University of Twente

Bachelor Thesis

Comparing different reconstruction algorithms by determining the MTF of the reconstructed images from a breast CT

Lize de Wit (s2488795)

Bachelor committee:
Prof.dr. I. Sechopoulos
M. Mikerov
Prof.dr.ir C.L. de Korte
L.C. Goris

January 18, 2024

Abstract

Breast cancer is the most commonly diagnosed cancer among Dutch women with a diagnosis percentage of 31%. Digital two-view mammography (DM) is the current standard imaging technique for breast cancer diagnosis and detection. But recently questioning concerning the suitability of DM has been rising, along with the potential of different imaging techniques. Breast CT (BCT) is one of these techniques and it the main topic of this report.
In BCT an image of the breast tissue is acquired using X-ray, the image then gets reconstructed and processed for evaluation. There are different reconstruction algorithms that can be used, like filtered back projection (FBP) and iterative reconstruction (IR). To find the best reconstruction algorithm, they can be compared using the reconstructed images and the determined modulation transfer function (MTF).
In this research, two experiments are done. The first experiment is to create a suitable phantom for further testing. Here measurements are done using different liquids and comparing their Hounsfield Unit values. The results showed that tap water is the best option, but also that further testing and research is necessary. The second experiment is to reconstruct images using Feldkamp-Davis-Kress (FDK) FBP with different filters and also reconstruct images using IR with different filters. After determining the MTF, the results are compared. Based on the results, FDK FBP with Shepp-Logan filter and additional filtering was determined to be the best reconstruction algorithm (combination). However, the used phantom was deemed unsuitable for accurate results and a reliable conclusion. Thus, further research should be done and the experiments need to be redone using a different phantom.

Contents

1 Introduction

Worldwide, breast cancer is the second most commonly diagnosed cancer with a diagnosis percentage of 11.6% of all cancer diagnoses. But when looking solely among women, breast cancer is the most commonly diagnosed and also the leading cause of cancer death [1]. In the Netherlands, the diagnosis percentage of breast cancer in women is 31% and in 2022 a 5-year prevalence of 66,400 was recorded. This means that out of the 88,772 women that got a breast cancer diagnosis between 2017 and 2022 in the Netherlands, only 66,400 women were still alive in 2022 [2].

To decrease the mortality related to breast cancer, a breast cancer screening program is implemented in most European countries. The standard screening technique used for this process is digital two-view mammography (DM), here the breast gets compressed and an X-ray image is made from two sides to get an even surface coverage [3, 4, 5, 6]. One of the main downsides of DM is that it calls for a two-dimensional projection of a three-dimensional object, this goes along with the need for compression. Compression of the breasts during this procedure is often considered to be painful and causes tissue overlap, which decreases the sensitivity of DM [3, 7]. In recent years, breast computed tomography (BCT) has emerged as a potential replacement technique for DM. In BCT a three-dimensional projection is made and no compression is needed, therefore offering solutions for the main problems with DM [3, 7].

1.1 The potential of breast CT

Imaging a breast tumor can be done in multiple different ways, depending on the particular demands. Some of the techniques to consider are DM, BCT, magnetic resonance imaging (MRI) and ultrasound (US). These techniques are used in a variety of settings. Along with the recent rising potential of BCT goes the questioning if the current standard technique, DM, is still the most suitable. DM is often compared to MRI, US and now BCT, and has previously been chosen as the standard because of its cost-effectiveness and proven reduction of breast cancer related mortality [3, 4, 5, 8]. MRI is proven to be effective for detecting breast cancer and can be used with contrast enhancement (CE-MRI), enabling for a highlighted display of a tumor [9]. It has a high sensitivity compared to DM and does not use X-ray, therefore not exposing patients to ionizing radiation. However, the cost, availability and the scan duration of MRI make it currently unsuitable for regular, general use [7, 8, 9]. US plays an important part in real-time imaging during procedures and is able to differentiate between benign and malignant tumors. It can be used in combination with DM to detect breast tumors, but it is not yet suitable for detection on its own [10, 11]. Furthermore, the use of US calls for a specialist, increasing the cost of general use [8]. BCT uses X-ray imaging to make a three-dimensional projection of the breast. It shares similarities with MRI in sensitivity and contrast enhancement options (CE-BCT), but it has a lower scan duration and cost. However, there is exposure to ionizing radiation for patients and it has a relatively low specificity [3, 7, 8].

1.2 Screening or monitoring

When implementing BCT as an imaging technique in detecting breast cancer, it should be considered that different kinds of imaging are necessary during screening and monitoring. Screening should be done in a very cost-efficient manner as it is a very regular and general procedure, also a low radiation exposure and relatively high specificity are desired. The aspects of BCT make it currently unfit for regular screening, but it does have potential for monitoring breast tumors during treatment [3, 12]. Compared to MRI and DM, BCT is a considerably better imaging technique for monitoring breast tumors, combining their advantages into one technique [7, 8]. Likewise, CE-BCT is very beneficial for monitoring tumor growth and would be a good option over CE-MRI because of the lower scan duration and lower cost [7, 12].

1.3 Aim and goal

The aim of this research is to imitate and model a phantom of a tumor in breast tissue and find the best reconstruction algorithm by determining the MTF of the phantom imaged by a breast CT. The goal of this report is to compare different reconstruction algorithms by the images created and the determined MTF. This leads to the following research question: **What is the best reconstruction algorithm based on the MTF of an image, of a phantom, created by a breast CT?** Other questions that were to be answered leading up to the research question were:

- How can the MTF be determined using Python?
- What is a suitable liquid for a home-made phantom and will this phantom give an adequate representation that can be used to get the desired results?

2 Theoretical background

2.1 Breast CT

BCT is an imaging technique which uses X-rays to image breast tissue. The core functioning of a BCT system is the same across all different brands, but most of the settings and mechanical elements are different [13]. As for this report, the 'Koning Co. breast CT' system [14] is described and shown in Figure 1. The BCT system consists of an exam table with an opening for the breast to go through for imaging. Underneath the table represents a regular CT system with an X-ray tube and a flat-panel detector, as is shown in Figure 2. The X-ray tube and detector are opposite of each other on a gantry and rotate 360 degrees around the breast during the scan [8, 15]. During the scan a number of projections are made, for the Koning Co. breast CT this number is 300 [14, 15]. Before starting the actual scan, a scout measurement is done with a low radiation dose to calculate the optimal dose level as a function of the thickness of the breast tissue. During this measurement two scout images are made at 0 and 90 degrees of rotation [15]. After acquiring an image, the image goes through reconstruction and processing before it can be evaluated.

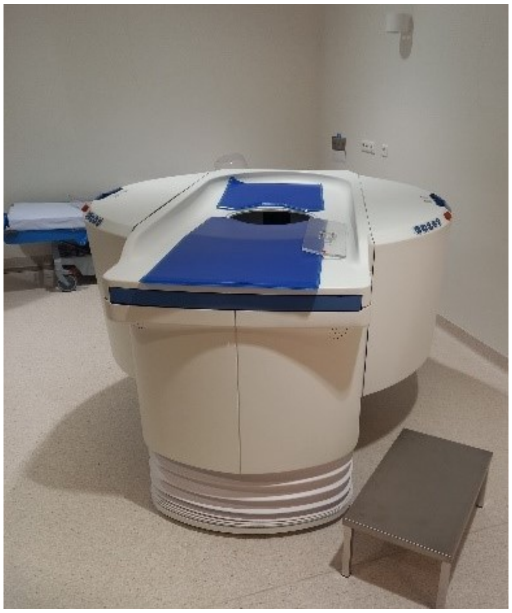


Figure 1: The Koning Co. breast CT that was used during the measurements done for this report. The opening is located near the middle of the exam table.

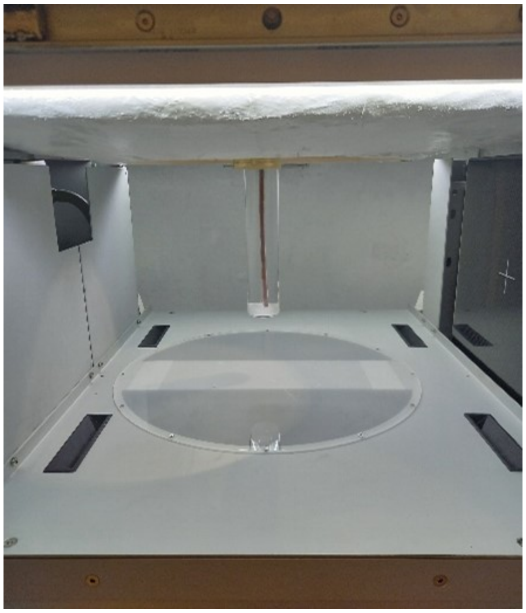


Figure 2: The inside of the breast CT with a home-made phantom suspended through the opening instead of a breast. The X-ray tube is on the left and the flat-panel detector is on the right.

2.1.1 Contrast-enhanced breast CT

The protocol for CE-BCT is similar to the protocol for BCT, except for the addition of contrast medium. Iodine contrast medium gets injected intravenously before the BCT scan to visualise tumor angiogenesis and enhance potential malignancies. This enables monitoring of tumor size and growth which is relevant for determining if the treatment of the patient is beneficial. An example of an image acquired with CE-BCT is shown in Figure 3. Here can be seen how the addition of contrast medium can increase the chance of detecting a tumor in breast tissue by enhancing and visualising tumor angiogenesis compared to the surrounding tissue [12, 15, 16, 17, 18].

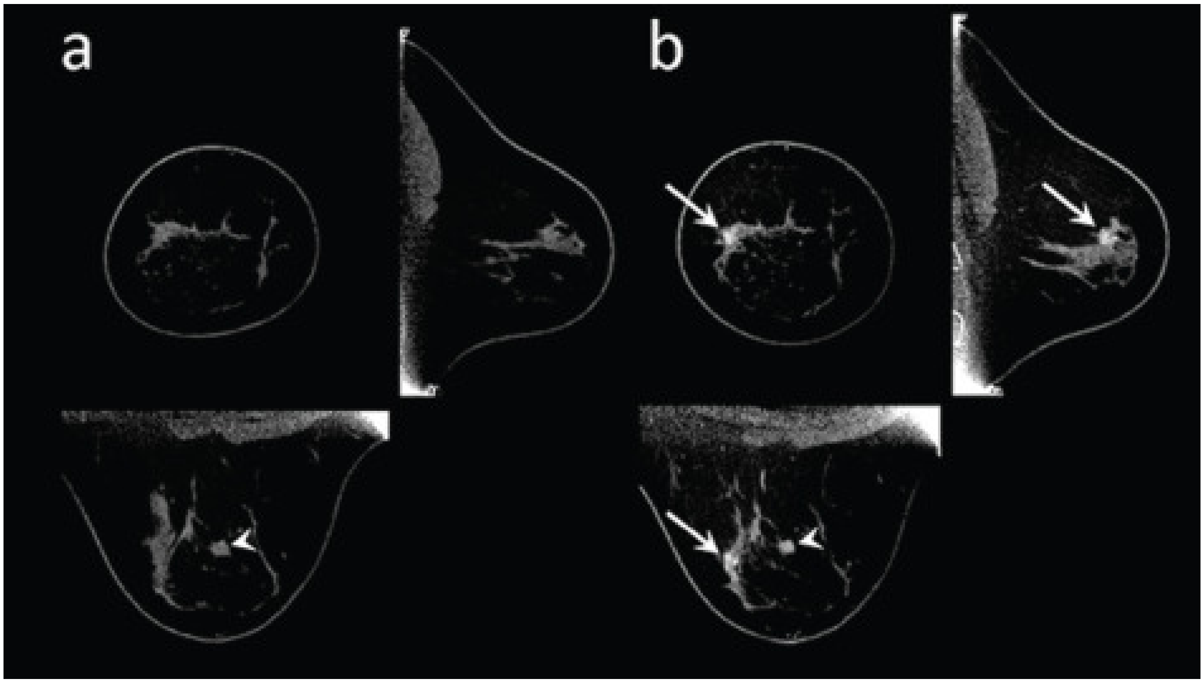


Figure 3: The effect of contrast medium in CE-BCT for detecting tumors in breast tissue, before (a) and after (b) using the contrast medium. The resulting enhanced tumor sites are indicated using arrows [18].

2.2 Modulation Transfer Function

The modulation transfer function (MTF) provides a measurement for the ability of a CT system to transfer contrast over spatial frequencies. This means that it provides information about the detectability of, for example, a tumor in breast tissue based on the difference in contrast. Therefore, a low MTF results in low object recognition [19]. The MTF is often used as an indicator for the spatial resolution of a system and image sharpness of CT, making it useful for evaluating the performance of a system [19, 20, 21]. The MTF can be calculated by taking the Fourier transform of the line-spread function (LSF), which is the derivative of the edge-spread function (ESF) [19, 22]. Further explaining of the determination and plotting of the ESF, LSF and MTF is done in Section 2.2.1. When plotting the MTF over the spatial frequency, as can be seen in Figure 8, conclusions can be drawn based on the values and shape of the graph. The y-axis, containing the MTF, ranges from 0.0 to 1.0 and is the resulting value of the modulation transfer ratio (MT). This ratio can be calculated as follows [23];

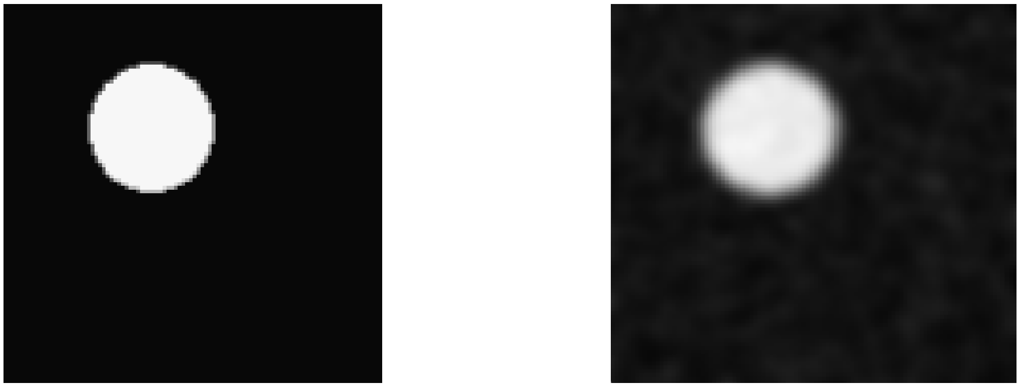
$$MT = \frac{M_{im}}{M_{obj}} \tag{1}$$

Here M_{im} represents the modulation in the image and M_{obj} represents the modulation in the object. The ratio values are often presented as percentages, with 1.0 being 100%, in a way to define the meaning of these values. When the MT is equal to 1.0 (100%) it means that the modulation in the image and the object are equal, signifying that the contrast got transferred perfectly from the object to the image. This would be an ideal result, but this rarely is the case [23]. The general rule of thumb is that a high MT (or MTF) is desirable over a low MT (or MTF).

2.2.1 Technical implementation

To get the MTF of a reconstructed image, using Python, certain steps are taken to go from detecting the relevant structures to taking the Fourier transform of the LSF. For explanatory purposes, a simplified process is shown below explaining the results and reasoning behind the steps taken. The Python code, for determining the MTF of a simplified image, is shown in Appendix A.

In BCT, and this research, an image will be acquired similar to Figure 4b. This is a simplified image compared to the actual acquired images, which are not a symmetrical circle and are subjected to non-artificial noise and blurring. This image is the result of adding certain amounts of noise and blurring onto Figure 4a by using the code presented in Appendix A.1.



(a) Simplified breast CT image of a contrast-enhanced tumor in breast tissue.

(b) Simplified breast CT image with artificially added noise and blurring.

Figure 4: Simplified and artificially created breast CT images.

To have Python detect and extract a structure within an image, in this case a circle, the image is put through thresholding and segmentation. This part of the Python code is shown in Appendix A.2. Thresholding is a way to separate the circle from the background by looking at the change in pixel intensity of the image. When an image has noise and blurring, the edges of a structure are no longer well defined. By using a thresholding technique, the edges get redefined but these redefined edges can differ from the actual edges of the structure. Segmentation is performed after thresholding to detect the redefined structure and select it so data can be gathered from that specific area. During segmentation the centre coordinates and the radius of the structure are determined and used to produce a circle, these values are based on the detected shape and estimations are made accordingly to neither overestimate or underestimate the size of the circle.

Next, the pixel values and distance from the centre get determined within the segmented circle. The function cycles through the pixels within the circle, determining their pixel value and corresponding coordinates. The distance of these pixels from the centre of the circle is calculated using Pythagoras and the x- and y-coordinates of the pixels. To make sure that the pixels belong to the circle, the calculated distance gets compared to the radius and if the distance is greater than the radius, the pixel value is not saved. Otherwise, the pixel value and corresponding distance are put into a matrix.
To acquire the ESF, the pixel values and their corresponding distances get sorted based on the distance from the centre. The pixel values then get plotted over the distance from the centre, and the resulting plot is shown in Figure 5.

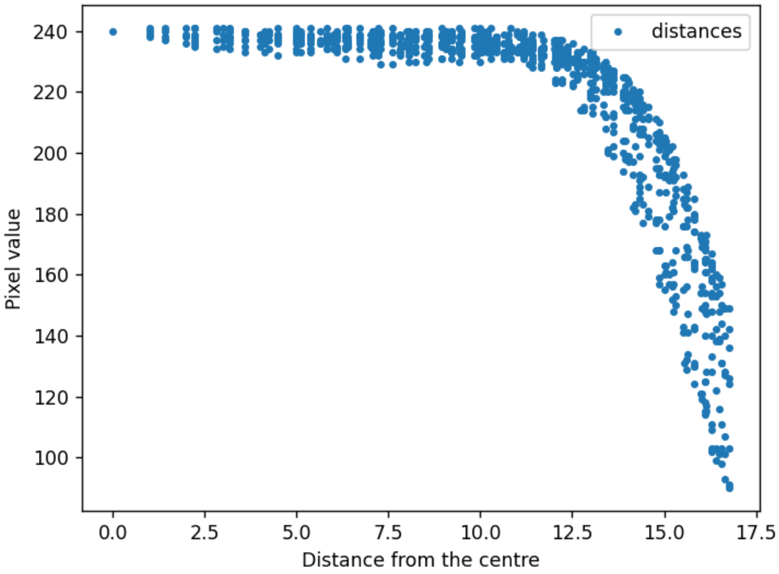


Figure 5: The ESF plotted.

In order to do a Fourier transformation, the data should be uniformly spaced. To achieve this, the ESF data points were re-binned as this cancels out uneven spacing between consecutive data points, as shown in Figure 6. During the re-binning, 30 'bins' were created with each bin corresponding to a distance between 0 and 30 evenly spaced apart. Then while cycling through the distances, a pixel value gets put in a bin when the corresponding, previously determined, distance matches the distance assigned to that bin. When no match is found and a bin is left empty, a mean value is determined and placed in the bin. Finally, the re-binned ESF gets plotted and the resulting plot is shown in Figure 7.

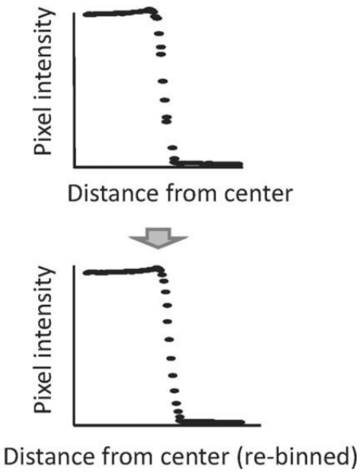


Figure 6: ESF plotted in a nonuniform way (top plot) and in a re-binned way (bottom plot) [19].

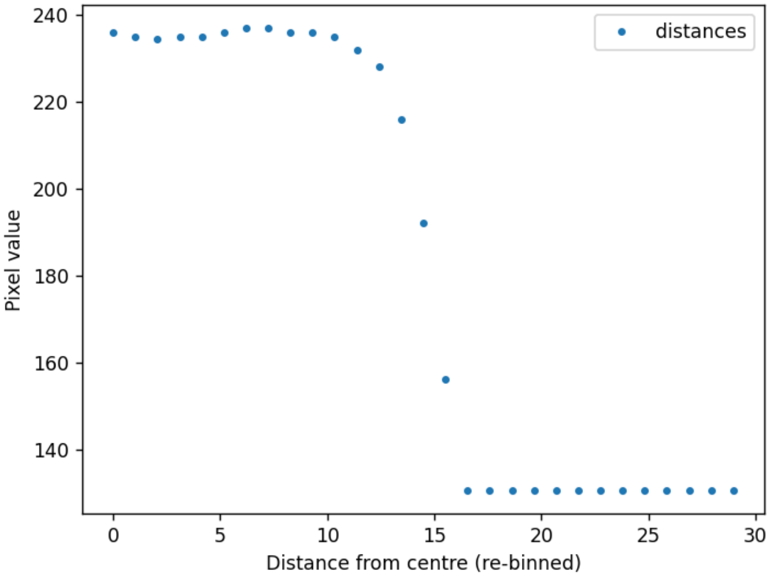


Figure 7: The re-binned ESF plotted.

To determine and plot the MTF, the derivative of the ESF is taken and the Fourier transform is done on the result. The Fourier transform represent the signal in the frequency domain as opposed to the time domain. The Python code for the Fourier transform is shown in Appendix A.3. After doing the Fourier transformation and plotting the results, the MTF is seen plotted over the spatial frequency in Figure 8.

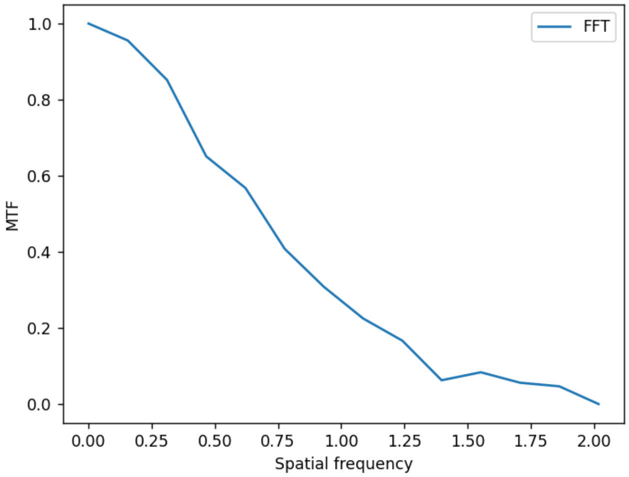


Figure 8: The MTF plotted over the spatial frequency.

2.3 Reconstruction algorithms

Various reconstruction algorithms are used to reconstruct CT images for evaluation. Depending on the characteristics of the imaging subject, different algorithms are favourable. The two main categories are filtered back projection (FBP) and iterative reconstruction (IR) [24].

FBP is the most common reconstruction algorithm used for CT [25]. It is an analytic reconstruction algorithm that relies on linear mathematical operations and relationships. FBP assumes that the acquired projection data from the CT scans is exact data and therefore produces an exact image of this data, but in reality the data is noisy. This noise is amplified by the FBP in the reconstructed images [26, 27]. Because this method reconstructs images based on an analytic equation, it is relatively fast compared to IR. However, FBP is not favourable compared to IR when using lower doses of radiation, because that causes the resulting reconstructed images to be noisier than at higher doses [24, 25, 26, 27].

IR was introduced as an alternative to FBP because of its ability to deliver reconstructed images with lower noise levels at a reduced radiation dose [24]. IR reconstructs images using a loop to minimize the error and reduce the differences between the acquired data and forward projected data [25, 26, 27]. One of the downsides of using IR is that it takes longer to reconstruct an image than FBP. This is due to repeating the same procedure for multiple iterations [26, 27].

3 Creating a physical phantom for testing

3.1 Goal of the experiment

The goal of this experiment was to create a physical phantom, representing breast tissue, that can be used in later measurements with the BCT. During this experiment, different liquids were tested in the BCT and the results were compared to an existing quality control (QC) phantom with similar attenuation values to breast tissue. Based on this, a liquid was chosen to use in further experiments with this home-made (HM) phantom.

3.2 Materials and methods

3.2.1 Breast tissue

In order to find the best representative liquid to use for the phantom, it is important to have a closer look at the type of tissues that breast tissue consists of. Breast tissue mainly consists of fatty tissue and fibroglandular tissue, the difference between these tissues becomes more apparent when comparing them on a CT scan or mammogram, as is seen in Figure 9. On the scan the fibroglandular tissue appears white, whilst the fatty tissue appears to be darker [29, 30]. Fibroglandular tissue is a dense type of tissue and consists of fibrous connective tissue and glandular tissue, which includes the milk glands. Because this is a dense tissue, it appears white on mammograms and (breast) CT scans. Fatty tissue is not as dense, appearing dark on mammograms and (breast) CT scans. Tumor tissue appears in a similar way as fibroglandular tissue on these scans, making the detection of a tumor in this type of tissue difficult compared to a tumor in fatty tissue [29, 30, 31].

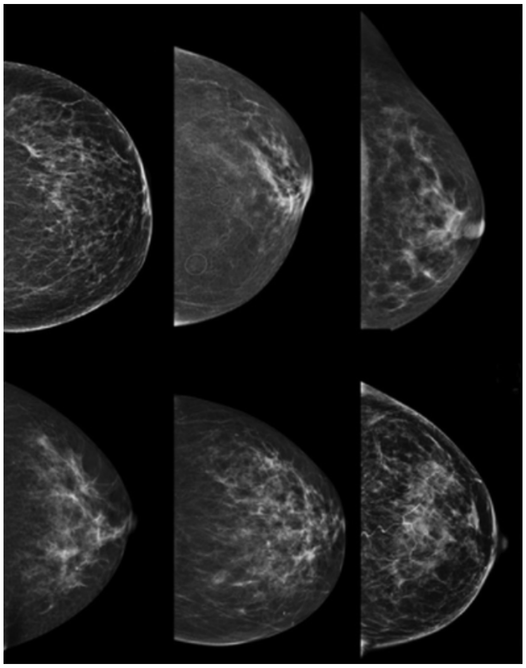


Figure 9: Mammogram with scattered areas of fibroglandular tissue. The light areas are fibroglandular tissue and the darker areas are fatty tissue [28].

3.2.2 Hounsfield Unit

The use of Hounsfield Units (HU) helps when differentiating between tissue types in reconstructed BCT images. These values act as a scale, comparing tissue attenuation coefficients with the attenuation coefficient of distilled water in the same conditions. For this reason, HU values are often used to help identify tissues that might look similar in a BCT scan [32]. The accuracy of a system when determining the HU is of importance when assessing the performance of a BCT system, as this could be a determining factor in detecting a tumor in breast tissue.

When recreating characteristics of breast tissue in a phantom used in BCT, choosing a substance with similar HU is of great importance. The HU values of breast tissue consists of ranges for the different types of tissue present within the breast tissue itself, the values differ based on the volumes in which they are present. For fatty tissue the HU values range from -210 to -100 HU, while for fibroglandular tissue the HU values range from -99 to +100 HU [29].

3.2.3 Data acquisition

The scans were acquired using a breast CT from Koning Co. and four different phantom options. The first phantom was the available QC phantom, specifically made to mimic breast tissue and to be used with the BCT. The other three phantoms had the same plastic tube casing, but different liquids were used each time, creating the different options for the HM phantom. The tube and part of the setup can be seen in Figure 2. The first liquid was regular olive oil, the second one was extra virgin olive oil and lastly tap water was used. All the scans were made using the same BCT settings, which are listed in Table 1.

Table 1: The BCT settings used in the first measurements.

Current X-ray tube	50 mA
Tube voltage	49 kVp
Estimated dose	5.8 mGy
Pulse width	5 ms
Scan duration	10 s
Pulses/rev	225
Total pulses	225
Filter	Aluminium filter

To determine what liquid had the most resemblance to breast tissue, and in this case the QC phantom, the HU values of the HM phantoms were compared to the acquired HU value of the QC phantom. The HU values were acquired by selecting a large region of interest (ROI) and measuring the average HU, in some cases the HU was measured for multiple ROIs along the vertical direction.

3.3 Results

In Table 2 the measured Hounsfield Unit values are displayed for the different phantoms. When these results get compared, the most fitting liquid can be chosen to use in the HM phantom for further experiments. In Table 2 can be seen that the HU values vary between the phantoms and also in the phantoms themselves. The QC phantom shows minimal variation, but the olive oil phantom shows a large difference of 81.8 HU along the height of the tube.

Table 2: The acquired Hounsfield Unit values for the different phantoms.

Phantom type	**Hounsfield Unit value**
QC phantom	-111.5 HU
	-116.3 HU
	-118.9 HU
Olive oil phantom	-244.5 HU
	-162.7 HU
Extra virgin olive oil phantom	-244.6 HU
Tap water phantom	45.4 HU

3.4 Discussion and conclusion

From the results can be seen that the HU values differ between the liquids for the HM phantom, the greatest difference is between tap water and olive oil. Extra virgin olive oil and regular olive oil showed a difference of around 0.1 HU, while the difference with tap water was around 290 HU. An important note is the varying HU values inside the phantoms themselves, which is a result of the spectrum varying over the detector. It can be seen that this difference is smaller for the QC phantom compared to the olive oil phantom, this can be explained by the difference in shape and size between the QC phantom and the HM phantoms. The QC phantom is shorter, vertically, than the olive oil phantom and has a round, half-sphere shape in contrast to the cylindrical tube shape of the olive oil phantom. When taking advantage of the varying spectrum and comparing the HU values of the different liquids with the HU values of the QC phantom, tap water was chosen to be used in the HM phantom. However, the difference between the HU of tap water and the QC phantom, which is around 156.9 HU, is still considered to be large and further testing should be done to find a more suitable liquid. A consideration that should be taken into account is the homogeneity of the liquids used, while actual breast tissue is a representation of heterogeneous tissue with the presence of fatty and fibroglandular tissue. A homogeneous liquid or substance will always represent a mean HU value of fatty and fibroglandular tissue, in contrast to the changing HU value in real breast tissue. This constant change in value adds a difficulty factor for detecting tumors which varies depending on the tumor location and surrounding tissue type.

4 Comparing reconstruction algorithms using the MTF

4.1 Goal of the experiment

The goal of this experiment was to reconstruct the images of the HM phantom, with an iodine rod inserted into it, using different reconstruction algorithms and additional filtering. The HM phantom should be a suitable, simplified representation of a contrast-enhanced tumor in breast tissue, in order for the experiments to have usable results. In addition, the acquired images and the MTF of these images were used to compare the different algorithms.

4.2 Materials and methods

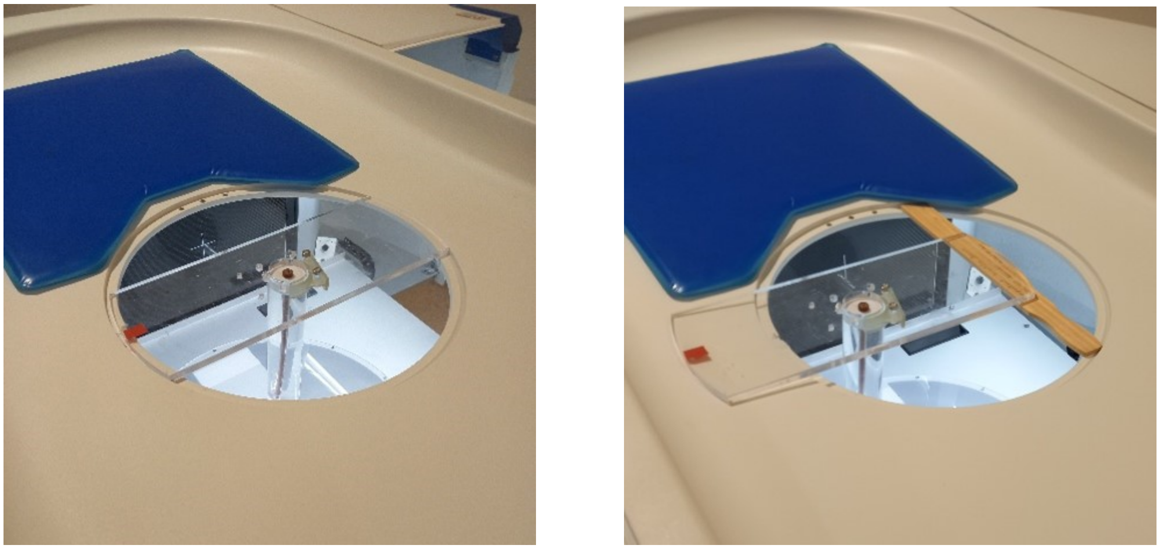
4.2.1 Data acquisition

In this experiment, the final HM phantom from Section 3 was used in combination with an iodine rod to mimic the contrast fluid used when monitoring tumor growth in breast tissue as explained in Section 2.1.1. This phantom is shown in Figure 2. Two groups of six measurements were done with the BCT, but only one of the twelve total scans was used for comparing the reconstruction algorithms. The first half of the first group of measurements was done with the settings from Table 1 and the phantom in a ”centred” position (CP), as shown in Figure 10a, while changing the scan duration to 10, 7 and 5 seconds. After these measurements, the settings were changed to the settings seen in Table 3 and more measurements were done with a scan duration of 10, 7 and 5 seconds.

Table 3: The BCT settings after changing the original settings.

Current X-ray tube	32 mA
Tube voltage	65 kVp
Estimated dose	3.7 mGy
Pulse width	5 ms
Scan duration	10 s, 7 s, 5 s
Pulses/rev	225
Total pulses	225

The second group of measurements was done with the HM phantom in ”off-centred” position (OCP), as shown in Figure 10b. This was achieved by moving the phantom 4 cm to the left from the middle position, to make this possible part of the phantom was balanced on a wooden bar. The same measurements were done as described before but with the phantom in OCP instead of CP. By doing these measurements, OCP and CP with different scan duration, the spatial resolution of the BCT was tested. It was assumed that the scan with the lowest scan duration (5 seconds) and the phantom in OCP would appear the most blurry.



(a) The home-made phantom set up in the "centred" position in the breast CT.

(b) The home-made phantom set up in the "off-centred" position in the breast CT.

Figure 10: The positioning of the phantom for the "centred" (a) and "off-centred" (b) measurements.

4.2.2 Data processing

After acquiring the images using the BCT they were reconstructed using different reconstruction algorithms, with and without additional filtering in the projection domain. Using the acquired data, the MTF was calculated and graphed using Python following the procedure explained in Section 2.2.1. The actual Python code used for determining the MTF of the reconstructed images is shown in Appendix B. The reconstruction algorithms used were FDK (Feldkamp-Davis-Kress) back projection with different filters, and IR with different filters and 100 iterations. The exact filters used are Cosine, Hamming, Hann, Ramp and Shepp-Logan filters for FDK back projection, and FISTA (Fast Iterative Shrinkage Thresholding Algorithm) 100, MLTR (Maximum Likelihood Transmission) 100 and SART (Simultaneous Algebraic Reconstruction Technique) 100 filters for IR.

4.3 Results

After evaluating the CP scans, it was decided not to include the OCP scans in the reconstruction process or further evaluate them due to issues with the setup.

4.3.1 The edge-spread function plots

During the process of determining the MTF, the ESF was plotted for all reconstructed images. This was done according to the methods described in Section 2.2.1. The resulting plots can be seen in Figure 11.

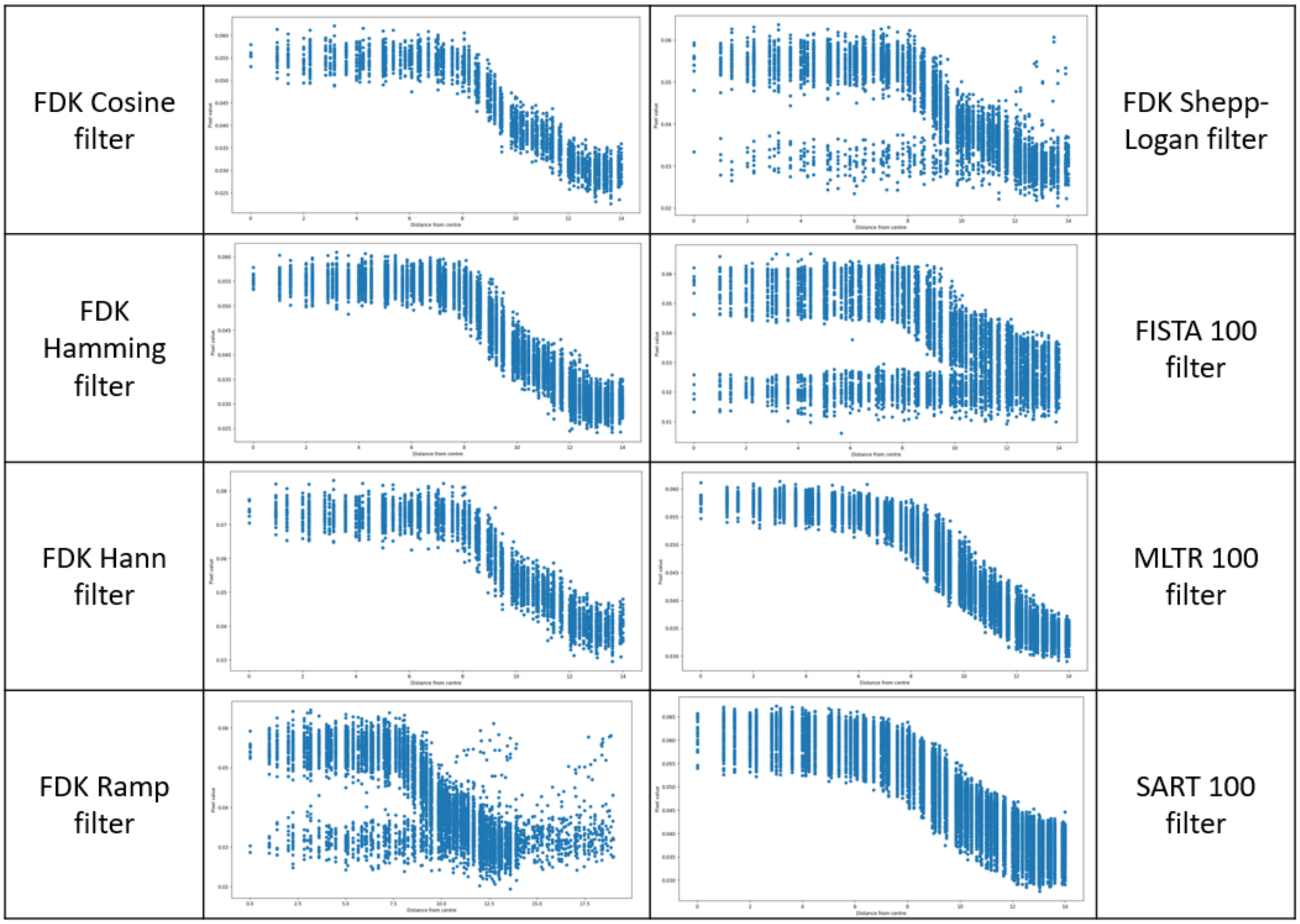


Figure 11: The ESF of all reconstruction methods used (all including additional filtering).

The data points are not uniformly spaced and there is noise visible, especially for the FDK Ramp filter and FDK Shepp-Logan filter with additional filtering. To create a uniform overview of the data, the data was re-binned. The result is shown in Figure 12.

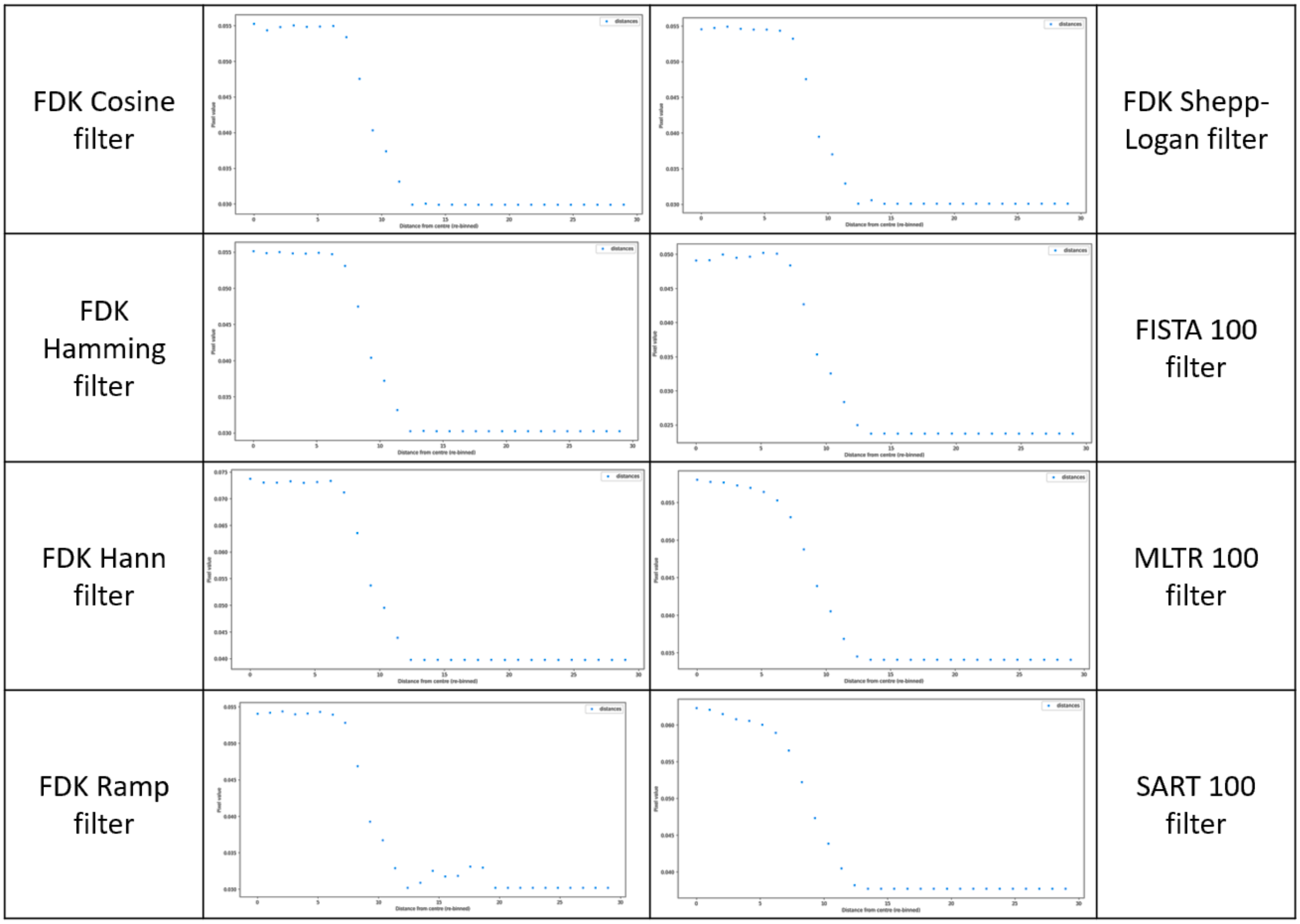


Figure 12: The re-binned ESF of all reconstruction methods used (all including additional filtering).

4.3.2 The reconstructed images and the MTF

In Figure 13 the images resulting from FDK back projection using different filters are shown. The first five images where acquired using the different filters from Section 4.2.2 and additional filtering. The sixth image is reconstructed using FDK and the Shepp-Logan filter, but without additional filtering of the projections. From the images can be seen that the amount of blurring and noise differs between the filters used. In all images the iodine rod is visible as a white circle on a dark background. The inside and outside of the circle are not one uniform color and the edges are not distinctly defined, but around the circle is a much darker rim visible that separates the undefined edges of the circle from the background. Figure 13F has less blurring, but seemingly more noise present.

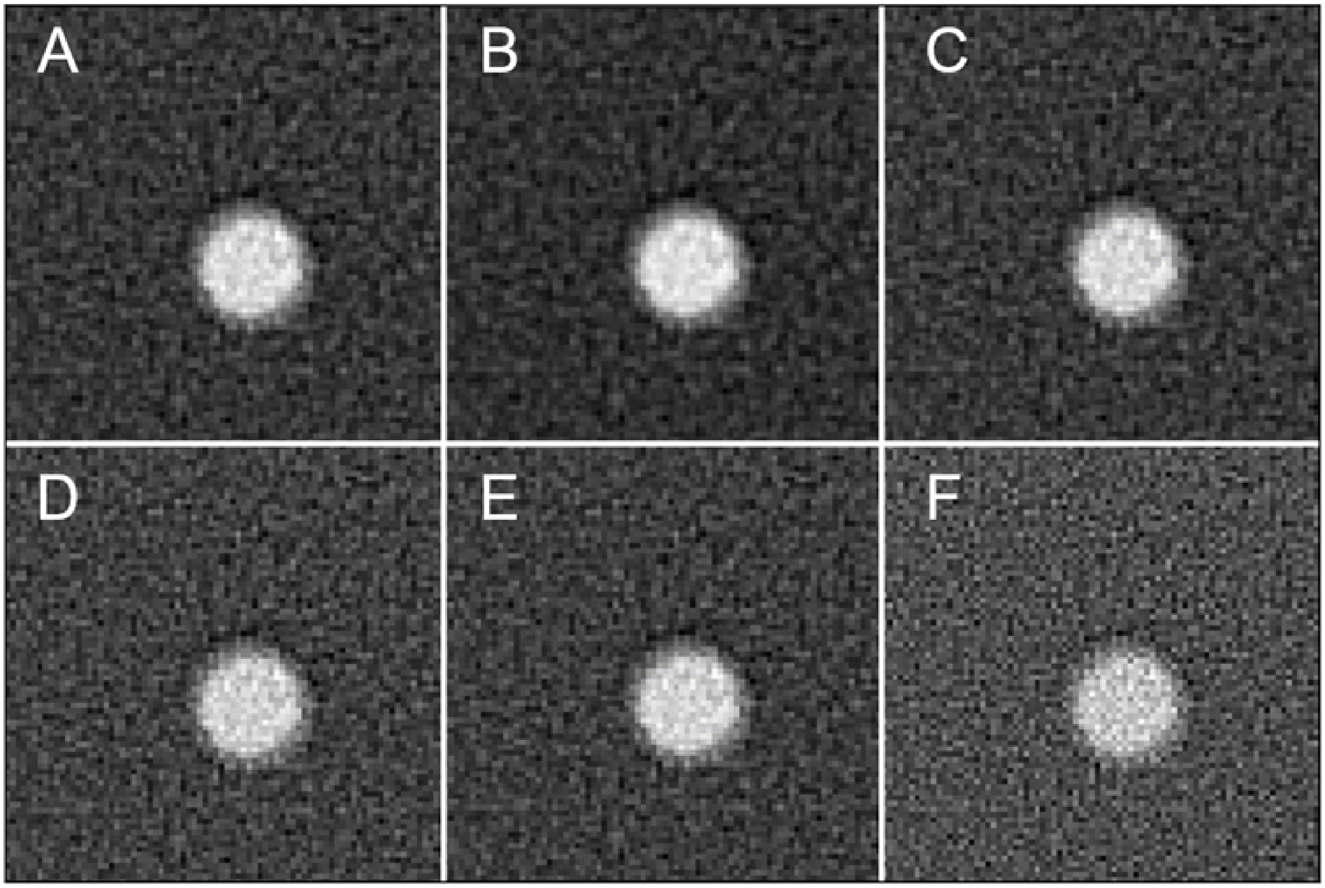


Figure 13: The reconstructed images acquired with the FDK back projection method and different filtering. A) FDK Cosine filter with additional filtering. B) FDK Hamming filter with additional filtering. C) FDK Hann filter with additional filtering. D) FDK Ramp filter with additional filtering. E) FDK Shepp-Logan filter with additional filtering. F) FDK Shepp-Logan filter without additional filtering.

To determine which FDK method should be displayed without additional filtering, the reconstructed images were compared initially. From this comparison, Figure 13D and Figure 13E were determined to be the clearest. To further compare these filters, the MTFs were graphed, as shown in Figure 14, and compared. From this comparison it was decided that the FDK with Shepp-Logan filter would be displayed without additional filtering.
In Figure 14 can be seen that all graphs have a dip between 0.75 and 1.00, afterwards most of the graphs go back up and down again with a smaller slope than they initially went down with. This creates a bump in the graphs.

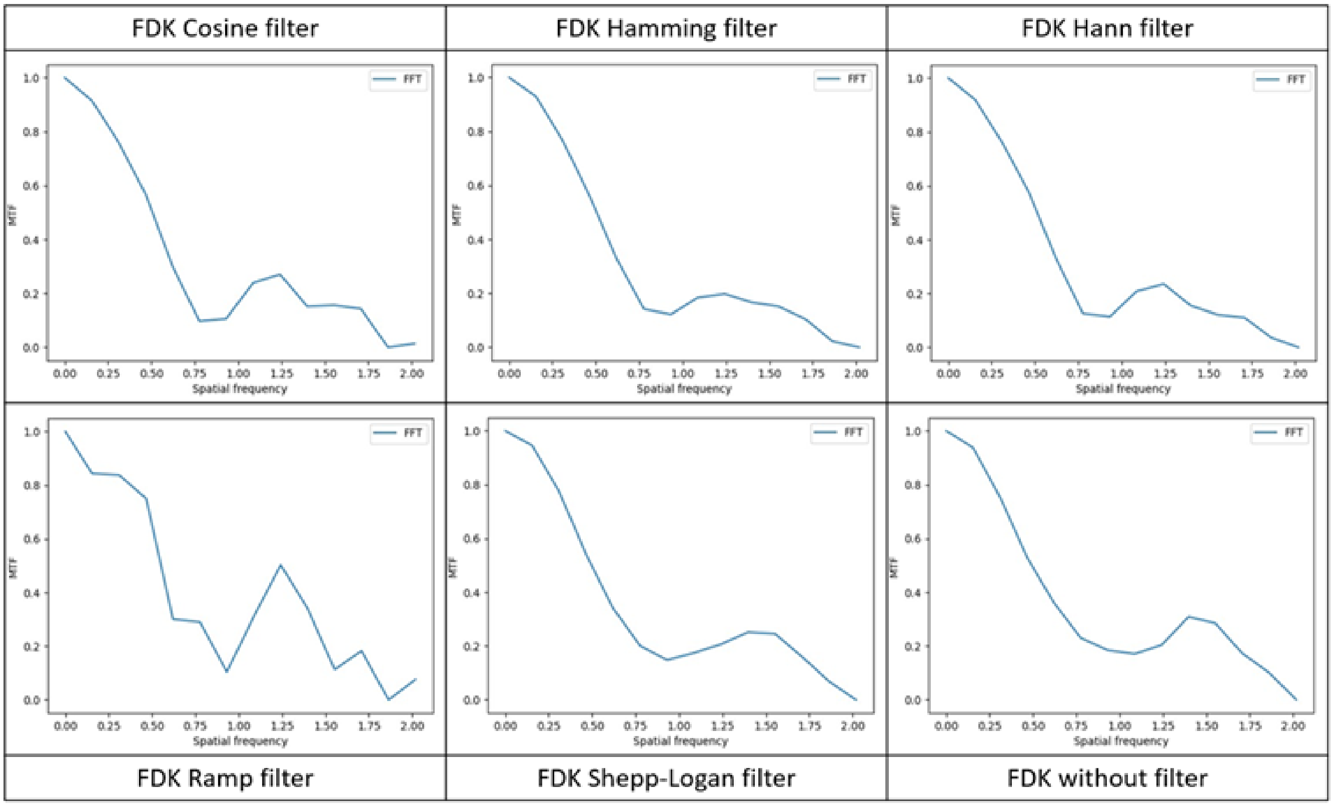


Figure 14: The graphs of the MTF over the spatial frequency for the FDK back projection method with the different filters and additional filtering. On the bottom right is the graph of the MTF over the spatial frequency for the FDK back projection method with the Shepp-Logan filter and without additional filtering.

For a better comparison between the MTF graphs, the graphs can be combined into one plot. The result can be seen in Figure 15.

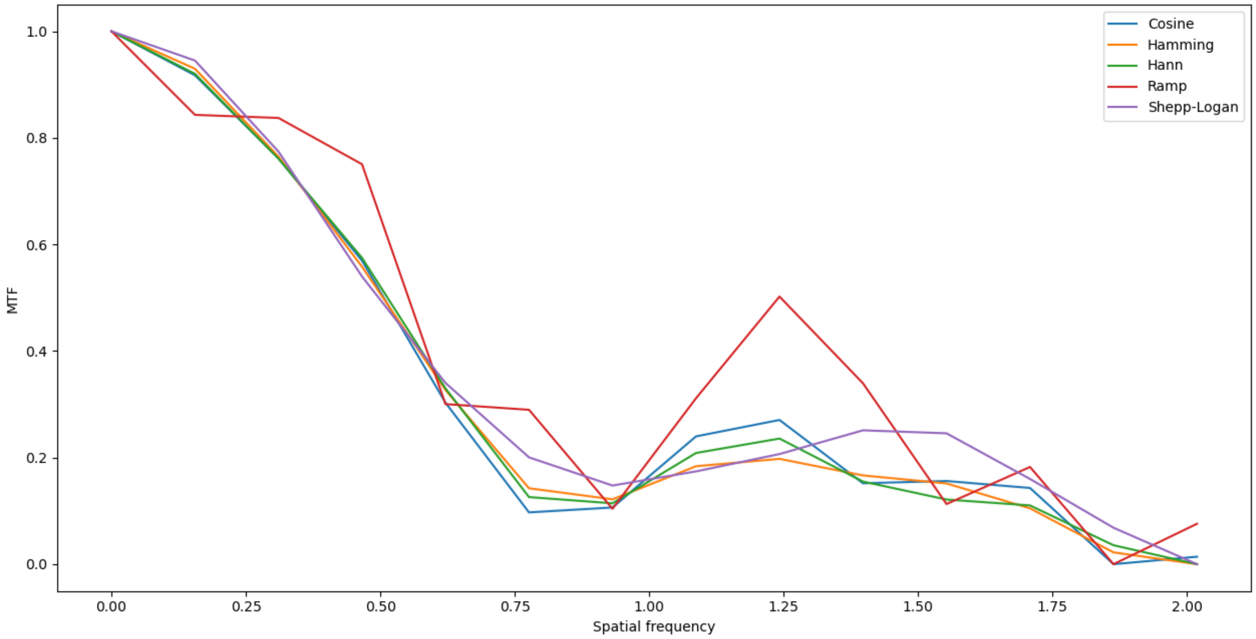


Figure 15: A collection of all (FDK) graphs of the MTF over the spatial frequency.

Figure 16 shows the images reconstructed using iterative reconstruction filters and additional filtering. In Figure 16A a dark rim is visible around the circle, similar to the rim seen in Figure 13, but in the other images there is no prominent dark rim visible.

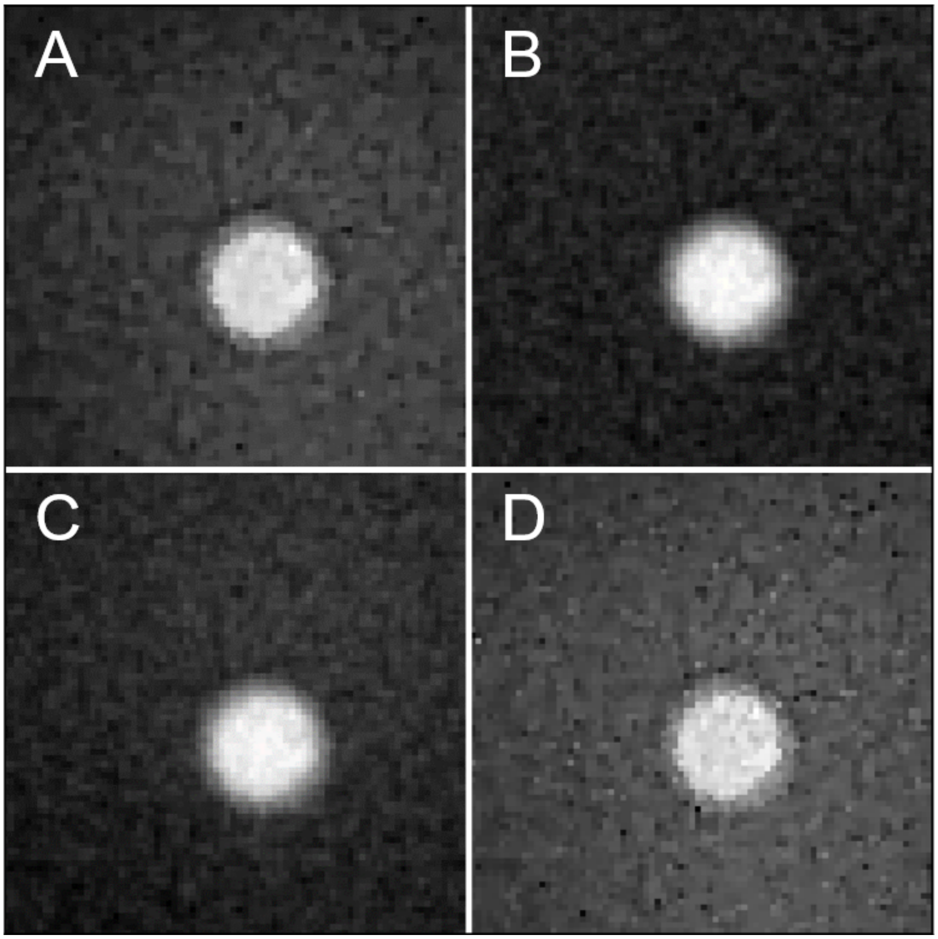
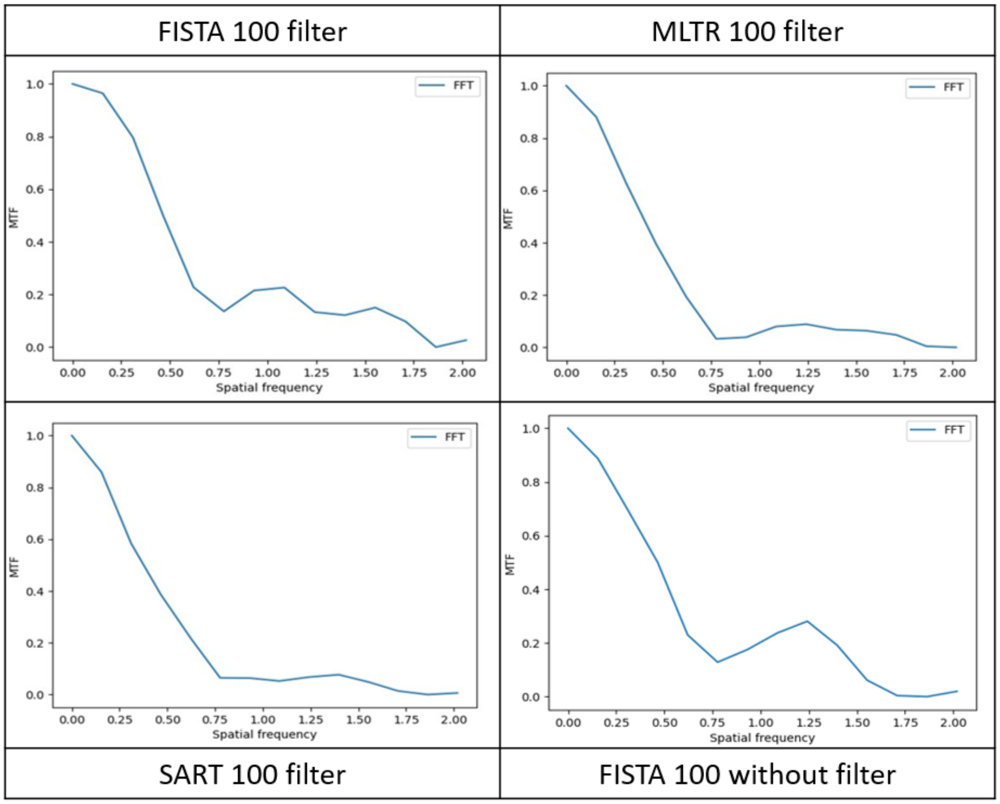


Figure 16: The reconstructed images acquired with different iterative reconstruction filters and additional filtering. A) FISTA 100 with additional filtering. B) MLTR 100 with additional filtering. C) SART 100 with additional filtering. D) FISTA 100 without additional filtering.

To determine which method should be depicted without additional filtering, the same procedure was followed as for the FDK method. The reconstructed images from Figure 16 were compared to each other by solely looking at the amount of noise and blurring visible. Afterwards, the graphs of the MTF, as shown in Figure 17, were compared and FISTA 100 was picked to be displayed without additional filtering.



FISTA 100 filter

MLTR 100 filter

SARS 100 filter

FISTA 100 without filter

Figure 17: The graphs of the MTF over the spatial frequency for the different iterative reconstruction methods.

Again, a combined plot is made for a better comparison between the MTF graphs. This can be seen in Figure 18.

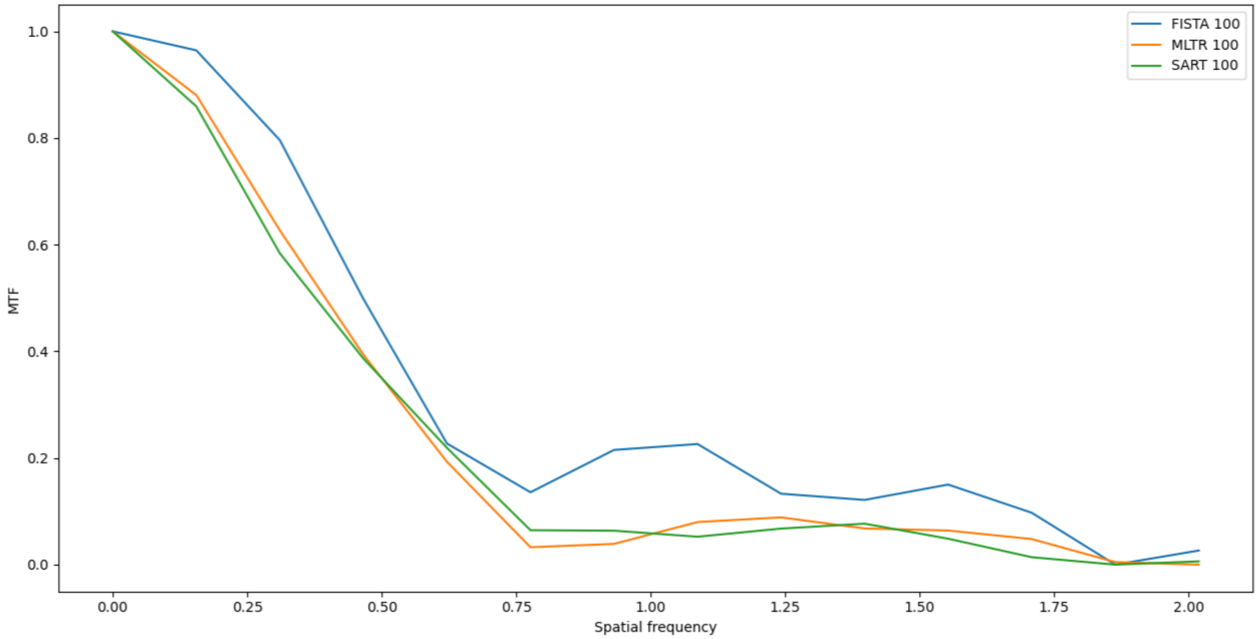


Figure 18: A collection of all (IR) graphs of the MTF over the spatial frequency.

As a final comparison, the two best MTF graphs are combined into one plot. The final plot can be seen in Figure

19. The reasoning behind choosing these specific graphs is explained in Section 4.4.

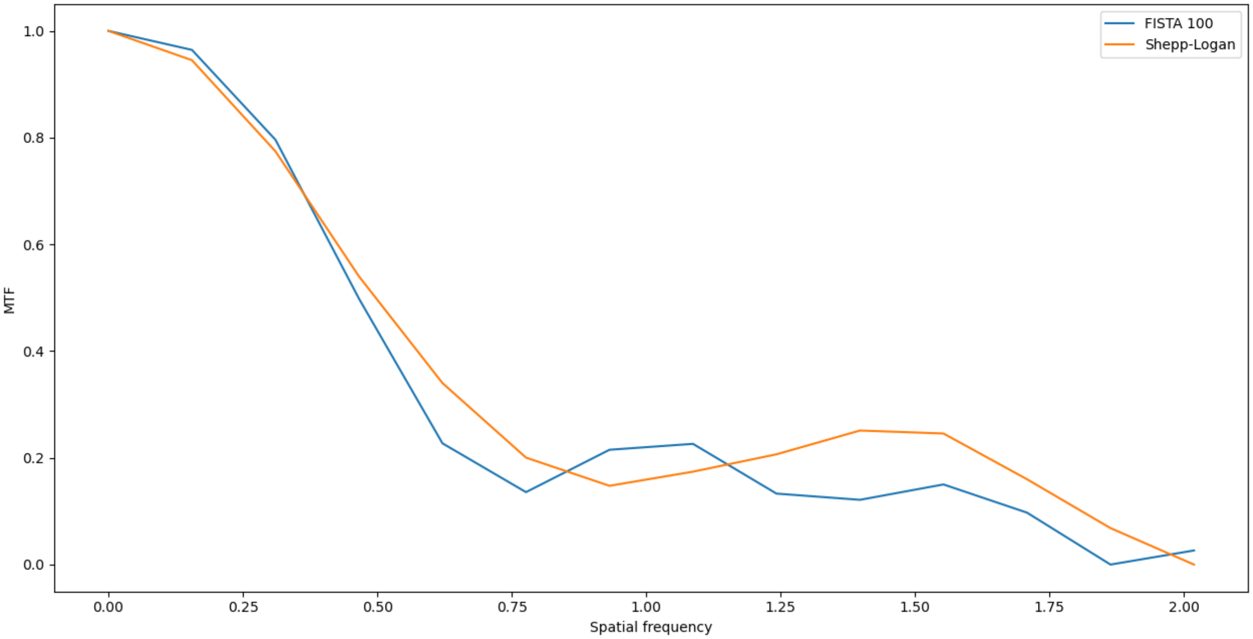


Figure 19: The MTF graphs of FISTA 100 and FDK Shepp-Logan combined into one plot for better comparing.

4.4 Discussion

In most of the reconstructed images, a high amount of noise is visible even after additional filtering. The images with the most amount of noise after additional filtering are Figure 13D and 13E (FDK with Ramp filter and Shepp-Logan filter including additional filtering) for FDK back projection, and Figure 16B (MLTR 100 with additional filtering) for IR. The noise can cause problems and influence the accuracy of the calculation of the MTF. One of these problems is that the noise makes it more difficult for the Python script to carry out the segmentation and determine the middle of the circle, here the white circle caused by the presence of the iodine rod, resulting in diverging results. The noise in the ESF plots, as seen in Figure 11, could have been caused by the noise in the reconstructed images. A potential solution would be to increase the X-ray tube current [33].

When looking at the effect of the additional filtering, the different reconstruction algorithms show differences in their changes. For FDK back projection the difference between additional filtering and no additional filtering is more visible than for IR. When comparing Figure 13E and 13F it can be seen that 13F is noisier than 13E, meaning that additional filtering reduces noise. The difference from additional filtering can also be seen when looking at the MTF graphs in Figure 14, specifically looking at the graphs in the bottom right and middle. The graphs start off in similar ways, but start displaying changes around the dip in the graph. The graph on the bottom right has a wider dip, starting around 0.75 and ending around 1.25, than the graph in the middle, which starts around 0.75 and ends around 1.15. But because the right graph rises with a steeper slope, both graphs end up with a peak around the same spatial frequency. The middle graph seems to have a lower peak around 1.37 than the right graph, resulting in a smaller visible bump. This bump is present in all MTF graphs of FDK back projection and IR, but it is most apparent in the FDK back projection graphs. The nature of this bump is still unclear, but as it is visible in all MTF graphs it is likely caused by a common error source in all measurements. This source could be the phantom itself or a mistake in the Python code, further research should be done to determine the actual cause.

In Figure 16A and 16D can be seen that the additional filtering in IR methods results in minimal noise reduction, the change is not prominent. However, this does not apply to the MTF graphs, when looking at Figure 17 the graphs corresponding to FISTA 100 with and without additional filtering show prominent differences. The additional filtering seems to have reduced the overall hight of the bump, but in stead separates the one bump into two bumps. In addition, the total width of the bump gets increased with additional filtering from approximately

1.0 to 1.12.

When comparing the reconstructed images of FDK back projection and IR, the differences are very prominent. The images acquired with FDK, seen in Figure 13, look more grainy than the images seen in Figure 16. Also the dark rim around the white circle is more pronounced in Figure 13 than in Figure 16. This dark rim is likely caused by the plastic layer around the iodine rod, this layer seems to have a different Hounsfield Unit value than the iodine and the surrounding water. The darker complexion suggests that the HU value is lower than the HU value of the tap water that was used, which seems to be correct as the HU value of plastic should be around -47 HU [34]. Therefore, it causes a dark rim to appear that clearly sets apart the white circle from the background and redefines the edges that might have been lost otherwise. This eliminates some of the problems caused by noise and blurring in the images. Even though this effect seems beneficial for recognizing the iodine rod and useful for segmentation, it has adverse effects on the outcome of the experiments and the conclusion of this report as it is no longer a suitable phantom representing a contrast-enhanced tumor in breast tissue. For any further research it is advised to either remove the plastic layer or choose a liquid with a HU value closer to the plastic than tap water.
However, this dark rim can be of use when comparing the reconstruction algorithms solely based on the reconstructed images as it presents a measure of the details that are lost during reconstruction. The dark rim is the result of the thin layer of plastic around the iodine rod, which was present during all measurements. But when comparing the images in Figure 13 and Figure 16, this dark rim has varying visibility. If the dark rim is not (or barely) visible it means that in a realistic setting, inconsistencies with a similar or smaller size are also not visible after reconstruction. The visibility is similar in the images shown in Figure 13 except for Figure 13F, here the dark rim is thinner than in the other images and is more difficult to differentiate from the background because of the increased amount of noise. Figure 13B and 13E seem to have the best visibility, in Figure 13B the dark rim stands out more than in Figure 13E but it also has more blurring. In Figure 13E the rim seems more detailed and slightly thinner than in Figure 13B, but not less visible. Therefore, Figure 13E is decided to have the overall best visibility of the dark rim. In Figure 16, the visibility of the dark rim is significantly better in Figure 16A than in the other images. Figure 16B has the least visible dark rim, as the background overall seems darker than the other images and closer in shade to the dark rim. When comparing Figure 13E and Figure 16A, it can be seen that the dark rim is more visible and detailed in Figure 13E.

When comparing the MTF graphs of FDK back projection, all graphs look similar except for the graph belonging to FDK with Ramp filter which does not seem to be a suitable option as the graph does not run as smoothly as the others. This is clearly visible in Figure 15. To pick the filter with the best MTF graph, it is important to look at the initial slope and the minimum MTF value this slope leads to. When looking at Figure 15 all initial slopes seem similar. But, around 0.50 on the x-axis, the slope of the graph belonging to FDK with Shepp-Logan filter starts to decrease compared to the other graphs. This results in a higher initial minimum of approximately 0.15. When following the rule of thumb from Section 2.2, the FDK with Shepp-Logan filter has the best MTF and MTF graph.

For IR it is less difficult to separate the different MTF graphs, as can be seen in Figure 18. The graphs belonging to IR with MLTR 100 filter and IR with SART 100 filter have a similar initial slope, but the graph belonging to IR with FISTA 100 filter has a less steep initial slope. This results in a higher MTF and is therefore the most favorable option.

Finally, when comparing the MTF graphs of FDK with Shepp-Logan filter and IR with FISTA 100 filter it will become clear which reconstruction algorithm and filter combination is the best. In Figure 19 can be seen that the FISTA 100 graph starts off a little higher than the Shepp-Logan graph, while the slope of the Shepp-Logan graph is less steep than the FISTA 100 graph. This means that the Shepp-Logan graph has a higher MTF over more spatial frequency than the FISTA 100 graph.

4.5 Conclusion

In conclusion, the additional filtering is beneficial for image reconstruction and evaluation, but there is room for improvement. The MTF is able to be determined of the reconstructed images using Python, but the graphs show the presence of a common error source of which the nature is unclear. And finally, the home-made phantom is deemed unsuitable for the purpose of representing a contrast-enhanced tumor in breast tissue because of the HU value of the plastic. Because of this, it is not possible to draw any reliable conclusions concerning the reconstruction ability, for a contrast-enhanced tumor, of the BCT and reconstruction algorithms based on the MTF or about the detectability of this tumor influenced by the spatial resolution. To be able to draw these conclusions, the measurements need to be redone using a different phantom.

However, when solely basing the conclusion on the comparison of the reconstructed images and the visibility of the dark rim, it can be concluded that the FDK back projection reconstruction algorithm in combination with the Shepp-Logan filter and additional filtering comes out on top. This is also the case for the MTF graphs.

Overall, when looking past the unsuitability of the phantom, it can be concluded that the FDK back projection reconstruction algorithm in combination with the Shepp-Logan filter and additional filtering is the best reconstruction algorithm (combination) based on the reconstructed images and MTF.

5 Bibliography

[1] Bray F, Ferlay J, Soerjomataram I, Siegel RL, Torre LA, Jemal A. Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: A Cancer Journal for Clinicians. 2018.

[2] Borstkanker in Nederland; 2023. Accessed: 2023-11-26. https://iknl.nl/borstkankercijfers.

[3] Germann M, Shim S, Angst F, Saltybaeva N, Boss A. Spiral breast computed tomography (CT): signal-to-noise and dose optimization using 3D-printed phantoms. European Radiology. 2020.

[4] Paap E, Holland R, den Heeten GJ, van Schoor G, Botterweck AAM, Verbeek ALM, et al. A remarkable reduction of breast cancer deaths in screened versus unscreened women: a case-referent study. Cancer Causes amp; Control. 2010.

[5] Tabar L, Yen MF, Vitak B, Chen HHT, Smith RA, Duffy SW. Mammography service screening and mortality in breast cancer patients: 20-year follow-up before and after introduction of screening. The Lancet. 2003.

[6] Massat NJ, Dibden A, Parmar D, Cuzick J, Sasieni PD, Duffy SW. Impact of Screening on Breast Cancer Mortality: The UK Program 20 Years On. Cancer Epidemiology, Biomarkers amp; Prevention. 2016.

[7] Zhu Y, O'Connell AM, Ma Y, Liu A, Li H, Zhang Y, et al. Dedicated breast CT: state of the art—Part II. Clinical application and future outlook. European Radiology. 2022.

[8] Boone JM, Kwan ALC, Yang K, Burkett GW, Lindfors KK, Nelson TR. Computed Tomography for Imaging the Breast. Journal of Mammary Gland Biology and Neoplasia. 2006.

[9] Gao Y, Heller SL. Abbreviated and Ultrafast Breast MRI in Clinical Practice. RadioGraphics. 2020.

[10] Tohno E, Ueno E. Ultrasound (US) diagnosis of nonpalpable breast cancer. Breast Cancer. 2005.

[11] Prasad SN, Houserkova D. THE ROLE OF VARIOUS MODALITIES IN BREAST IMAGING. Biomedical Papers. 2007.

[12] Akashi-Tanaka S, Fukutomi T, Sato N, Miyakawa K. The role of computed tomography in the selection of breast cancer treatment. Breast Cancer. 2003.

[13] Sarno A, Mettivier G, Russo P. Dedicated breast computed tomography: Basic aspects. Medical Physics. 2015.

[14] Kaiser B. Koning Breast CT — Improved Breast Imaging Technology - Koning Health; 2023. Accessed: 2023-11-27. https://www.koninghealth.com/.

[15] Zhu Y, O'Connell AM, Ma Y, Liu A, Li H, Zhang Y, et al. Dedicated breast CT: state of the art—Part I. Historical evolution and technical aspects. European Radiology. 2021.

[16] Wienbeck S, Fischer U, Luftner-Nagel S, Lotz J, Uhlig J. Contrast-enhanced cone-beam breast-CT (CBBCT): clinical performance compared to mammography and MRI. European Radiology. 2018 March;28(9):3731–3741. Available from: http://dx.doi.org/10.1007/s00330-018-5376-4.

[17] Prionas ND, Lindfors KK, Ray S, Huang SY, Beckett LA, Monsky WL, et al. Contrast-enhanced Dedicated Breast CT: Initial Clinical Experience. Radiology. 2010 September;256(3):714–723. Available from: http://dx.doi.org/10.1148/radiol.10092311.

[18] Prionas ND, Aminololama-Shakeri S, Yang K, Martinez SR, Lindfors KK, Boone JM. Contrast-enhanced dedicated breast CT detection of invasive breast cancer preceding mammographic diagnosis. Radiology Case Reports. 2015;10(2):936. Available from: http://dx.doi.org/10.2484/rcr.v10i2.936.

[19] Richard S, Husarik DB, Yadava G, Murphy SN, Samei E. Towards task-based assessment of CT performance: System and object MTF across different reconstruction algorithms: Towards task-based assessment of CT performance. Medical Physics. 2012 June;39(7Part1):4115–4122. Available from: http://dx.doi.org/10.1118/1.4725171.

[20] Kayugawa A, Ohkubo M, Wada S. Accurate determination of CT point-spread-function with high precision. Journal of Applied Clinical Medical Physics. 2013 July;14(4):216–226. Available from: http://dx.doi.org/10.1120/jacmp.v14i4.3905.

[21] Oostveen LJ, Boedeker KL, Brink M, Prokop M, de Lange F, Sechopoulos I. Physical evaluation of an ultra-high-resolution CT scanner. European Radiology. 2020 February;30(5):2552–2560. Available from: http://dx.doi.org/10.1007/s00330-019-06635-5.

[22] Itrić K, Modrić D, Milković M. Edge spread function for the paper-ink system. Nordic Pulp amp; Paper Research Journal. 2018 August;33(3):542–547. Available from: http://dx.doi.org/10.1515/npprj-2018-3054.

[23] Ahmed SN. In: Position-sensitive detection and imaging. Elsevier; 2015. p. 435–475. Available from: http://dx.doi.org/10.1016/B978-0-12-801363-2.00007-3.

[24] Matsuura K, Ichikawa K, Kawashima H. Task-specific spatial resolution properties of iterative and deep learning-based reconstructions in computed tomography: Comparison using tasks assuming small and large enhanced vessels. Physica Medica. 2022 March;95:64–72. Available from: http://dx.doi.org/10.1016/j.ejmp.2022.01.009.

[25] Willemink MJ, Leiner T, de Jong PA, de Heer LM, Nievelstein RAJ, Schilham AMR, et al. Iterative reconstruction techniques for computed tomography part 2: initial results in dose reduction and image quality. European Radiology. 2013 January;23(6):1632–1642. Available from: http://dx.doi.org/10.1007/s00330-012-2764-z.

[26] Fleischmann D, Boas FE. Computed tomography—old ideas and new technology. European Radiology. 2011 January;21(3):510–517. Available from: http://dx.doi.org/10.1007/s00330-011-2056-z.

[27] Mileto A, Guimaraes LS, McCollough CH, Fletcher JG, Yu L. State of the Art in Abdominal CT: The Limits of Iterative Reconstruction Algorithms. Radiology. 2019 December;293(3):491–503. Available from: http://dx.doi.org/10.1148/radiol.2019191422.

[28] Winkler NS, Raza S, Mackesy M, Birdwell RL. Breast Density: Clinical Implications and Assessment Methods. RadioGraphics. 2015 March;35(2):316–324. Available from: http://dx.doi.org/10.1148/rg.352140134.

[29] Akuoko DEM, Osorio DEV, Van Herk PM, OTHERS. Hounsfield-based Automatic Evaluation of Volumetric Breast Density on Radiotherapy CT-Scans. 2021. Available from: https://arxiv-org.ezproxy2.utwente.nl/pdf/2110.04929.pdf.

[30] Boyd NF, Martin LJ, Bronskill M, Yaffe MJ, Duric N, Minkin S. Breast Tissue Composition and Susceptibility to Breast Cancer. JNCI Journal of the National Cancer Institute. 2010 July;102(16):1224–1237. Available from: http://dx.doi.org/10.1093/jnci/djq239.

[31] Fibroglandular Density: Scattered, Concerns What It Means; 2023. Accessed: 2023-11-29. https://my.clevelandclinic.org/health/articles/22874-fibroglandular-density.

[32] Shah JP, Mann SD, McKinley RL, Tornai MP. Characterization of CT Hounsfield Units for 3D acquisition trajectories on a dedicated breast CT system. Journal of X-Ray Science and Technology. 2018 August;26(4):535–551. Available from: http://dx.doi.org/10.3233/XST-17350.

[33] Mberato AJGM, Ratini NN, Yuliara IM, Irhas R. ANALYSIS OF THE EFFECT OF THE X-RAY TUBE CURRENT ON UNIFORMITY IMAGE NOISE VALUES. Indonesian Physical Review. 2022 December;6(1):11–19. Available from: http://dx.doi.org/10.29303/ipr.v6i1.183.

[34] Choi HJ, Lee HJ, Kang SG. The Clinical Significance of Hounsfield Number of Metallic and Non-Metallic Foreign Bodies in the Soft Tissue. Soonchunhyang Medical Science;16(2):226-30. Available from: https://jsms.sch.ac.kr/upload/pdf/SMS_16_2_226_230.pdf.

A Appendix

A.1 Generating an image

```
import numpy as np
import cv2
from skimage.util import random_noise
from skimage.filters import gaussian
from skimage.util import img_as_ubyte

DEBUG = True



def load_png_image(path='testing2.png'):
    img=cv2.imread(path)
    # it is also possible to use cv2.imread('name',0) to get a greyscale.
    img_gray=cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    if DEBUG:
        cv2.namedWindow("Display", cv2.WINDOW_NORMAL)

        cv2.imshow('Display',img_gray)
        # when no waitkey is used, the image disappears immediately after appearing. So go be able to actually look at the aquired image, waitKey(0) should be used.
        cv2.waitKey(0)
        # the image can be seen as an array of values (grayscale values), when not automated these can be used for manually setting a threshold.
        print(np.array(img_gray))
        print(np.array(img_gray).shape)



    return img_gray

def make_noise(img_gray):
    var=np.arange(0.01,0.1,0.01)
    #var=[0.01]
    #var=[0.06]
    #print(var)
    sigma=range(1,10)
    #sigma=[1]

    for v in var:
        for s in sigma:
            v=np.round(v,2)
            #otherwise it gives no rounded 6 and 7
            #img_gray_ns = random_noise(img_gray,mode='gaussian',seed=None,clip=True,var=v) #should be var=v
            img_gray_ns = random_noise(img_gray,mode='gaussian',seed=None,clip=True,var=0.02)
            img_gray_ns = img_as_ubyte(img_gray_ns,force_copy=False)

            #img_gray_ns = gaussian(img_gray_ns, sigma =s, output=None, mode='nearest', cval=0, preserve_range=False,truncate=4.0) #should be simga=s
            img_gray_ns = gaussian(img_gray_ns, sigma =2, output=None, mode='nearest', cval=0, preserve_range=False,truncate=4.0)
            img_gray_ns = img_as_ubyte(img_gray_ns,force_copy=False)
            #print(v)
            #print(s)
            #print(img_gray_ns)

            np.save('.\data\image_{0}_{1}.npy'.format(v,s),img_gray_ns)

    if DEBUG:
        img_gray_ns = random_noise(img_gray,mode='gaussian',seed=None,clip=True,var=0.02)
        img_gray_ns = img_as_ubyte(img_gray_ns,force_copy=False)

        img_gray_ns = gaussian(img_gray_ns, sigma =2, output=None, mode='nearest', cval=0, preserve_range=False,truncate=4.0)
        img_gray_ns = img_as_ubyte(img_gray_ns,force_copy=False)
        cv2.namedWindow("Display2", cv2.WINDOW_NORMAL)

        cv2.imshow('Display2',img_gray_ns)
        # when no waitkey is used, the image disappears immediately after appearing. So go be able to actually look at the aquired image, waitKey(0) should be used.
        cv2.waitKey(0)

    return img_gray_ns
```

A.2 Determining the ESF

```
import cv2
import numpy as np
import math
import matplotlib.pyplot as plt

DEBUG = False

def thresholding(img_gray_ns):
    #ret,thresh=cv2.threshold(img_gray_ns,0,255,cv2.THRESH_TRIANGLE)
    ret,thresh=cv2.threshold(img_gray_ns,0,255,cv2.THRESH_OTSU)
    #for inverse: cv2.THRESH_BINARY_INV|cv2.THRESH_OTSU

    if DEBUG:
        print(ret)
        cv2.namedWindow("Display3", cv2.WINDOW_NORMAL)
        cv2.imshow('Display3',thresh)
        cv2.waitKey(0)

    return thresh


def center_coordinates(thresh):
    contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    # an empty list is created so a list can be made with the coordinates of the centroids.
    coordinates=[]

    # a for-loop is made for c in contours, here the weighted average (moments) are calculated and used for the determination of the centroids. This loop is repeated for
    # multiple (all) circles until no more are found.
    if DEBUG:
        cv2.namedWindow("Display4", cv2.WINDOW_NORMAL)

    for c in contours:
        M=cv2.moments(c)

        # the general formula for the x and y coordinates of the centroid, respectively, is: cX = M10/M00 and cY = M01/M00
        cX= int(M[’m10’]/M[’m00’])
        cY= int(M[’m01’]/M[’m00’])

        _,radius = cv2.minEnclosingCircle(c)
        radius = int(radius)
        print(’radius=’,radius)


        if DEBUG:
            # circles are drawn with certain radius and the determined coordinates. These circles are not visible.
            cv2.circle(thresh, (cX,cY), 5, (200, 200, 200), -1)
            # for visibility the centroids are marked.
            cv2.putText(thresh, ’centroid’, (cX - 25, cY - 25), cv2.FONT_HERSHEY_SIMPLEX, 0.5,(200,200,200),1)

            cv2.imshow(’Display4’, thresh)
            cv2.waitKey(0)

        # the coordinates are added into the previously empty list (as groups) by using append. When new coordinates are found, they are added to the list.
        #print(cX,cY)
        coordinates.append(cX)
        coordinates.append(cY)



    print(coordinates)
    return coordinates, radius

def get_value(img_gray_ns, coordinates, radius):
    the_values = []
    distance= []
    cX_1 = coordinates[0]
    print("cX",cX_1)
    cY_1 = coordinates[1]
    print("cY",cY_1)
    x_range = range(79) #100
    y_range = range(79) #100
    for x in x_range:
        for y in y_range:
            #print(x)
            #print(y)
            dx=abs(x-cX_1)
            dy=abs(y-cY_1)
            if dx**2 + dy**2 <= radius**2:
                values=img_gray_ns[y][x]
                #print(values)
                the_values.append(values)
                dist=math.sqrt((dx**2)+(dy**2))
                distance.append(dist)
    print("values",np.array(the_values).size)
    print("distance",np.array(distance).size)
    val_dist=np.column_stack((the_values, distance))

    print(np.array(val_dist).shape)
    plt.figure(0)
    plt.plot(val_dist[:,1],val_dist[:,0], '.', label="distances")
    plt.xlabel("Distance from the centre")
    plt.ylabel("Pixel value")
    plt.legend()
    plt.show()

    return values, dist, val_dist

def get_plot_values(val_dist):
    val_dist = val_dist[val_dist[:,1].argsort()] #sorts input values distribution based on second column which is distance from center

    distances = val_dist[:, 1]
    values = val_dist[:, 0]
    max_distance = 30
    distance_bins = 30 #number of bins for re-binning

    new_distances = np.linspace(0, max_distance, distance_bins) #creates new set of distances evenly spaced between 0 and max_distance
    new_values = np.zeros(distance_bins)
    for i in range(len(new_distances)-1): #loop iterates over new distances and calculates mean value in each bin
        bin_left = new_distances[i] #left and right boundaries of current bin
        bin_right = new_distances[i+1]
        values_in_bin = values[(distances >= bin_left) & (distances <= bin_right)] #extracts value that falls within current bin

        if len(values_in_bin) == 0: #if bin is empty the mean value is set to minimum value in new_values (excluding zero), otherwise its set to median of values in the bin
            mean_value_in_bin = np.min(new_values[new_values > 0])
        else:
            mean_value_in_bin = np.median(values_in_bin)

        new_values[i] = mean_value_in_bin #mean value is assigned to corresponding position in new_values

    plt.figure(0)
    plt.plot(new_distances[:-1],new_values[:-1],'.', label='distances')
    plt.xlabel("Distance from centre (re-binned)")
    plt.ylabel("Pixel value")
    plt.legend()
    plt.show()

    return new_values[:-1]
#values distribution is sorted by distance, re-binned values and plot re-binned values
```

A.3 The Fourier transform

```
import numpy as np
import matplotlib.pyplot as plt


# s=np.ones(100)

# s[-50:] = 0



# plt.figure(0)
# plt.plot()
# plt.show()

def do_fourier(val_2):
    val_2 = np.diff(val_2)
    fft_signal = np.fft.fft(val_2)
    freq_signal = np.fft.fftfreq(len(val_2), d=0.23)

    fft_signal = np.fft.fftshift(np.abs(fft_signal))
    freq_signal = np.fft.fftshift(freq_signal)

    mask = freq_signal >= 0
    fft_s =fft_signal[mask]
    freq_s = freq_signal[mask]

    return fft_s, freq_s
#fft_signal[mask], freq_signal[mask]


#fft_s, freq_s = do_fourier(val_dist - np.mean(val_dist))
def printing(fft_s,freq_s):
    # print(fft_s)
    abs_fft=np.abs(fft_s)
    max = np.amax(abs_fft)
    min = np.amin(abs_fft)
    norm_fft = (abs_fft - min)/(max-min)

    plt.figure(1)
    plt.plot(freq_s, norm_fft, label='FFT')
    #plt.plot(freq_s, np.abs(fft_s), label='FFT')
    plt.xlabel("Spatial frequency")
    plt.ylabel("MTF")
    plt.legend()
    plt.show()
    return

plt.show()
```

```
list_of_files = [’image_0.01_6.npy’, ’image_0.01_7.npy’, ’image_0.06_1.npy’, ’image_0.09_6.npy’]

for el in list_of_files:
    # if ’6.npy’ in el:
    #     if ’0.09_’ not in el:
    #         print(el)

    noise = float(el.split(’_’)[1])
    if noise <= 0.08:
        if ’6.npy’ in el:
            if ’0.09_’ not in el:
                print(el)
    # print(noise)
```

A.4 Script for executing

```
from generate import load_png_image, make_noise
from circle import thresholding, center_coordinates, get_value, get_plot_values
from fourier import do_fourier, printing
import numpy as np

path=’testing2.png’
img_gray=load_png_image(path)
img_gray_ns=make_noise(img_gray)
thresh = thresholding(img_gray_ns)
coordinates,radius = center_coordinates(thresh)
#radius = center_coordinates(thresh)
values, dist, val_dist = get_value(img_gray_ns,coordinates,radius*1.05)
val_2 = get_plot_values(val_dist)
fft_s, freq_s = do_fourier(val_2 - np.mean(val_2))
thing = printing(fft_s,freq_s)
```

B Appendix

B.1 Python code

```python
import numpy as np
from circle import thresholding, center_coordinates, get_value, get_plot_values
import cv2
import matplotlib.pyplot as plt
import skimage
from skimage.draw import circle_perimeter, disk


def load_ct_data(file_path):
    ct_image = np.load(file_path)[:, 120:240, 100:200]
    #ct_image = cv2.normalize(ct_image, None, 255, 0, cv2.NORM_MINMAX, cv2.CV_8U)
    print(ct_image.shape)
    low, medium, high = extract_regions(ct_image)
    return low, medium, high

#slices

def extract_regions(ct_image):
    high_1, high_2 = 575, 600
    medium_1, medium_2 = 315, 345
    low_1, low_2 = 240, 265

    high = ct_image[high_1:high_2]
    medium = ct_image[medium_1:medium_2]
    low = ct_image[low_1:low_2]

    return low, medium, high


def values_dist_3D(image):

    vd_ = np.zeros((image.shape[0], 1023, 2)) # you might need to change 689 and 2 depending on the data

    for i in range(image.shape[0]):
        cx, cy, radius = segment(image[i])

        if len(cx) == 1: #checks if only one structure is detected in current slice
            if cx > 40:
                if (radius >= 8):
                    _, _, val_dist = get_value(image[i], [cx[0], cy[0]], radius + 5)

                    print("Shape of values", val_dist.shape)

                    vd_[i, :len(val_dist), :] = val_dist

    print("vdshape",vd_.shape)

    return vd_

```

```python
#get the values of the pixels that are in the segmented section with the center coordinates and within radius. Detecting structures in image and getting values.

def segment(slice_2D):
    chan_vese = skimage.segmentation.chan_vese(slice_2D, mu=0.01, lambda1=0.9)
    # chan_vese = skimage.segmentation.clear_border(chan_vese)

    hough_radii = np.arange(1, 15, 1)
    hough_res = skimage.transform.hough_circle(chan_vese, hough_radii)
    accums, cx, cy, radius = skimage.transform.hough_circle_peaks(hough_res, hough_radii, total_num_peaks=1)

    return cx, cy, radius

#chan vese segmentation algorithm that aims to segment objects based on intensity differences
#shape is detected by contrast, transformation is done where centre coordinates are assumed/determined and radius to produce a circle based on the detected shape without over or underestimating

if __name__ == '__main__':
    l, m, h = load_ct_data('./data/m1_FDK_shepp_no_filter.npy')
    l = l[:, :80, :80]
    m = m[:, :80, :80]
    h = h[:, :80, :80]
    plt.imshow(h[10], cmap='gray')
    plt.show()
    vd_ = values_dist_3D(h)
    vd_ = vd_.reshape((vd_.shape[0]*vd_.shape[1], 2))
    mask = vd_[:, 0] > 0
    vd_ = vd_[mask]


    plt.plot(vd_[:, 1], vd_[:, 0], 'o')
    plt.xlabel("Distance from centre")
    plt.ylabel("Pixel value")
    plt.show()

    from fourier import do_fourier, printing
    val_2 = get_plot_values(vd_)
    fft_s, freq_s = do_fourier(val_2 - np.mean(val_2))
    thing = printing(fft_s, freq_s)

#CT dataset loaded, slice extracted and displayed in grayscale. Value distribution calculated and plotted. Fourier performed on plotted values.
```