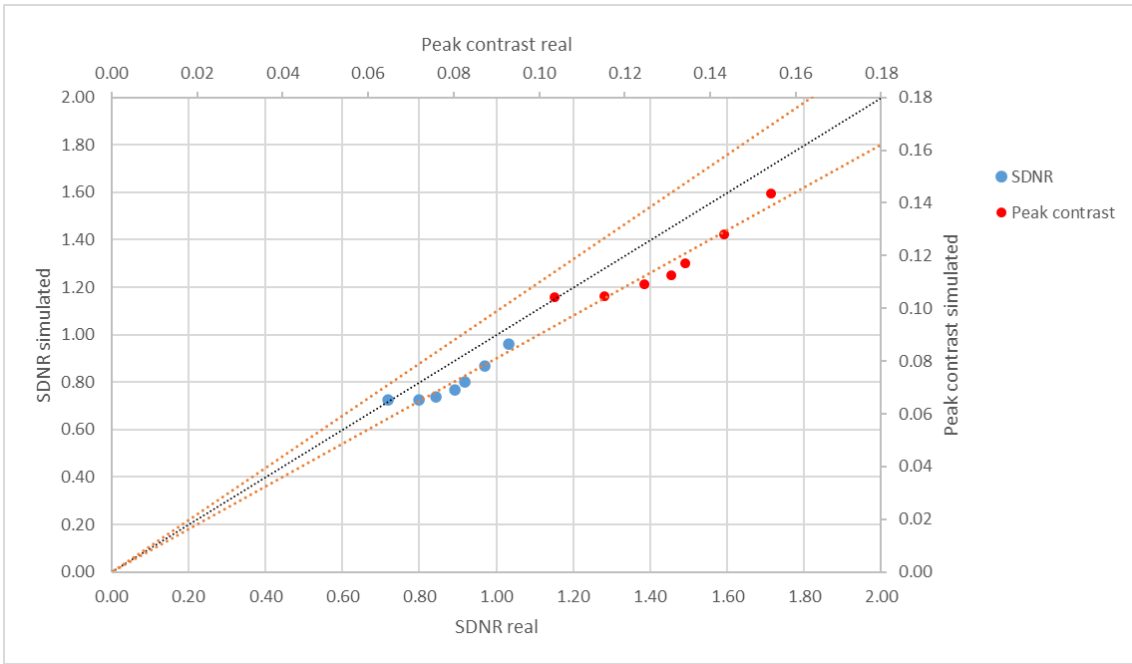


Figure 16: SDNR and peak contrast ratios of simulated and real aluminum sheet in 2D mode

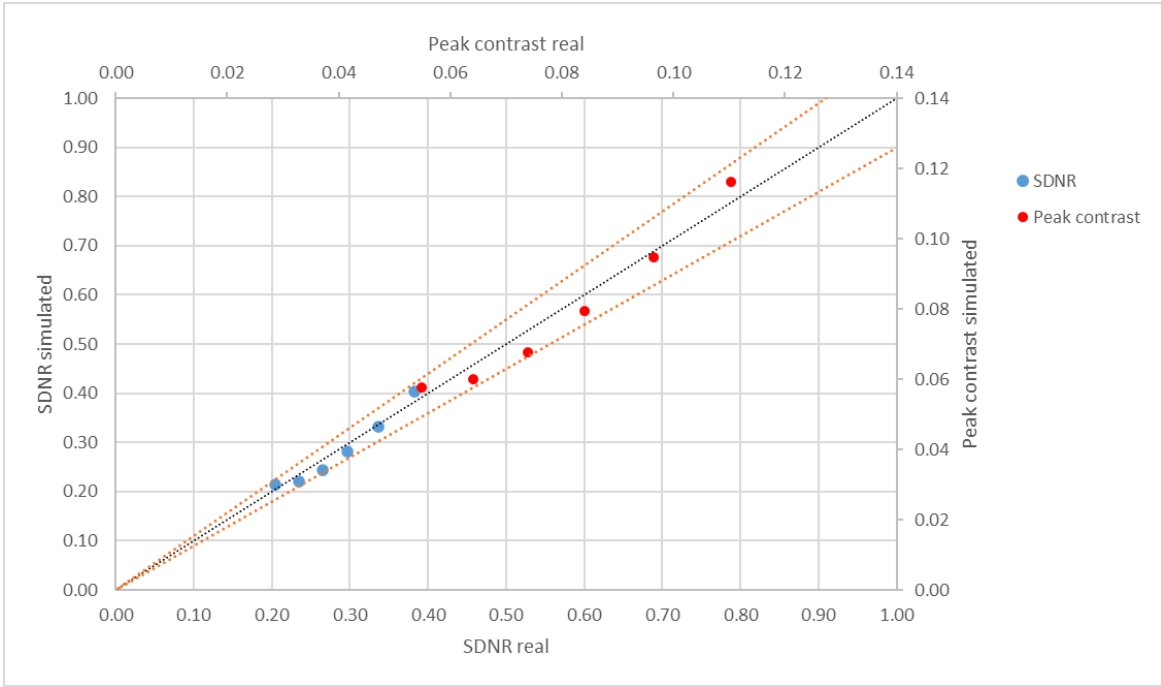


Figure 17: SDNR and peak contrast ratios of simulated and real aluminum sheet in DBT mode

Figure 18 shows the output image of the 2D simulation with a PMMA thickness of 40 mm and a tube voltage of 28 kV. The upper square is the real aluminum sheet and the lower square the simulated one. In terms of visual appearance, it is difficult to distinguish between the images of the real and simulated sheet.

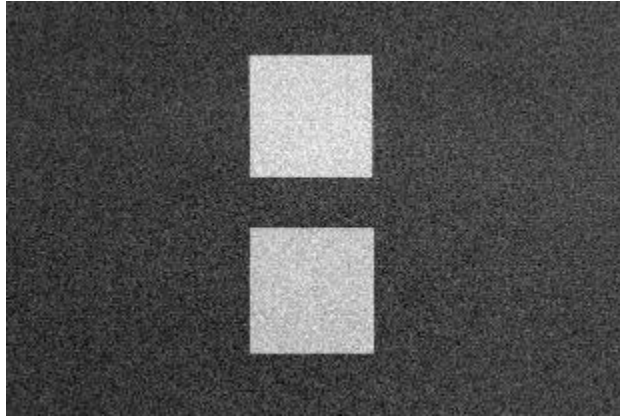


Figure 18: Output image of the 2D simulation with a PMMA thickness of 40 mm and a tube voltage of 28 kV

Figure 19 and figure 20 show the graphs of the SDNR ratio between the MATLAB simulated aluminum sheet and the Python simulation in 2D mode and DBT mode, respectively. Results of the 2D simulation show a maximum error of 9% for the SDNR and peak contrast. The error in DBT mode shows a maximum error of 10%. These error margins indicate that the simulations performed with both tools match well.

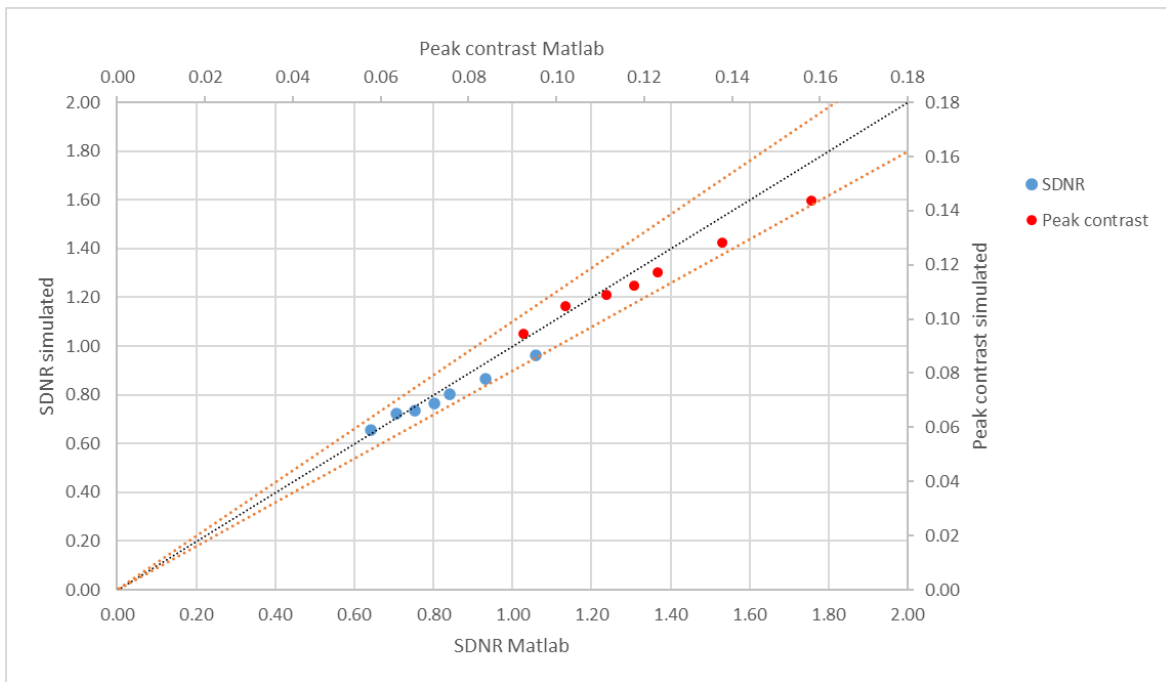


Figure 19: SDNR and peak contrast ratios of python and MATLAB simulations of a aluminum sheet in 2D mode

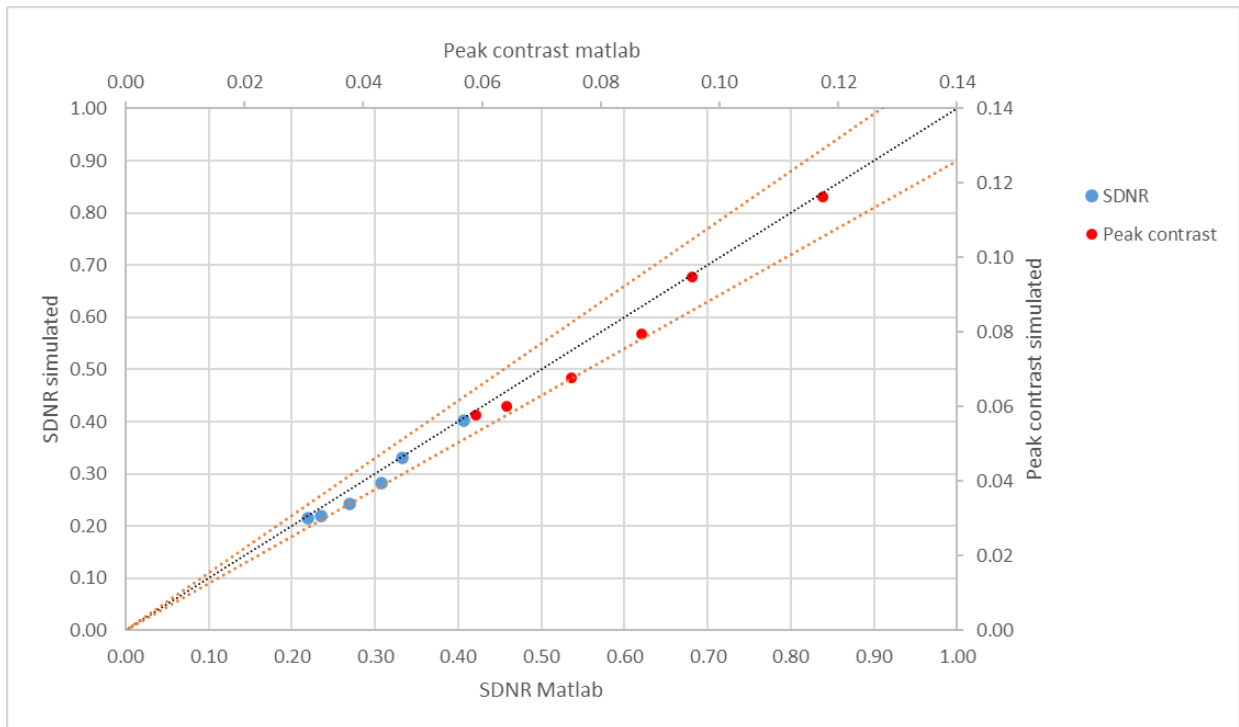


Figure 20: SDNR and peak contrast ratios of python and MATLAB simulations of a aluminum sheet in DBT mode

In addition to the calculation of contrast and SDNR in the simulated images, the profile through the aluminum sheet in the x-direction in the central projection is measured. Figure 21 shows the profile along the x-direction of the MATLAB simulation and the Python simulation. The images are from the simulation of the 40 mm PMMA with a voltage setting of 28 kV in DBT mode. When comparing the two profiles it shows that they are very similar in shape. This can be confirmed by visual inspection of figure 22 which shows the result of both simulations.

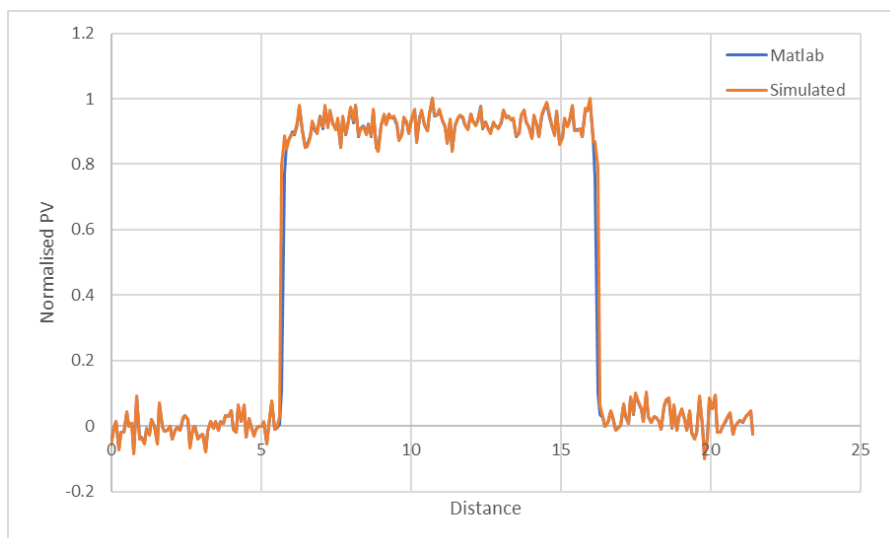


Figure 21: Profile through the aluminum sheet along the x-axis of the MATLAB and Python simulation

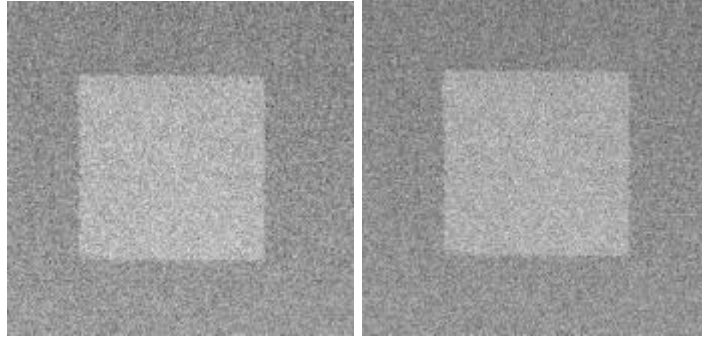


Figure 22: Central projection of the aluminum sheet simulation in DBT mode (40 mm PPMA, 28 kV) of MATLAB (left) and Python (right)

3.2.4 Aluminum sphere

Table 3 shows the signal, peak contrast and SDNR ratios of the simulated image in comparison to the real image for both the projection and the reconstructed image. The maximum error is 3% for both image types. This means that the simulated and the real aluminum sphere match well. Table 4 shows the same results but for the MATLAB and the Python tool. For the projection it shows an error of 11%, this is just outside the error margin. When looking at the reconstruction the error is 2% indicating a good agreement.

Table 3: Signal, Peak contrast and SDNR ratios of the Python simulation and the real small aluminum sphere

	kV	Signal	Peak contrast			SDNR		
		Sim/real	real	Sim	Sim/real	real	Sim	Sim/real
Projection	28	1.02	0.07	0.06	0.97	0.19	0.19	0.99
reconstruction	28	1.00	0.24	0.25	1.03	12.77	13.09	1.03

Table 4: Signal, Peak contrast and SDNR ratios of the MATLAB and Python simulations of a small aluminum sphere

	kV	Signal	Peak contrast			SDNR		
		Sim/MATLAB	MATLAB	Sim	Sim/MATLAB	MATLAB	Sim	Sim/MATLAB
Projection	28	1.00	0.06	0.06	1.11	0.17	0.19	1.11
reconstruction	28	1.00	0.22	0.22	1.02	11.11	11.29	1.02

Figure 23 shows the ASF of the sphere simulated with MATLAB and Python. The profile along the z-axis around the peak aligns well, meaning the simulated objects match between the simulations. This claim is supported when looking at the central projection of both the simulations we see these are hard to distinct. When looking at the outer parts of the graph, they match less well. This means that the background of the objects is different which is due to a difference in positioning of the simulated object with respect to the background. Figure 24 shows the projection generated by the simulation.

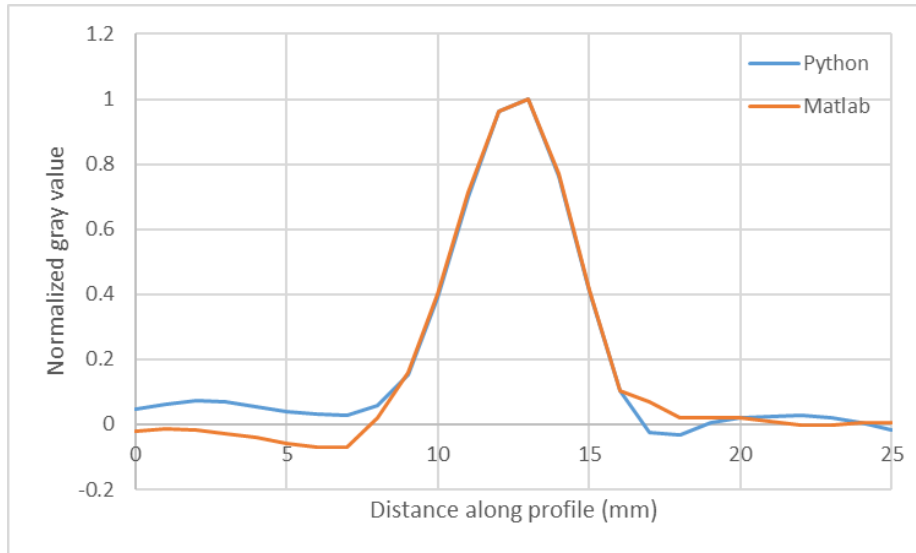


Figure 23: Profile in the z-direction of the Python and MATLAB simulation of a small aluminum sphere

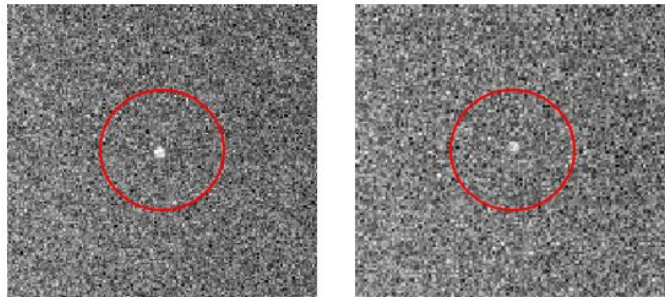


Figure 24: Projection of the small aluminum sphere generated from the simulations from the MATLAB tool (left) and the Python tool (right)

3.3 Application: clinical cases

3.3.1 Different breast density types

Figures 25, 26, 27 and 28 Show the result of the clinical cases with varying breast type and lesion density. The least dense breast is on the left (density = density of glandular tissue) and the densest breast (density = 1.1*density of glandular tissue) is on the right. On the right breast the simulated lesion is indicated with a red circle. For breast types A and B, the lesion is very good visible for each density. When looking at breast types C and D the less dense lesion becomes hard to distinct from the background to the point that the least dense lesion is completely not visible in the type D breast. Looking at the densest lesions a black edge can be seen around the lesion. This is an artifact due to the density.

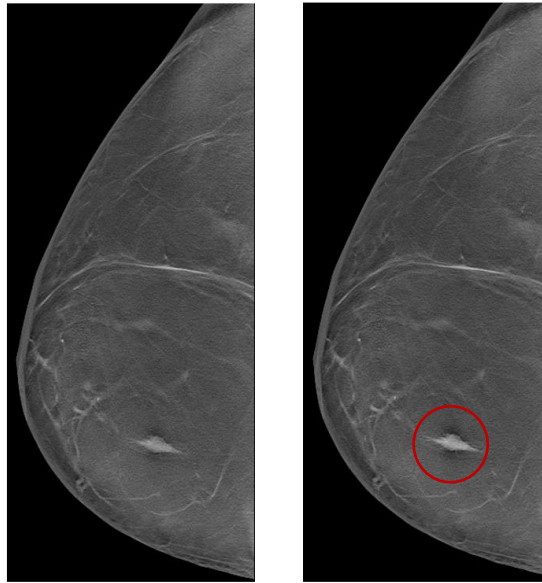


Figure 25: Reconstruction of the simulation of type A breast with different densities of the lesion

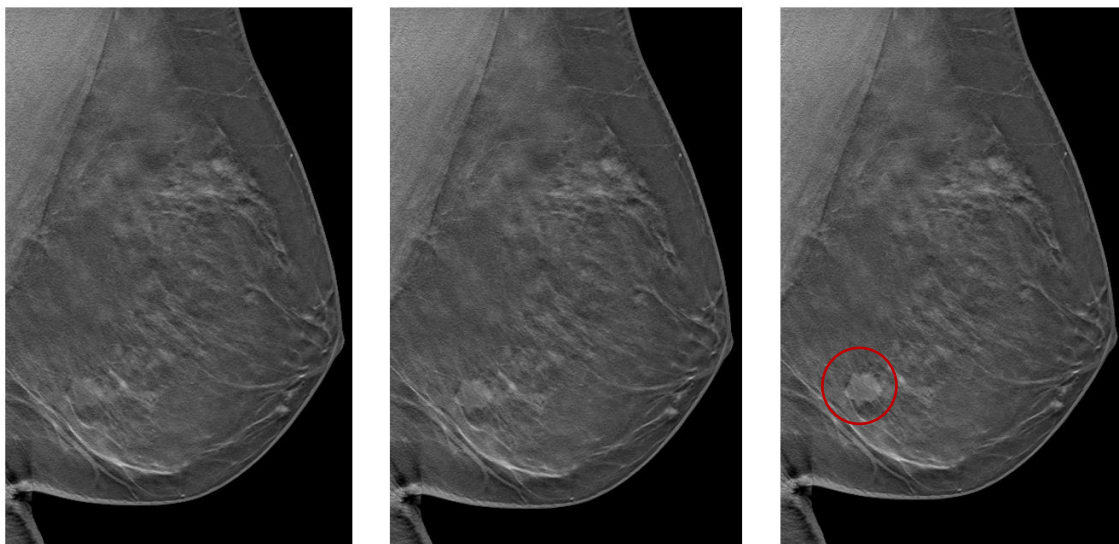


Figure 26: Reconstruction of the simulation of type B breast with different densities of the lesion

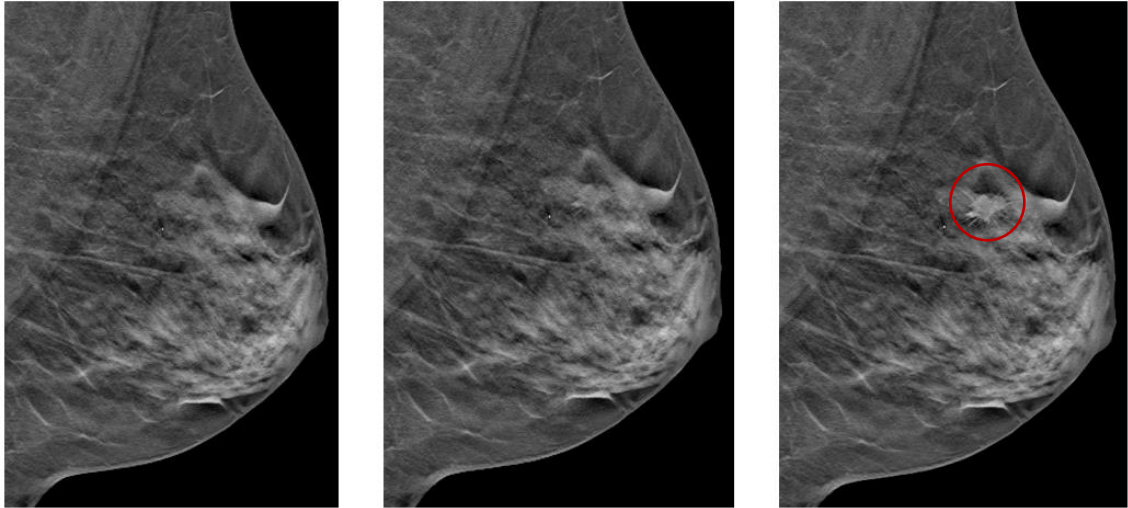


Figure 27: Reconstruction of the simulation of type C breast with different densities of the lesion

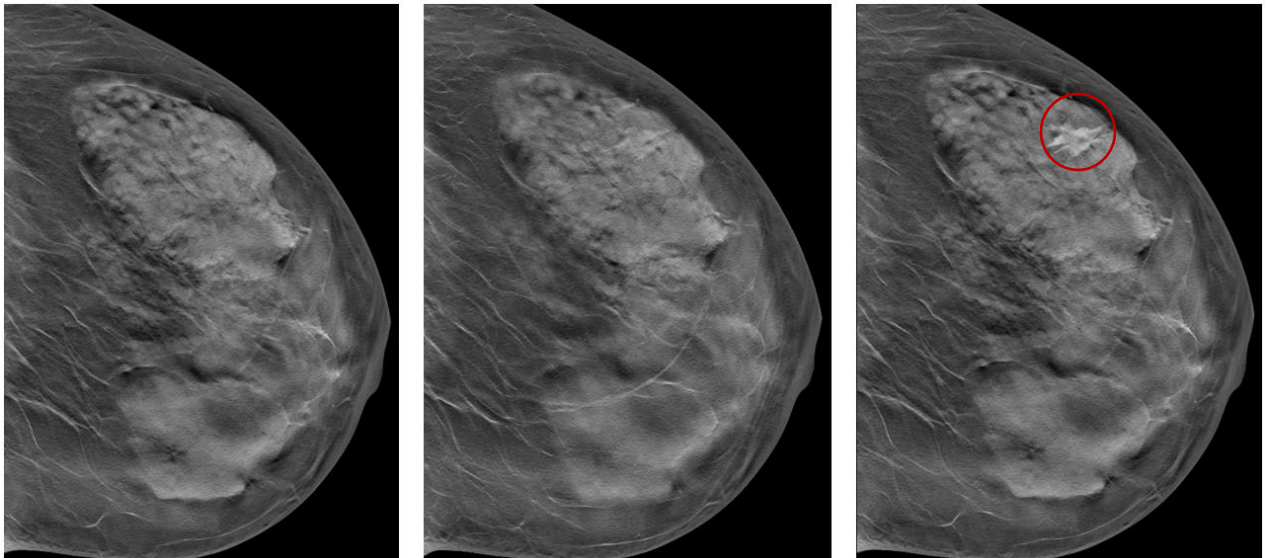


Figure 28: Reconstruction of the simulation of type D breast with different densities of the lesion

3.3.2 Spiculated and non-spiculated lesion

Figure 29 and figure 30 Show the reconstruction of the simulation of the spiculated lesion case and the non-spiculated lesion case together with an enlarged picture of the lesion and the central slice of the voxel model. The image of the spiculated lesion shows good visibility of the lesion overall and the different spicules can be distinct from the background. The non-spiculated also shows good visibility and can be distinct from the background.

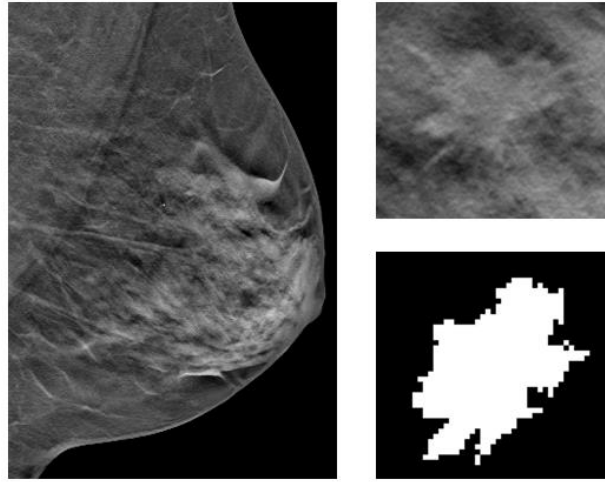


Figure 29: Reconstructed image of the clinical case with the spiculated lesion

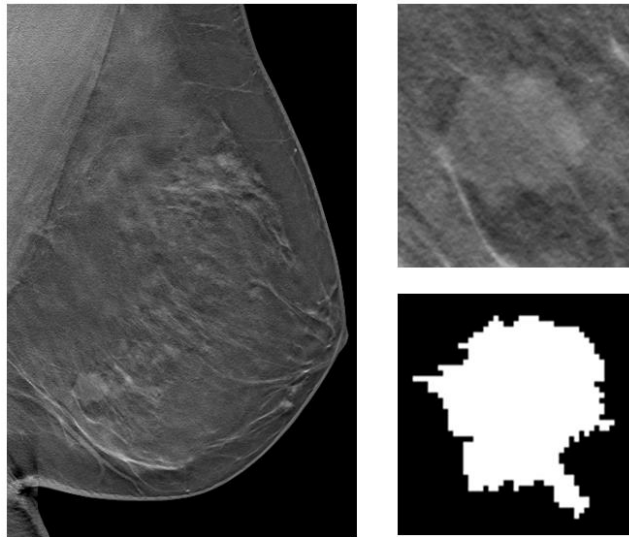


Figure 30: Reconstructed image of the clinical case with the non-spiculated lesion

4 Discussion

4.1 Converting the MATLAB framework to the Python environment

One of the most important added features is the ability to perform multiple simulations without the intermediate input of the user. An application where this can be useful is for the creation of AI training data. When training AI, a great number of images is required. It is important that the ground truth of these images is known which in most cases is the position of the lesion together with the diagnosis like lesion type, BIRADS classification, outcome etc. There is a lot of screening data available, however these are not annotated in a structured way. It is very time consuming for radiologists to create an annotated dataset for AI. When using simulated images, the location and shape of the lesion is defined by the user and thus the ground truth is known. With the use of the new Python tool big datasets, where the ground truth is known, can be easily made, which is not possible in MATLAB.

Another application of the developed simulation tool is to study the impact of different scan parameters on the image quality. In this case, the user often wants to change one or two parameters in between different simulations. For example, in the section of clinical applications, only the density of the simulated lesion was changed for every breast type. This is where the Python tool is easier to use in contrary to the MATLAB tool. When running a new simulation in MATLAB the settings go back to default, for example the default tube voltage in MATLAB is 28 kV. When you want to run three simulations with a voltage of 30 kV the user has to change the voltage to 30 kV every time you run a new simulation. This is the case for most of the parameters. In the Python tool, when running a new simulation, the settings from the previous simulation are retained. This means that when running consecutive simulations where only one or two parameters are changed, the Python tool saves a lot of time and clicking.

4.2 Validation of the Python tool

The ability to modify the sharpness of a simulated object was verified by measuring the MTF of a simulated object and comparing this to a real object. Looking at the results, a close agreement was seen for the MTF calculated from real and simulated edges with only a maximum error of 4.83% for 2D mode and 9.22% for DBT mode. The largest error is for the most extreme case which is with the lesion positioned 6 cm above the table.

The accuracy of the images generated with Python simulation framework, has been verified for a range of system parameters. A thin aluminum sheet of 0.2 mm was used to assess the simulation accuracy of a high contrast large area object. The ratio of the peak contrast and SDNR between the Python and MATLAB simulation in 2D or DBT mode is minimum 0.90 and maximum 1.03. These errors are just enough to meet the desired accuracy of 10%. The visual comparison of the simulations shows that this falls in line with the results. The same thing can be said when looking at the profiles of the 2 simulations as these matches well. The ratio for peak contrast and SDNR between the Python tool and the real image for the aluminum sheet is minimum 0.86 and maximum 1.05. These results do not meet the desired accuracy of 10%. These errors can be explained by the used input parameters. As mentioned before certain input parameters that are system specific are needed to simulate an object. These input parameters include the MTF and SPR parameters which are provided by the UZ Leuven as they have measured them before. After further inspection, these parameters were not entirely

correct which can introduce an error. This error on top of the error made in comparison with MATLAB can accumulate in a bigger error.

A small aluminum sphere of 0.5 mm was used to determine the accuracy of a small high contrast object. The ratio between the MATLAB and Python simulation for peak contrast and SDNR was at least 1.02 or at most 1.11. This error is too big to fall within the margin of 10%. After further inspection these errors are due to a difference in insertion position meaning the object is inserted on a different background than the object simulated in MATLAB. This is a mistake of the coordinate system implemented in the code. Smaller objects are more prone to change in peak contrast and SDNR as a consequence of a different position because a small change in position will make that the surrounding background is completely different in contrary to bigger objects which will only have a part of their background changed. Comparing visually, the two simulated spheres are hard to distinct. These claims are supported when looking at the ASF graph, where we see that the pixel values of the object (peak) match well, but the background does not. The ratio of the real and simulated object is between 0.97 and 1.03 meaning a good correspondence with only a maximum error of 3%.

There is a distinct difference between the MATLAB simulation and the Python simulation, in some cases the error falls outside the desired margin of 10%. When converting the code all the used formulas and calculations in the Python tool were the same as the MATLAB code so the result is expected to be the same. The verification shows this is not the case. As mentioned before the difference in result can be due to a difference in position. When calculating, MATLAB automatically rounds the numbers. Python also does this but retains more significant numbers. especially when the roundings are carried through multiple calculations this can cause a bigger error.

4.3 Application: clinical cases

The results show that the density of both the breast and the lesion can be changed easily to mimic clinical cases with different types of breasts. The visibility of the lesions also changes with its density meaning this is correctly implemented in the tool. For denser lesion a density artifact can be observed.

Looking closer at the results of breast type D, the lesion cannot be distinguished from the background when the density is too low. This is a good example of how these simulations can be applied for optimization studies. For example when a new mammography unit needs to be tested we want to know at which density the lesion will not be visible anymore. To do this multiple simulation can be run with varying density. This can also be done by changing other parameters like tube voltage.

A more specific application is the testing of synthetic mammogram. When searching for a lesion in a reconstructed sliced view of a DBT the radiologist must scroll through multiple slices of a 3D volume to identify the lesion, this can take a lot of time. An alternative to these images is synthetic mammograms. A synthetic mammogram is a 2D mammogram generation from the data of a DBT. This can lighten the workload for radiologists and because only a DBT is needed lower the dose for the patient[32], [33]. The simulation tool can be used to simulate multiple DBT acquisitions with different lesion densities. The synthetic mammogram can then

be made from these simulations to see the effectiveness for different densities and at what density the lesion will not be visible anymore.

4.4 Future development

After looking at the validation results and the simulated clinical cases, it is clear that there are still some small mistakes in the Python code. The next step will be looking at further development of the tool to correct these imperfections and to look at future applications where the tool can be used to possibly prove its usefulness.

One of the more important developments is the correction of small errors. In some cases, there is still an error bigger than 10% when comparing it to MATLAB simulations and real images. This error is probably due to a mistake with the insertion coordinates in the code. The next step would be to correct this error and search for any other possible errors.

Other than fixing the small errors, other extra features could be added. For example, the ability to simulate 3D objects with a mix of materials instead of just one. This can be useful when wanting to simulate certain alloys by combining different metals. It can also be used to combine different types of tissue with other materials to have a more accurate depiction of the material of a certain mass.

The feature described previously is the ability to simulate homogeneously mixed materials. Following this another additional feature is simulating an object where the user can create different parts of the object from different materials. For example, a sphere where the upper half is made from aluminum and the lower half made from breast tissue. This can be used to simulate more complex 3D objects. Currently there is only the possibility to simulate homogeneous lesions, while in reality lesions are often heterogeneous.

This year, a new mammography unit will be installed in Leuven. This is the first chance for the simulation tool to be applied for VCT. Currently the tool possesses the means to simulate presently used mammography units in the department. Another mammography unit can easily be added to the tool. Once the system is added, the tool can be used to compare the current techniques to the new one and see if the performance is better. The advantage here instead of conventional clinical trials is that it is faster, and we know the ground truth.

5 Conclusion

This study shows that the MATLAB tool can be converted to Python. The resulting tool can simulate geometric shapes of different sizes or custom voxel models in a real background image of a breast. Multiple simulation parameters can be set to closely mimic a real mammographic unit. Although the operation of the simulation has remained the same as the MATLAB tool, it turns out that the tool was in need of renewal. These innovations include a more user-friendliness of the tool but more importantly the ability to use the tool on a larger scale. This is necessary to be able to use the tool for virtual clinical trials.

The validation shows that the Python tool can simulate small high contrast objects and large area contrast objects whose results are comparable to the results of the MATLAB tool. Validating the tool and comparing it with real images, reveals that the results of the small high contrast objects agree well, however, with the large area contrast objects there is a substantial error. Comparing the MTF between the Python tool and the real images shows that the Python tool is able to modify the objects' sharpness to make the blurring more realistic. When we start comparing the results visually, it is difficult to distinguish them which beats a rather good agreement.

The simulation tool has the ability to simulate clinical cases. This is done by simulating a custom voxel model of a real lesion. The simulation tool shows potential to be used in applications of virtual clinical trials. These trials can include the testing of a new mammography unit, the training of AI and the testing of synthetic mammography

The resulting Python simulation framework is a complete tool that can be used for large scale virtual clinical trials. Nonetheless, next steps should be taken to further develop the tool in order to remove minor errors from the code.

Bibliography

- [1] Agentschap Zorg en Gezondheid, “Bevolkingsonderzoek Borstkanker | Vlaanderen.be,” 2017. <https://www.vlaanderen.be/bevolkingsonderzoek-borstkanker> (accessed Dec. 21, 2022).
- [2] K. kenniscentrum voor de Gezondheidszorg, “Borstkankerscreening op maat van je risico,” 2012. <https://kce.fgov.be/nl/borstkankerscreening-op-maat-van-je-risico> (accessed Jan. 20, 2023).
- [3] S. W. Duffy *et al.*, “Mammography screening reduces rates of advanced and fatal breast cancers: Results in 549,091 women,” *Cancer*, vol. 126, no. 13, pp. 2971–2979, Jul. 2020, doi: 10.1002/cncr.32859.
- [4] B. N. Hellquist *et al.*, “Effectiveness of population-based service screening with mammography for women ages 40 to 49 years,” *Cancer*, vol. 117, no. 4, pp. 714–722, Feb. 2011, doi: 10.1002/cncr.25650.
- [5] American Cancer Society, “Breast Cancer Early Detection and Diagnosis | How To Detect Breast Cancer,” 2023. <https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection.html> (accessed Dec. 21, 2022).
- [6] C. Klabunde, F. Bouchard, S. Taplin, A. Scharpantgen, and R. Ballard-Barbash, “Quality Assurance for Screening Mammography: An International Comparison,” 2001.
- [7] C. Mercer, P. Hogg, and J. Kelly, “Digital Mammography A Holistic Approach Second Edition,” 2015. [Online]. Available: <https://ashcards.springernature.com/login>
- [8] D. R. Dance, S. Christofides, A. D. A. Maidment, I. D. Mclean, and K. H. Ng, “Diagnostic Radiology Physics: A Handbook for Teachers and Students,” Vienna, 2014.
- [9] J. S. Amisha, J. Wang, T. Yamada, and L. L. Fajardo, “Digital Mammography A Review of Technical Development and Clinical Applications,” *Imaging in Breast Cancer*, vol. 4, pp. 63–70, 2003.
- [10] A. M. Rocha García and D. Mera Fernández, “Breast tomosynthesis: State of the art,” *Radiología (English Edition)*, vol. 61, no. 4, pp. 274–285, Jul. 2019, doi: 10.1016/j.rxeng.2019.03.008.
- [11] G. Gennaro *et al.*, “Combination of one-view digital breast tomosynthesis with one-view digital mammography versus standard two-view digital mammography: Per lesion analysis,” *Eur Radiol*, vol. 23, no. 8, pp. 2087–2094, Aug. 2013, doi: 10.1007/s00330-013-2831-0.
- [12] C. J. Rose, “Statistical Models of Mammographic Texture and Appearance,” 2005.
- [13] L. Niklason, “Chapter 2 - Physics and Development of Breast Tomosynthesis,” in *Breast Tomosynthesis*, L. E. Philpotts and R. J. Hooley, Eds. Elsevier, 2018, pp. 5–8. doi: <https://doi.org/10.1016/B978-0-323-35827-9.00002-2>.
- [14] X. Wang, J. Li, and C. Qian, “Semantic Label Prediction of Mammography Based on CC and MLO Views,” in *2020 IEEE 6th International Conference on Computer and*

- Communications, ICC 2020*, Dec. 2020, pp. 1461–1465. doi: 10.1109/ICC51575.2020.9345195.
- [15] N. Houssami *et al.*, “Breast screening using 2D-mammography or integrating digital breast tomosynthesis (3D-mammography) for single-reading or double-reading - Evidence to guide future screening strategies,” *Eur J Cancer*, vol. 50, no. 10, pp. 1799–1807, 2014, doi: 10.1016/j.ejca.2014.03.017.
- [16] G. Bicchierai *et al.*, “A review of breast imaging for timely diagnosis of disease,” *Int J Environ Res Public Health*, vol. 18, no. 11, Jun. 2021, doi: 10.3390/ijerph18115509.
- [17] C. Reis, A. Pascoal, T. Sakellaris, and M. Koutalonis, “Quality assurance and quality control in mammography: A review of available guidance worldwide,” *Insights into Imaging*, vol. 4, no. 5, pp. 539–553, Oct. 2013. doi: 10.1007/s13244-013-0269-1.
- [18] D. J. Brenner *et al.*, “Cancer risks attributable to low doses of ionizing radiation: Assessing what we really know,” *Proc Natl Acad Sci U S A*, vol. 100, no. SUPPL. 2, pp. 13761–13766, Nov. 2003, doi: 10.1073/pnas.2235592100.
- [19] T. Kretz, K. R. Mueller, T. Schaeffter, and C. Elster, “Mammography Image Quality Assurance Using Deep Learning,” *IEEE Trans Biomed Eng*, vol. 67, no. 12, pp. 3317–3326, Dec. 2020, doi: 10.1109/TBME.2020.2983539.
- [20] P. Elangovan *et al.*, “Design and validation of realistic breast models for use in multiple alternative forced choice virtual clinical trials,” *Phys Med Biol*, vol. 62, no. 7, pp. 2778–2794, Mar. 2017, doi: 10.1088/1361-6560/aa622c.
- [21] F. di Franco *et al.*, “GEANT4 Monte Carlo simulations for virtual clinical trials in breast X-ray imaging: Proof of concept,” *Physica Medica*, vol. 74, pp. 133–142, Jun. 2020, doi: 10.1016/j.ejmp.2020.05.007.
- [22] A. Badal, D. Sharma, C. G. Graff, R. Zeng, and A. Badano, “Mammography and breast tomosynthesis simulator for virtual clinical trials,” *Comput Phys Commun*, vol. 261, 2021, doi: 10.17632/k5x2bsf27m.1.
- [23] B. Andreu, D. Sharma, C. Graff G., R. Zeng, and A. Badano, “Mammography and breast tomosynthesis simulator for virtual clinical trials,” *Comput Phys Commun*, vol. 261, no. 107779, 2021.
- [24] N. W. Marshall and H. Bosmans, “Performance evaluation of digital breast tomosynthesis systems: comparison of current virtual clinical trial methods,” Leuven, 2022.
- [25] A. Badano *et al.*, “Evaluation of Digital Breast Tomosynthesis as Replacement of Full-Field Digital Mammography Using an In Silico Imaging Trial,” *JAMA Netw Open*, Nov. 2018, doi: 10.1001/jamanetworkopen.2018.5474.
- [26] L. Vancoillie, N. Marshall, L. Cockmartin, J. Vignero, G. Zhang, and H. Bosmans, “Verification of the accuracy of a hybrid breast imaging simulation framework for virtual clinical trial applications,” *Journal of Medical Imaging*, vol. 7, no. 04, pp. 1–17, Apr. 2020, doi: 10.1117/1.jmi.7.4.042804.

- [27] Robert L. Siddon, “Fast calculation of the exact radiological path for a three-dimensional CT array,” Mar. 1985.
- [28] S. Naeem Ahmed, “Physics and Engineering of Radiation Detection,” Ontario, 2007.
- [29] E. Shaheen, F. Zanica, F. Sisini, G. Zhang, J. Jacobs, and H. Bosmans, “Simulation of 3D objects into breast tomosynthesis images,” *Radiat Prot Dosimetry*, vol. 139, no. 1–3, pp. 108–112, Apr. 2010, doi: 10.1093/rpd/ncq084.
- [30] Mayo Clinic, “Dense breast tissue: What it means to have dense breasts - Mayo Clinic,” Feb. 25, 2022. <https://www.mayoclinic.org/tests-procedures/mammogram/in-depth/dense-breast-tissue/art-20123968> (accessed Jan. 12, 2023).
- [31] P. R. Bakic *et al.*, “Virtual clinical trial of lesion detection in digital mammography and digital breast tomosynthesis,” Mar. 2018, p. 5. doi: 10.1117/12.2294934.
- [32] J. J.J., E. Gianotti, and Y. Chen, “Evaluation of a computer-aided (CAD)-enhanced 2D synthetic mammogram: comparison with standard synthetic 2D mammograms and conventional 2D digital mammography,” *Clin Radiol*, vol. Vol. 73, no. Issue 10, pp. 886–892, 2018.
- [33] G. van Schie, R. Mann, M. Imhof-Tas, and N. Karssemeijer, “GenErating Synthetic Mammograms From Reconstructed Tomosynthesis Volumes,” *IEEE Trans Med Imaging*, vol. 32, no. 12, pp. 2322–2331, Dec. 2013, doi: 10.1109/TMI.2013.2281738.