Artifact Reduction in Computed Tomography

PhD Thesis

Csaba Olasz
Supervisors: László G. Varga, PhD and Antal Nagy, PhD

Doctoral School of Computer Science
Department of Image Processing and Computer Graphics
Faculty of Science and Informatics
University of Szeged

Szeged
2024

Contents

List of Figures

List of Tables

Abbreviations

AVG	Average
CNN	Convolutional Neural Networks
CT	Computed Tomography
FBP	Filtered Backprojection
GT	ground truth
HHom	Haralick Homogeneity
MAE	Mean Absolute Error
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
ROI	Region of Interest
SIRT	Simultaneous Iterative Reconstruction Technique
SNR	Signal-to-Noise Ratio
SSIM	Structural Similarity
STD	Standard Deviation

List of Symbols

α	the angle of the projection
δ	the distance between neighboring rays
p	the column vector of the projections indexed in detector-major order
W	the projection matrix
x	the column vector of the reconstructed image indexed in row-major order
f	function of the unknown object
f'	the reconstructed image with size of n by n
I_D	the number of X-ray photons sensed by the detector
I_S	number of photons leaving the source
m	number of projection lines in a detector-major order: $m = m_p \cdot m_d$
m_d	number of projection lines at a given angle
m_p	number of projections from different angles
n	the height and width of the reconstructed image
t	the distance of a projection line from the origin

Chapter 1

Preliminaries

Computed Tomography (CT) is an imaging technique in which an object can be reconstructed slice-by-slice without damaging the object [32]. To achieve this, one exposes the object of study to some kind of penetrating wave or radiation (e.g., X-ray, seismic waves, ultrasound, neutron beams, radio waves, etc.) and uses mathematical tools to derive the structure of the object from the changes of the waves passing through the object or reflected from it [24]. CT has a wide scope of applications considering fields such as medical imaging [23, 34], geology [27], non-destructive testing [7, 8, 33], etc.

In the case of transmission X-ray tomography, measurements are taken with penetrating X-ray radiation. As the radiation passes through the object, it attenuates proportionally to the linear attenuation coefficient of the material it crosses. There are multiple sensors placed behind the object, where the total attenuation of an X-ray beam is detected at a given view angle. One can reconstruct the attenuation coefficient at any given point of an object if a large enough number of measurements are available at different view angles.

Measurements used by a tomographic method are usually affected by distortions [25]. Many sources of distortions exist, among which the most common problems are physical phenomena related to data acquisition, such as beam hardening and Compton scattering [17]. Measured values can also be affected by electrical noise [13], and geometric inconsistencies in the CT scanner can cause additional artifacts [4, 10]. Furthermore, the reconstruction method itself and the discrete representation of the resulting images may also introduce errors [64].

This thesis summarizes the research of the author in the field of transmission X-ray tomography. The binding concept of the conducted research was to improve the quality of the reconstructed image with the reduction of previously mentioned artifacts by improving existing reconstruction algorithms and the fusion of CT and Convolutional Neural Networks (CNN) techniques.

In Chapter 1 I summarize the fundamental knowledge from the existing literature

crucial to understanding this theses and describe the metrics that we used for the objective evaluation of the results.

The number of projections and detector elements in a CT is limited. In addition, reconstruction is usually performed on a discrete grid. These restrictions of the model create artifacts in the reconstructed images. In Chapter 2, I show that the type of interpolation used during forward projection and backprojection strongly influences these artifacts (i.e. *interpolation errors*). The connection between pixel coverage and interpolation error is also obtained to get a better understanding of the interpolation errors, and I also propose an effective correction method to reduce the effects of interpolation errors in reconstructed images. I tested the proposed method in a comprehensive experiment, where I found that the proposed correction method can significantly improve the quality of the reconstructed images.

Combining computer tomography with deep learning methods is an effective way to achieve improved image quality and artifact reduction in reconstructed images. In Chapter 3, I present three novel neural network architectures for tomographic reconstruction with reduced effects of beam hardening and electrical noise. In the case of two novel networks, multiple skip connections maintained a strong bound between the parts of the neural network that allowed the projection data and the reconstructed image to be improved together. The proposed methods were tested on datasets generated by me. The datasets contain physically correct simulated data and show strong signs of beam hardening and electrical noise. We also performed a numerical evaluation on the reconstructed images corrected by neural networks according to three error measurements. The experimental results showed that the reconstruction step used at the end of the neural network or in skip connections in neural networks improves the quality of the reconstructions.

The ideas, figures, tables, and results included in this thesis were published in scientific papers (listed at the end of the thesis). In a nutshell, the author's contributions are the followings:

Chapter 2.: I implemented a framework to test popular interpolation methods in various settings. I organized multiple test series and evaluated the results. I proved the connection between pixel coverage and the interpolation error by experiments. I showed that the proposed correction of the projection matrix can reduce the interpolation error of the reconstructed images.

Chapter 3.: I created a physically correct database using GATE simulation software. I modified the original U-net and applied it at different stages of the CT image processing pipeline. I proposed three novel CNN architectures, where the image reconstruction step is used as a non-trainable neural network layer. I tested the algorithms on the generated dataset and showed that the new architectures outperformed the others, where there was none or only one reconstruction layer inside the network.

1.1 Fundamentals

1.1.1 The reconstruction problem

The general task of transmission tomography is to determine the internal structure of an unknown object with the constraint that we know only the projections of the object in given directions [32]. The attenuation coefficients of the material in a two-dimensional cross-section is represented by the $f : \mathbb{R}^2 \to \mathbb{R}$ function. The projections of f can be obtained by the Radon transform as

$$[\mathcal{R}f](\alpha, t) = \int_{-\infty}^{\infty} f(t\cos(\alpha) - q\sin(\alpha), t\sin(\alpha) + q\cos(\alpha))dq \ , \tag{1.1}$$

where the (α, t) pair determines a line in the two-dimensional space, by giving its direction and distance from the origin, respectively. Figure 1.1 illustrates an object and one of its projections using parallel beam geometry, where a projection consists of projection lines that have the same direction α. The two-dimensional image called the sinogram is a collection of projections acquired from different angles. In Figure 1.2 one can see the central slice of the Shepp-Logen head phantom [55] and its sinogram, where α is ascending on the vertical axis and the projections are represented with grayscale values along the t axis. With Equation 1.1 the mathematical description of the reconstruction problem can be defined as finding a f', such that

$$[\mathcal{R}f'](\alpha, t) = [\mathcal{R}f](\alpha, t) \ , \tag{1.2}$$

for all measured (α, t) projection lines.

In CT scanners, the object of study is located between the X-ray source (S) and the detector (D), resulting in a limited span of f. Therefore, the range to be integrated can be simplified to the $[S, D]$ interval in Equation 1.1.

The values I_D and I_S are measured by CT scanners, where I_D is the number of X-ray photons sensed by the detector and I_S corresponds to the number of photons leaving the source. The relation between I_D, I_S and Equation 1.1 is given by the Beer-Lambert law as $I_D(\alpha, t) = I_S \, e^{-[\mathcal{R}f](\alpha,t)}$, leading to

$$[\mathcal{R}f](\alpha, t) = -\ln\left(\frac{I_D(\alpha, t)}{I_S}\right) \ . \tag{1.3}$$

For practical applicability, we have to make two restrictions on the reconstruction. First, f' has a bounded support, that is, $f'(u, v) = 0$, with $(u, v) \notin [-n/2, n/2)^2$ for a large enough constant integer n. Second, f' takes a constant value on each unit square-shaped area determined by the two-dimensional integer lattice, that is, $f'(u + a, v + b) = f'(u + c, v + d)$, for all $u, v \in \mathbb{Z}$ and for all $a, b, c, d \in [0, 1)$. With

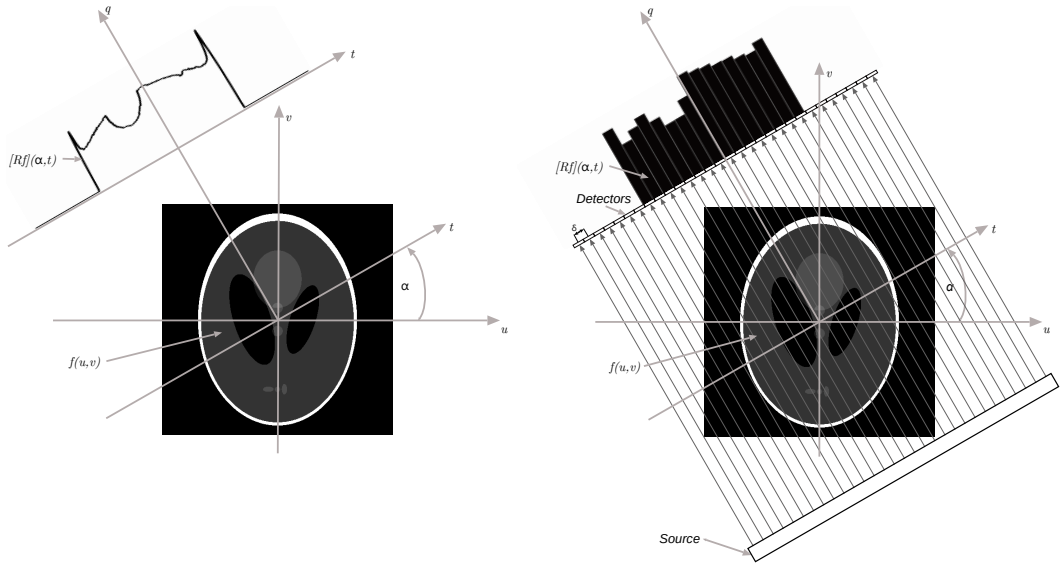

Figure 1.1: *Left, illustration of the Radon transform. Right, acquisition of a projection that highlights the discretized nature of the task.*

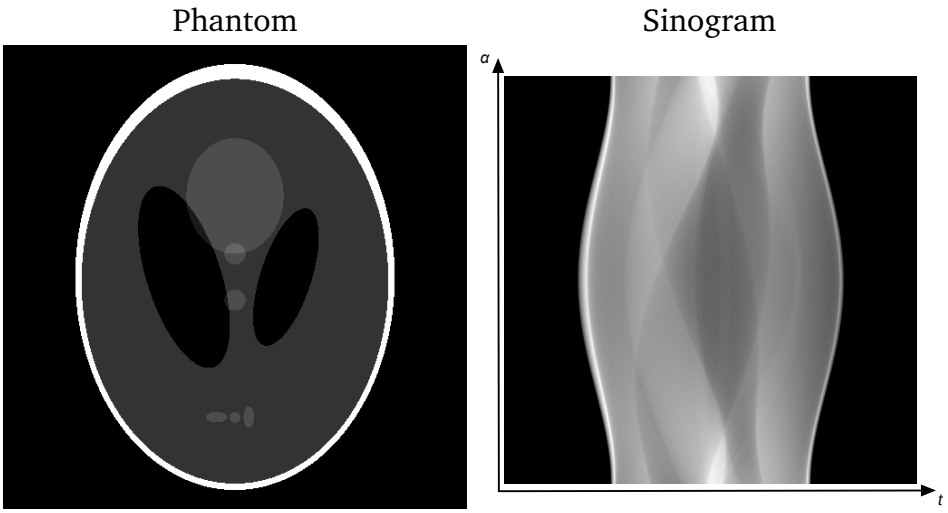

Figure 1.2: *Shepp-Logen phantom and its sinogram*

these restrictions, the task is to reconstruct an $n \times n$ sized digital image consisting of uniform distinct pixels with sides of unit length. We also note that the measurements are available on a discrete grid due to the fact that the detector consists of multiple detector elements placed next to each other (see Figure 1.1).

With the above restrictions, we can define the reconstruction problem as a system of equations as follows: let us say that we have a $n \times n$ image $\boldsymbol{x}$ on a square grid indexed in a row-major order, as shown in Figure 1.3. Furthermore, let the sinogram be a vector $\boldsymbol{p}$, with projection angles m_p, each containing m_d detector values indexed

in detector-major order. In this way, let p_i ($i \in m$, $m = m_p \cdot m_d$) be the i-th ray as shown in Figure 1.3. The following linear equation can be stated:

$$\sum_{j=1}^{n^2} W_{ij}\, x_j = p_i, \quad i = 1, 2, ..., m \ , \tag{1.4}$$

where ***W*** is the projection matrix (or system matrix) with the dimension of $m \times n^2$, which contains the connection between all rays and pixels. Basically, W_{ij} is a weight that defines the contribution of the x_j pixel in the projection p_i. There are different ways to calculate the weights of ***W*** discussed in detail in Section 2.2.

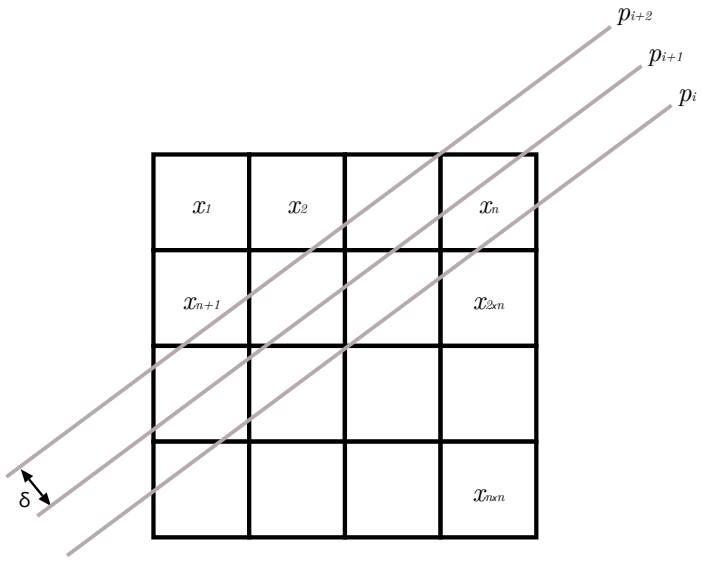

Figure 1.3: *The reconstruction problem as a system of linear equations. x_n is the pixels of an image on a square grid. p_i is a ray at a given angle and position. δ represents the distance between neighboring rays.*

The algorithms that can reconstruct the images from its projections can be divided into two groups, namely the analytical and the algebraic approaches. The next two subsections will give an overview of these groups in general and specify the solutions that we used in the studies of this dissertation.

1.1.2 Analytic approach

The *backprojection* algorithm is one step of the Filtered Backprojection (*FBP*) which is the most popular reconstruction algorithm [32]. The backprojection means that one takes the sinogram values one by one and adds them to the pixels of the reconstructed image that the projection line crosses corresponding to the given detector position (t) and direction (α). The problem with that is the blurring effect appearing on the

reconstructed image (see Figure 1.4). The explanation and solution of the blurring effect originate from the two-dimensional Fourier-Slice Theorem [24]:

Theorem 1: *The one-dimensional Fourier transform of a $f(u, v)$ function at angle α is the straight line of the two-dimensional Fourier transform of the $f(u, v)$ function that has an angle α and passes through the origin.*

Theorem 1 shows that the one-dimensional Fourier transform of the projections are equal to a line of the two-dimensional Fourier transformed reconstructed image. However, the projections cover the space unevenly, so that the low frequencies are over-, while the high frequencies are under-represented in the Fourier domain. Therefore, the reconstruction results in blurred images without sharp edges with simple backprojection. The blurring effect can be eliminated by filtering the projections in the Fourier domain. The filter should be a high-pass filter that compensates for the sparser sampling at higher frequencies, such as the well-known Ram-Lak filter [24].

The *FBP* combines *backprojection* with the filtering step [25]. The algorithm starts with the one-dimensional Fourier transform of the projections. Then, one should multiply the result with an appropriate filter element-wise in frequency domain. After that, the filtered projections are transformed back to the image domain with the one-dimensional inverse Fourier transform. Finally, the filtered projections are backprojected. *FBP* is fast and can give images of good quality but requires a large number of projections. It is also the most popular reconstruction algorithm in CT although it is sensitive to noise and other distortions in the projections that occur during data acquisition due to physical phenomena. Note that during the backprojection phase, we use so-called interpolation methods, which differentiate the interactions between the projection lines and the image pixels. Above, we assumed that the values of the sinogram will be added equally to all pixels that the given projection line crosses. In Section 2.2 we introduce interpolation methods that understand that the contributions of the crossed pixels are not necessarily equal to the total attenuation of an X-ray.

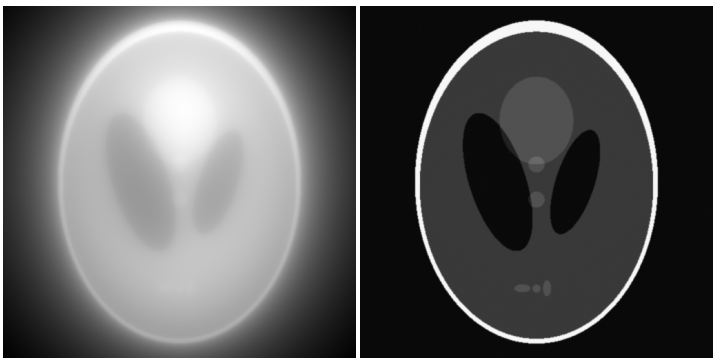

Figure 1.4: *Result of backprojection (left) and FBP (right).*

1.1.3 Algebraic approach

There are many algebraic reconstruction algorithms that utilize the algebraic representation of the reconstruction problem [3, 24, 32, 60]. The algorithm I used to solve the equation 1.4 was the Simultaneous Iterative Reconstruction Technique (*SIRT*) [60], which is a popular algebraic method for CT image reconstruction. It usually takes more computational time to get a result with *SIRT* as it works in an iterative way, so it converges to a solution step-by-step. However, it proved to be superior compared to *FBP* [69] because of its potential to improve image quality with repeated steps as long as necessary. I let the algorithm run for a fixed number of 100 iterations. I used the following implementation with the notations of 1.4:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{C}\mathbf{W}^T\mathbf{R}(\mathbf{p} - \mathbf{W}\mathbf{x}^k) \quad , \tag{1.5}$$

where k is iteration index, and $\mathbf{x}^k$ is a vector of the intermediate reconstructed image – or in our case $\mathbf{x}^{100}$ is the result –, $\mathbf{p}$ is a vector of the projections, and $\mathbf{W}^T$ is the transposed projection matrix. Furthermore, $\mathbf{C}$ and $\mathbf{R}$ are diagonal matrices giving additional weighting to ensure the convergence of the algorithm. The $\mathbf{C}$ and $\mathbf{R}$ matrices are constructed as

$$\mathbf{R} \in \mathbb{R}^{m \times m} \quad , \quad R_{ii} = \frac{1}{\sum_{j=1}^{n^2} W_{ij}} \quad , \tag{1.6}$$

$$\mathbf{C} \in \mathbb{R}^{n^2 \times n^2} \quad , \quad C_{jj} = \frac{1}{\sum_{i=1}^{m} W_{ij}} \quad , \tag{1.7}$$

where n^2 is the number of pixels and m is the number of rays as earlier. The goal of Equation 1.5 is to produce less and less difference between $\mathbf{p}$ and the projections of $\mathbf{x}^k$, where $\mathbf{x}^k$ is the actual solution and $\mathbf{p}$ is the projection data of the image to be reconstructed.

Extending the above formalism, iterative algorithms are sometimes applied with non-linear projection- and back-projection steps that cannot be formulated with a projection matrix. In this case, the multiplication with $\mathbf{W}$ and $\mathbf{W}^T$ in (1.5) can be replaced by applying the forward projection and backprojection operations, respectively, and the C and R matrices should be calculated in a way to maintain convergence. Moreover, in some cases of tomography unmatched forward projection – backprojection pairs are also in use. The main advantage of this approach is the reduced computational cost of the reconstruction process, while it is still possible to maintain convergence of the algorithms [58, 70].

1.1.4 Metrics

To evaluate the quality of the reconstructed image, we needed objective measurements. The metrics used in this thesis are well known in the field of image processing

and in the CT image modality [9, 26, 36, 43, 67]. Note that not every metric was used in every experiment.

The Mean Absolute Error (*MAE*) and Mean Squared Error (*MSE*) are pixel-wise metrics for comparing the reconstructed result to a ground truth image [63]. The formula for *MAE* is given as

$$MAE(f', \hat{f}) = \frac{\sum_{u=1}^{n} \sum_{v=1}^{n} |f'_{uv} - \hat{f_{uv}}|}{n^2} \quad , \tag{1.8}$$

while for *MSE*

$$\mathtt{MSE}(f', \hat{f}) = \frac{\sum_{u=1}^{n} \sum_{v=1}^{n} \left(f'_{uv} - \hat{f_{uv}}\right)^2}{n^2} \quad , \tag{1.9}$$

where f' is a reconstructed $\hat{f}$ is the ground truth (GT) image. We are referring to the GT image in the future as the perfect image of the object to be reconstructed. If the reconstructed image matches perfectly to its GT, both *MSE* and *MAE* give a value of zero. The *MAE* provide us with a general evaluation of how the two images defer on average. In contrast, *MSE* is more sensitive to outliers above the average differences because the pixel differences are squared. This difference is more relevant when we use them as loss functions, as the *MSE* could provide a larger gradient in the training phase, but it may take the final results to be optimized heavily for outliers rather than the generally good result.

The Peak Signal-to-Noise Ratio (*PSNR*) is measuring the basic goodness of the results in decibel units (dB) compared to the GT [63]. The projection data are affected by noise in practice, and the *PSNR* represents how strongly the GT appears in the noise on a logarithmic scale.

$$PSNR(f', \hat{f}) = 10 \log_{10} \left(\frac{\mathrm{MAX}_I}{MSE(f', \hat{f})} \right) \quad , \tag{1.10}$$

where MAX_I is the maximal possible value of the reconstructed image.

The Structural Similarity (*SSIM*) index also requires a GT image, but the original aim of this metric was to model the visual perception of the human eye [62]. The formula of *SSIM* has three terms:

$$SSIM(f', \hat{f}) = [l(f', \hat{f})]^{\alpha} \cdot [c(f', \hat{f})]^{\beta} \cdot [s(f', \hat{f})]^{\gamma} \quad , \tag{1.11}$$

namely the $\mathtt{l}(f', \hat{f})$ luminance, the $\mathtt{c}(f', \hat{f})$ contrast and the $\mathtt{s}(f', \hat{f})$ structural terms. The pictures to be compared are denoted with f' and $\hat{f}$ as previously notated. The contribution of the three terms can be regulated by the exponents α, β, and γ. As we wanted to focus here on the terms related to contrast and structure, we have set $\alpha = 0$ while $\beta, \gamma = 1$. Originally, the range of the measure is $SSIM(f', \hat{f}) \in (-1, 1]$. If

SSIM$(f', \hat{f}) = 1$, then the two images compared are identical. If *SSIM*$(f', \hat{f})$ is negative, then there is similarity, but in a complementer manner. If *SSIM*$(f', \hat{f}) = 0$, then no similarity was found between the two images. Since reconstruction techniques are not intended to produce complementer reconstructions, negative values should be regarded as errors, so negative values were replaced by zero. The final value of *SSIM* is the average of the locally calculated similarity values. The *SSIM* is also capable of generating similarity maps, where one can observe the similarities of the images locally for each pixel. We will also use such tools to highlight interpolation errors. We will present the *SSIM* maps in inverted form for better visibility.

The Haralick Homogeneity (*HHom*) is one of the Haralick texture features [22]. The Haralick texture features are calculated from the co-occurence matrix of the image to be evaluated without GT. This matrix contains the frequency of two pixel intensities appearing together next to each other at a given offset. We always used the offset [0, 1], therefore, the two pixels to be evaluated are the pixel of interest and its right neighbor. Regarding the other parameters of the co-occurrance matrix, calculation we used 255 gray levels. The value of *HHom* is in the $[0, 1]$ interval. For a constant image *HHom*$(f') = 1$.

Chapter 2

Discretization artifacts

Real world objects have structures of infinite detail which one cannot capture technically and cannot handle computationally. Therefore, the reconstructed image and the projections are usually modeled on a discrete grid in transmission X-ray tomography, as we introduced in Chapter 1. Due to the discrete representations, one can only give an approximate formulation of the projection geometry. In Chapter 1.1.2 we assumed that the values of the sinogram uniformly backprojected to the pixels that the corresponding projection line crosses. But differentiation can be made based on the nature of the crossing (for example, the line passes through the center of the pixel or only touches the corner of it) using the weights of the projection matrix. The way we determine the weights are the so-called 'interpolation methods'. The different kinds of interpolations result in different structured artifacts in the reconstructed image. Figure 2.1 shows an example of an interpolation error in the reconstructed image. What we call interpolation error is the patterns originating from the center of the image and widening as the corners are getting closer.

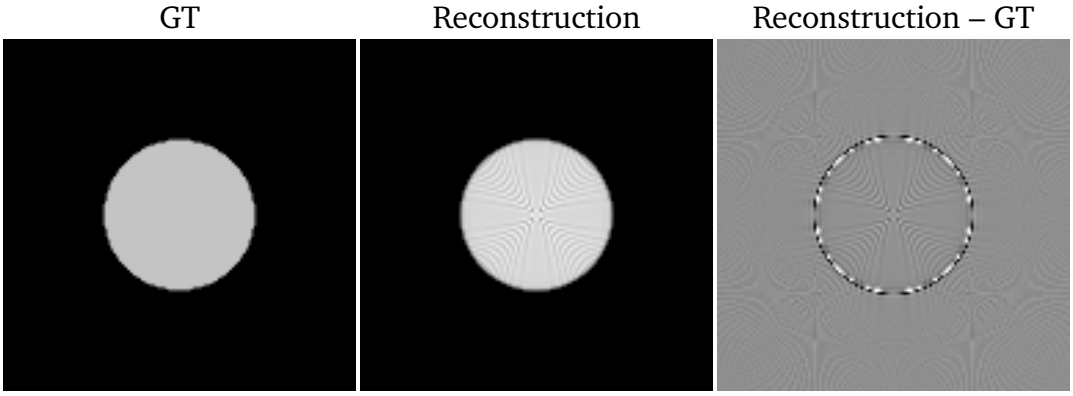

Figure 2.1: *Left, the ground truth (GT) image. In the middle, the reconstructed image. Right, GT minus the reconstructed image. The display window is [-0.15, 0.15] for the Raw – Corrected case and [0.8, 1.06] for the others.*

This chapter is built around interpolation methods. First, I introduce the related research in Section 2.1. Then, I describe the interpolation methods used in Section 2.2. I organized our results into two studies. In the *Comparison study* – presented by Section2.3 – eight interpolation methods were compared in our test dataset in multiple different settings. After that, a correction technique is proposed in the *Correction Study* to reduce the errors caused by the interpolation methods. Details of the correction technique and the results of this second study are provided in Section 2.4. Our final thoughts related to Chapter 2 are summarized in Section 2.5.

2.1 Related works

There are many interpolation methods in the literature and researchers have done extensive work to examine and develop them. For example, the authors of [18] compared three linear interpolation approaches. In [12] the projection matrix was analyzed on real data created by the Siddon [57] and Joseph [31] methods. The researchers of binary tomography have developed algorithms that work with hexagonal [42] or triangular [39] pixels. In non-binary tomography, alternatives to the square grid representation of the image space were introduced in [37]. The contribution of the authors of [66] was the hexagonal-2x-oversampled scheme of the square grid, which is similar to the hexagonal pixels in [42]. In [21] the connection between pixels and projection lines was defined with the help of local basis functions. The authors of [41] invented their own distance-driven method. Researchers not only investigated the different interpolation methods (i.e. possible connections between pixels and projection lines), but also experimented with different projection line or pixel representations with non-square grids. In the end, every method has advantages and disadvantages, depending on the task and subjective preferences.

In the field of Cross-Hole Tomography [56], the geometric configuration of the measurements is very different from that in transmission X-ray tomography because the reconstructed area is scarcely covered by projection lines. Due to its different nature, the spatial coverage of pixels by projection lines is widely discussed [27] in this field.

During our investigation of the interpolation methods, we found a connection between the pixel coverage of the projections (detailed in 2.4.1) and the structured artifacts of the given interpolation method. To capitalize on that, we propose an artifact removal approach based on balancing the pixel coverage – an indicator of the imbalance of the coverage of pixels by projection beams – within projections. With correction, the pixel coverage becomes uniform, which reduces structural artifacts in the reconstructed results. Note that the authors of [27] use an algorithm to divide the reconstructed image – consisting of equal pixel size – into distinct pixel sizes with uniform pixel coverage. This is similar to our correction presented here, but we find

it easier to leave the pixel with the same size and rather correct the weights of the pixels using the calculated pixel coverage.

2.2 Interpolation methods

Various interpolation techniques exist in the literature, with different characteristics and performance. We used the ASTRA Toolbox [59] and the Matlab software [2] to test the merits of different interpolation methods. For Gaussian Distance Interpolation (*GDI*) we used our own implementation because this option is not available in ASTRA or Matlab.

In the case of ASTRA Line interpolation (*Line*), projection beams are modeled by straight lines [5]. The weight W_{ij} is the length of the section of the p_i ray that travels through the x_j pixel.

We tested also the ASTRA Linear interpolation (*Linear*). Here, to calculate W_{ij}, we first need to take into account the slopes of the beams [18]. Let u be the horizontal axis and v be the vertical axis on an Euclidean coordinate system holding our image **x**. Let us also assume that the projection beam is a line defined by the equation $v = \mu \cdot u + b$ with some slope μ and constant b. In this way, we have two cases. When $|\mu| > 1$ (i.e., the projection beam is closer to vertical), we go through the rows of the image, and in each row the coefficients of W_{ij} are calculated to perform a linear interpolation of the two pixels closest to the intersection of the beam and the row. In $|\mu| <= 1$ we do the same, but by stepping through the columns of the image. Therefore, the W_{ij} weight of the f_j-th pixel will be d_2 looking at Figure 2.2. The advantage of this interpolation technique is that it is relatively easy to implement and is computationally efficient compared to some other techniques.

The interpolation technique we call *GDI* uses a parameterised Gauss function [21] to provide the weight W_{ij}. As illustrated in Figure 2.2 the weight of W_{ij} of x_j is defined by the Gaussian of the distance between the projection line p_i and the center of the pixel x_j. In our implementation, the Gaussian σ parameter of *GDI* was set to 0.7 and we only calculated the coefficient for pixels within 2 units of distance from the projection lines. For the other pixels, the weights were set to zero.

The ASTRA Strip interpolation (*Strip*) behaves as if the p_i ray has width [5]. The weight W_{ij} of pixel x_j is the area in which the p_i and the x_j-th pixels overlap. This technique has a more complex and computationally demanding implementation.

Figure 2.2 illustrates the calculation of the weights of the projection matrix with the four interpolation methods already discussed.

We used four interpolation methods included in the Matlab software, namely the *Nearest*, *Spline*, *Cubic*, *MLinear*. The Matlab implementation of these methods perform the interpolations on the projection values directly rather than calculating the projection matrix beforehand [2, 32]. Consequently, the projection matrix was

replaced with the corresponding forward projection and backprojection operators with these methods as explained in Section 1.1.3. It is also important to mention that the forward projection in Matlab divides the pixels of the image into four sub-pixels to increase the resolution [6, 38]. We will refer to this technique as *supersampling*.

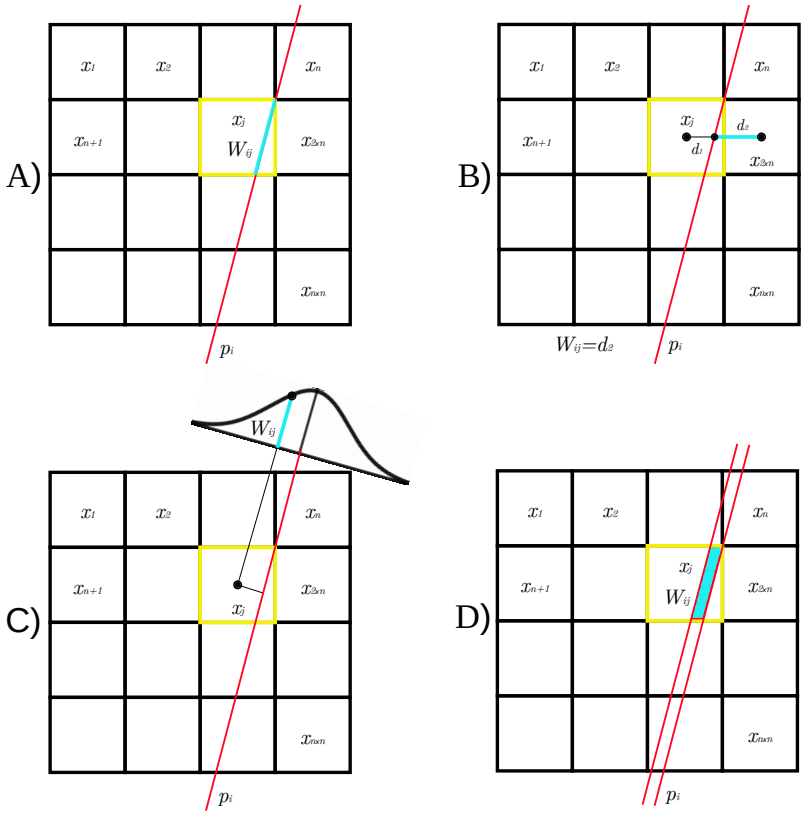

Figure 2.2: *Interpolation models examined in our experiments. A) Line, B) Linear, C) GDI and D) Strip.*

2.3 Comparison study

In the first series of experiments, we tested the interpolation methods listed in Section 2.2 to examine and enumerate their interpolation errors and compare their performance. The examined reconstruction algorithms were Filtered Backprojection (*FBP*) with Ram-Lak [32] filter and Simultaneous Iterative Reconstruction Technique (*SIRT*) with one hundred iterations [60].

2.3.1 Test dataset of the comparison study

Our test dataset can be divided into two parts: geometrical phantoms and head phantoms. We generated the sinograms of all test images, calculating the interactions of the lines of the geometrical shapes with the projection lines, with the help of [68]. These sinograms will be inputs for reconstruction algorithms using the corresponding interpolation methods. We generated sinograms with 143, 285, and 428 projections in the case of the geometrical image database and 4550 projections with the head phantoms. We sampled the projections positioned equiangularly on the half-circle. Using only 143 projections might show aliasing, but the evaluation with 285, 428 and 4550 projections must have enough projections for an accurate reconstruction, because according to the sampling criteria [32] the sufficient number of projections is

$$m_p = \frac{n}{\sqrt{2}}\pi \quad . \tag{2.1}$$

With our phantom images, m_d equals 180 in the case of geometrical phantoms, while 2896 with head phantoms.

The geometrical phantoms have a size of 128 $\times$ 128 pixels and have one object on them. The object can be a circle, ellipse, ring, rectangle, square, or triangle, which are basic geometrical structures. In addition to the structure, the objects are variable in size, location, and (if not rotation invariant) rotation angle for a more complete insight. Figure 2.3 shows some examples of our geometrical phantoms. In the case of geometrical phantoms we specified one Region of Interest (ROI) with the help of a mask, which excludes all object pixels and their few pixel vicinities. Therefore, the black pixels in Figure 2.4 a) are ignored during the calculation of the error metrics.

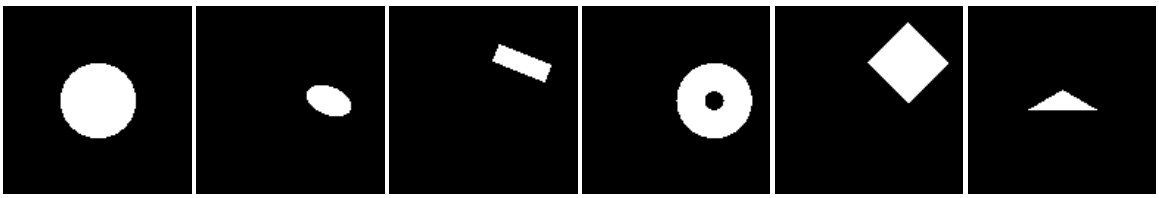

Figure 2.3: *Examples of our geometrical phantoms.*

Apart from the geometrical phantoms, we used the Sheep-Logan [55] and the Forbild [68] head phantoms, which are classical test images in computed tomography. We examined the head phantoms with the size of 2048 $\times$ 2048 pixels for better visibility. There are several objects in the head phantoms. For evaluation we determined 15 ROIs with the size of 36 $\times$ 36 pixels for both head phantoms shown in Figure 2.4. The ROIs are covering homogeneous regions according to GT images. 11 ROIs from the 15 are located inside the head phantoms.

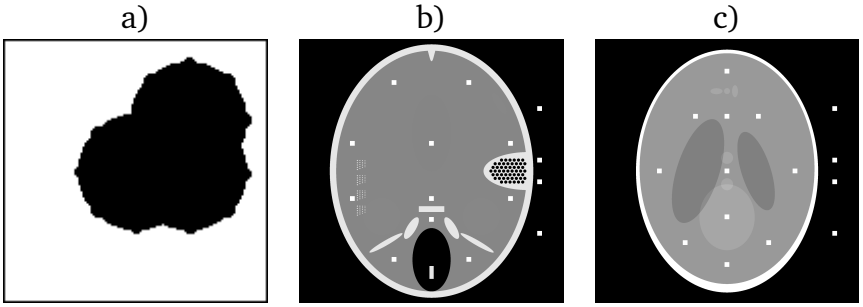

Figure 2.4: *The mask used for geometrical images shown in a), while b) and c) are illustrations of Forbild and Shepp-Logan head phantoms and the used ROIs (white squares).*

2.3.2 Random noise on projections

In practice, tomographic imaging is always exposed to some type and level of noise. Commonly accepted solution in simulation studies [13] is to model shot noise with Poisson distribution, while electrical noise of the detector is usually modeled by Gaussian distribution. We examined both types of distributions in our experiments, one at a time. Altogether, we have three noise levels with each distribution and the noiseless case. We set the noise levels based on the Signal-to-Noise Ratio (SNR) of the simulated projections. The noise levels related to the geometrical phantoms are 36 dB, 42 dB, and 48 dB, while in the case of the high-resolution head phantoms 54 dB, 60 dB, and 66 dB.

2.3.3 Results of the geometrical phantoms

First, we evaluated the results of geometrical phantoms in the comparison study. During these evaluations, we will use the average error value calculated for all geometrical phantoms. The results of geometrical phantoms are summarized in Figure 2.5 and Figure 2.6. Moreover we provided the calculated *MAE* values in Figure 4.1 located in the Appendix (i.e. Chapter 4).

For 285 projections, looking at the noiseless part of the diagrams in Figure 4.1 all the average *MAE* values are below 0.03, showing that the quality of the reconstructions is acceptable. According to the *SSIM* value (see Figure 2.5) the *Nearest* method performs the worst in most cases. Besides that, the *Nearest*, *Linear* and *Line* performed better with *SIRT* then *FBP*. This suggests that the reconstruction algorithms are exposed to interpolation error to different extent. The results of the other methods are close to each other, but a notable difference was observed for the benefit of the *GDI* with both algorithms. Based on the *HHom* measurement (see Figure 2.6) the *GDI* method yields the best results with a notable difference.

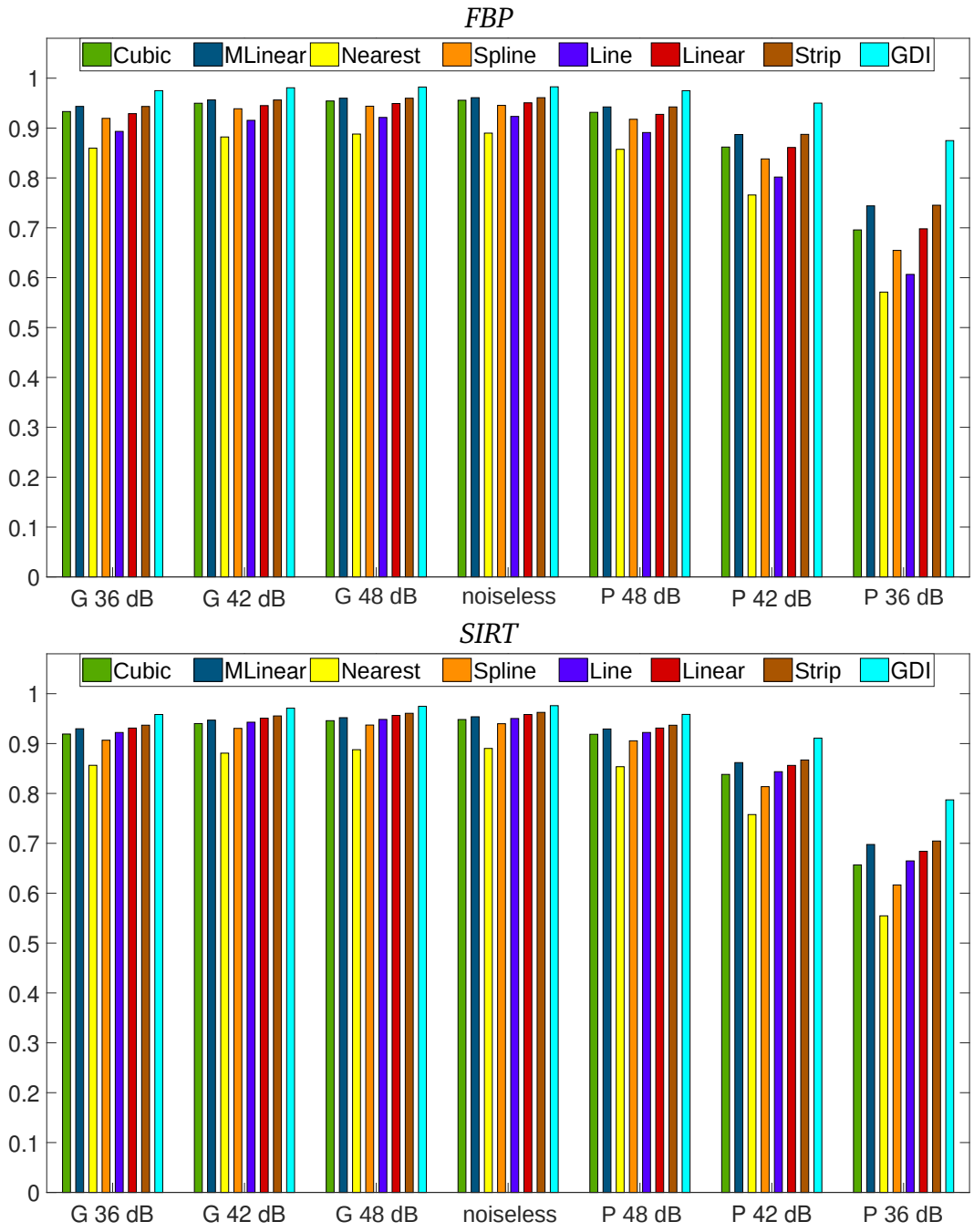

Figure 2.5: *The diagrams show the average SSIM for the geometrical phantoms with 285 projections. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown to the left, Poisson noise to the right side.*

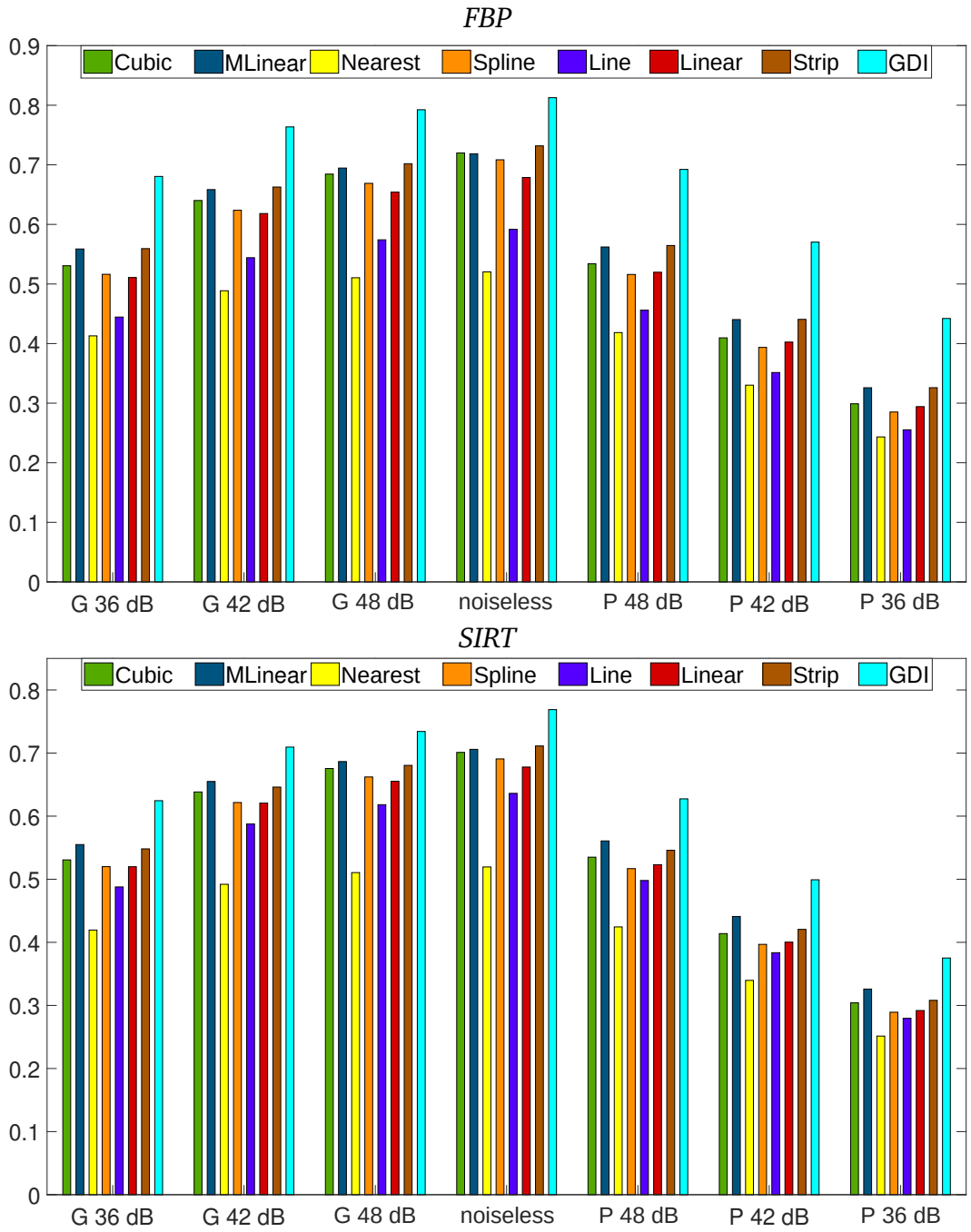

Figure 2.6: *The diagrams show the average HHom for the geometrical phantoms with 285 projections. The center of the diagrams shows the noiseless case. The effect of Gaussain noise is shown on the left and Poisson noise on the right.*

If we look at the effects of noise in the diagrams in Figures 2.5 and Figures 2.6, we can say that most of the reconstruction algorithm - interpolation method combinations are performing worse with increasing noise. This is, however, not to the same extent. Moreover, it seems that the Poisson type of noise has a greater effect on the reconstruction quality than the Gaussian type. The *MLinear*, *Strip* and *GDI* methods preform the best among all. Initially, the *GDI* method yields the highest *SSIM* and *HHom* values, which decreased to a lesser extent than the others; therefore, in the end it has gained a lot of advantage. An example of the effect of noise on reconstructions can be seen in Figure 2.7. The effect of the interpolation error is clearly visible with medium-level noise, and it is still noticeable with higher levels of noise.

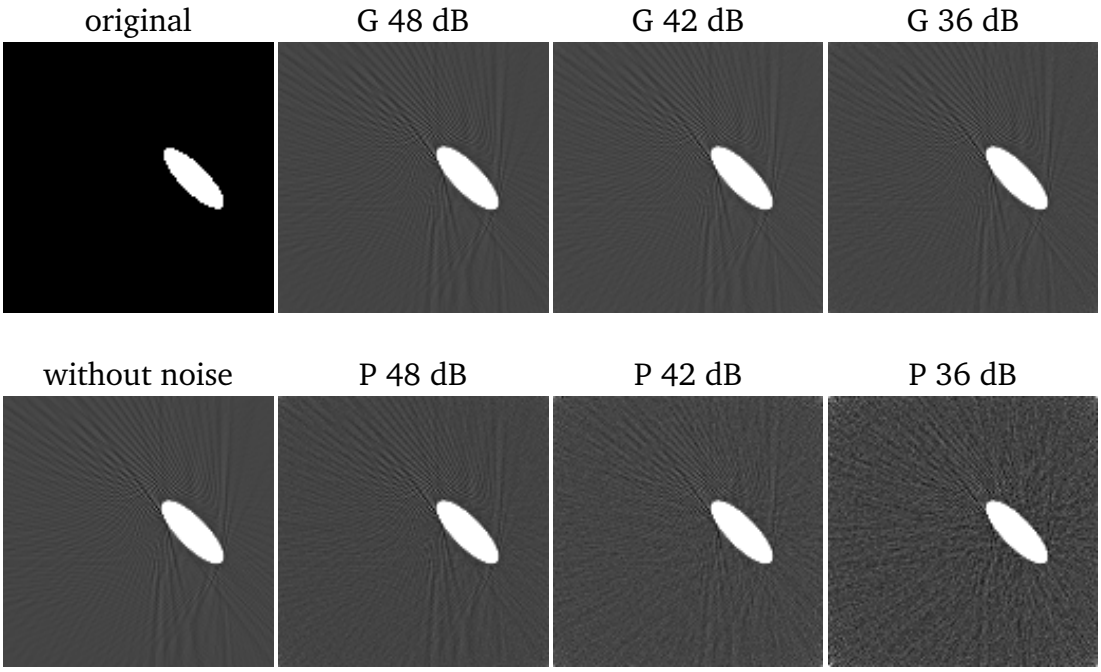

Figure 2.7: *The effect of noise in case of 285 projections using the SIRT reconstruction algorithm with Line interpolation. In the first column we present the original image, and its reconstruction without noise. From the second column, one can see the images with Gaussian noise in the first row, and with Poisson noise in the second row. The amount of noise grows from left to right.*

With 428 projections, the results are essentially the same as with 285 angles. The quality of the reconstructions is constantly good according to the *MAE*, while the relations between the performance of the interpolation techniques did not change. With 143 projections, some deterioration can be observed in every case. One can see an example in Figure 2.8 to this statement, which was expected since 143 projections are below the sampling criteria. Apart from that, no significant changes occurred.

One can notice in Figure 2.9, that the error is stronger along the line of the object

edges, which implies that the object structure influences the quality of the reconstruction and the interpolation error itself.

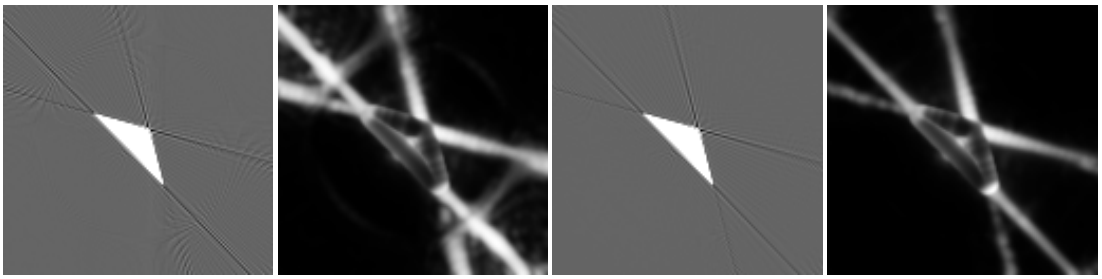

Figure 2.8: *This figure illustrates the deterioration of the quality with Line method due to the reduced projection number using FBP. The first two images are the reconstructed images with 143 projections and its SSIM map, respectively. The second two images are the reconstruction with 285 projections and its SSIM map, respectively.*

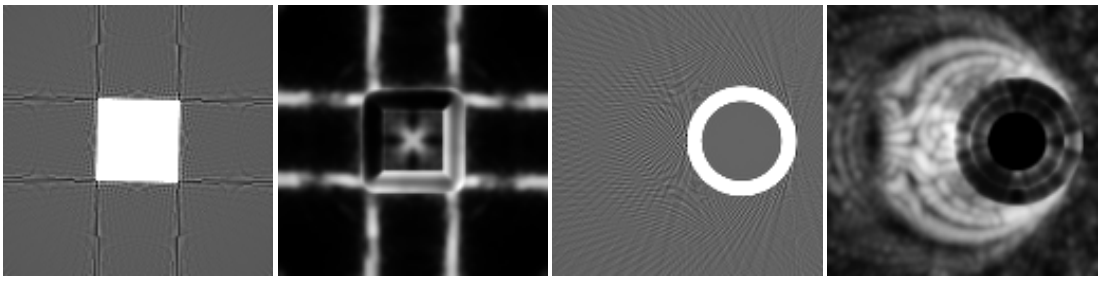

Figure 2.9: *The structure of the object influencing interpolation error. The first and third images show the result of the Line method without noise with 285 projections and FBP algorithm. The second and fourth images show the corresponding SSIM maps.*

2.3.4 Results of the head phantoms

In case of examining the head phantoms in the comparison study, the error values were calculated in distinct ROIs located in homogeneous regions. The *SSIM* and *HHom* values presented in Figure 2.10 and Figure 2.11 are the average of all ROIs with *FBP*. Regarding the Forbild head phantom, the noiseless results look similar to the geometrical phantoms according to the *SSIM* and *HHom*. With increasing noise, the *SSIM* values are decreasing to a different extent depending on the interpolation methods. At the highest noise level, the *GDI* has the best performance followed by *Strip* and *MLinear* with *FBP*.

In the case of the noiseless results of the Sheep-Logan head phantom, the differences between the interpolation methods are barley noticeable with *SSIM*. However,

in the noisy case the observed relations were similar to those of the Forbild head phantom. With *HHom* the situation is just the opposite. One can see big differences without noise, which is decreasing with increasing noise. We argue that this phenomenon is natural, because the unstructured noise spoils homogeneity of the image, resulting in low *HHom* value with all methods. Noise, on the other hand, can amplify the structured interpolation noise in proportion to its initial size without noise.

In the case of *SIRT*, the corresponding figures can be found in the Appendix (i.e., Chapter 4). With *SIRT*, the reconstructed images are more similar to GT and more homogeneous than with *FBP*. According to the *SSIM* and *HHom* the differences among the methods are smaller than with *FBP*. Again, this suggests that the iterative algorithm makes the reconstruction less sensitive to the interpolation method. With increasing noise, the *HHom* values are decreasing, while the differences are decreasing between the methods.

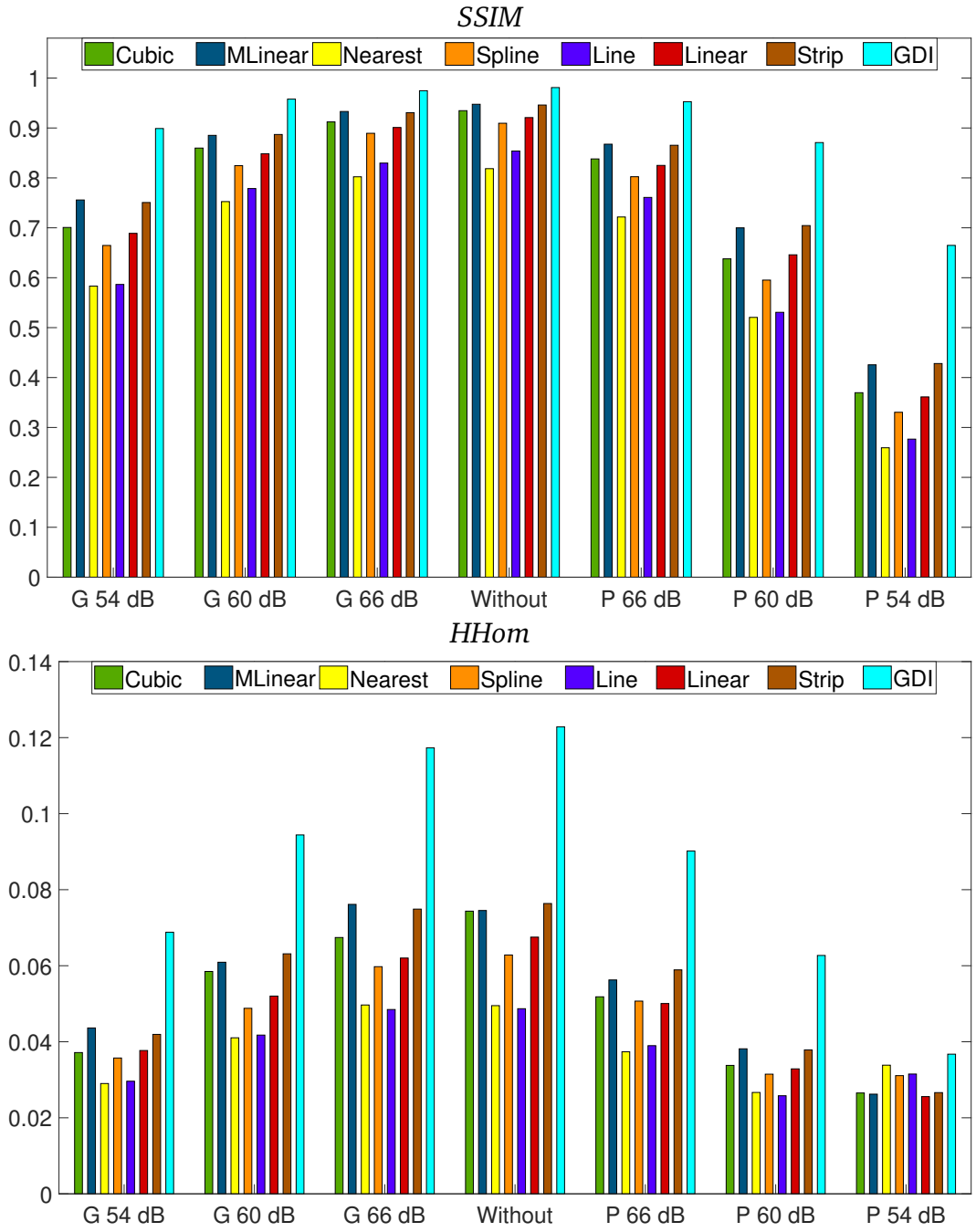

Figure 2.10: *The average values for all ROIs of the Forbild head phantom with FBP. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown to the left, Poisson noise to the right side.*

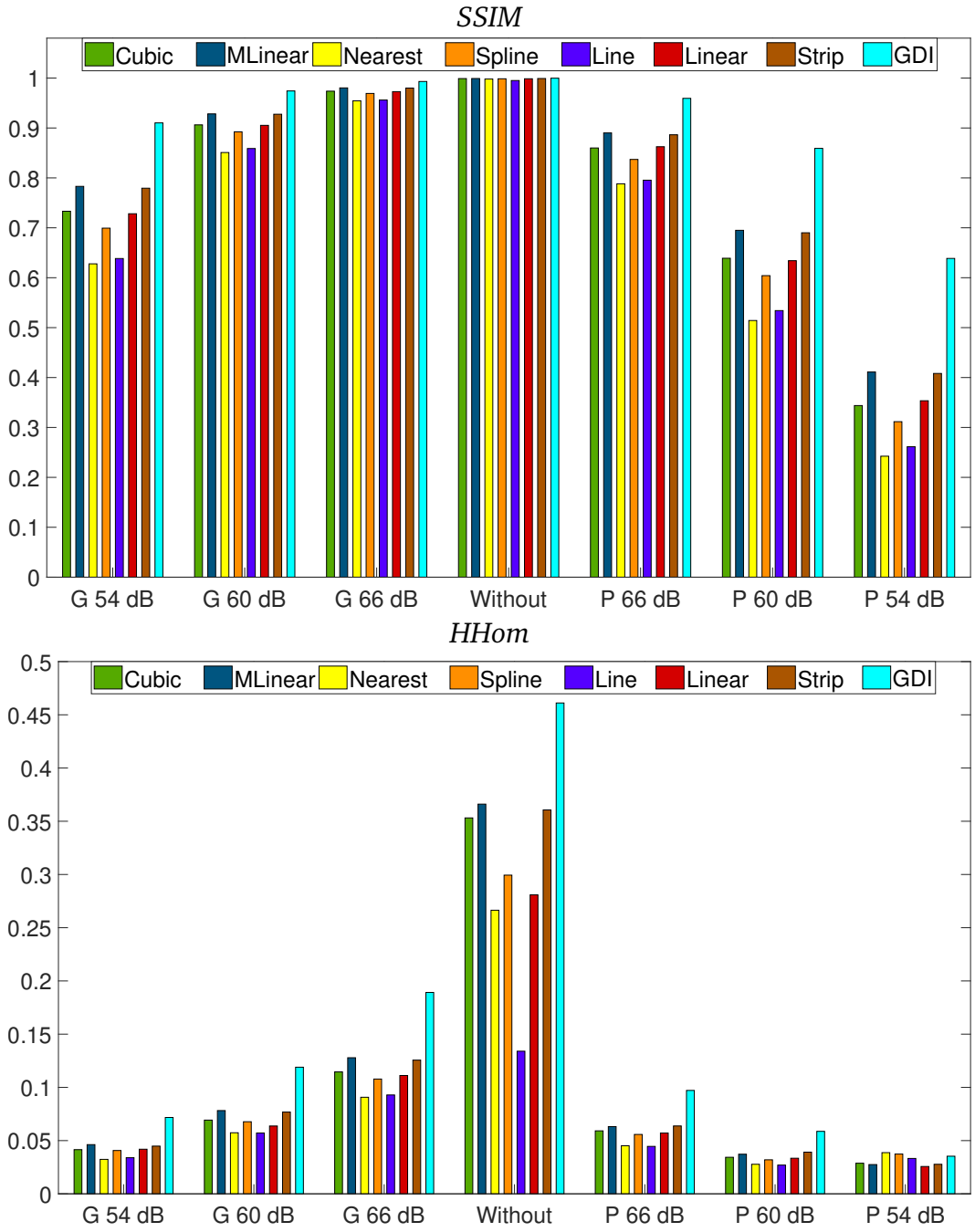

Figure 2.11: *The average values for all ROIs of the Forbild head phantom with FBP. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown to the left and the Poisson noise to the right.*

2.4 Correction study

In Section 2.3 we performed a comparison study with different interpolation methods and concluded that *GDI* resulted in the most accurate reconstructions. Others reasoned that images with *Strip* interpolation can be more appealing to human eyes [21] and we argue that it better represents the real word situation. Therefore, we further examined these methods and also included the *Line* and *Linear* interpolations because of their popularity. In this second study about interpolation errors, our main goal was to understand the root cause of the interpolation errors and to reduce the artifacts.

In this study, we applied the same noise levels that we introduced in Section 2.3.2 in the case of geometrical phantoms. In the tests both the *FBP* and *SIRT* algorithms were included, but here we used the algebraic approach, which allowed us to manipulate the stored weights between the projection lines and pixels. Looking at other parameters, we made some changes. This fine-tuning allowed us to focus more on the interpolation error and the effectiveness of the correction we proposed. Table 2.1 summarizes all parameters of this second study. Some of the parameters have already been explained, and some will be explained in detail in the following subsections.

Table 2.1: *The variable parameters we used in Section 2.4. (FBP: Filtered backprojection, SIRT: Simultaneous Iterative Reconstruction Technique, GDI: Gaussian Distance Interpolation, G: Gaussian, P: Poisson, MAE: Mean Absolute Error, SSIM: Structural Similarity, HHom: Haralick Homogeneity)*

Parameter name	**Tested settings**
Algorithm	*FBP*, *SIRT*
Interpolation	*Line, Linear, Strip, GDI*
Geometry	*Basic, View++, Ray++*
Noise	G 36 dB, G 42 dB, G 48 dB, Noiseless, P 48 dB, P 42 dB, P 36 dB
Normalization	*Raw, Corrected*
Metric	*MAE*, *SSIM*, *HHom*

2.4.1 The proposed correction using pixel coverage

We found that the interpolation artifacts are diverse properties of the interpolation methods but are closely related to how much the rays contribute to a pixel in the projection geometry. The concept of obtaining the contribution is what we will call pixel coverage. The pixel coverage can be calculated in multiple ways by considering one ray or a group of rays. Let us define two versions of the pixel coverage using the notations introduced in (1.4). Let the global coverage of the j-th pixel of $\mathbf{x}$ be

defined as

$$\mathcal{C}_j = \sum_{i=1}^{m} W_{ij} \quad , \tag{2.2}$$

giving the sum of coefficients in all the projections. Moreover, let Θ_k be the set of projection line indices i belonging to the projection angle k, and let the $\mathcal{C}_{j,\Theta_k}$ directed coverage of the j-th pixel be calculated as

$$\mathcal{C}_{j,\Theta_k} = \sum_{i \in \Theta_k} W_{ij} \quad . \tag{2.3}$$

For each pixel this gives us the sum of projection coefficients within a projection. We can also say that the global and directed pixel coverage are related as

$$\mathcal{C}_j = \sum_{k=1}^{m_p} \mathcal{C}_{j,\Theta_k} \quad . \tag{2.4}$$

The directed and global pixel coverage determines how much a pixel contributes to a projection or the sum of projections. It also reflects how much intensity is propagated into the specific pixel when reconstructed. We assume that imbalances in the local and global pixel coverage show how uneven the representation of pixels in the geometry is, and thus can cause artifacts.

To show this phenomenon, we calculated projection matrices with the same geometries but different interpolations. We then calculated the global coverage of each f'_j pixel for each interpolation. The resulting global coverage maps are shown in the first row of Figure 2.12. In the second row, one can see the reconstructions (*Raw*) minus the ground truth (GT), where no correction was applied. One can observe that the patterns in the first and second rows of Figure 2.12 correlate with each other. In the following, the global coverage map is what we refer to simply as pixel coverage.

Our proposal is to even out the global coverage map by dividing the weights of the projection matrix in every given view angle by the corresponding directed coverage, such as:

$$W'_{ij} = W_{ij} / \mathcal{C}_{j,\Theta_k} \quad | \quad i \in \Theta_k \quad . \tag{2.5}$$

Basically, we normalize the weights by the sum of the weights within the angles. After normalization, the directed coverage map of W'_{ij} is equal to one for all angles. The effect of the correction is shown in the third row of Figure 2.12 (*Corrected* - GT), where the normalized projection matrix was used for reconstruction. As the pixel coverage map contains only ones in the case of the *Strip* interpolation originally, we expect no significant change after correction in that case.

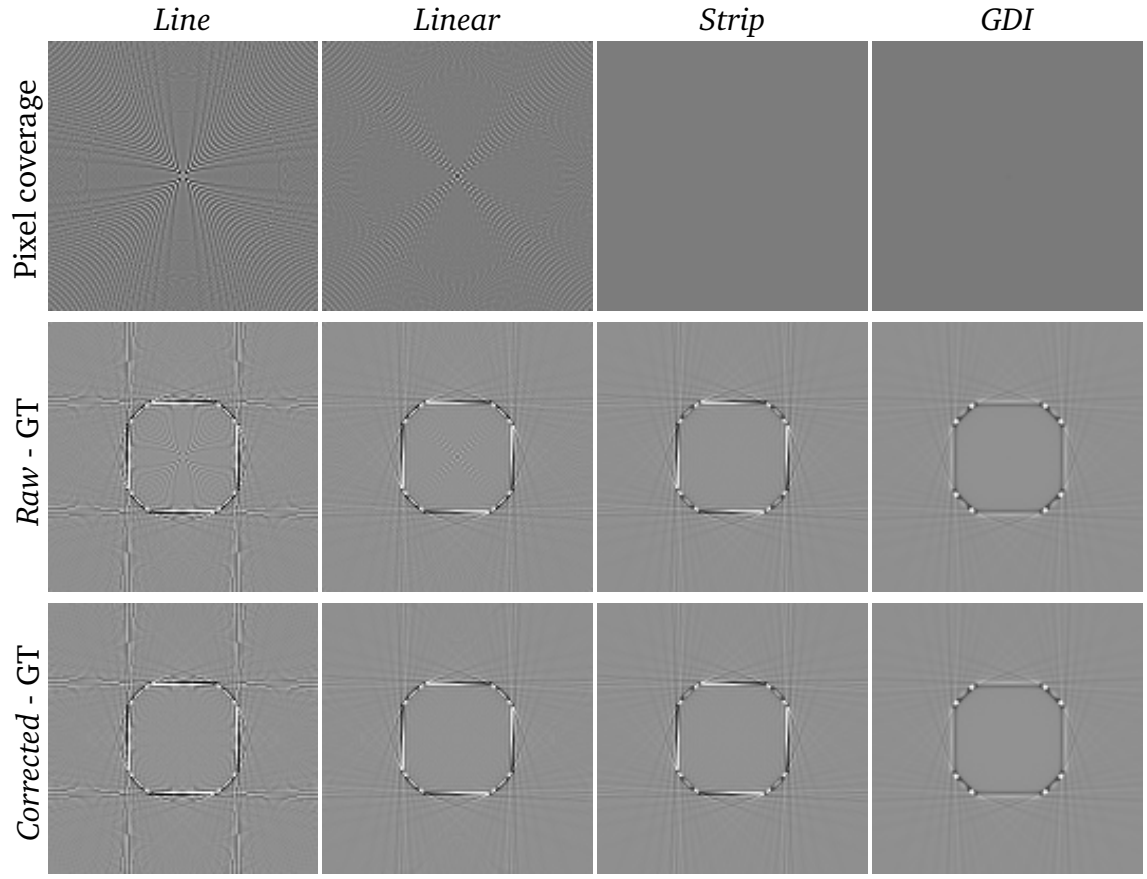

Figure 2.12: *The first raw shows the pixel coverage according the indicated interpolation method. Pixel coverage images are normalized between the interval [0, 1]. Standard deviations of the pixel coverage maps: Line: 0.0682, Linear: 0.0423, Strip: 0.00002, GDI: 0.00037. The second and third raw contain the corresponding difference images of the stop_sign phantom (see Figure 2.13), where Raw is the uncorrected reconstruction, Corrected is the reconstruction with correction and GT is the ground truth, stop_sign in Figure 2.13. The display window of the difference images is [-0.15, 0.15].*

2.4.2 Geometry

The number of detectors (m_d) was calculated by 2.1, but we consider the value of m_d as a variable in this study. In the case of an image $n \times n$, the diagonal length is $\sqrt{2}n$. Therefore, we used projection lines $2\sqrt{2}n$ to cover the entire diagonal length (that is, $m_d = 2\sqrt{2}n$).

We used three different settings considering the geometry, all of which had at least the required number of projection lines and the minimal number of projection angles to obtain a theoretically perfect reconstruction. In the case that we will call "*Basic*" geometry, we set the number of lines to 366 and the number of views to 285. When we used the geometry called "*Ray++*", we increased the number of

lines to 732, while the number of viewing angles remained the same. The so-called "*View++*" geometry was implemented with 570 view angles and 366 lines. Taking into account the distance between lines (that is δ in Figure 1.3), we have set the value 0.5 for *Ray++* and 1.0 for the rest. Table 2.2 summarizes the exact values of the geometries.

Table 2.2: *Parameter values of the tested geometries.*

	Number of lines	**Number of views**	**Distance among lines**
Basic	366	285	1
View++	366	570	1
Ray++	732	285	0.5

2.4.3 Test phantoms

Our test database consisted of 12 phantoms with a size of 128 $\times$ 128 pixels. The intensities of the images were selected from the $[0, 1]$ interval. All images are shown in Figure 2.13. We built four groups of phantoms to construct a data set for structured experiments. First, we placed the same circle in the images, but in slightly different places (see the first row of Figure 2.13). Second, we altered the various shapes in some images to ensure that the objects had roughly the same area (first two images in the second row of Figure 2.13). Third, we used binary and nonbinary phantoms, including the Sheep-Logen[55] and Forbild [68] head phantoms (see the third row of Figure 2.13). Note that here the head phantoms have the size of 128 $\times$ 128 pixels as the other phantoms and the noise levels are also the same. Due to this small size, we generated the Forbild phantom without the details that represent the ears. The projections of the images were produced by an analytical computer simulation that calculates the line intersection of the geometric shapes with the projection lines; therefore, the projection values were not calculated from rasterized phantoms. In this task, we utilized the codes of [68] such as in Section 2.3.

The artifacts – caused by the interpolation error – can be detected on homogeneous regions of objects. In the first study described in Section 2.3, we determined one ROI for the geometrical phantoms and multiple ROIs for head phantoms. Here, we follow a different approach and focus on the inside of the objects. We calculated all metrics only inside the largest object of the image. To do that, we applied per-image masks. Initially, the pixels outside the largest object of the image are assigned to background pixels. Then, we morphologically eroded the foreground pixels to exclude the pixels of the borders. The border pixels of the object are naturally highly variable, which we do not consider as a heterogeneous region. Although the border reconstruction capability of the interpolation methods can be an interesting question,

it was not in the scope of the study carried out. Figure 2.14 shows an example of the GT image and its mask. The mask determines the region where the metrics were calculated. Only the blue pixels will be included in the calculation of the metrics.

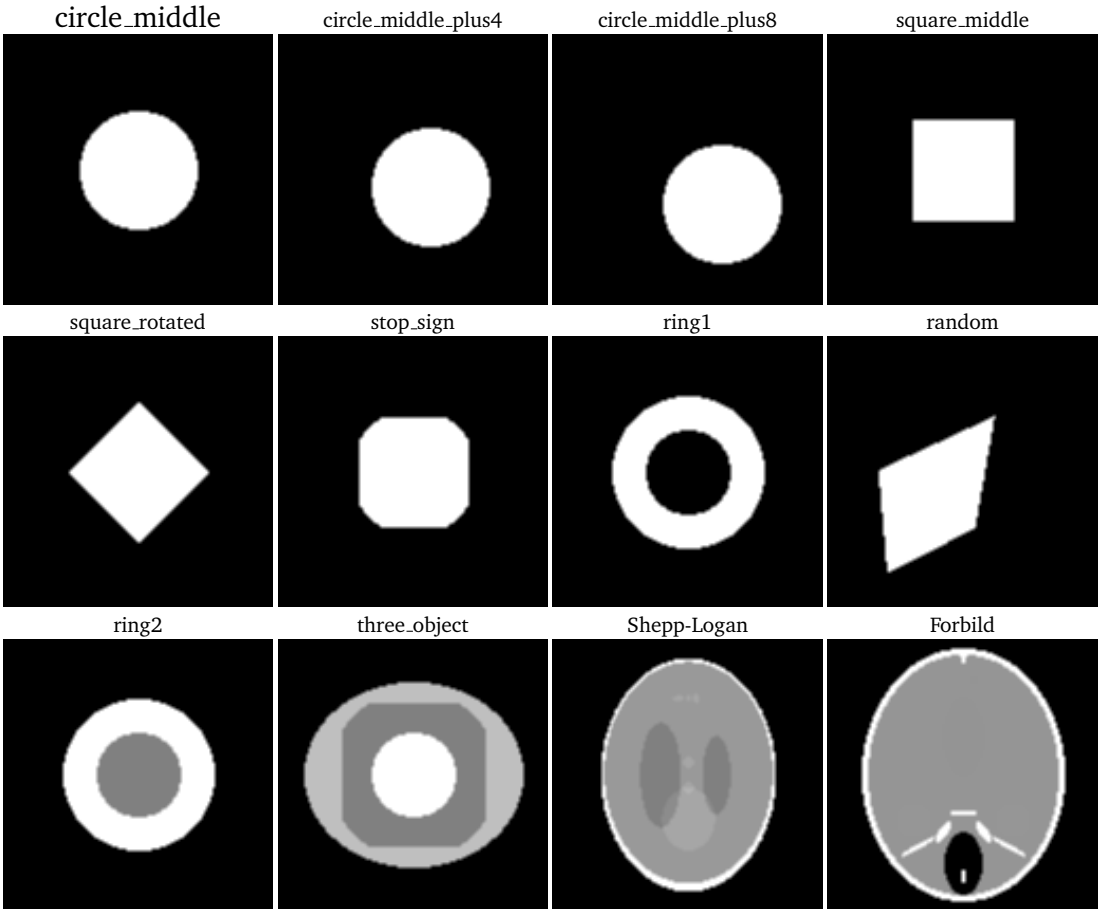

Figure 2.13: *Phantoms of the Correction Study.*

Figure 2.14: *Example of a phantom and its mask. (Colors: black - background; white - object pixels of the square_middle phantom (see Figure 2.13); blue - mask) We included only the pixels with the color of blue at the calculation of the metrics.*

2.4.4 Results with *Standard* geometry

In this section, we will present our results and share our finding. Table 2.3 contains the results for all phantoms individually in a setup, where we use *FBP* reconstruction, *Basic* projection geometry, *Noiseless* projection data, and evaluate results with *HHom*. Statistics are provided for both the *Raw* version of the interpolation methods, and the global coverage map balanced variances obtained by (2.5). Moreover, Table 2.3 shows the Average (AVG) and Standard Deviation (STD) of the corresponding columns. To better visualize the magnitude of STD compared to the difference in interpolation methods, we plotted the AVG and STDs together in Figure 2.15. We noticed that the STD is relatively high compared to the difference between interpolation methods, but we argue that this is due to the variability of the phantoms in our dataset (see Table 2.3). To support this conclusion, we calculated the Pearson correlation among the interpolation methods, which were calculated including the entire result set. Table 2.4 shows the pairwise correlations between the methods. We understand that the high value of the pairwise correlation (always above 0.9474) means that the interpolation methods follow the same trend throughout the test phantoms. Therefore, in the following sections we will discuss only the AVG values. In Table 2.3, the reconstructions produced by the corrected projection matrix resulted in increased *HHom* values compared to the uncorrected (*Raw*) case with *Line* and *Linear* interpolations. This suggests that the interpolation error can be reduced by the proposed correction step.

Table 2.3: *The HHom values of all phantoms with the settings: FBP reconstruction, Noiseless data and Basic projection geometry. Average (AVG) and Standard Deviation (STD) values are presented. In the case of Raw, we used the uncorrected projection matrix. Phantoms marked with * have roughly the same area.*

FBP*, *Basic*, Noiseless, *HHom	***Line***		***Linear***		***Strip***		***GDI***	
	Raw	***Corrected***	***Raw***	***Corrected***	***Raw***	***Corrected***	***Raw***	***Corrected***
AVG	**0.4291**	**0.6261**	**0.5590**	**0.7004**	**0.7463**	**0.7463**	**0.8940**	**0.8821**
STD	**0.0799**	**0.1820**	**0.0961**	**0.1610**	**0.1467**	**0.1467**	**0.1326**	**0.1162**
circle_middle*	0.4961	0.9438	0.6750	0.9470	0.9514	0.9514	0.9756	0.9756
circle_plus4*	0.4870	0.8210	0.6147	0.8629	0.8831	0.8831	0.9978	0.9607
circle_plus8*	0.5063	0.7839	0.6166	0.8195	0.8350	0.8350	0.9883	0.9169
square_middle*	0.3796	0.4925	0.5425	0.6367	0.7707	0.7707	0.9830	0.9878
square_rotated	0.3151	0.4867	0.4702	0.6327	0.7462	0.7462	0.9675	0.9688
stop_sign*	0.4301	0.6035	0.5530	0.6488	0.6751	0.6751	0.9892	0.9338
ring1	0.4639	0.6607	0.6387	0.8087	0.8533	0.8533	0.9601	0.9655
ring2	0.4863	0.7584	0.6387	0.8164	0.8386	0.8386	0.8560	0.8564
random	0.3595	0.4521	0.4598	0.5291	0.6012	0.6012	0.9273	0.8619
three_object	0.4722	0.6145	0.5806	0.6821	0.7050	0.7048	0.7148	0.7837
Shepp-Logen	0.4865	0.6059	0.5824	0.6603	0.6914	0.6914	0.7778	0.7793
Forbild	0.2662	0.2896	0.3356	0.3601	0.4045	0.4045	0.5904	0.5945

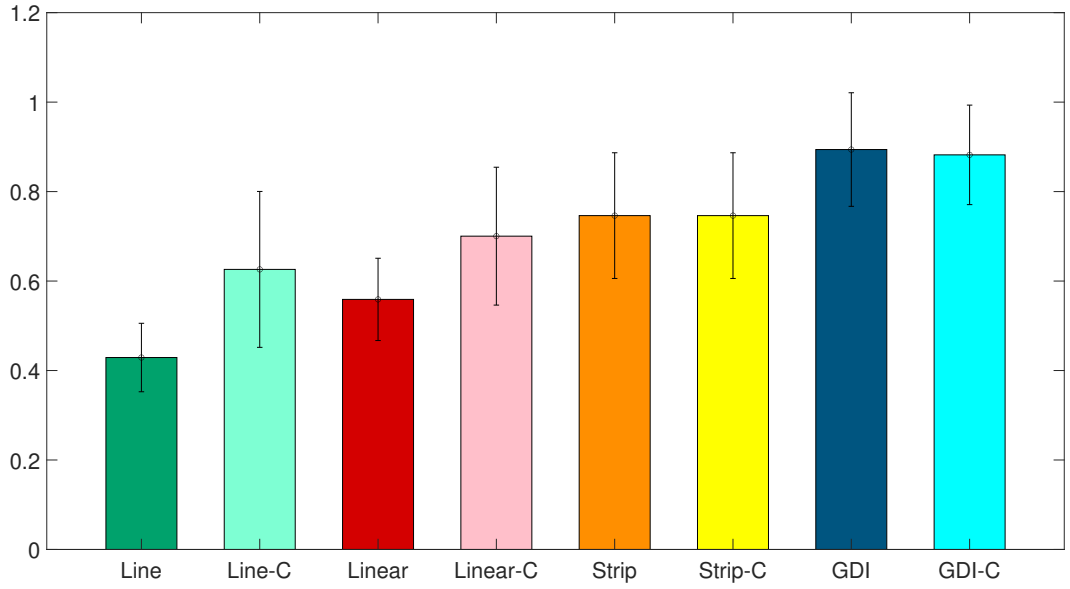

Figure 2.15: *The average values from Table 2.3 (settings: FBP, Noiseless, Basic, and HHom). The Corrected versions are marked with the "-C" suffix. Standard deviations are shown as error bars.*

Table 2.4: *The Pearson correlation between the interpolation methods.*

Full dataset		***Line***		***Linear***		***Strip***		***GDI***	
Pearson correlation		***Raw***	***Corrected***	***Raw***	***Corrected***	***Raw***	***Corrected***	***Raw***	***Corrected***
Line	***Raw***	1.0000	0.9982	0.9963	0.9961	0.9843	0.9843	0.9612	0.9613
	Corrected	0.9982	1.0000	0.9945	0.9963	0.9886	0.9886	0.9605	0.9605
Linear	***Raw***	0.9963	0.9945	1.0000	0.9994	0.9815	0.9815	0.9776	0.9777
	Corrected	0.9961	0.9963	0.9994	1.0000	0.9853	0.9853	0.9772	0.9773
Strip	***Raw***	0.9843	0.9886	0.9815	0.9853	1.0000	1.0000	0.9474	0.9475
	Corrected	0.9843	0.9886	0.9815	0.9853	1.0000	1.0000	0.9474	0.9475
GDI	***Raw***	0.9612	0.9605	0.9776	0.9772	0.9474	0.9474	1.0000	1.0000
	Corrected	0.9613	0.9605	0.9777	0.9773	0.9475	0.9475	1.0000	1.0000

2.4.5 Reconstruction algorithms and noise

Let's look at the results considering only the *Basic* geometry from the geometries we detailed in Section 2.4.2. Figures 2.16 and 2.17 show two grouped bar charts each with the average values of the *MAE* and *SSIM* metrics. With *HHom* the resulted charts are located in the Appendix referred as 4.4. Regarding the reconstruction algorithm, one can see that the *SIRT* has higher *SSIM* and *HHom*, except for a few cases with zero or low level Gaussian noise using the *GDI*. This better performance of the *SIRT* is expected according to the literature [69]. With *MAE* the images produced by *SIRT* have less error even at the highest level of Gaussian noise, but with increasing the strength of the Poisson noise the *FBP* has smaller values of average except for *Line* interpolation. A rather interesting phenomenon also occurs when looking at the *Noiseless* and right-hand side (Poisson noise) part of the bar chart with *MAE*.

Compared to the noiseless case, one can see a drop in the *MAE* values when we added a low level of noise (48 dB). In our opinion, when we randomly added the low-level noise to the image, it could change the values of individual pixels with the same volume but in the opposite way as the interpolation artifacts did. Nevertheless, it can be just a simple smoothing effect caused by the small inconsistencies in the projection matrix due to random noise. With that exception stated, generally we can say that adding noise to the projections has a negative effect on the image quality according to all metrics. Moreover, the quality of the reconstructed image dropped at a higher rate when Poisson noise was added. Examples of reconstructed images from noisy and noiseless data can be found in Figure 2.18.

Switching to the difference between the *Raw* and the *Corrected* versions (third column in Figure 2.18) we can see that applying the correction step efficiently removed the structural patters from the images, indicating the removal of interpolation artifacts. In the difference image, the effect of correction is clearly visible despite the type of noise. Still, only looking at the reconstructed images in the Poisson noise affected case, we can say that strong noise can dominate the interpolation errors, making them invisible on the results. On one note, this might have been the reason why interpolation errors are not in mainstream research. When the random noise in the measured data is strong, the interpolation errors are not significant compared to other distortions. With improving data acquisition techniques, on the other hand, good quality imaging techniques in the future may make it more vital to account for interpolation errors in tomographic reconstruction.

For a further detailed analysis of the proposed correction technique, Figure 2.19 shows a phantom reconstructed by all interpolation methods with and without correction. We also provide the *MAE* maps in the second and fourth rows (that is, essentially the pixelwise absolute difference between the phantom and the reconstructed image). The *Line* and the *Linear* interpolation methods are most affected by the interpolation error. The structures of the interpolation errors are clearly visible in those images. This supports the explained connection between the interpolation error and the pixel coverage in Section 2.4.1. One can recognize the same patterns as we presented in Figure 2.2. As the *Strip* interpolation originally has a uniform pixel coverage, the effect of the correction is not visible. Regarding the balance in the pixel coverage, the *Strip* interpolation is a perfect interpolation technique. Note that despite the *Strip* model being perfectly balanced in terms of pixel coverage, the resulted image with *GDI* is still likely to be more appealing. We argue that this is because the *GDI* has an inbuilt smoothing capability that can suppress other artifacts not caused by coverage imbalance but coming from the filter of the Filtered backprojection algorithm. Therefore, it is not directly connected to the pixel coverage.

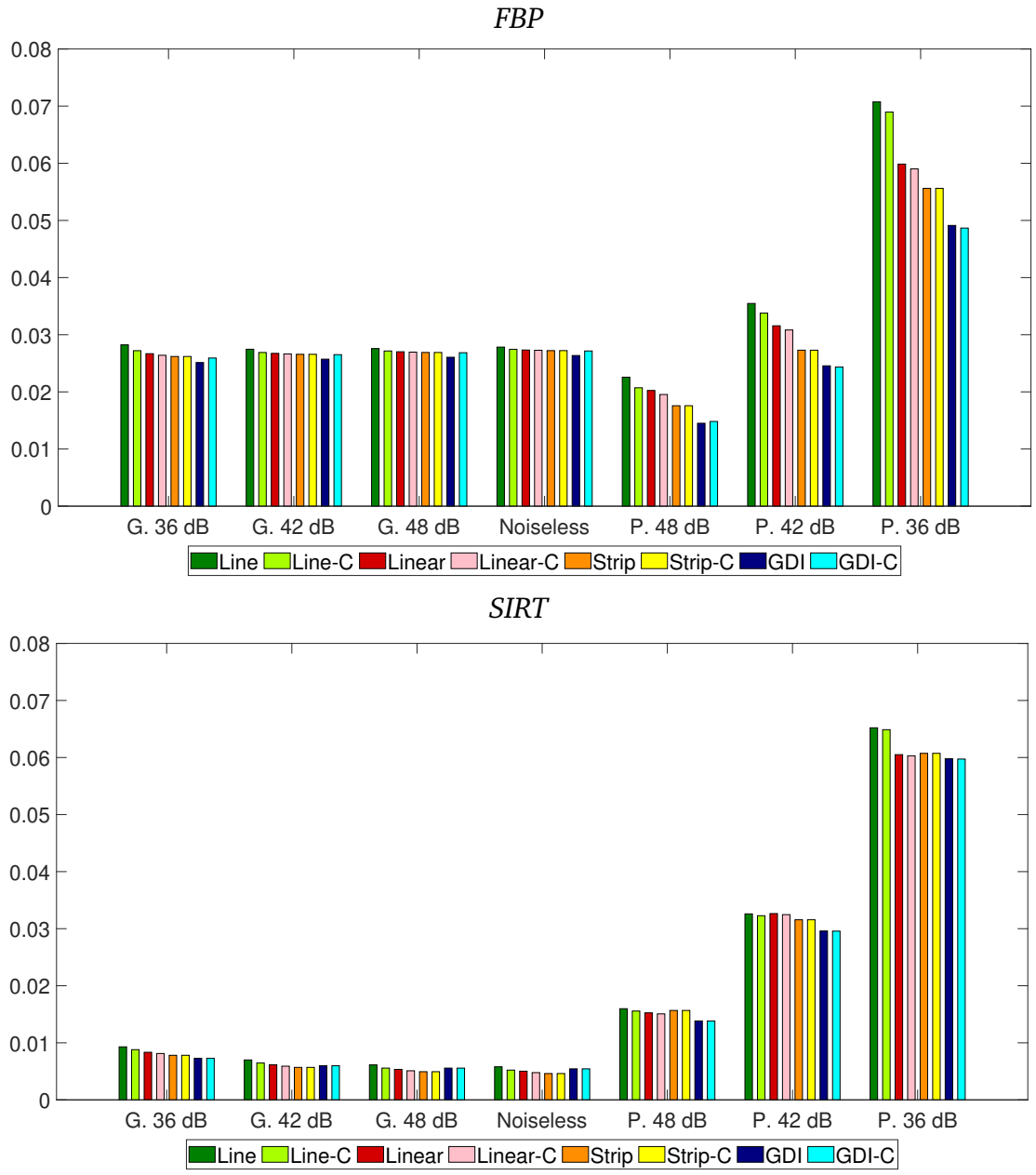

Figure 2.16: *The values are MAE averages of the entire dataset with Basic geometry. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown to the left and the Poisson noise to the right. The four interpolation methods are represented with and without correction in each group. The Corrected versions are marked with the "-C" suffix.*

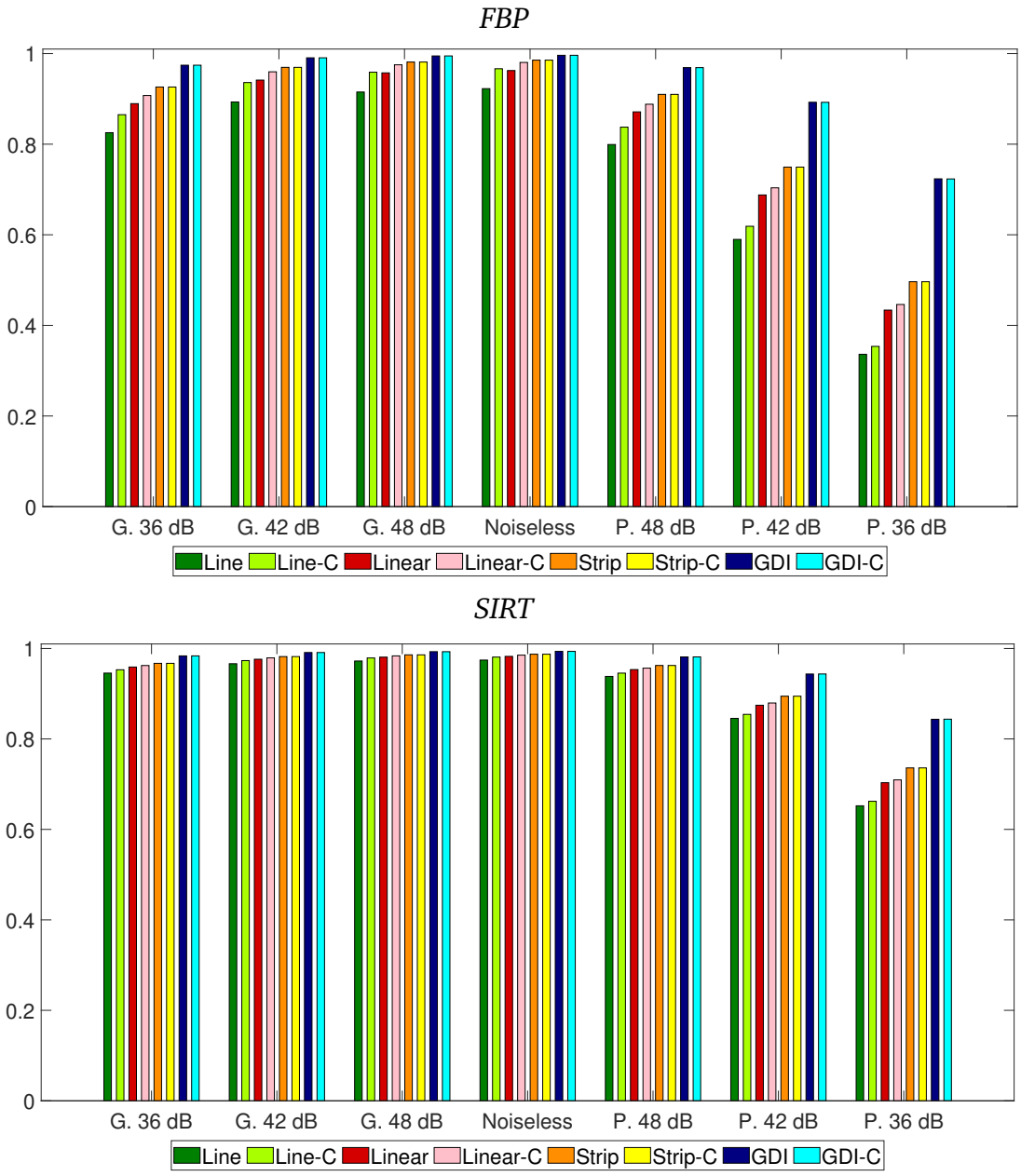

Figure 2.17: *The values are SSIM averages of the entire dataset with Basic geometry. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown to the left and the Poisson noise to the right. The four interpolation methods are represented with and without correction in each group. The Corrected versions are marked with the "-C" suffix.*

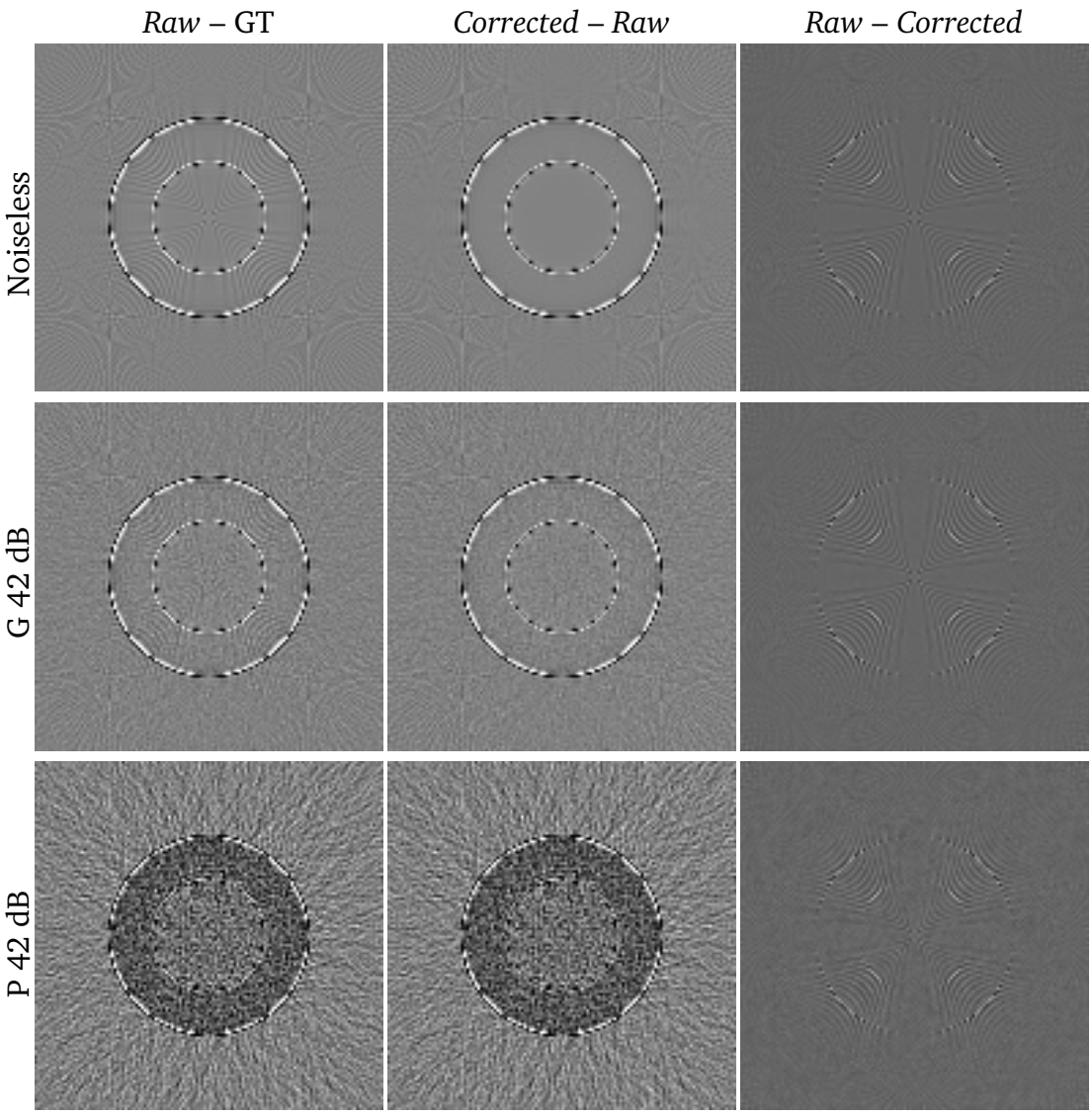

Figure 2.18: *Resulted images of ring2 from 2.13 with 42 dB noise. The settings: Basic geometry, FBP and Line interpolation. The figure contains results of noiseless case with medium-level Gaussian and Poisson noise. The Raw and Corrected versions are also shown in the first and second columns, respectively, while the third column is the difference between the two. The display window is [-0.06, 0.06] for the Raw – Corrected case and [0.47, 1.06] for the others.*

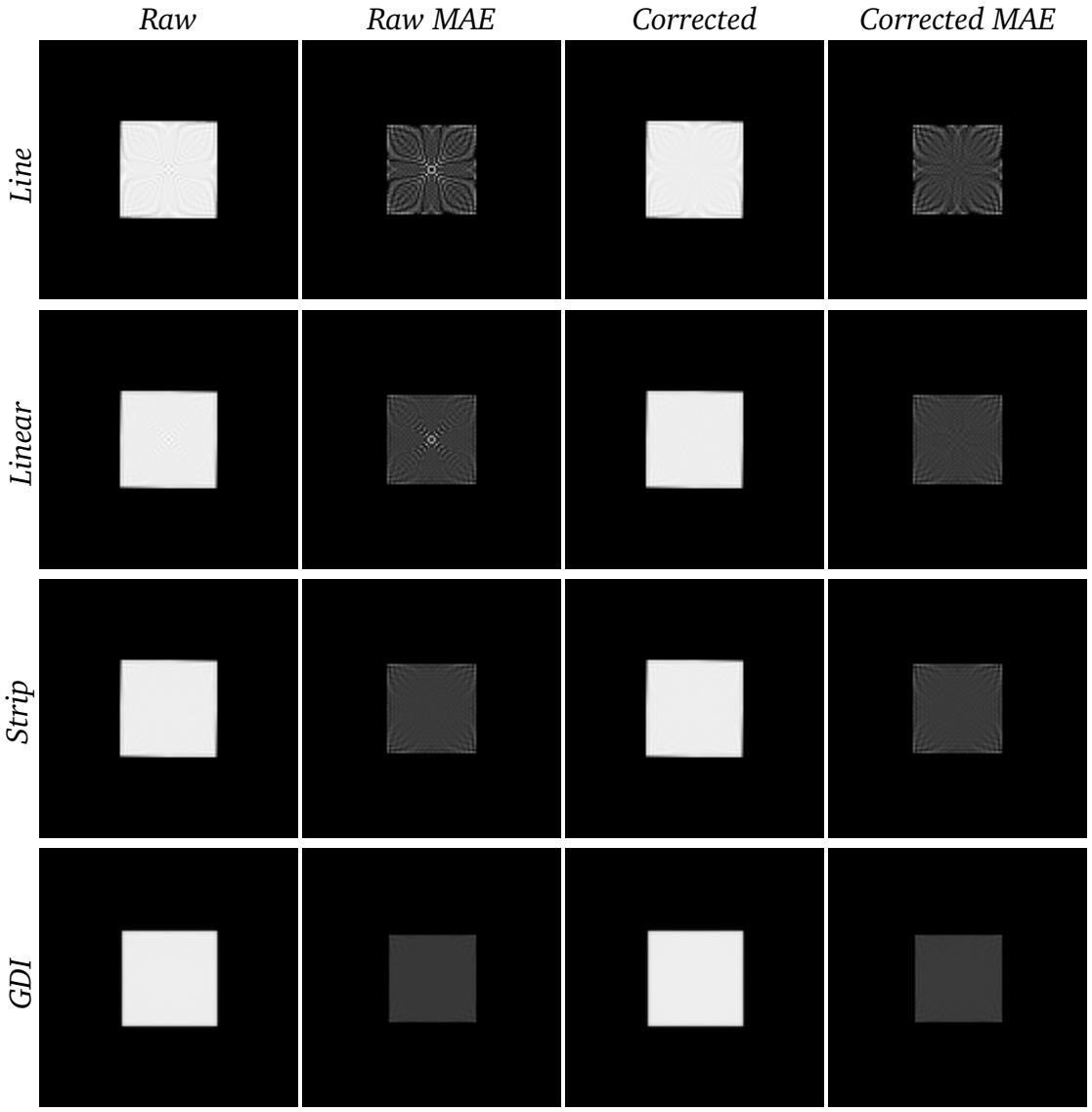

Figure 2.19: *Resulted images of square_middle from 2.13. The settings: Basic geometry, FBP and Noiseless. All interpolation methods are presented with and without correction with the corresponding MAE map. The display window is [0, 0.09] for the MAE maps and [0.47, 1.06] for the reconstructed images.*

2.4.6 Results with *View++* and *Ray++* geometries

We were interested in how the increased number of view angles or projection rays above the necessary number may affect the quality of the reconstructed image. We found that the pixel coverage improves when there are more rays or view angles. Still we can say that the improvement is more significant if we increased the number of rays. Figure 2.20 shows the coverage of the pixels with *Line* interpolation according to the three geometries. The main trends remain the same when switching from *Basic* to *View++* geometry, but the STD decreased from 0.0682 to 0.0658 due to doubled view angels. One can see that some of the noise outside the center of the image disappeared from the images of the *View++* geometry. In the case of the *Ray++* geometry, a more significant improvement in STD and a change in the pixel coverage pattern were recognized. Note that the number of rays and view angles cannot be arbitrarily raised in real-world applications. These numbers are physically limited by the equipment that is used to acquire the data. We also point out that the necessary number of rays and view angles depend on the size of the reconstructed image. Therefore, the physical limits of the equipment have a greater effect if an image with higher resolution is reconstructed. For example, the required number of ray and view angles according to Section 2.4.2 is 1138 and 1449 respectively, in the case of a 512×512 pixels sized image.

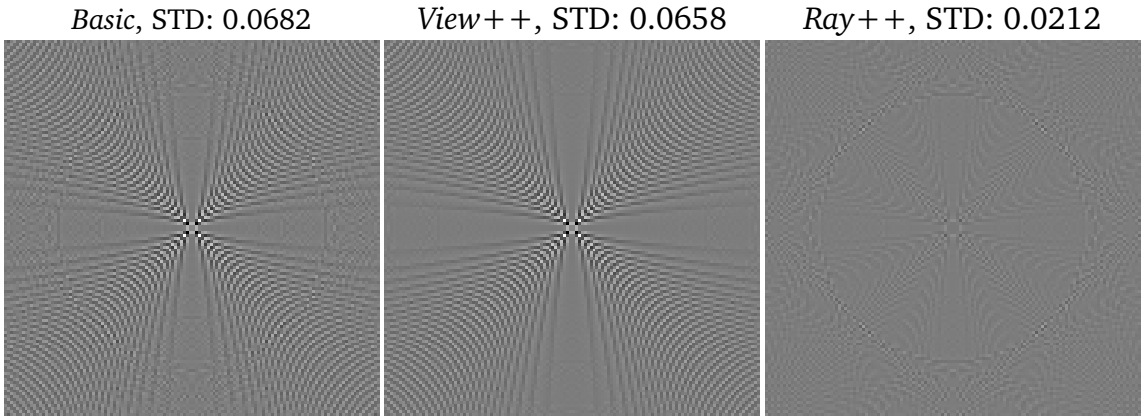

Figure 2.20: *The pixel coverage of the Line interpolation with the three geometries. The images were normalized between the [0, 1] interval.*

Figures 2.21 and 2.22 show the values *HHom* using the *View++* and *Ray++* geometries. The results of *SSIM* are omitted, because their conclusions coincide with the results of *HHom*. The calculated values of *MAE* can be found in the Appendix (Section 4), where Figures 4.5 and 4.6 contain the results using the *View++* and *Ray++* geometries respectively. The *View++* and *Ray++* geometries result in values and trends similar to the *Basic* geometry. As predicted by the pixel coverage, looking

at the *MAE* with the *Ray++* geometry, the effect of the correction is not significant. The *Corrected* and uncorrected versions of the interpolation method move together regardless of the noise or the algorithm. However, when examining the *HHom* values, the *Corrected* versions of the *Line* and *Linear* methods are showing improvement in the noiseless and the Gaussian noise affected cases. Significant improvement appears only with the *Ray++* and with the *Line* and *Linear* interpolation methods compared to the *Basic* geometry. Note that the interpolation error was originally worse with *Line* and *Linear*.

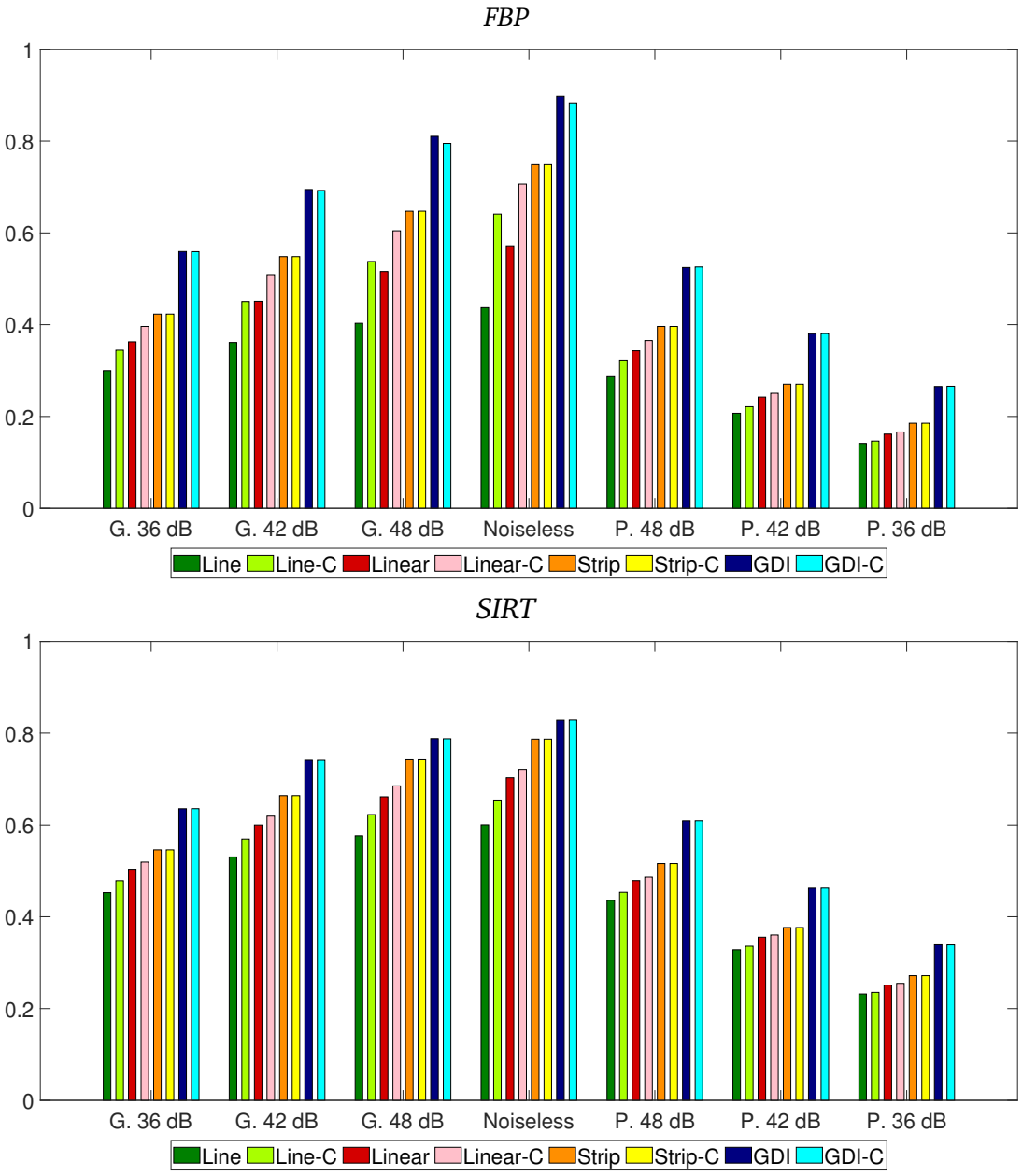

Figure 2.21: *The average HHom of the entire dataset with View++ geometry. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown to the left and the Poisson noise to the right. The four interpolation methods are represented with and without correction in each group. The Corrected versions are marked with the "-C" suffix.*

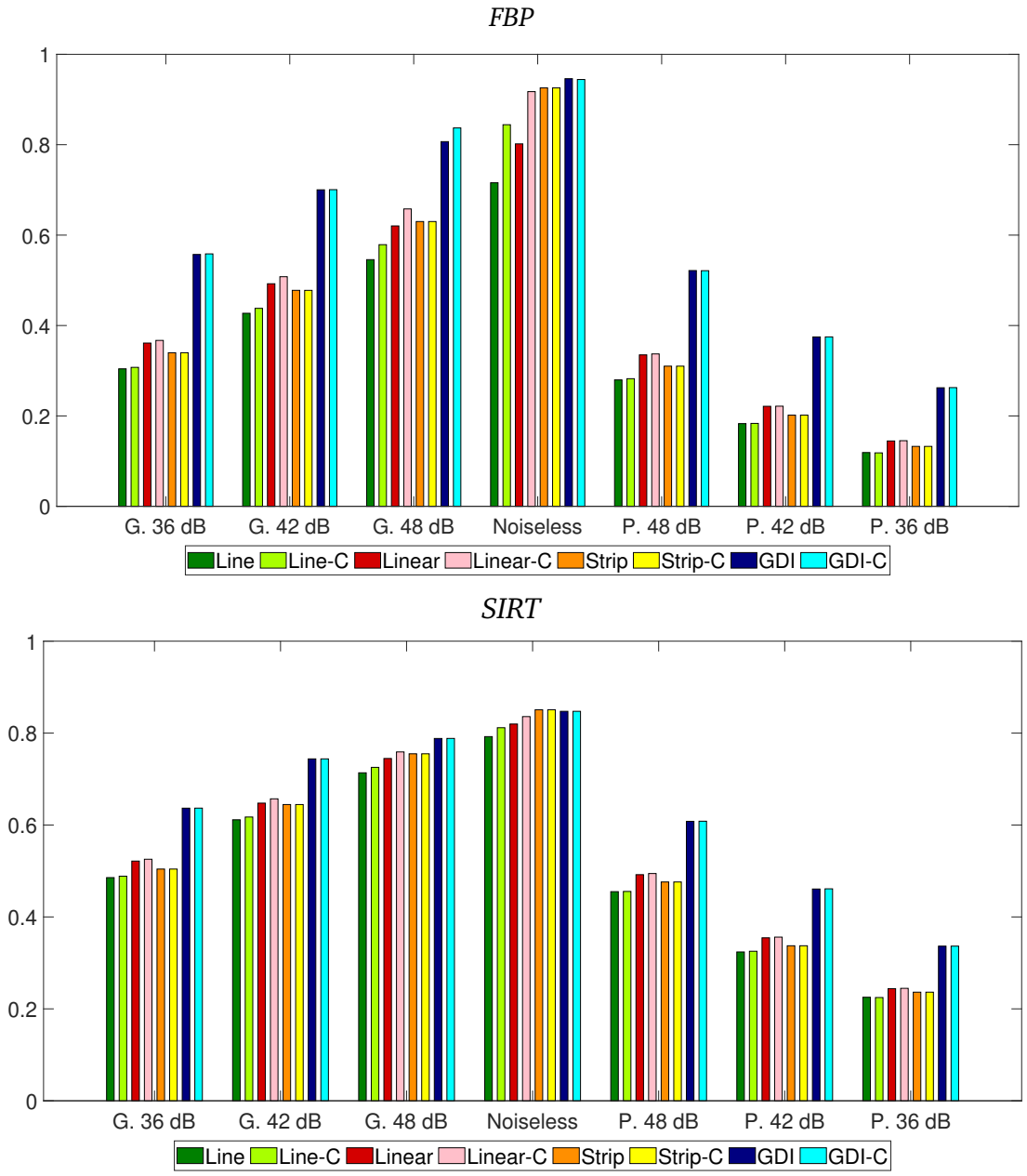

Figure 2.22: *The average HHom of the entire dataset with Ray++ geometry. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown to the left and the Poisson noise to the right. The four interpolation methods are represented with and without correction in each group. The Corrected versions are marked with the "-C" suffix.*

2.5 Discussion and concluding remarks

In the studies on interpolation error presented in Sections 2.3 and 2.4, we examined the effects of using discretized approximate models in transmission X-ray tomography.

Considering the first study, our observations are that the iterative algorithm was more robust to interpolation errors, and we can say that selecting the proper interpolation method for modeling the projection is important to gain good quality reconstructions. The more rudimentary interpolation methods like *Nearest* or *Line* produced significantly more artifact. In our experiments we found that the best projection geometries were the *Linear* interpolation-based one with pixel supersampling and the Gaussian-blob based (*GDI*) distance driven approaches.

In the second study, we proposed applying the pixel coverage to get a better understanding of the interpolation errors and also to visualize the different artifact structures caused by the different interpolation methods. We proposed an effective correction formula for the projection matrix to eliminate interpolation errors from the reconstructed images. We tested our proposed method in a comprehensive experiment and found that our proposed correction method significantly improved the quality of the image if the applied interpolation was the *Line* or the *Linear*. Although the increased noise levels can suppress the artifacts caused by the discretization, the calculated metrics indicated improvements due to our proposed correction at higher level of noise. Nevertheless, we also concluded that some discretized projection interpolation models, by design, behave better in terms of interpolation errors. We also found that increasing the projection lines in the projection geometry can improve the image quality in terms of interpolation error, but the increased number of view angles was not effective in achieving the same level of improvement.

The experiments we elaborated on in this chapter were made in a context of transmission X-ray tomography, but we think that our results could be useful in the broad landscape of tomography.

The author of this PhD thesis is responsible for the following contributions presented in this chapter:

I/1. I implemented a framework to test popular interpolation methods in various settings. The framework considers different projection geometries, seven noise levels, and two reconstruction algorithms. I evaluated the results of multiple test series on self-created datasets and popular head phantoms based on three error metrics.

I/2. I found that the iterative reconstruction algorithm was less sensitive to the applied interpolation method than the analytical reconstruction algorithm.

I/3. I proved the connection between pixel coverage and the interpolation error by experiments. I applied the pixel coverage to get a better understanding of the interpolation errors and also to visualize the different artifact structures caused by the different interpolation methods.

I/4. I used a correction formula for the projection matrix based on pixel coverage to eliminate interpolation errors from the reconstructed images and I gave experimental validation of the proposed correction method that can reduce the interpolation error.

I/5. I concluded that the interpolation error of the methods can be reduced by increasing the number of projection lines, but the increased number of view angles resulted in only minor changes.

Chapter 3

Artifact reduction using U-net based neural networks

In one major topic of my research, I was working on combining the Filtered Back-projection (*FBP*) with an outstandingly versatile and useful deep learning tool called U-net [53]. The U-net is an artificial neural network with a special structure, but generally speaking, the U-net is part of the CNN [54] and deep learning [35] families as one of the sub-fields of artificial intelligence. These kinds of techniques have achieved outstandingly good results in the field of computer vision and digital image processing in the last decade. Similarly, the application of CNNs became a trend in CT research. I present my work in this chapter, where I combined the two fields in a unique way to provide highly accurate methods for the case of tomography when the projection data show strong signs of beam hardening and random electrical noise.

Beam hardening is a physical phenomenon [32], which occurs because the lower energy photons of the polychromatic radiation are absorbed with a higher probability in the material of the object studied than the higher energy photons. Therefore, if polychromatic radiation is passing through an object, it will lose a greater proportion of its lower energy photons, so the ratio of higher energy photons will increase relative to the lower energy photons. As this happens, one can say that the beam becomes harder, which means that the inner layers of the object will interact with radiation having a different characteristic. Beam hardening artifacts appear in two forms, such as cupping (the interior of the object appearing to be darker) and dark or light streaks (see Figures 3.2 and 3.3).

Another type of distortion that occurs during the acquisition of data in tomography is called electrical noise. Electrical noise is a random factor in the measurement process that causes random changes in the measured values [13, 24]. In the reconstruction, it can cause streaks and random changes in the pixel values.

In order to cope with these distortions of the measurements (beam hardening and electrical noise), we introduce several changes to the original U-net structure

[53]. Moreover, we designed three novel U-net based methods for reducing artifacts in the reconstructions. We needed a dataset to train neural networks; therefore, we created a database with a virtual CT scanner called GATE [28, 29], producing realistic simulations of projection data.

This chapter is organized as follows. First, I summarize the related research in Section 3.1. After that, I describe our data generation process in Section 3.2 and the structures of the neural networks tested in Section 3.3. I present the results of two experiments in 3.4 and 3.5. Finally, the conclusions are drawn in Section 3.6.

3.1 Related works

The literature provides a variety of options (e.g., in [61]) for the combination of computer tomography and deep learning methods. There are approaches for the reduction of various artifacts, for example, beam hardening [20, 40, 51] and metal artifact [15, 16, 26, 50, 71]. Furthermore, the researchers in [14, 19, 65] were interested in performing the reconstruction with neural networks, while others applied deep learning as a pre- or post-processing tools before or after the reconstruction in [11, 36, 43].

In this chapter, I present the results of the research in which I participated in developing three novel deep convolutional neural network architectures for image reconstruction from projections. These methods provide end-to-end solutions taking projection data as an input and producing reconstructed images on their outputs. In comparison, the end-to-end solution in [65] has simpler architecture than ours and also it contains a fully connected layer. The end-to-end reconstruction architecture in [14] has a more complicated architecture with three well-separated parts. The first part is working on the projection data with convolution layers. The second part performs the reconstruction, while the third part is working on the reconstructed images with convolution and deconvolution layers. This method maintains a weak connection between the parts working with the projection data and the reconstructed images. Our methods were built on the U-net structure by incorporating the reconstruction at the end of the neural network or in skip connections. Using the reconstruction step as skip connections we achieved, that the reconstruction is not before or after the main part of the network, but in the middle of a U-net structure, thus the connection is stronger, which makes the training easier and more efficient.

3.2 Data generation

We used computer-simulated artificial data in this study, generating the projections of software phantoms. Our physically correct projection data were generated using the

GATE software [28, 29] without Compton scattering [17]. We used parallel beam geometry and photon-counting detectors in the GATE model. The projection data were generated with 596 angles, each containing 362 detectors. 400 000 photons left the source at each projection angle. The projection data was reconstructed as 256×256 images.

We have split our versatile dataset into two partitions, that we call *Dataset A* and *B*, in order to get a simple notation. For a more convenient discussion, from this point on, we will be mentioning these two groups of data as different datasets. The two datasets are different in the shape and material composition of the phantom object.

3.2.1 Dataset A

In the case of *Dataset A*, we used five different X-ray sources assigned in equal proportions to 5 000 phantoms to be more realistic. We used prefiltering during the generation of source characteristics, as it is a common way to reduce beam hardening [30]. The characteristics of the sources were calculated by [52] software. The sources differ only in the thickness of a pre-hardening aluminum filter. Figure 3.1 shows the characteristic of the sources. Each phantom was present in the dataset using one selected source. Furthermore, we calculated the images of Figure 3.2 showing an example of all the sources used with one phantom, which way the effect of using different sources is more visible. Figure 3.3 shows the intensity value profiles along the yellow lines in Figure 3.2. As one can see, the effects of beam hardening are decreasing toward the wider aluminum filters, while the electrical noise is increasing.

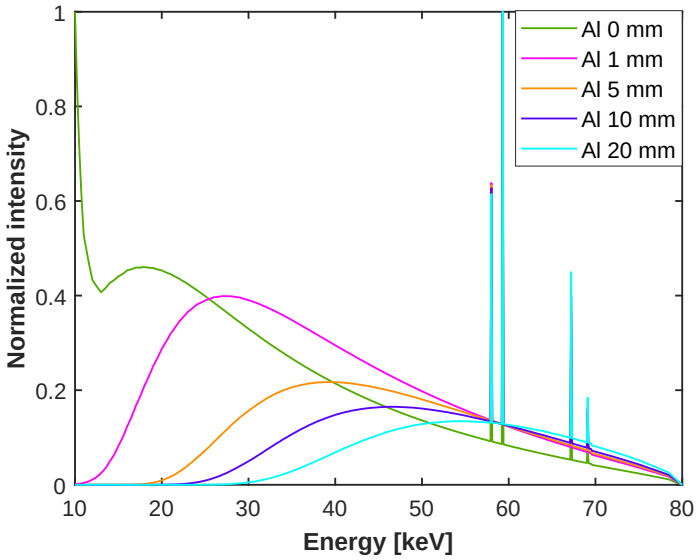

Figure 3.1: *The characteristics of the X-ray sources with different aluminum thicknesses.*

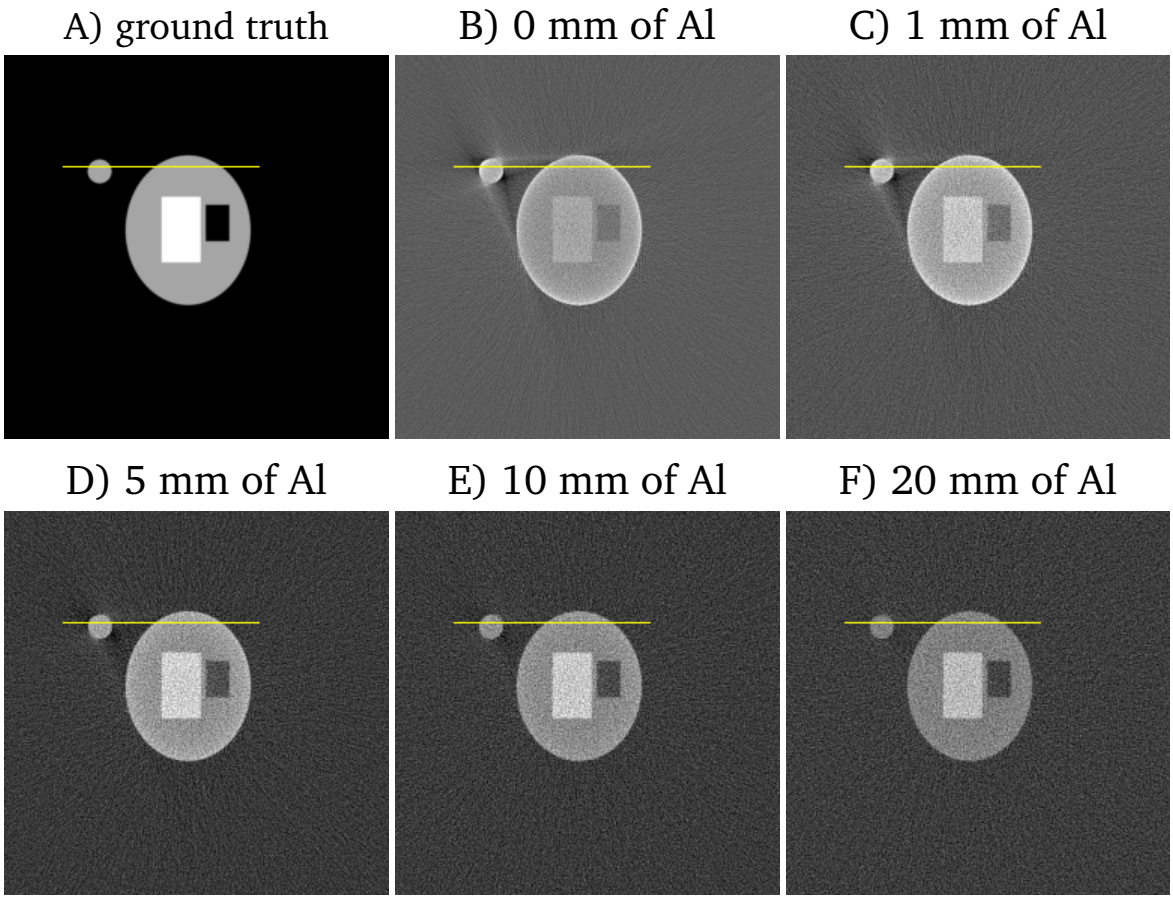

Figure 3.2: *The ground truth phantom and its reconstructions from distorted projection data acquired using different pre-hardening filters.*

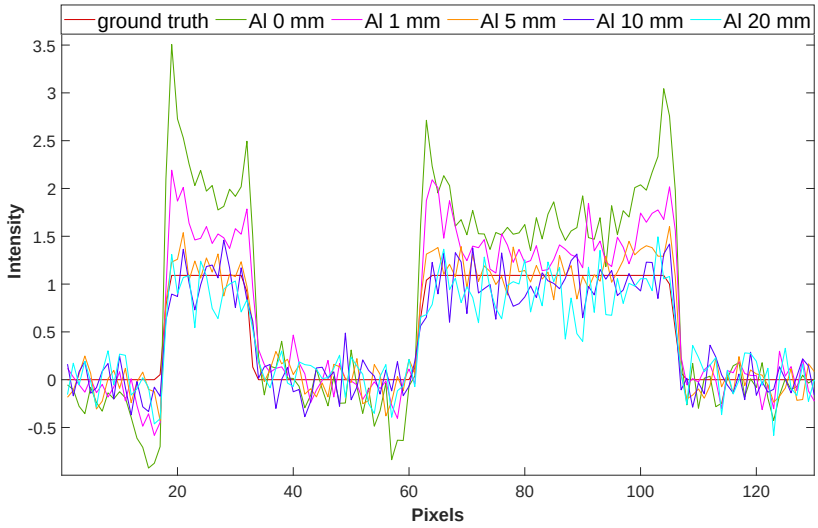

Figure 3.3: *Intensity value profiles along the yellow lines in Figure 3.2.*

The phantoms were generated as a combination of randomly chosen shapes, i.e., circles, ellipses, and rectangles. The objects may contain each other, but partial overlap was not allowed according to the GATE software restrictions. One object can consist of only one material from the following set: air, spine bone, rib bone, skull, blood, cartilage, kidney, kidney stone, and adipose. Moreover, the objects are various in size and location, which parameters were chosen randomly during phantom generation. The phantoms were generated in 4 groups according to their object counts. Each group contained a quarter of all the images (i.e., 1250 images), and groups were generated with a maximal object count of 4, 5, 6, or 7 objects. Figure 3.4 shows a few examples from this dataset.

For a ground truth projection set of the *Dataset A*, we calculated analytically correct projections with the help of [68] without any noise. We set the values of the materials to their mass linear attenuation coefficient. To this end, we performed specific GATE measurements, where the quotient of the detector intensity behind the object and the source intensity (in other words, the transmission value) were in the $[0.4995; 0.5005]$ interval. After conducting the measurements, we calculated the linear attenuation coefficient of the materials for each X-ray source based on (1.3) which was divided by the density of the material.

Before training, we applied (1.3) to the projection data. We then normalized the intensities to be in the [0, 1] interval. We used five different normalizing factors that correspond to the five sources.

3.2.2 Dataset B

A second dataset was produced using hand-drawn shapes as templates from [1] with the same simulation method as *Dataset A*. This dataset was excluded from the training and only used in the testing of the methods. Figure 3.5 presents a few examples from this dataset, which consists of 66 phantoms with various non-basic geometrical shapes. Results were generated with the source prefiltered with 5 mm aluminum. The 66 icon phantoms can be partitioned into three groups with 22 phantoms in each group. The object or objects of the phantoms in the first group (*"I"*) consist of rib bone while the background is always air. The material of the objects in the second group (*"II"*) can be air, spine bone, rib bone, skull, blood or adipose, while air, adipose, teeth, skull, ribs spongiosa or rib bone in the third group (*"III"*). This means that in the third group there are two materials (teeth and ribs spongiosa), which were not used during the training of the networks. In each group, random cracks were created and every phantom appears with and without the cracks. The cracks consist of air. Here, we also performed normalization with a distinct scaling factor in each group. The dimensions of the projection data and reconstructed images were the same as with *Dataset A*.

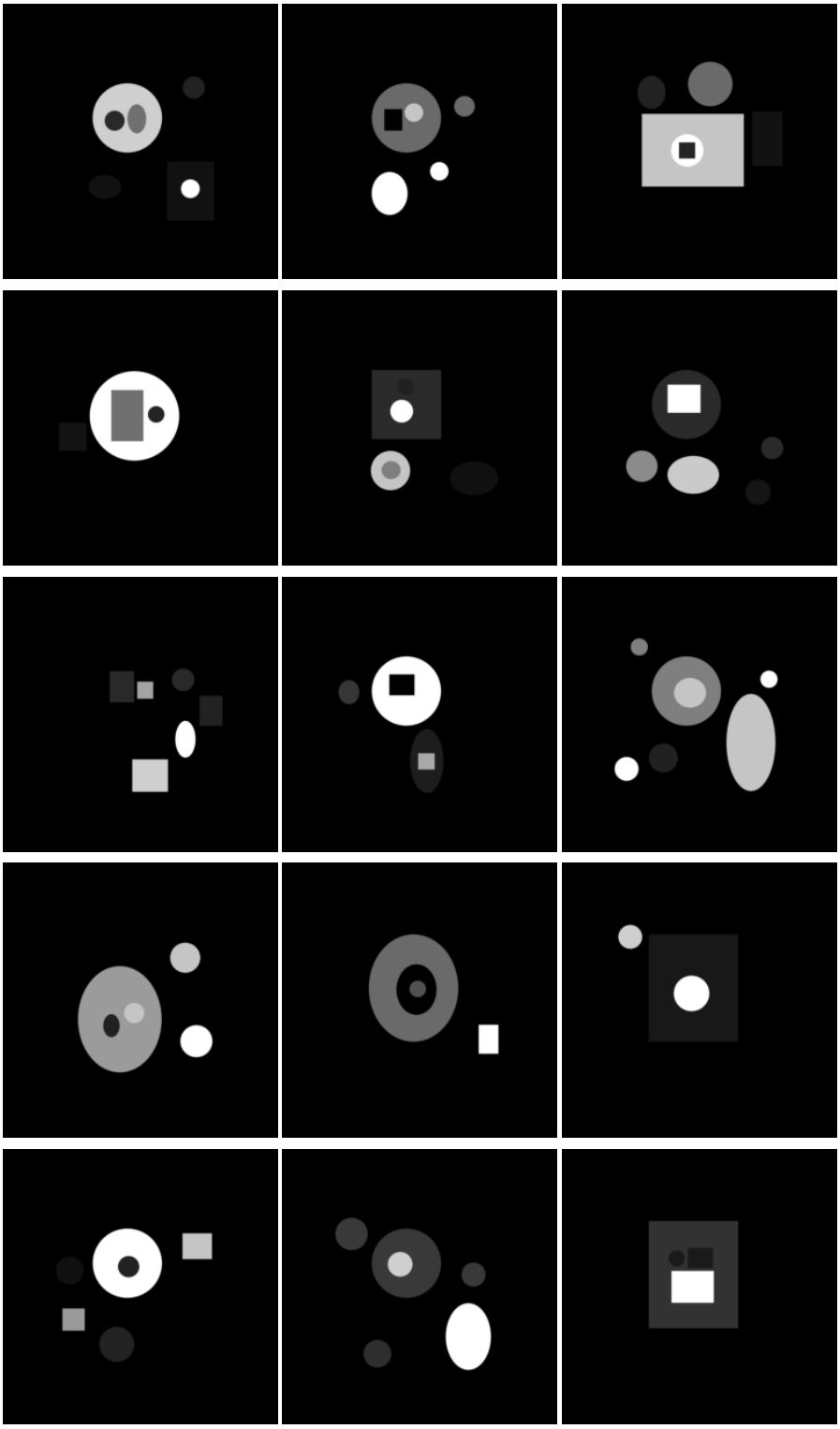

Figure 3.4: *Examples from the Dataset A.*

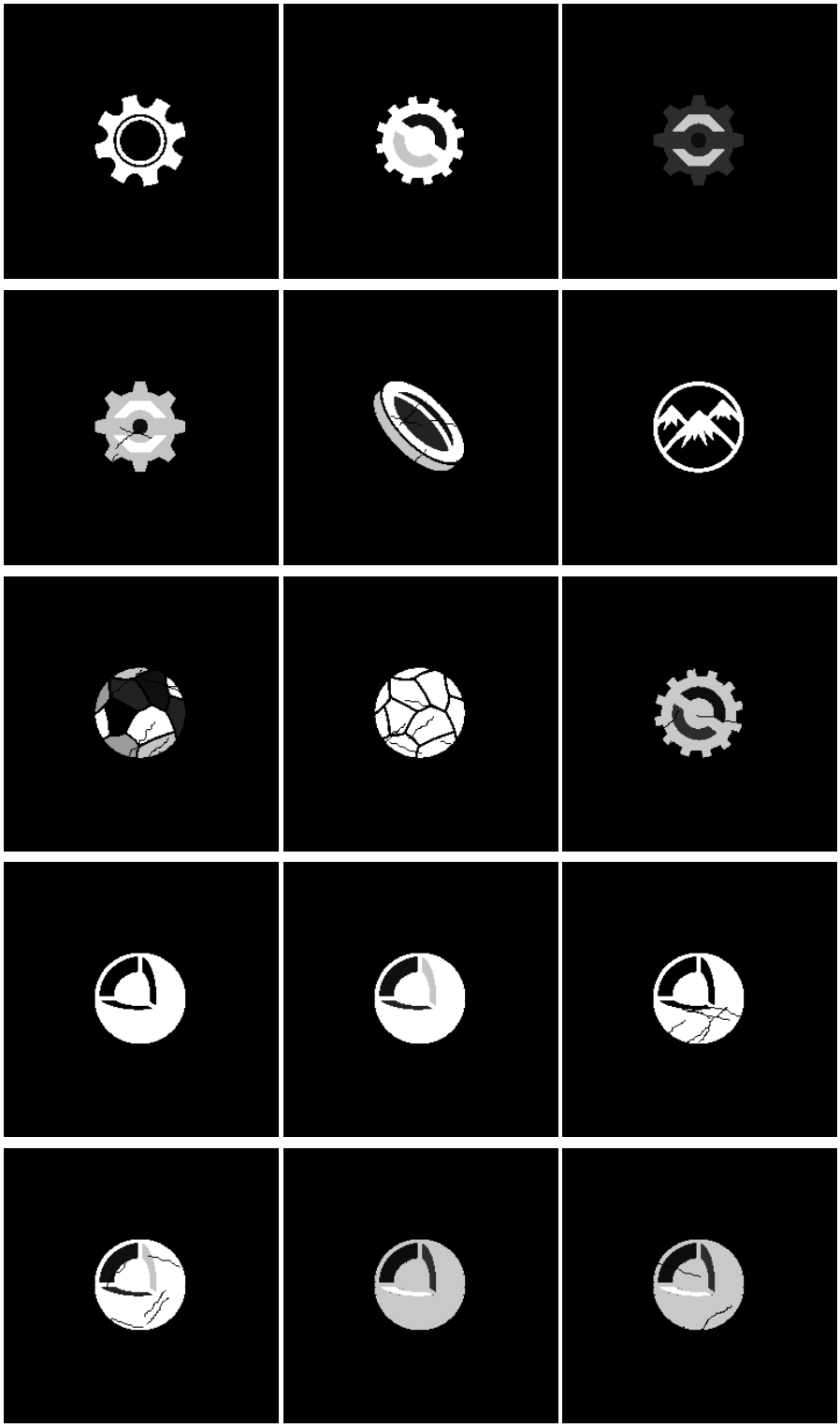

Figure 3.5: *Examples from the Dataset B.*

3.3 The used U-net based architectures

In our studies, we examined six neural networks. All of these networks are based on the special deep-, fully-convolutional neural network structure called U-net [53]. The name of this network originates from the special structure of the network as one can arrange the layers in a U-shape. The original U-net is consisting of one contracting and one expanding paths. During the contracting path, multiple convolutional layers are located after each other followed by a rectifier linear unit. A maxpooling layer is applied after two convolutional layers, which will reduce the dimension of the input and allow the network to extract features in broder receptive fields. During the expanding path the upsampling layers reset the original size of the input, while the spatial information and extracted features concatenated using skip connections. Our tested networks are the modifications of the original U-net. The structures of the networks are shown in Figure 3.6. Furthermore, to clarify the differences between the networks in terms of inputs and outputs, we made Table 3.1.

Table 3.1: *The inputs and outputs of networks.*

<table>
<tr><th></th><th>Input</th><th>Ground truth</th><th>Output</th></tr>
<tr><td>DSENet</td><td rowspan="2">corrupted projection data</td><td>corrupted minus ground truth projection data</td><td>the error of the projection data</td></tr>
<tr><td>SinoNet</td><td>ground truth projection data</td><td>corrected projection data</td></tr>
<tr><td>TomoNet1</td><td rowspan="3">corrupted projection data
and reconstructed image</td><td rowspan="3">ground truth projection data and phantom</td><td rowspan="4">corrected reconstructed image</td></tr>
<tr><td>TomoNet2</td></tr>
<tr><td>TomoNet3</td></tr>
<tr><td>ReconNet</td><td>corrupted reconstructed image</td><td>ground truth phantom</td></tr>
</table>

SinoNet and ***ReconNet***: The structure of the *SinoNet* and the *ReconNet* are the simplest ones. They have a U-shape consisting of the contracting path (left side) and the expansive path (right side) and they also have an additional average pooling connection which is concatenated to the contracting path. The difference between *SinoNet* and *ReconNet* are the inputs and outputs. The *SinoNet* is working on only the projection data. The reconstruction step can be carried out as an additional step after processing the projection data. On the other hand, the usage of *ReconNet* takes place after the reconstruction as a post-processing step. Results using only projection data or reconstructed images in a U-net-like structure can be found in [11, 15, 36, 43, 50] and [16, 20, 40, 51] respectively.

DSENet: In terms of structure *DSENet* and *SinoNet* are the same. The main difference between the networks are the input, output and ground truth data (see Table 3.1). The inputs of *DSENet* are corrupted projections and its goal is to produce corrupted projections minus ground truth projections on the output. In other words, the *DSENet* output is the distortion itself; therefore, it has to be subtracted from the distorted projection data to get the corrected projection data. This network was inspired by the DSE in [40].

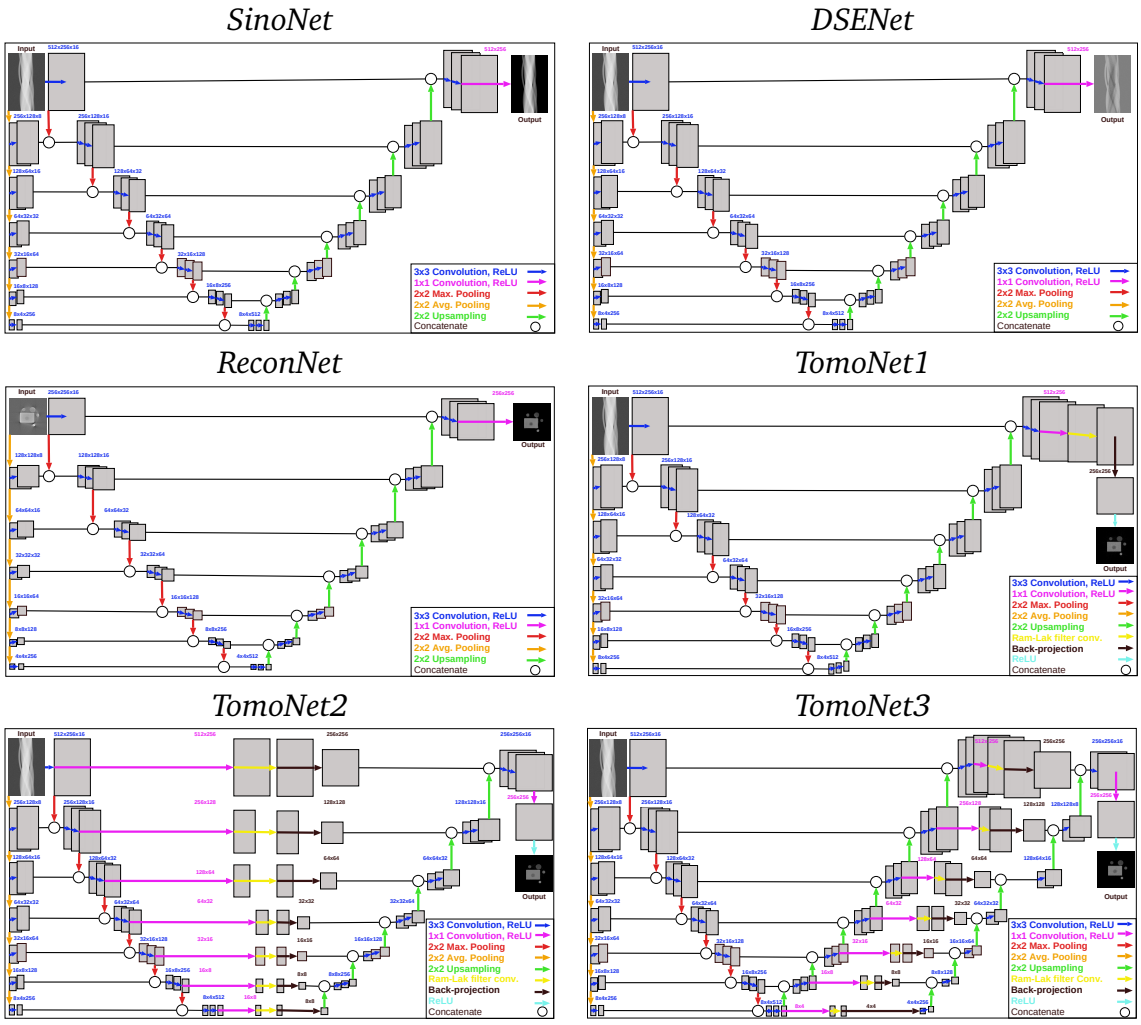

Figure 3.6: *The structures of neural networks.*

TomoNet1: The *TomoNet1* starts with the *SinoNet* structure, but we added new elements corresponding to the *FBP*. The new elements are similar to the construct in [19, 65]. The first element is a convolution with a Ram-Lak filter in projection space. The parameters of this element are non-trainable parameters. The backprojection is the differentiable second element, which is part of neural network as a non-trainable layer. The third element is a ReLU (rectified linear unit) activation layer. The network prefers positive values with the ReLU, which is optimal for the reconstruction tasks. The *TomoNet1* accepts projection data as input and provides reconstructed images as output. The main novelty of this network against the *SinoNet*, *ReconNet* and *DSENet* was that the reconstruction and the network did not separate the processing steps. The network contains the *FBP* as a differentiable layer, so the training process includes the effects of the *FBP*.

TomoNet2: In the case of *TomoNet2*, we mixed the *SinoNet*, *ReconNet* and *Tomo-*

Net1 in a well-organized structure. The three elements correspond to the *FBP* located between the contracting and expansive paths in every level of the U-shape. Accordingly, the projection data of the contracting path transformed into reconstructed images at the expansive path. The *TomoNet2* uses projection data as input and provides reconstructed images as output.

TomoNet3: The *TomoNet3* uses projection data as input and provides reconstructed images as output too, but here the contracting path and the expansive path of the U-shape operate with projection data. Nevertheless, the *FBP* is present at every level, forming a new expansive path.

3.4 The *FBP* at the end of the neural network

In this section we present the results of [46], where our first idea of combining the U-net with the *FBP* (i.e. *TomoNet1*) was tested. In addition, the tests included the following models: *SinoNet*, *ReconNet* and *DSENet*. *Dataset B* was not part of this research.

The measurements were performed on an NVIDIA GeForce RTX 2080 GPU using CUDA 10.1 and cuDNN v7. We used the TensorFlow Keras library running on Python v3.6. We split our database into three parts, namely, train, validation, and test. The proportions were 70%, 20% and 10%, respectively. This means, that we used 3500 phantoms for training, 1000 phantoms for validation and 500 phantoms for testing. The presented results were made with the same random split, but shuffle was allowed among epochs during training and validation.

3.4.1 Training of the neural networks

In this Section we provide detailed information about the training of the deep convolutional neural networks. We trained the networks several times with different hyperparameters. The optimal configuration of hyperparameters resulted in the best image quality shown in Table 3.2 for every networks.

Table 3.2: *Hyperparameters of the networks of Section 3.4*

	Loss function	Optimizer	AMS Grad	Early Stopping	Activation function	Initial learning rate	Epochs	Batch size
SinoNet	Mean Squared Error	Adam	True	True	ReLU	0.0001	40	13
ReconNet							79	7
DSENet							61	43
TomoNet1						0.001	42	7

The loss function diagrams of all networks can be seen in Figure 3.7. In the case of *DSENet* and *SinoNet* the loss and validation loss decreased rapidly during the first few epochs, then it changed slowly. For *ReconNet* it took a while to converge, but the

result ended up as the previous ones. With *TomoNet1* the validation loss function had more fluctuation and some higher spikes, but this is acceptable because the reconstruction step is integrated into this network. We applied early stopping and, in the end, used the network snapshot that resulted in the best validation performance. In this way, we completely excluded the potential effect of overfitting from our results.

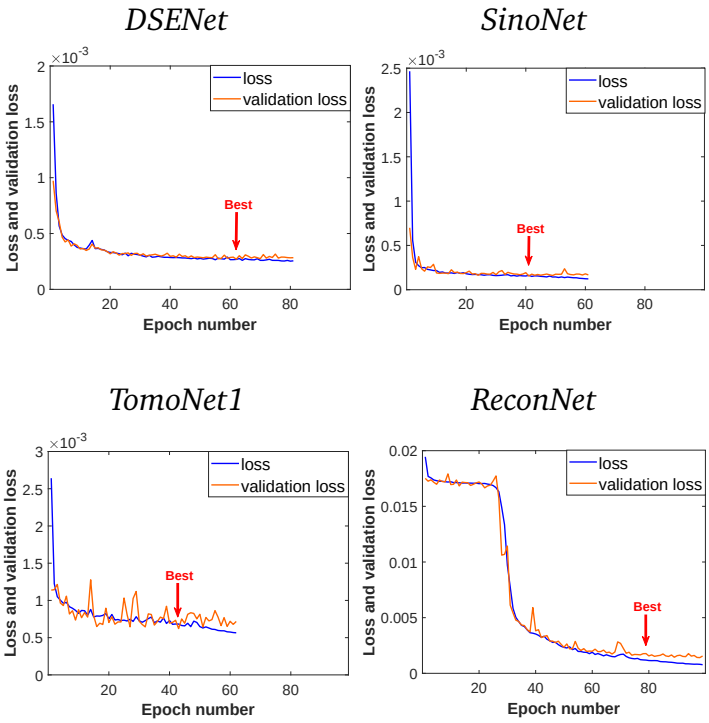

Figure 3.7: *The loss and the validation loss values of the networks in Section 3.4 during the training phase. The red arrows mark the best learning state of the networks that were used for prediction.*

3.4.2 Evaluation of the results

We performed the evaluation of the methods considering the reconstructed images of the test database. The phantoms of this database were not shown to the networks during training or validation. The average error and dispersion belonging to the test phantoms are shown in Table 3.3.

Our general observation is that every method performed well in streak artifact reduction. With *DSENet* the electrical noise stayed high and *ReconNet* often did not keep the shape of the objects. While *SinoNet* and *TomoNet1* reduced the electrical noise and preserved the shape of the objects at the same time. Overall, the *TomoNet1* outperformed the other methods, but all the algorithms yielded a significant improvement in comparison of the uncorrected images. Although the *SSIM* values

Table 3.3: *The results in Section 3.4. The best values are marked with boldface characters.*

Al [mm]	Error type	*No correction*		*DSENet*		*SinoNet*		*TomoNet1*		*ReconNet*	
		AVG	STD	AVG	STD	AVG	STD	AVG	STD	AVG	STD
0	***PSNR***	20.9252	4.0782	23.3071	1.9631	30.9057	4.1767	**32.9277**	4.5268	29.2246	3.6058
	SSIM	0.8442	0.0774	0.8856	0.0298	0.9836	0.0087	**0.9882**	0.0070	0.9876	0.0066
	MSE	0.0116	0.0092	0.0051	0.0021	0.0012	0.0009	**0.0008**	0.0010	0.0017	0.0017
1	***PSNR***	24.1043	3.9576	24.2138	1.9614	32.5226	5.0239	**34.9553**	5.4176	31.8609	4.1983
	SSIM	0.8857	0.0653	0.8989	0.0253	0.9877	0.0070	**0.9916**	0.0056	0.9912	0.0052
	MSE	0.0058	0.0054	0.0042	0.0020	0.0010	0.0012	**0.0007**	0.0011	0.0010	0.0012
5	***PSNR***	22.7526	1.8531	23.6960	0.8985	33.2657	3.8949	**36.2942**	4.2668	32.4376	3.9297
	SSIM	0.8501	0.0397	0.8841	0.0133	0.9878	0.0059	**0.9924**	0.0042	0.9913	0.0042
	MSE	0.0058	0.0027	0.0044	0.0009	0.0007	0.0005	**0.0004**	0.0004	0.0008	0.0009
10	***PSNR***	20.9796	1.6583	22.5474	1.2293	32.6290	4.2584	**35.0155**	4.4830	31.8752	4.3151
	SSIM	0.7988	0.0424	0.8572	0.0226	0.9866	0.0065	**0.9928**	0.0041	0.9910	0.0048
	MSE	0.0086	0.0031	0.0058	0.0016	0.0008	0.0008	**0.0005**	0.0007	0.0010	0.0010
20	***PSNR***	18.8682	1.3609	21.5029	1.1524	31.5836	4.2697	**33.7631**	4.5012	31.2386	4.6241
	SSIM	0.7234	0.0462	0.8304	0.0258	0.9846	0.0066	**0.9908**	0.0042	0.9906	0.0047
	MSE	0.0136	0.0041	0.0073	0.0018	0.0011	0.0010	**0.0007**	0.0007	0.0012	0.0012
AVG	***PSNR***	21.5835	3.4374	23.0994	1.8045	32.1394	4.4240	**34.5345**	4.8039	31.2619	4.2749
	SSIM	0.8233	0.0798	0.8727	0.0344	0.9860	0.0073	**0.9911**	0.0055	0.9903	0.0054
	MSE	0.0091	0.0064	0.0053	0.0021	0.0010	0.0009	**0.0006**	0.0008	0.0012	0.0013

of the *ReconNet* are really close to the results of the *TomoNet1*, this phenomena is probably caused by the strong smoothing effect of *ReconNet*.

Note that the correction task became harder with using 10- or 20-mm aluminum prefiltering. In this case objects suffered much less beam hardening, but the electrical noise increased owing to the lack of low energy photons. Consequently, the methods must correct a barely visible effect of beam hardening with the presence of high electrical noise with 20 mm aluminum prefiltering. We can say that the methods could still improve the image quality despite the huge amount of random noise.

In the case of *SinoNet* and *DSENet* one can observe a drop of *MSE* values when moving from the loss values of Figure 3.7 to the evaluation in Table 3.3. This is mainly because the values in Figure 3.7 are produced by the *MSE* loss function evaluating the differences in the projection data during training. The comparison was carried out on the reconstructed images. Looking at the *ReconNet* and *TomoNet1* networks this difference cannot be observed because their outputs are the reconstructed images. Figure 3.7 shows that the *SinoNet* gained lower error values, but in the end, it preforms worse than *TomoNet1*. In our opinion this is because, using the *FBP* block in the *TomoNet1* made it possible to get less error on the final image, while the aim of *SinoNet* is to optimize the projection data regardless of the reconstructed result.

Figure 3.8 displays the resulting images of the methods and the error maps of a phantom. The projections of the phantom were generated using 1 mm aluminum prefiltering. On the first phantom, there are a lot of streak artifacts without correction, which are highlighted by the error maps. The error maps highlight the shape-distorting effect of the *ReconNet*. Moreover, they confirm our observation on the

inability of *DSENet* to decrease the electrical noise. The performance of the *SinoNet* and *TomoNet1* are similar looking at the error maps, but differences are visible in favor of the *TomoNet1*. Figure 3.9. shows the intensity value profiles along the lines in Figure 3.8.

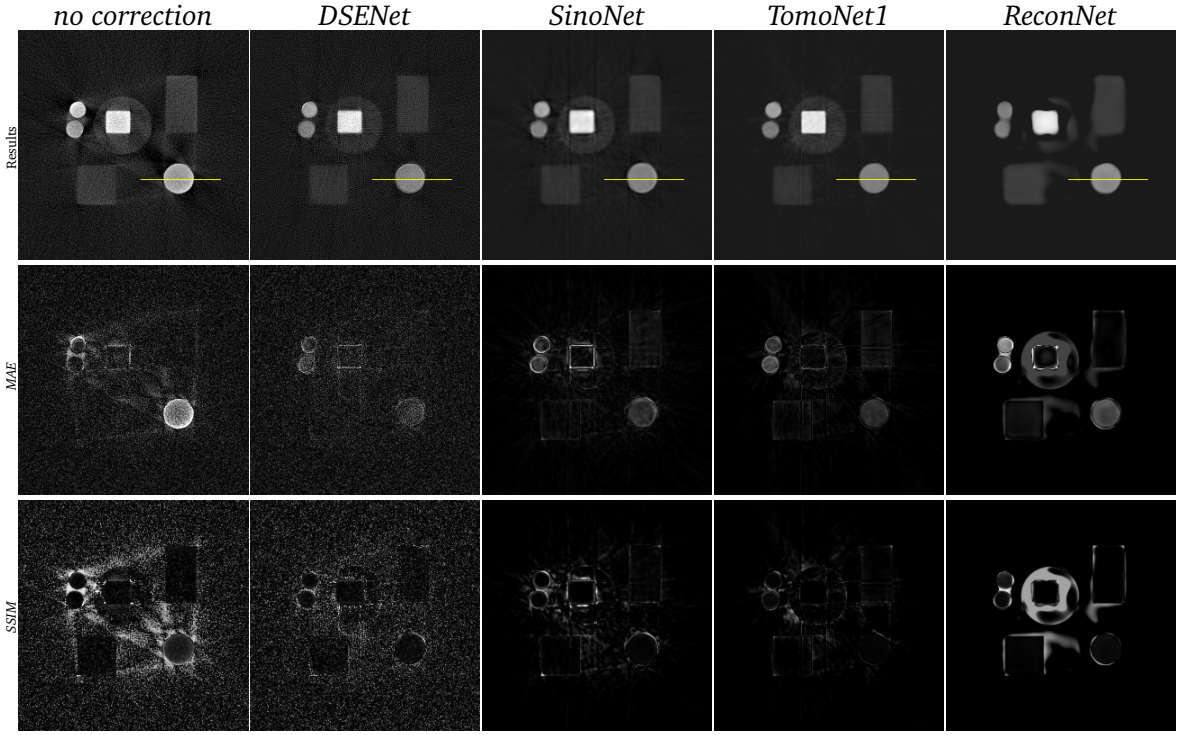

Figure 3.8: *An example of the resulted reconstructions of the neural networks with MAE and SSIM error maps. SSIM maps are inverted for better visibility; therefore, a darker pixel means better result at the given location. Display windows are [-0.1, 1.18], [0, 0.5] and [0, 1] for the results, MAE maps and SSIM maps, respectively.*

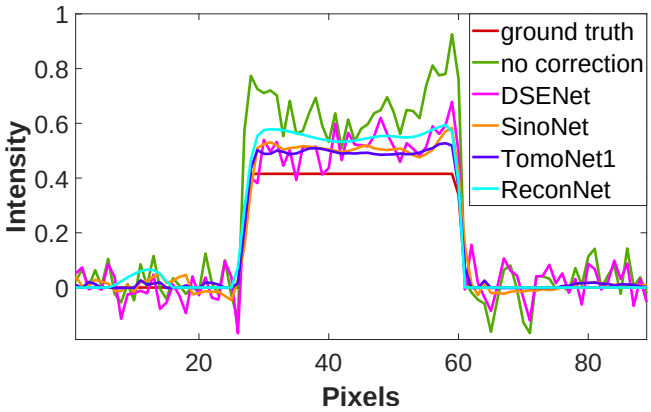

Figure 3.9: *Intensity value profiles along the yellow lines shown in Figure 3.8.*

Because of the high STD values in Table 3.3 we made further investigations. Figure 3.10 shows a summarized rankings of the methods that include all the test phantoms. We summed up, how many 1st, 2nd, 3rd, 4th, and 5th best results were achieved by the methods. Note that the same rankings were experienced taking the *PSNR* and *MSE* values.

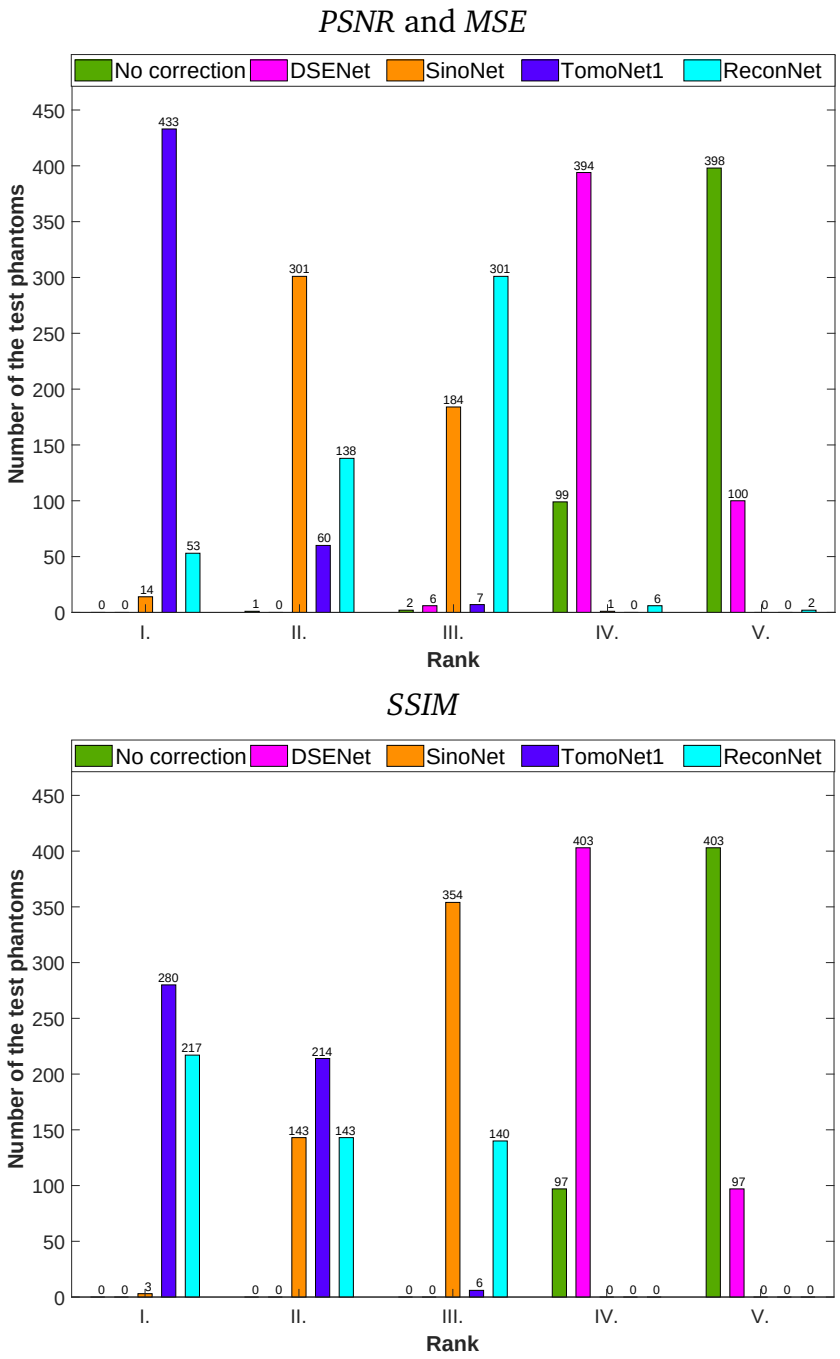

Figure 3.10: *The achieved ranks by the methods by all the measured errors.*

Figure 3.10 with the *PSNR* and *MSE* statistics shows the clear dominance of the *TomoNet1*. From the 500 test phantoms *TomoNet1* performed the best in 433 cases and never was in the 4th or 5th place. This means that the *TomoNet1* outperformed the other methods certainly in 86.6 percent of the cases.

According to the *SSIM* the *ReconNet* was in 217 cases the best, but there were also in 143 and 140 cases where it was only the 2nd and 3rd best. Therefore, we conclude low trustworthiness for *ReconNet* and once again, the efficient electrical noise removal effect of the *ReconNet* may have had a great impact on the *SSIM*. On the other hand, *TomoNet1* yielded the best result in 280 and the 2nd in 214 cases, which is the 98.8 percent of the cases. So, the *TomoNet1* seems to be slightly better than the other methods according to the *SSIM*, and much better when considering *PSNR* or *MSE*.

Table 3.4 shows the total score of the methods for all errors. The total score was calculated by the formula

$$TotalScore = \sum_{rank=1}^{5} N_i \, rank \, , \tag{3.1}$$

where $rank \in [1, 2, 3, 4, 5]$ is the number corresponds to the rank and N_i is the number of the test cases at the given method and rank. Therefore, a lower Total Score means better results. According the summarized rankings, we got the same results as before. The *TomoNet1* proved to be the best choice, the *DSENet* was giving slightly less accurate results, while the *SinoNet*, and *ReconNet* fell behind. In general, all the methods gave better results than keeping the images untouched and three of the methods (*TomoNet1*, *DSENet* and *SinoNet*) always improved the results.

Table 3.4: *The achieved $TotalScore$ by the methods by all the measured errors. The smaller is better.*

Error type	*No correction*	*DSENet*	*SinoNet*	*TomoNet1*	*ReconNet*
PSNR and *MSE*	2393	2094	1172	**574**	1266
SSIM	2403	2097	1351	**708**	923

3.5 Using *FBP* as skip connections in the modified U-net

In this section we present the results of [47], where two novel neural networks (*TomoNet2* and *TomoNet3*) were introduced, and we compared them to the previously discussed neural network structures, i.e. *SinoNet*, *ReconNet* and *TomoNet1*.

3.5.1 Training of the neural networks

In this research, we included *Dataset B* in the testing phase, so the phantoms of *Dataset B* remained unseen during training and validation. Otherwise, we used the same framework as in Section 3.4. The training was performed on an NVIDIA GeForce RTX 2080 GPU using CUDA 10.1 and cuDNN v7. We used the TensorFlow Keras library running on Python v3.6. We used only Dataset A for training and validation. We have split the 5 000 images of Dataset A into three partitions, namely, train, validation and test. The proportions were 70%, 20%, and 10% respectively. The presented results were made with the same split, but shuffle was allowed among epochs during training and validation.

We trained the networks several times with different hyper parameters and we have chosen the final settings leading to the best reconstruction quality. These settings are summarized in Table 3.5. We applied early stopping, which stopped the training if the validation loss stopped decreasing for twenty epochs, and resets the network snapshot that resulted in less validation performance. In this way, we completely excluded the compromising effect of overfitting from our results.

Table 3.5: *Hyperparameters of the networks of Section 3.5*

	SinoNet	*ReconNet*	*TomoNet1*	*TomoNet2*	*TomoNet3*
Loss function	Mean Squared Error				
Optimizer	Adam				
AMSGrad	True				
Early Stopping	True				
Activation function	ReLU				
Initial learning rate	0.0001			0.001	0.0001
Epochs	48	82	43	105	107
second/Epochs	43	42	60	61	70
Batch size	43	7	43		

The loss and validation function diagrams of all networks can be seen in Figure 3.11. We marked (with a red arrow) the best state, which was saved by early stopping. The loss and validation loss of the models decreased rapidly during the first few epochs, then they moved slowly toward the zero line. Although, the curves of loss and validation loss functions moved away from each other before the stopping point in the case of *ReconNet*, we considered the curves to be satisfactory in all respects.

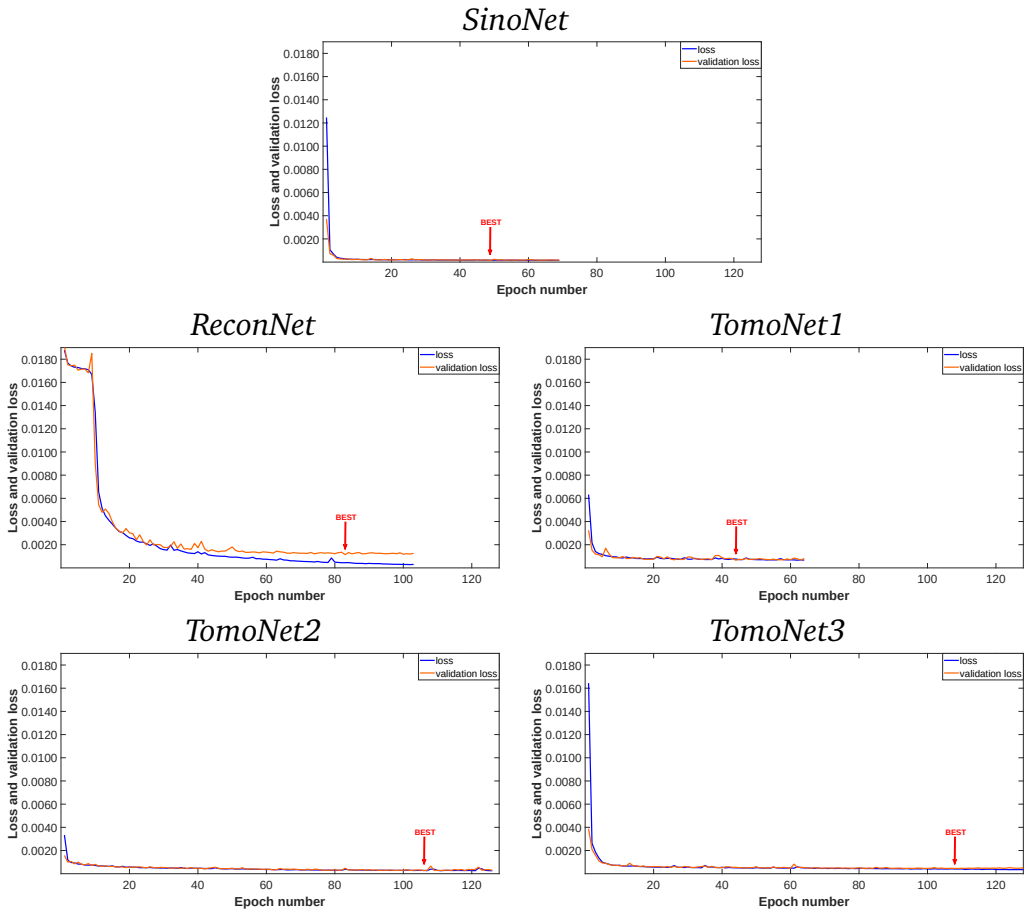

Figure 3.11: *The normalized loss and the validation loss values of the networks in Section 3.5 during the training phase. The red arrows mark the best learning state of the networks that were used for prediction.*

3.5.2 Evaluation of the results

We compared the networks with each other and also included a *FBP* algorithm using the Hann filter. We chose the Hann filter based on a test on *Dataset A* examining the Ram-Lak, Shepp-Logen, Cosine, Hamming, and Hann filters. Table 3.6 shows the results of the test. Clearly, the performance of the Hann filter seems to be the best; therefore, we included only the columns of the Hann filter in the upcoming tables.

Tables 3.7 and 3.8 show the average and standard deviation values of the methods with *Dataset A* and *B* by category. In the case of *Dataset A*, the categories correspond to the sources used. In Table 3.8 the category *"I"* denotes the phantoms with two intensities (background is air and object is rib bone). The phantoms belonging to row *"II"* contain four to six materials, which were seen by the networks during training. In the phantoms of category *"III"* the number of different materials can vary between four and six, from which two have not been seen by the networks during training. In

Table 3.6: *AVG and STD according to Dataset A by categories with three types of errors. The first column corresponds to the five sources having prehardening aluminum filters of different thicknesses (i.e. the categories). The last row of the first column shows the overall AVG. The columns of Ram-Lak, Shepp-Logen, Cosine, Hamming and Hann corresponds of the five most common filter used in FBP.*

Al [mm]	Error type	*RamLak*		*SheppLogen*		*Cosine*		*Hamming*		*Hann*	
		Average	STD	Average	STD	Average	STD	Average	STD	Average	STD
0	***PSNR***	20.9252	4.0782	22.3924	4.4433	22.2455	4.4017	23.0518	4.6553	**23.1297**	4.6824
	SSIM	0.8441	0.0774	0.8874	0.0621	0.8836	0.0634	0.9045	0.0563	**0.9066**	0.0556
	MSE	0.0116	0.0092	0.0087	0.0072	0.0089	0.0074	0.0077	0.0066	**0.0076**	0.0065
1	***PSNR***	24.1043	3.9576	26.9589	4.8238	26.6664	4.7169	28.5640	5.6155	**28.7827**	5.7529
	SSIM	0.8856	0.0653	0.9306	0.0482	0.9267	0.0497	0.9483	0.0410	**0.9504**	0.0401
	MSE	0.0058	0.0054	0.0036	0.0038	0.0037	0.0039	0.0029	0.0033	**0.0028**	0.0032
5	***PSNR***	22.7526	1.8531	26.8156	2.3309	26.3808	2.2566	29.5870	3.0570	**30.0248**	3.2310
	SSIM	0.8500	0.0397	0.9230	0.0281	0.9165	0.0292	0.9535	0.0230	**0.9573**	0.0224
	MSE	0.0058	0.0027	0.0024	0.0015	0.0026	0.0015	0.0014	0.0011	**0.0013**	0.0011
10	***PSNR***	20.9796	1.6583	25.2375	1.9055	24.7761	1.8634	28.2783	2.4215	**28.7768**	2.5669
	SSIM	0.7988	0.0424	0.8974	0.0266	0.8882	0.0282	0.9421	0.0186	**0.9478**	0.0176
	MSE	0.0086	0.0031	0.0033	0.0015	0.0036	0.0016	0.0017	0.0010	**0.0016**	0.0010
20	***PSNR***	18.8682	1.3609	23.4168	1.4644	22.9103	1.4480	26.8872	1.7619	**27.4971**	1.8609
	SSIM	0.7233	0.0462	0.8545	0.0292	0.8417	0.0313	0.9194	0.0176	**0.9282**	0.0158
	MSE	0.0136	0.0041	0.0048	0.0017	0.0054	0.0018	0.0022	0.0011	**0.0020**	0.0010
Average	***PSNR***	21.5835	3.4374	24.9265	3.8725	24.5691	3.8010	27.1495	4.5412	**27.5021**	4.6894
	SSIM	0.8232	0.0799	0.8991	0.0503	0.8921	0.0529	0.9328	0.0406	**0.9372**	0.0399
	MSE	0.0091	0.0064	0.0047	0.0046	0.0050	0.0047	0.0034	0.0043	**0.0032**	0.0042

addition, we calculated the overall average in both tables and provided the standard deviation of each category.

Compared to the basic *FBP*, all of the networks improved the quality of the reconstructed images according to the average measurements in Table 3.7. Moreover, Tables 3.7 and 3.8 show that *TomoNet2* outperformed the others, but it was closely followed by *TomoNet3*.

For a better insight, we summed up, how many 1^{st}, 2^{nd}, 3^{rd}, 4^{th}, 5^{th} and 6^{th} best places were achieved by the networks. In this way, the 1^{st} place means that the given network gave the best result. Figures 3.12 and 3.13 show detailed statistics about this ranking.

In the case of *Dataset A*, the statistics are depicted in Figure 3.12. We observed clear dominance in favor of the *TomoNet2*. *TomoNet2* gave the best in 396 cases according to *PSNR* and *MSE* and in 374 based on *SSIM* out of the 500 test phantoms preserved for testing. This means, that the *TomoNet2* consequently outperformed the other methods in 74.8 percent of the cases, but we argue that it is closer to 79.2 based on *PSNR* and *MSE*. This ranking corresponds to the results seen in Table 3.7, which shows the dominance of one of our proposed methods in each category.

In the case of *Dataset B*, we can say, that *TomoNet2* had the best performance with all error measurements looking at Figure 3.13. Note that *ReconNet* only made the reconstruction worse looking at the *PSNR* and *MSE* diagram.

Table 3.7: *AVG and STD according to Dataset A by categories. The first column corresponds to the five sources having prehardening aluminum filters of different thicknesses (i.e., categories). The last row of the first column shows the overall average.*

Al	Error	*FBP*		*SinoNet*		*ReconNet*		*TomoNet1*		*TomoNet2*		*TomoNet3*	
[mm]	type	Average	STD	Average	STD	Average	STD	Average	STD	Average	STD	Average	STD
0	***PSNR***	23.1297	4.6824	30.6019	4.1088	30.6378	4.0829	32.7588	4.2920	**35.9307**	4.4062	34.7219	4.7548
	SSIM	0.9066	0.0556	0.9838	0.0091	0.9913	0.0052	0.9884	0.0063	**0.9970**	0.0019	**0.9969**	0.0017
	MSE	0.0076	0.0065	0.0012	0.0009	0.0013	0.0014	0.0008	0.0009	**0.0004**	0.0004	0.0006	0.0007
1	***PSNR***	28.7827	5.7529	32.4271	4.9667	33.2776	4.6385	34.3974	5.2409	**37.9697**	5.1554	36.9896	5.7430
	SSIM	0.9504	0.0401	0.9887	0.0064	0.9936	0.0041	0.9912	0.0048	**0.9976**	0.0018	**0.9974**	0.0018
	MSE	0.0028	0.0032	0.0010	0.0011	0.0008	0.0011	0.0007	0.0010	**0.0003**	0.0005	0.0005	0.0008
5	***PSNR***	30.0248	3.2310	33.3140	3.8925	34.6786	3.9260	35.2207	3.9602	**40.9700**	3.8661	38.7901	4.4705
	SSIM	0.9573	0.0224	0.9887	0.0052	0.9946	0.0028	0.9903	0.0038	**0.9986**	0.0008	0.9980	0.0012
	MSE	0.0013	0.0011	0.0007	0.0005	0.0005	0.0004	0.0005	0.0005	**0.0001**	0.0001	0.0002	0.0004
10	***PSNR***	28.7768	2.5669	32.5171	4.1650	33.9986	4.8718	34.5703	4.3657	**38.8459**	4.8006	36.7349	5.4773
	SSIM	0.9478	0.0176	0.9869	0.0059	0.9946	0.0032	0.9911	0.0042	**0.9981**	0.0012	0.9972	0.0017
	MSE	0.0016	0.0010	0.0008	0.0008	0.0008	0.0016	0.0006	0.0007	**0.0003**	0.0004	0.0005	0.0008
20	***PSNR***	27.4971	1.8609	31.3519	3.9710	32.9442	5.0721	33.0634	4.3449	**37.7962**	4.1573	35.4217	4.9142
	SSIM	0.9282	0.0158	0.9844	0.0056	0.9940	0.0032	0.9878	0.0048	**0.9972**	0.0015	0.9964	0.0019
	MSE	0.0020	0.0010	0.0011	0.0009	0.0010	0.0016	0.0008	0.0008	**0.0003**	0.0003	0.0005	0.0007
Average	***PSNR***	27.5021	4.6894	31.9951	4.3509	33.0133	4.7227	33.9611	4.5637	**38.1958**	4.7906	36.4728	5.2715
	SSIM	0.9372	0.0399	0.9865	0.0070	0.9935	0.0041	0.9897	0.0051	**0.9977**	0.0016	0.9972	0.0018
	MSE	0.0032	0.0042	0.0010	0.0009	0.0009	0.0013	0.0007	0.0008	**0.0003**	0.0004	0.0005	0.0007

Table 3.8: *AVG and STD according to Dataset B by categories. The category I contains phantoms with two intensities. The phantoms belonging to the category of II can be constructed only of known materials showed to the networks during training. The phantoms of III contain at least one material unseen from the networks during training. The last row of the first column shows the overall average.*

Category	Error	*FBP*		*SinoNet*		*ReconNet*		*TomoNet1*		*TomoNet2*		*TomoNet3*	
	type	Average	STD	Average	STD	Average	STD	Average	STD	Average	STD	Average	STD
I	***PSNR***	22.9847	0.8083	23.4377	1.5777	7.6204	9.6137	25.2104	2.0990	**26.8979**	1.9009	26.0787	2.2368
	SSIM	0.9373	0.0074	0.9728	0.0070	0.9517	0.0189	0.9756	0.0068	**0.9875**	0.0045	**0.9878**	0.0051
	MSE	0.0051	0.0009	0.0048	0.0014	1.9855	5.5222	0.0033	0.0010	**0.0022**	0.0008	0.0027	0.0011
II	***PSNR***	25.0866	1.7419	25.3617	1.6025	14.6258	6.7897	26.9704	1.6068	27.6045	1.4897	**27.8985**	1.8768
	SSIM	0.9476	0.0088	0.9796	0.0052	0.9584	0.0139	0.9823	0.0049	**0.9901**	0.0020	0.9896	0.0030
	MSE	0.0033	0.0012	0.0031	0.0011	0.1230	0.2245	0.0021	0.0008	**0.0018**	0.0006	**0.0018**	0.0007
III	***PSNR***	28.4547	2.8829	26.2879	3.5975	21.1095	3.8025	27.1091	3.6830	**28.8856**	2.3983	27.1993	3.5178
	SSIM	0.9706	0.0107	0.9789	0.0081	0.9642	0.0089	0.9845	0.0044	**0.9881**	0.0046	0.9850	0.0052
	MSE	0.0017	0.0009	0.0031	0.0021	0.0105	0.0074	0.0027	0.0020	**0.0015**	0.0008	0.0025	0.0019
Average	***PSNR***	25.5087	3.0051	25.0291	2.6918	14.4519	8.9573	26.4300	2.7199	**27.7960**	2.1048	27.0588	2.7061
	SSIM	0.9519	0.0166	0.9771	0.0074	0.9581	0.0151	0.9808	0.0066	**0.9886**	0.0040	0.9875	0.0049
	MSE	0.0034	0.0017	0.0037	0.0018	0.7063	3.2713	0.0027	0.0014	**0.0018**	0.0008	0.0023	0.0014

Table 3.9 shows the total score of the networks for all errors. The total score was calculated using Equation 3.1, where $rank \in \{1, 2, 3, 4, 5, 6\}$. As a reminder, lower $TotalScore$ means better results. According to the summarized rankings, *TomoNet2* provided the best quality of reconstructed images.

Table 3.9: *The gained $TotalScore$ by the methods for all errors. The smaller is better.*

$TotalScore$	Error type	*FBP*	*SinoNet*	*ReconNet*	*TomoNet1*	*TomoNet2*	*TomoNet3*
Dataset A	***PSNR*** and ***MSE***	2917	2316	1940	1641	**633**	1053
	SSIM	2996	2427	1634	1911	**633**	899
Dataset B	***PSNR*** and ***MSE***	251	291	396	196	**111**	141
	SSIM	364	256	343	198	**99**	126

Figures 3.14 and 3.15 present examples from *datasets A* and *B*. The *FBP* reconstruction shows strong signs of beam hardening artifacts and electrical noise with the *FBP* in Figure 3.14. Errors have been highlighted by the *MSE* and *SSIM* error maps. The usage of *ReconNet* led to highly homogeneous objects, but *ReconNet* did not preserve the shapes of the objects. *SinoNet* and *TomoNet1* kept the edges well, but they also left a significant amount of artifacts in the images. *TomoNet2* and *TomoNet3* gave the best looking results according to the images in Figure 3.14 and Figure 3.15. Note that *TomoNet2* resulted in more homogeneous objects, which is preferable according to the ground truth image. We argue that the error measurements are consistent with what is seen in the images.

The reconstruction in Figure 3.15 also shows signs of beam hardening artifacts and electrical noise with the *FBP*. This phantom is from the "two unseen materials" category and it also contains three cracks. Here again, *ReconNet* was struggling to keep the shape of the objects, and as a result, it gave back an amorphous form. *SinoNet* yielded a decent result but left some streak artifacts on the image. The results of *TomoNet1* and *TomoNet3* are blurry, but the cracks are all along recognizable especially with *TomoNet1*. The reconstructed images are sharper according to the results of *TomoNet2*, and the surrounding object with the lower attenuation coefficient is better preserved. However, the cracks have disappeared in some places.

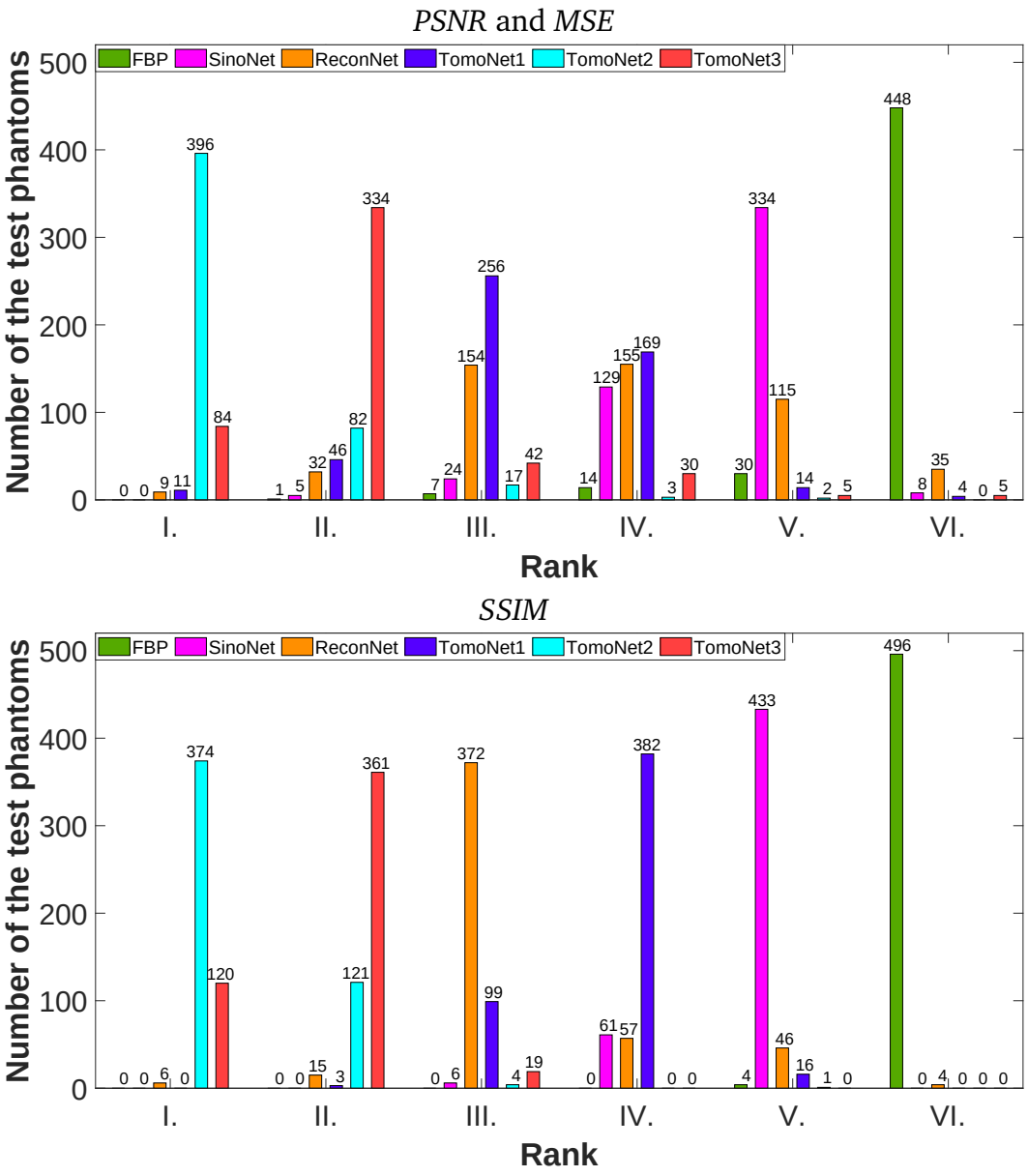

Figure 3.12: *The achieved ranks by the networks on Dataset A.*

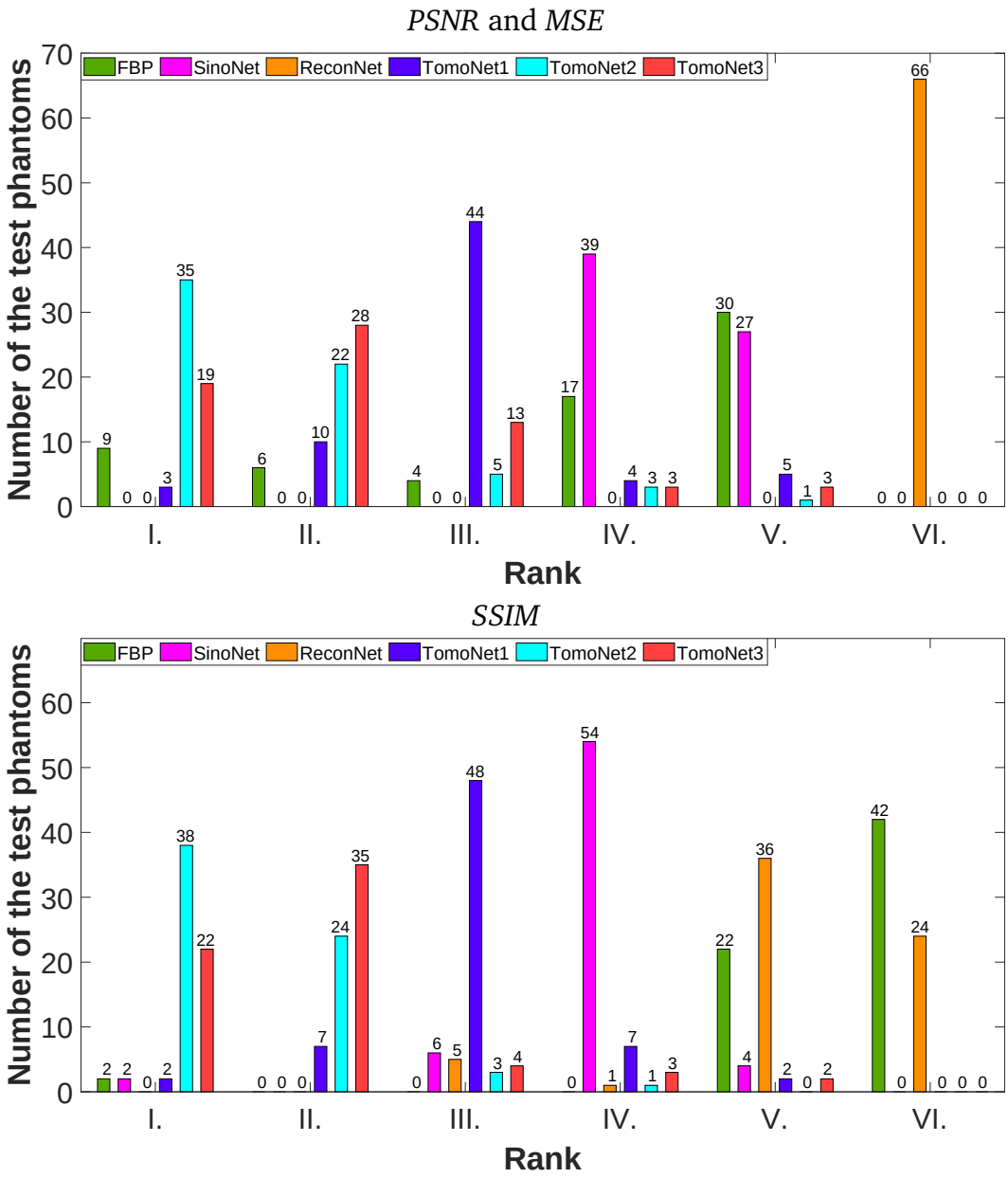

Figure 3.13: *The achieved ranks by the networks on Dataset B.*

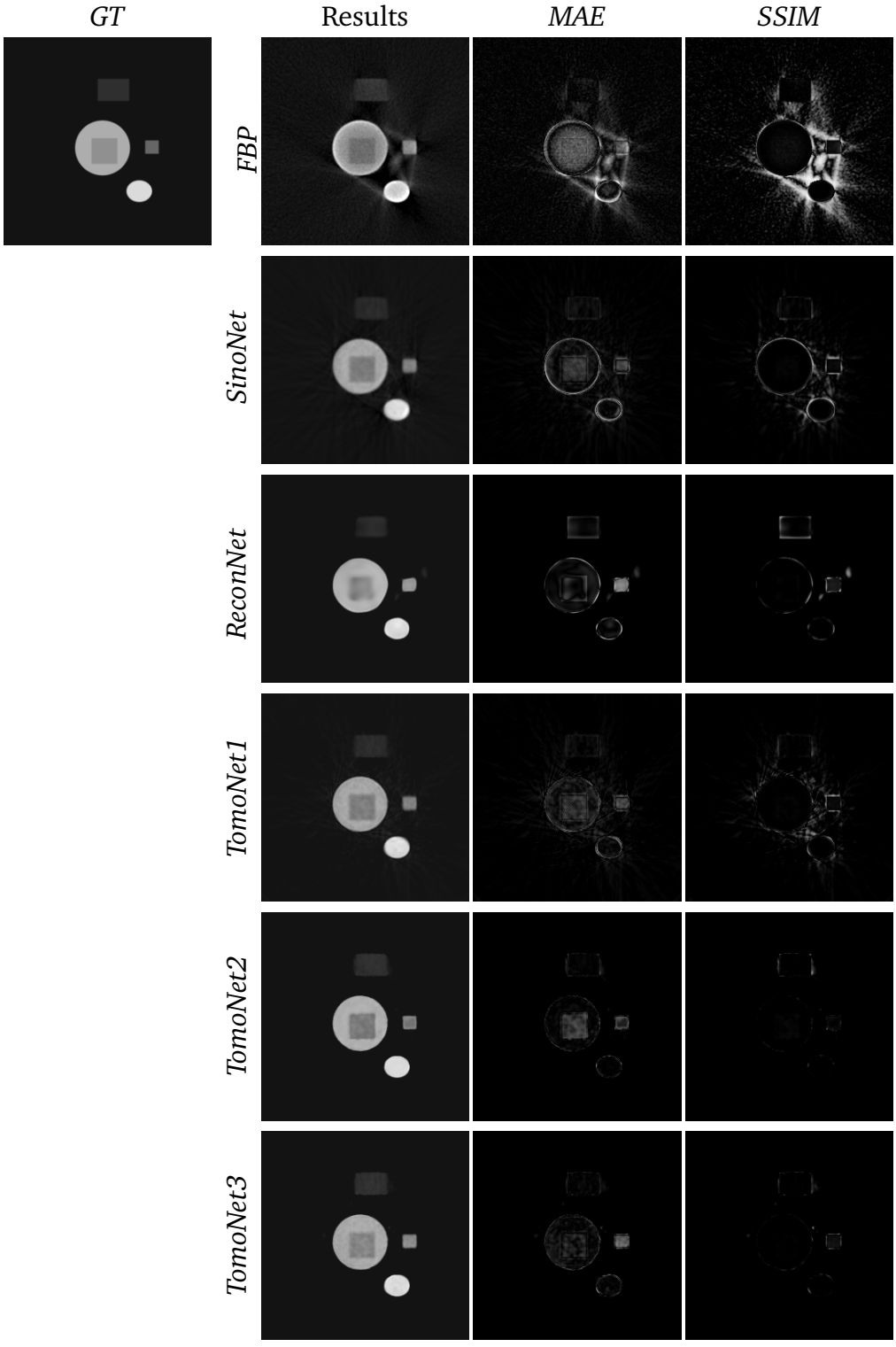

Figure 3.14: *An example from the resulted reconstructions of Dataset A with MAE and SSIM error maps. SSIM maps are inverted for better visibility. Darker pixel means better result in the given location in the case of error maps. Display windows are [-0.1, 1.18], [0, 0.77] and [0, 1] for the Results, MAE maps and SSIM maps.*

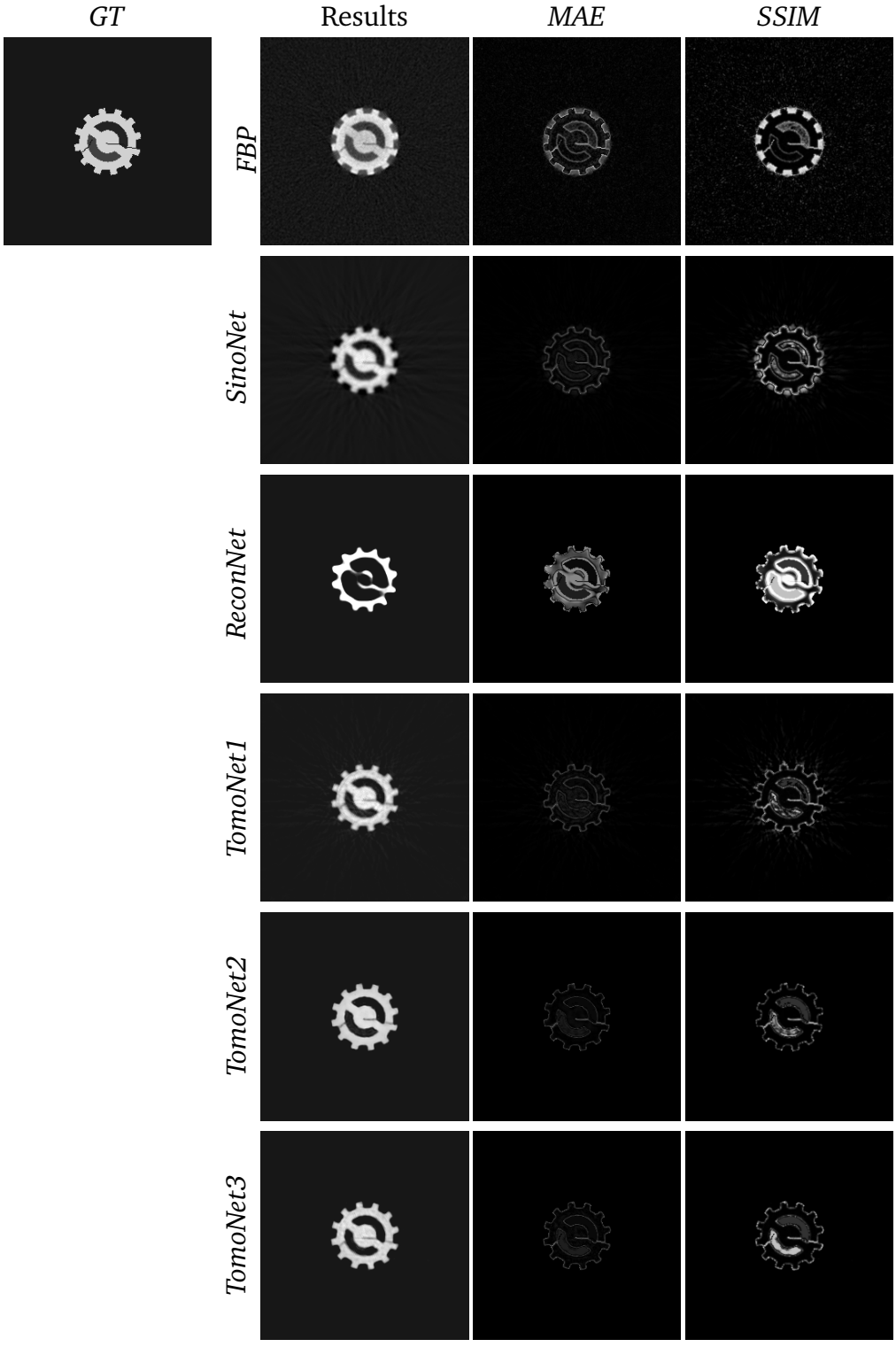

Figure 3.15: *An example from the resulted reconstructions of Dataset B with MAE and SSIM error maps. SSIM maps are inverted for better visibility. Darker pixel means better result in the given location in the case of error maps. Display windows are [-0.1, 0.98], [0, 1.48] and [0, 1] for the Results, MAE maps and SSIM maps.*

3.6 Discussion and concluding remarks

In this chapter, we showed our research that combined CT methods with CNN algorithms.

In Section 3.4, we demonstrated how deep convolutional neural networks are capable of correcting beam hardening artifacts. Our first aim was to find the ideal place of a neural network in the Computer Tomography image processing pipeline. We found that the best results were achieved by the *TomoNet1* which combines a reconstruction with the pre-processing step on the projection data with the U-net. To our knowledge this is the first time, when the *FBP* was the part of a modified U-net in order to improve the outputs distorted by beam hardening. Lastly, our objective was to successfully correct beam hardening in the presence of high electrical noise. Our *TomoNet1* algorithm performed the best in this task.

In Section 3.5, we presented the results with our proposed two novel neural networks for image reconstruction from projections, where the *FBP* used as skip connections (*TomoNet2* and *TomoNet3*). The novelty of the network is that they contain multiple back-projection layers, which provide a strong connection among the two main parts of the network working on projection data and working on reconstructed images, respectively.

In the literature, there are many publications that use deep learning techniques combined with tomography, but these solutions operate mainly as pre- or post-processing steps added before or after the reconstruction. Looking at our results, it is clear that the end-to-end solutions yielded better results.

Here we list our five major observations based on the Section 3.5 to highlight the merits of our proposed methods:

- In our opinion, the reason for the superiority of the new models is, that if the network sees the whole path of the data, then it can better optimize the output and learn more complex features.

- We also argue that the methods improved the image quality regardless of the used physical prefiltering according to Table 3.7, although the highest improvement was detected in the case of the 5 mm aluminum filter.

- The ranking analysis revealed that the networks and the averages are reliable because there is a clearly dominant network in each place. This means that there is a network in every given place, which ends up at the given place more than 50% of the cases. Only *ReconNet* made an exception in the case of *Dataset A* with *PSNR* and *MSE*, because its ranks are spread out over the rankings (i.e., 3-th place in 30.8%, 4-th place in 31%, 5-th place in 23%).

- Testing the networks against “unknown shapes” (i.e. shapes not present when training the networks) in *Dataset B* definitely caused a drop in the performance

of all networks. Especially for the *ReconNet*, in which *PSNR* and *MSE* values became worse as if we had not done anything (i.e. *FBP*). As a reminder, we did not use the phantoms of *Dataset B* during training; therefore, we concluded that the *ReconNet* strongly relies on shape priors and it could not learn the patterns general enough to be able to gain good results on the unseen phantoms of *Dataset B*.

- The introduction of new materials did not cause a problem. All methods handled the phantoms with unseen material well, according to the minor difference between the category of "II" and "III" in Table 3.8.

The author of this PhD thesis is responsible for the following contributions presented in this chapter:

II/1. I created a physically correct database using GATE simulation software. The focus of the created database was the cupping and streak artifacts caused by beam hardening and the coexisting electrical noise.

II/2. I modified the original U-net and applied it at different stages of the image processing pipeline of computer tomography. I proposed a CNN structure (*TomoNet1*) where the reconstruction step was attached to the pre-processing CNN as a non-trainable layer. I showed by experiments that the proposed method yielded superior results.

II/3. I proposed two CNN architectures (*TomoNet2* and *TomoNet3*), where the image reconstruction step is used as a non-trainable neural network layer in the form of skip connections. I tested the algorithms on two datasets and showed that the new architectures outperformed the others, where there was no or only one reconstruction layer inside the network.

Chapter 4

Appendix

4.1 Supplementary figures to Chapter 2

4.1.1 Comparison Study

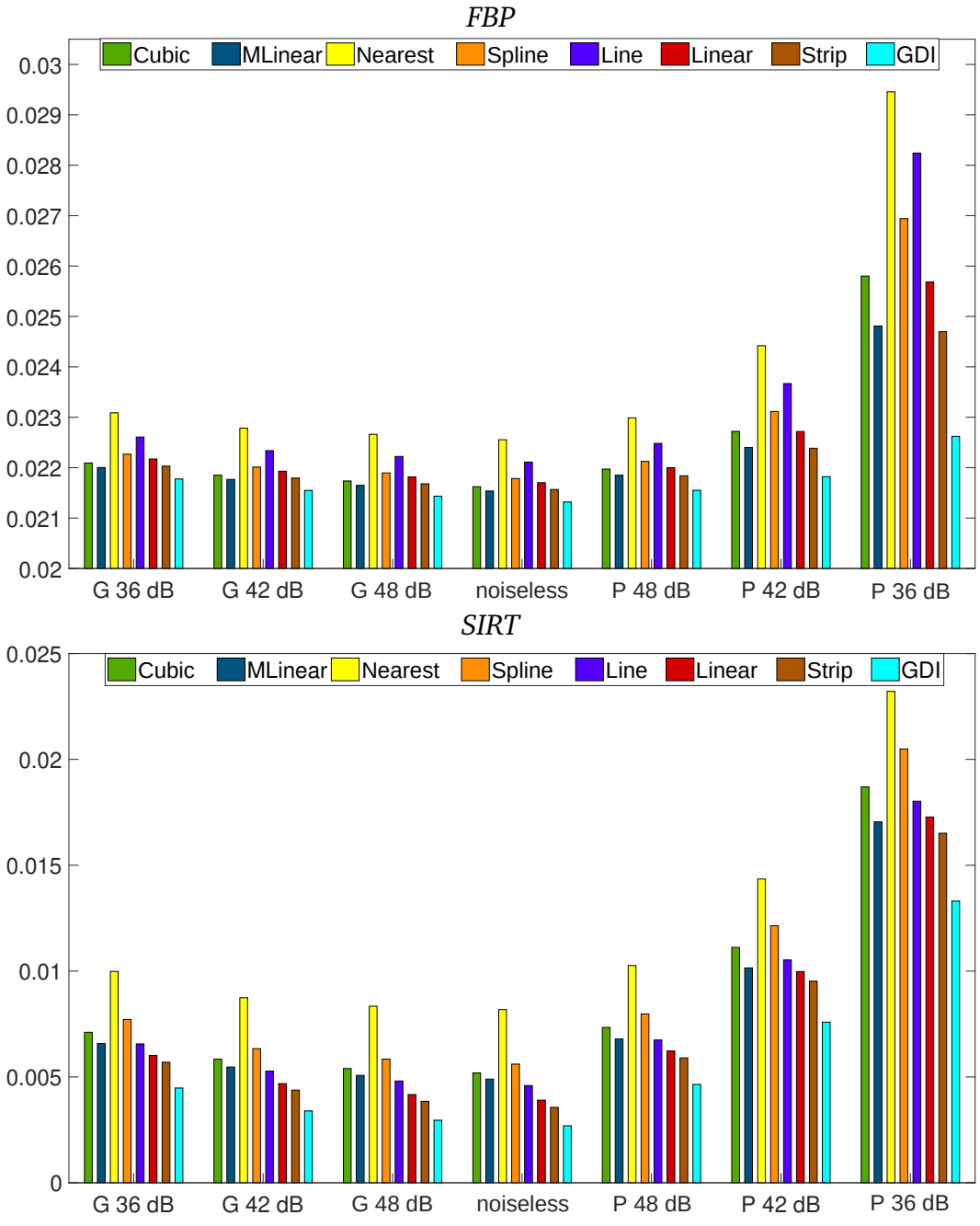

Figure 4.1: *The diagrams show the average MAE for the geometrical phantoms with 285 projections. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown on the left and Poisson noise on the right.*

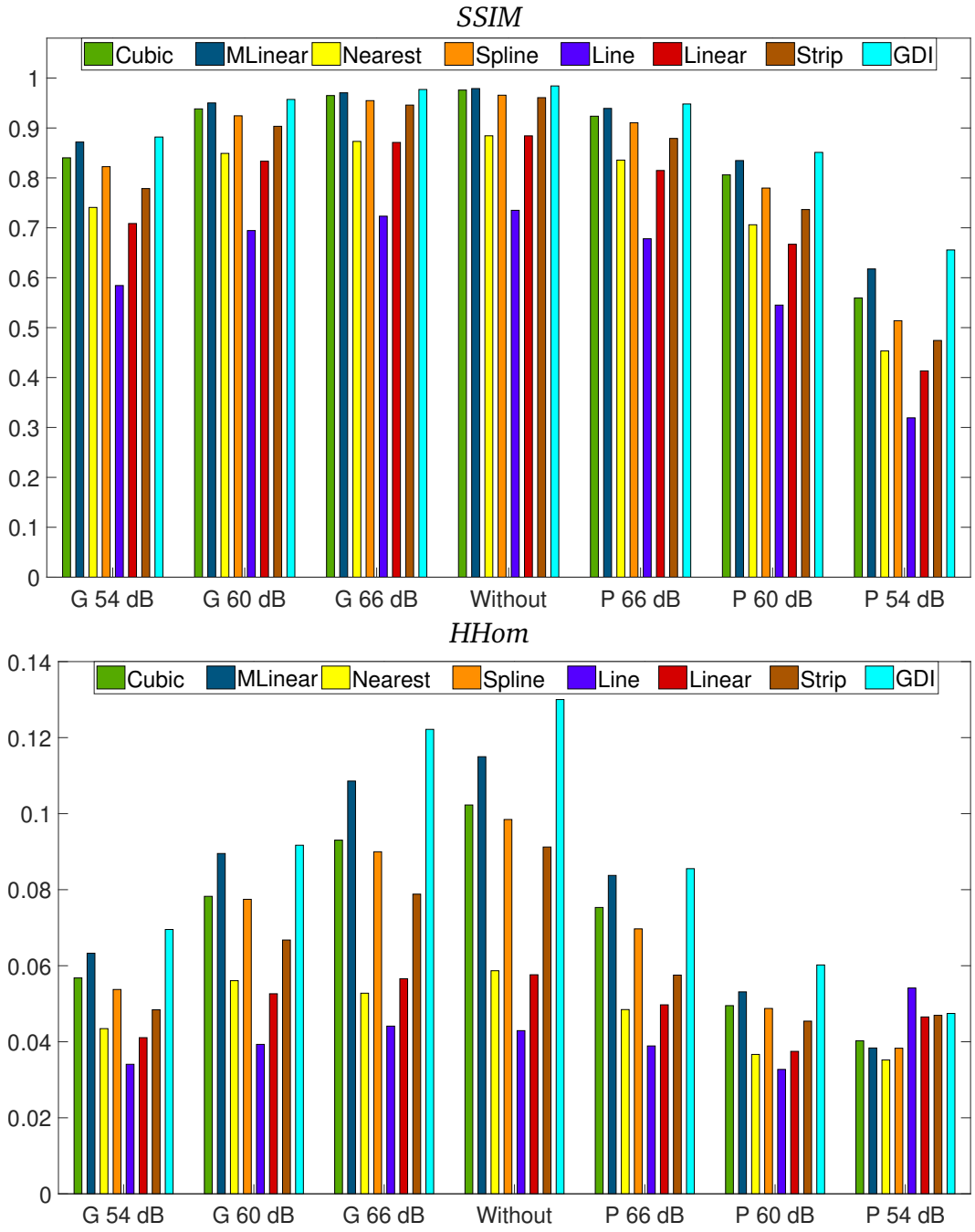

Figure 4.2: *The average values for all ROIs of the Forbild head phantom with SIRT. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown on the left, Poisson noise on the right.*

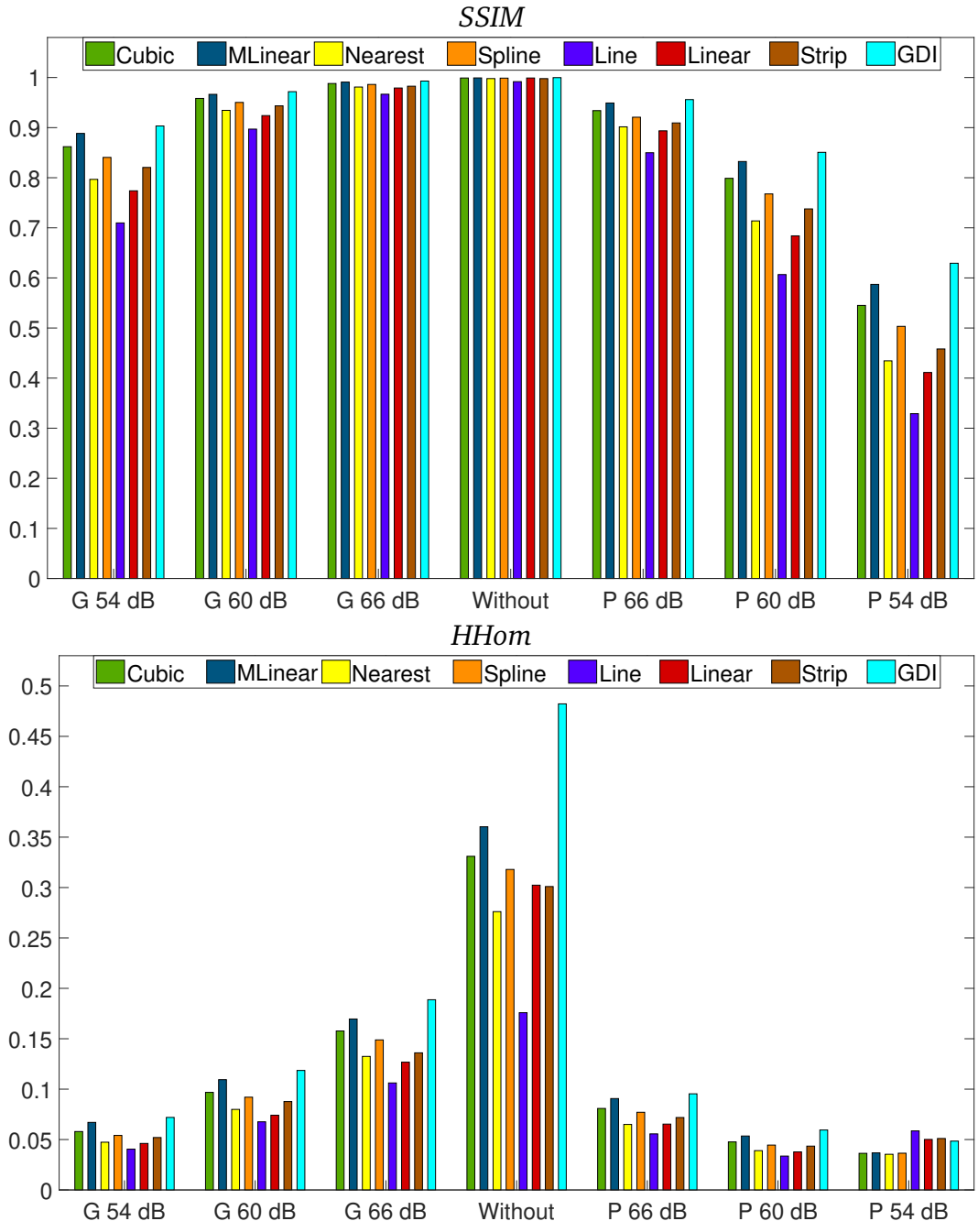

Figure 4.3: *The average values for all ROIs of the Shepp-Logen head phantom with SIRT. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown on the left and the Poisson noise on the right.*

4.1.2 Correction Study

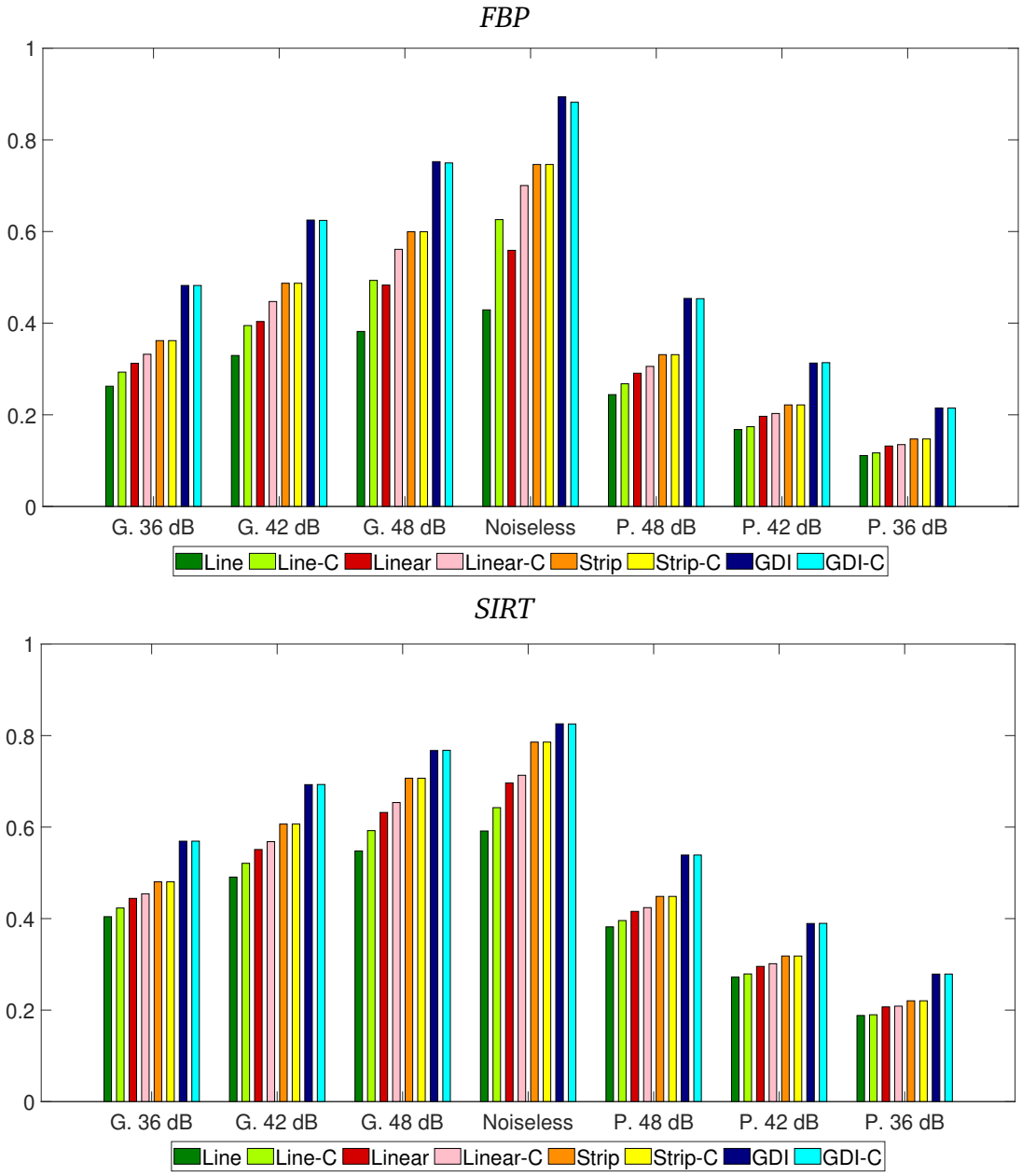

Figure 4.4: *The values are HHom averages of the entire dataset with Basic geometry. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown to the left and the Poisson noise to the right. The four interpolation methods are represented with and without correction in each group. The Corrected versions are marked with the "-C" suffix.*

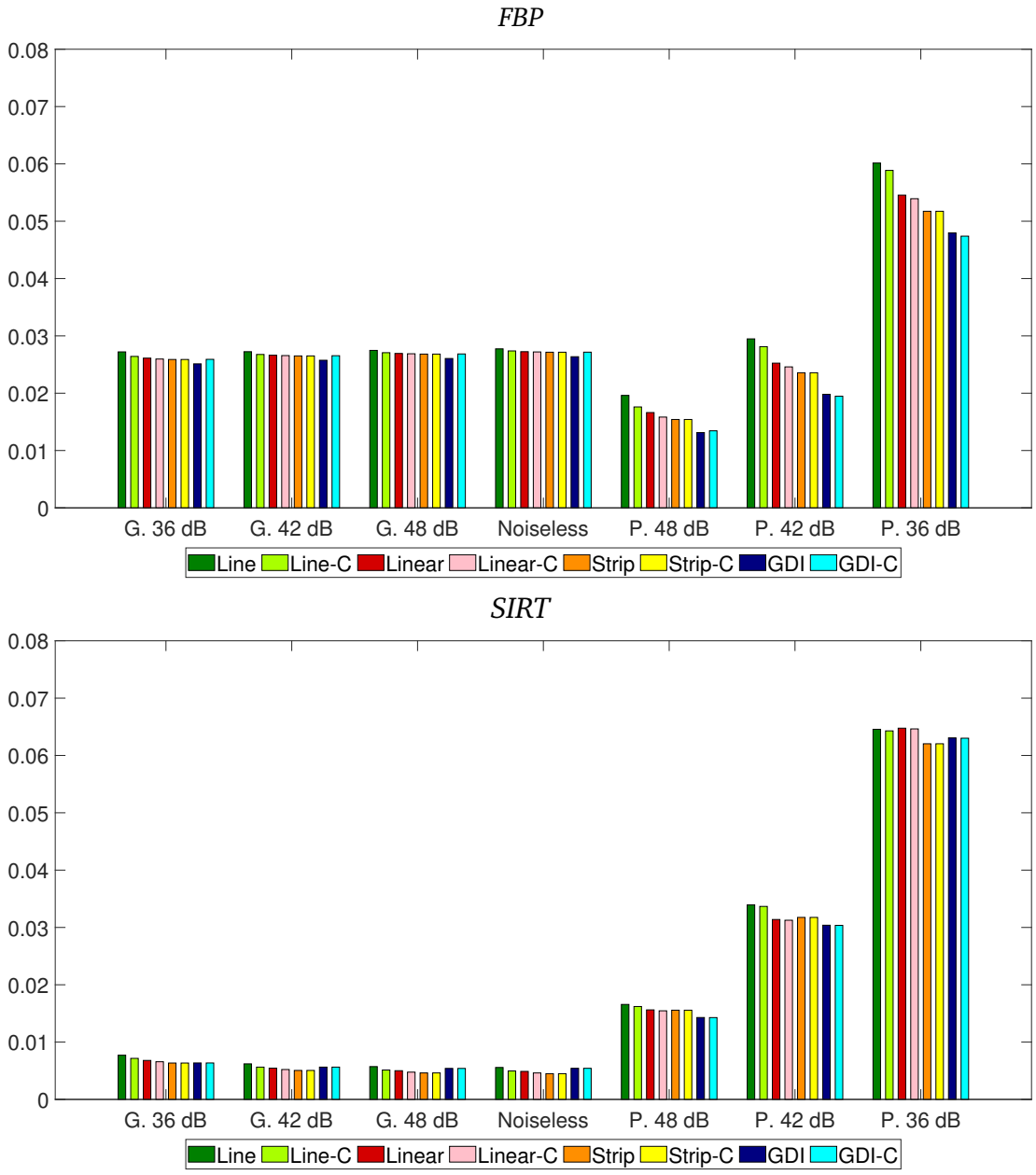

Figure 4.5: *The average MAE of the entire dataset with View++ geometry. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown on the left and the Poisson noise on the right. The four interpolation methods are represented with and without correction in each group. The Corrected versions are marked with the "-C" suffix.*

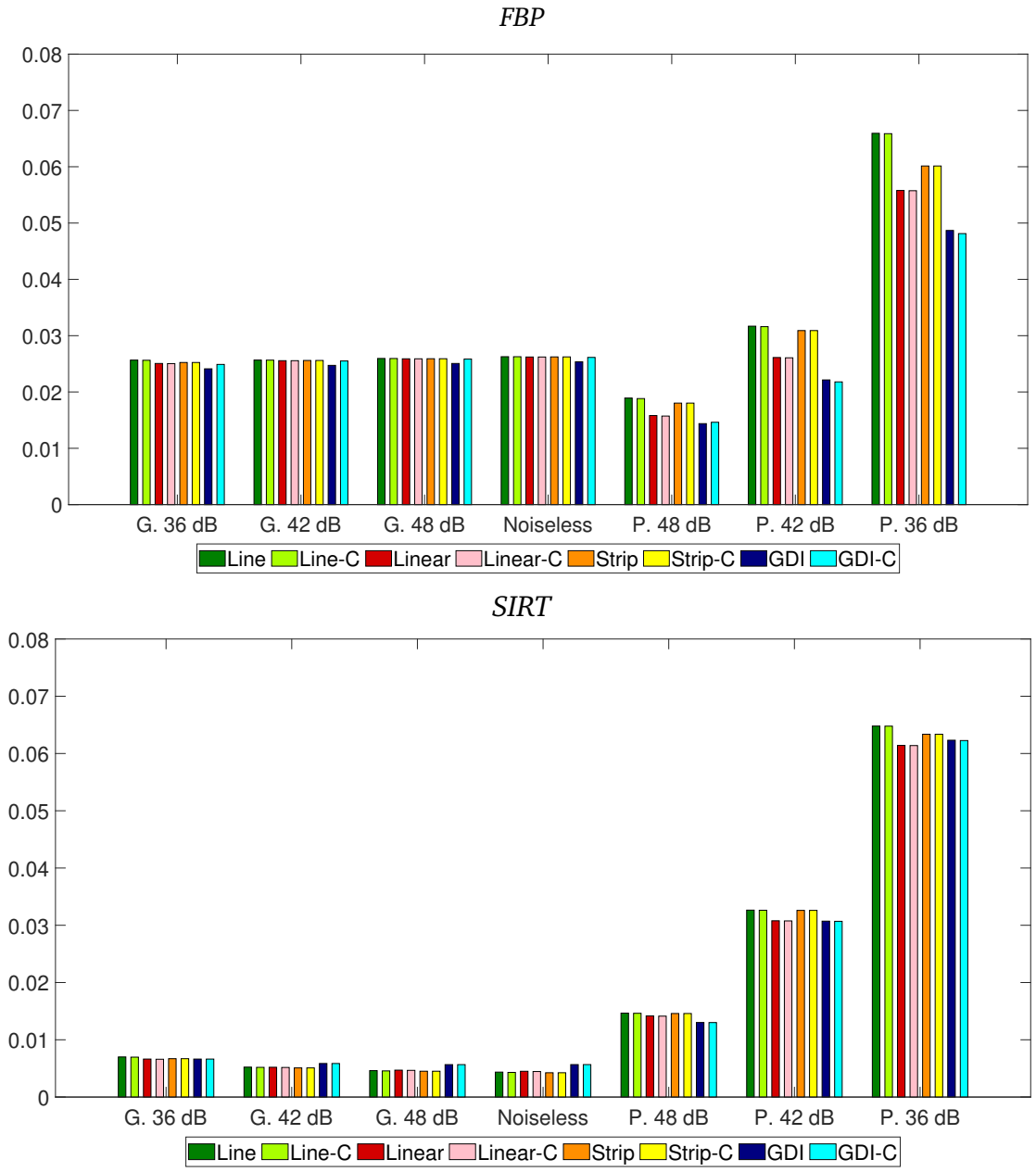

Figure 4.6: *The average MAE of the entire dataset with Ray++ geometry. The center of the diagrams shows the noiseless case. The effect of the Gaussain noise is shown to the left and the Poisson noise to the right. The four interpolation methods are represented with and without correction in each group. The Corrected versions are marked with the "-C" suffix.*

Bibliography

[1] Game-icons.net. https://game-icons.net. Accessed: 2021-03-16.

[2] Matlab r2022a. The MathWorks, Inc., Natick, Massachusetts, United States.

[3] A.H. Andersen and A.C. Kak. Simultaneous algebraic reconstruction technique (sart): A superior implementation of the art algorithm. *Ultrasonic Imaging*, 6(1):81–94, 1984.

[4] Zsolt Balogh, Tamás Huszár, and Benedek Kis. Iterative ring artifact removal method for helical computed tomography scans. *Journal of computer assisted tomography*, 44:796–805, 09 2020.

[5] Folkert Bleichrodt, Tristan van Leeuwen, Willem Jan Palenstijn, Wim van Aarle, Jan Sijbers, and Joost Batenburg. Easy implementation of advanced tomography algorithms using the astra toolbox with spot operators. *Numerical Algorithms*, 71(3):673–697, March 2016.

[6] Ronald N Bracewell. *Two-Dimensional Imaging*, pages 505–537. Prentice Hall, 1nd edition, 1995.

[7] B Chalmond, F Coldefy, and B Lavayssière. Tomographic reconstruction from non-calibrated noisy projections in non-destructive evaluation. *Inverse Problems*, 15(2):399, apr 1999.

[8] B Chalmond, F Coldefy, and B Lavayssière. Tomographic reconstruction from non-calibrated noisy projections in non-destructive evaluation. *Inverse Problems*, 15(2):399, apr 1999.

[9] Rohit Chityala, Kenneth Hoffmann, Sam Rudin, and Daniel Bednarek. Region of interest (roi) computed tomography (ct): Comparison with full field of view (ffov) and truncated ct for a human head phantom. *Proceedings - Society of Photo-Optical Instrumentation Engineers*, 5745:583–590, 04 2005.

[10] Linda Croton, Gary Ruben, Kaye Morgan, David Paganin, and Marcus Kitchen. Ring artifact suppression in X-ray computed tomography using a simple, pixel-wise response correction. *Optics Express*, 27:14231, 05 2019.

[11] Jianbing Dong, Jian Fu, and Zhao He. A deep learning reconstruction framework for X-ray computed tomography with incomplete data. *PLOS ONE*, 14(11):1–17, 11 2019.

[12] Liubov A. Flores, Vicente Vidal, and Gumersindo Verdú. System matrix analysis for computed tomography imaging. *PLoS ONE,* 10, 2015.

[13] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.

[14] Yongshuai Ge, Ting Su, Jiongtao Zhu, Xiaolei Deng, Qiyang Zhang, Jianwei Chen, Zhanli Hu, Hairong Zheng, and Dong Liang. Adaptive-net: deep computed tomography reconstruction network with analytical domain transformation knowledge. *Quantitative Imaging in Medicine and Surgery*, 10(2), 2020.

[15] Muhammad Usman Ghani and W. Karl. Deep learning based sinogram correction for metal artifact reduction. *Electronic Imaging*, 2018:4721–4728, 01 2018.

[16] Lars A. Gjesteby, Qingsong Yang, Yan Xi, Hongming Shan, Bernhard E. H. Claus, Yannan Jin, Bruno De Man, and Ge Wang. Deep learning methods for ct image-domain metal artifact reduction. In *Optical Engineering + Applications*, 2017.

[17] Gary H. Glover. Compton scatter effects in ct reconstructions. *Medical Physics*, 9:860–867, 1982.

[18] Katharina Hahn, Harald Schoendube, Karl Stierstorfer, Joachim Hornegger, and Frederic Noo. A comparison of linear interpolation models for iterative ct reconstruction. *Medical Physics*, 43:6455–6473, 12 2016.

[19] Kerstin Hammernik, Tobias Würfl, Thomas Pock, and Andreas Maier. A deep learning architecture for limited-angle computed tomography reconstruction. In Klaus Hermann Maier-Hein, geb. Fritzsche, Thomas Martin Deserno, geb. Lehmann, Heinz Handels, and Thomas Tolxdorff, editors, *Bildverarbeitung für die Medizin 2017*, pages 92–97, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.

[20] David C. Hansen, Guillaume Landry, Florian Kamp, Minglun Li, Claus Belka, Katia Parodi, and Christopher Kurz. Scatternet: A convolutional neural network for cone-beam ct intensity correction. *Medical Physics*, 45(11):4916–4926, 2018.

[21] Kenneth M. Hanson and George W. Wecksung. Local basis-function approach to computed tomography. *Appl. Opt.*, 24(23):4028–4039, Dec 1985.

[22] Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973.

[23] Gabor Herman and Attila Kuba. Discrete tomography in medical imaging. *Proceedings of the IEEE*, 91:1612 – 1626, 11 2003.

[24] Gabor T. Herman. *Fundamentals of Computerized Tomography: Image Reconstruction from Projections*. Springer Publishing Company, Incorporated, 2nd edition, 2009.

[25] Jiang Hsieh. *Computed Tomography: Principles, Design, Artifacts, and Recent Advances*. Press Monographs. SPIE, 2015.

[26] Xia Huang, Jian Wang, Fan Tang, Tao Zhong, and Yu Zhang. Metal artifact reduction on cervical ct images by deep residual learning. *BioMedical Engineering OnLine*, 17, 11 2018.

[27] Armando Luis Imhof, Carlos Adolfo Calvo, and Juan Carlos Santamarina. Seismic data inversion by cross-hole tomography using geometrically uniform spatial coverage. *Revista Brasileira de Geofísica*, 28(1):79–88, Jan 2010.

[28] S Jan, D Benoit, E Becheva, T Carlier, F Cassol, P Descourt, T Frisson, L Grevillot, L Guigues, L Maigne, C Morel, Y Perrot, N Rehfeld, D Sarrut, D R Schaart, S Stute, U Pietrzyk, D Visvikis, N Zahra, and I Buvat. Gate v6: a major enhancement of the gate simulation platform enabling modelling of ct and radiotherapy. *Physics in Medicine & Biology*, 56(4):881, jan 2011.

[29] S Jan, G Santin, D Strul, S Staelens, K Assié, D Autret, S Avner, R Barbier, M Bardiès, P M Bloomfield, D Brasse, V Breton, P Bruyndonckx, I Buvat, A F Chatziioannou, Y Choi, Y H Chung, C Comtat, D Donnarieix, L Ferrer, S J Glick, C J Groiselle, D Guez, P-F Honore, S Kerhoas-Cavata, A S Kirov, V Kohli, M Koole, M Krieguer, D J van der Laan, F Lamare, G Largeron, C Lartizien, D Lazaro, M C Maas, L Maigne, F Mayet, F Melot, C Merheb, E Pennacchio, J Perez, U Pietrzyk, F R Rannou, M Rey, D R Schaart, C R Schmidtlein, L Simon, T Y Song, J-M Vieira, D Visvikis, R Van de Walle, E Wieërs, and C Morel. Gate: a simulation toolkit for pet and spect. *Physics in Medicine & Biology*, 49(19):4543, sep 2004.

[30] Robert J. Jennings. A method for comparing beam-hardening filter materials for diagnostic radiology. *Medical Physics*, 15(4):588–599, 1988.

[31] Peter M. Joseph. An improved algorithm for reprojecting rays through pixel images. *IEEE Transactions on Medical Imaging*, 1(3):192–196, 1982.

[32] Avinash C. Kak and Malcolm Slaney. *Principles of Computerized Tomographic Imaging*. Society for Industrial and Applied Mathematics, 2001.

[33] S. Krimmel, J. Baumann, Z. Kiss, A. Kuba, A. Nagy, and J. Stephan. Discrete tomography for reconstruction from limited view angles in non-destructive testing. *Electronic Notes in Discrete Mathematics*, 20:455–474, 2005. Proceedings of the Workshop on Discrete Tomography and its Applications.

[34] Attila Kuba, Gabor Herman, Samuel Matej, and Andrew Todd-Pokropek. Medical applications of discrete tomography, discrete mathematical problems with medical applications. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science, AMS*, 55:195–208, 01 2000.

[35] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

[36] Hoyeon Lee, Jongha Lee, Hyeongseok Kim, Byungchul Cho, and Seungryong Cho. Deep-neural-network-based sinogram synthesis for sparse-view ct image reconstruction. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 3(2):109–119, 2019.

[37] R M Lewitt. *Physics in Medicine & Biology*, 37(3):705, mar 1992.

[38] Jae S. Lim. *Two-Dimensional Signal and Image Processing*, pages 42–45. Prentice Hall, 1nd edition, 1990.

[39] Tibor Lukić and Benedek Nagy. Deterministic discrete tomography reconstruction by energy minimization method on the triangular grid. *Pattern Recognition Letters*, 49:11–16, 2014.

[40] Joscha Maier, Stefan Sawall, Michael Knaup, and Marc Kachelrieß. Deep scatter estimation (dse): Accurate real-time scatter estimation for X-ray ct using a deep convolutional neural network. *Journal of Nondestructive Evaluation*, 37, 07 2018.

[41] Bruno De Man and Samit Basu. Distance-driven projection and backprojection in three dimensions. *Physics in Medicine & Biology*, 49(11):2463, may 2004.

[42] Samuel Matej, Avi Vardi, Gabor T. Herman, and Eilat Vardi. *Binary Tomography Using Gibbs Priors*, pages 191–212. Birkhäuser Boston, Boston, MA, 1999.

[43] M Nauwynck, Shabab Bazrafkan, Anneke van Heteren, Jan De Beenhouwer, and Jan Sijbers. Ring artifact reduction in sinogram space using deep learning. pages 486–489, 08 2020.

[44] Csaba Olasz. Addressing discretization artifacts in tomography by accessing and balancing pixel coverage of projections. In Reneta P. Barneva, Valentin E. Brimkov, and Giorgio Nordo, editors, *22st International Workshop on Combinatorical Image Analysis*, Lecture Notes in Computer Science. Springer International Publishing, 2024, (Accepted).

[45] Csaba Olasz, László G. Varga, and Antal Nagy. Evaluation of the interpolation errors of tomographic projection models. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Daniela Ushizima, Sek Chai, Shinjiro Sueda, Xin Lin, Aidong Lu, Daniel Thalmann, Chaoli Wang, and Panpan Xu, editors, *Advances in Visual Computing*, pages 394–406, Cham, 2019. Springer International Publishing.

[46] Csaba Olasz, László G. Varga, and Antal Nagy. Beam hardening artifact removal by the fusion of FBP and deep neural networks. In Xudong Jiang and Hiroshi Fujita, editors, *Thirteenth International Conference on Digital Image Processing (ICDIP 2021)*, volume 11878, page 1187817. International Society for Optics and Photonics, SPIE, 2021.

[47] Csaba Olasz, László G. Varga, and Antal Nagy. Novel u-net based deep neural networks for transmission tomography. *Journal of X-Ray Science and Technology*, 30:1–19, 11 2021.

[48] Csaba Olasz, László Varga, and Antal Nagy. Ct scanner calibration based on optimization. In *5th Winter School of PhD Students in Informatics and Mathematics*, pages 37–38, 2018.

[49] Csaba Olasz, László Varga, and Antal Nagy. Tomográfiai modellek interpolációs hibájának vizsgálata. In *Képfeldolgozók és Alakfelismerők társaságának 13. konferenciája*, 2021.

[50] Hyoung Suk Park, Yong Eun Chung, Sung Min Lee, Hwa Pyung Kim, and Jin Keun Seo. Ct sinogram-consistency learning for metal-induced beam hardening correction. *Medical Physics*, 45:5376–5384, 2017.

[51] Ruben Pauwels, Wenchao Cao, Bo Wang, Yongshun Xiao, and Wim Dewulf. Exploratory research into reduction of scatter and beam hardening in industrial computed tomography using convolutional neural networks. *e-Journal of Nondestructive Testing*, 24, 03 2019.

[52] G Poludniowski, G Landry, F DeBlois, P M Evans, and F Verhaegen. Spekcalc: a program to calculate photon spectra from tungsten anode X-ray tubes. *Physics in Medicine & Biology*, 54(19):N433, sep 2009.

[53] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[54] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[55] L. A. Shepp and B. F. Logan. The fourier reconstruction of a head section. *IEEE Transactions on Nuclear Science*, 21(3):21–43, 1974.

[56] R.E. Sheriff and L.P. Geldart. *Exploration Seismology*. Cambridge University Press, 1995.

[57] Robert L. Siddon. Fast calculation of the exact radiological path for a three-dimensional ct array. *Medical physics*, 12 2:252–5, 1985.

[58] Emil Sidky, Per Christian Hansen, Jakob Joergensen, and Xiaochuan Pan. Iterative image reconstruction for ct with unmatched projection matrices using the generalized minimal residual algorithm. page 43, 10 2022.

[59] Wim van Aarle, Willem Jan Palenstijn, Jeroen Cant, Eline Janssens, Folkert Bleichrodt, Andrei Dabravolski, Jan De Beenhouwer, K. Joost Batenburg, and Jan Sijbers. Fast and flexible X-ray tomography using the astra toolbox. *Opt. Express*, 24(22):25129–25147, Oct 2016.

[60] A. van der Sluis and H.A. van der Vorst. Sirt- and cg-type methods for the iterative solution of sparse linear least-squares problems. *Linear Algebra and its Applications*, 130:257–303, 1990.

[61] Ge Wang. A perspective on deep imaging. *IEEE Access*, 4:8914–8924, 2016.

[62] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[63] Zhou Wang and Alan Bovik. *Modern Image Quality Assessment*. Morgan & Claypool, 1st edition, 2006.

[64] Stefan Wesarg, Matthias Ebert, and Thomas Bortfeld. Parker weights revisited. *Medical Physics*, 29(3):372–378, 2002.

[65] Tobias Würfl, Mathis Hoffmann, Vincent Christlein, Katharina Breininger, Yixin Huang, Mathias Unberath, and Andreas K. Maier. Deep learning computed

tomography: Learning projection-domain weights from image domain in limited angle problems. *IEEE Transactions on Medical Imaging*, 37(6):1454–1463, 2018.

[66] Fang Xu and K. Mueller. A comparative study of popular interpolation and integration methods for use in computed tomography. In *3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2006.*, pages 1252–1255, 2006.

[67] Mei Yap, Shalini Vinod, Ivan Ho Shon, A Fowler, Michael Lin, Gabriel Gabriel, and Lois Holloway. The registration of diagnostic versus planning fluorodeoxyglucose positron emission tomography/computed tomography in radiotherapy planning for non-small cell lung cancer. *Clinical oncology (Royal College of Radiologists (Great Britain))*, 22:554–60, 09 2010.

[68] Zhicong Yu, Frédéric Noo, Frank Dennerlein, Adam Wunderlich, Günter Lauritsch, and Joachim Hornegger. Simulation tools for two-dimensional experiments in X-ray computed tomography using the forbild head phantom. *Physics in Medicine & Biology*, 57(13):N237, jun 2012.

[69] Lotfi Zarour and Galina Malykhina. *Comparison of Analytical BP-FBP and Algebraic SART-SIRT Image Reconstruction Methods in Computed Tomography for the Oil Measurement System*, pages 335–343. 01 2020.

[70] Gengsheng Zeng and Grant Gullberg. Unmatched projector/backprojector pairs in an iterative reconstruction algorithm. *IEEE transactions on medical imaging*, 19:548–55, 06 2000.

[71] Yanbo Zhang and Hengyong Yu. Convolutional neural network based metal artifact reduction in X-ray computed tomography. *IEEE Transactions on Medical Imaging*, PP, 09 2017.

Summary

The PhD thesis presents the research of the author in the field of transmission X-ray tomography. The binding concept of the conducted research was to improve the quality of the reconstructed image by improving existing reconstruction algorithms and fusion of CT and Convolutional Neural Networks (CNN) techniques.

The dissertation consists of two major parts. In Chapter 2 we analyzed the interpolation errors that is originating from the discrete representation of the projection data and the reconstructed image. In Chapter 3 we combined Computed Tomography (CT) with Convolutional Neural Networks (CNN) in a unique way to improve the image quality for the case of tomography when the projection data shows strong signs of beam hardening and random electrical noise.

Discretization artifacts

The number of projections and detector elements in a CT is limited. In addition, reconstruction is usually performed on a discrete grid. These restrictions of the model create artifacts on the reconstructed images. In Chapter 2, we show that the type of interpolation used during the backprojection is strongly influences these artifacts (i.e., *interpolation errors*). Connection between pixel coverage and interpolation error is also obtained to get a better understanding of the interpolation errors. We propose a correction method to reduce the effects of interpolation errors in reconstructed images. We tested our proposed method in a comprehensive experiment, where we found that our proposed correction method can significantly improve the quality of the image. I also found that increasing the number of projection lines in the projection geometry can improve the image quality in terms of interpolation error, but the increased number of the view angles were not effective in achieving the same level of improvement.

Artifact reduction using U-net based neural networks

Combining computer tomography with deep learning is an effective way to achieve improved image quality and artifact reduction in reconstructed images. In Chapter 3, I present three novel neural network architectures for tomographic reconstruction with reduced effects of beam hardening and electrical noise. The reconstruction step is located inside of the proposed neural networks. In the case of two networks, multiple skip connections maintained a strong bound between the parts of the neural network that allowed the projection data and the reconstructed image to be improved together. We tested the proposed methods on datasets that were generated on our own. The datasets contain physically correct simulated data and show strong signs of beam hardening and electrical noise. We also performed a numerical evaluation of the neural networks on the reconstructed images according to three error measurements. Our experimental results showed that the reconstruction step used at the end of the neural network or in skip connections in neural networks improves the quality of the reconstructions.

Contributions of the Thesis

In the **first thesis group**, the contributions are related to the errors of discrete projection geometries with different interpolation methods. I examined interpolation methods in various settings to find their advantages and disadvantages. I proved the connection between pixel coverage and the interpolation errors and showed how to reduce the effects of interpolation errors using the pixel coverage. The corresponding results in details can be found in Chapter 2. These results were published in one conference proceedings [45] and accepted for publication in another conference proceedings [44].

I/1. I implemented a framework to test popular interpolation methods in various settings. The framework considers different projection geometries, seven noise levels, and two reconstruction algorithms. I evaluated the results of multiple tests on our datasets and popular head phantoms based on three error metric.

I/2. I found that the iterative reconstruction algorithm was less sensitive to the applied interpolation method than the analytical reconstruction algorithm.

I/3. I proved the connection between pixel coverage and the interpolation error by experiments. I applied the pixel coverage to get a better understanding of the interpolation errors and also to visualize the different artifact structures caused by the different interpolation methods.

I/4. I used a correction formula for the projection matrix based on pixel coverage to eliminate interpolation errors from the reconstructed images and I proved by experiments that the proposed correction method can reduce the interpolation error.

I/5. I showed that the interpolation error of the methods can be reduced by increasing the number of projection lines, but the increased number of view angles resulted in only minor changes.

In the **second thesis group**, the contributions are related to the fusion of computed tomography algorithms and deep convolutional neural networks. The neural networks used are based on the well-known U-net architecture. The combined algorithms were compared on the basis of their noise and artifact reducing capabilities. The corresponding results in details can be found in Chapter 3. These results were published in two conference proceedings [46] and a journal paper [47].

II/1. I created a physically correct database using GATE simulation software for training and testing the neural networks. The focus of the created database was the cupping and streak artifacts caused by beam hardening and the coexisting electrical noise.

II/2. I modified the original U-net and applied it at different stages of the image processing pipeline of computer tomography. I proposed a CNN structure (*TomoNet1*) where the reconstruction step was attached to the pre-processing CNN as a non-trainable layer. I showed by experiments that the proposed method yielded superior results in the comparison.

II/3. I proposed two neural network architectures (*TomoNet2* and *TomoNet3*), where the reconstruction step is used as a non-trainable neural network layer in the form of skip connections. I tested the algorithms on two datasets and showed that the new architectures outperformed the others, where there was no or only one reconstruction layer inside the network.

Összefoglalás

Jelen doktori disszertáció a szerző kutatói tevékenységét mutatja be a transzmissziós számítógépes tomográfia területén. Az elvégzett kutatások célja az volt, hogy a rekonstruált képek minőségét javítsam meglévő módszerek fejlesztésével, illetve új technikák kidolgozásával.

A disszertáció két fő részből áll. A 2. Fejezet-ben az interpolációs hibák elemzésével foglakoztam, amelyek a vetületi adatok és a rekonstruált kép diszkrét reprezentációjából ered. A 3. Fejezetben a számítógépes tomográfia és konvolúciós neurális hálózatokat egyedi módon kombináltam a rekonstruált kép minőségének javítására sugárkeményedéssel és elektromos zajjal nagy mértékben terhelt vetületi adatok esetén.

Diszketizációs műtermékek

A vetületek és detektor elemek száma korlátozott egy CT berendezésben. Továbbá, a rekonstukciót általában diszkrét rácson hajtják végre. A modell ezen megszorításai műtermékeket okoznak a rekonstruált képeken. A 2. Fejezetben bemutatom, hogy a visszavetítés során használt különböző interpolációs módszerek nagy mértékben befolyásolják ezen műtermékeket (más néven *interpolációs hibákat*). Megfigyeltem, hogy kapcsolat van a pixelek lefedettsége és az interpolációs hiba között, amely segítette az interpolációs hibák jobb megértését. Javasoltam egy módszert az interpolációs hibák hatásának csökkentésére a rekonstruált képeken. A módszert egy átfogó kísérletben megvizsgáltam, amelyben azt találtam, hogy a javasolt módszer képes szignifikánsan javítani a képek minőségét. Továbbá megfigyeltem hogy a vetítő sugarak számának növelése a vetületi geometriában képes az interpolációs hibákat csökkenteni, de a vetületi irányok számának növelése nem tudta elérni ugyanazt a szintű javulást.

Műtermékek csökkentése U-net alapú neurális hálózatokkal

A mély tanulás módszereinek használata a számítógépes tomográfiában egy hatékony mód a rekonstruált képek minőségének javítására. A 3. Fejezetben három új módszert mutatok be a sugárkeményedéssel és elektromos zajjal terhelt vetületek rekonstrukciójának javítása érdekében. A javasolt új módszerek esetén a rekonstrukciós lépés a konvolúciós neurális hálózaton belül helyezkedik el. Két neurális hálózat esetén, számos residuális kapcsolat lehetővé tette, hogy a módszerek a vetületeket és a rekonstruált képet is tudják együttesen javítani. A módszereket olyan adathalmazokon vizsgáltam, amelyeket fizikailag korrekt módon szimuláltam. Az adatokat nagy mértékű elektromos zaj és sugárkeményedés jeleit mutatják. Az objektív kiértékelést három hiba mérték segítségével végeztem a rekonstruált képeken. A kísérletek alapján a rekonstrukciós lépés alkalmazása a neurális hálózat végén vagy rezidulális kapcsolatokként javította a rekonstruált kép minőségét.

A disszertáció tézisei

Az **első téziscsoportban** a hozzájárulások a különböző interpolációs módszereket használó diszkrét vetületi geometriák hibáihoz kapcsolódik. Megvizsgáltam több különböző interpolációs módszert számos beállítás mellett, annak érdekében, hogy a módszerek előnyeit és hátrányait megtaláljam. Bizonyítottam, az interpolációs hiba és a pixelek lefedettsége közötti kapcsolatot és bemutattam hogyan lehet az interpolációs hibát csökkenteni a pixelek lefedettségének használatával. A kapcsolódó eredmények részletes bemutatása a 2 Fejezetben található. Az eredmények publikálásra kerültek egy konferencia kiadványban [45] és befogadták publikálásra egy másik konferencia kiadványban [44].

I/1. Implementáltam egy keretrendszert, amellyel alkalmas interpolációs módszerek tesztelésére számos különböző beállítással. A keretrendszerben használható több vetületi geometria, hét zajszint és két rekonstrukciós algoritmus. A tesztek kiértékelését saját adathalazokon végeztem, valamint két fejfantomon, amelyet számos publikációban használtak. Az objektív kiértékelést három metrika segítségével végeztem.

I/2. Megállapítottam, hogy az iteratív rekonstrukciós algoritmus kevésbé volt érzékeny az alkalmazott interpolációs módszerre mint az analitikus rekonstrukciós algoritmus.

I/3. Kísérletileg bizonyítottam a kapcsolatot a pixelek lefedettsége és az interpolációs hiba között. A pixelek lefedettségét alkalmaztam az interpolációs hiba jobb megértésére és a különböző interpolációs módszerek által okozott struktúrált műtermékek vizualizálására.

I/4. A pixelek lefedetségén alapuló korrekciós formulát alkalmaztam a vetítési mátrixon annak érdekében, hogy eliminálja az interpolációs hibát a rekonstruált képekről. Kisérletileg igazoltam, hogy a javasolt korrekciós formula képes az interpoláció hiba csökkentésére.

I/5 Megmutattam, hogy az interpolációs hiba csökkenthető a vetítő sugarak számának növelésével, de a vetületi irányok növelése csak jelentéktelen változásokat eredményezett.

A **második téziscsoportban** a hozzájárulások a számítógépes tomográfia és a mély konvolúciós neurális hálózatok fúziójához kapcsolódnak. A használt neurális hálózatok a népszerű U-net struktúrán alapszanak. A módszerek a rekonstruált képen mért elektromos zaj és műtermék csökkentő képességeik alapján lettek összehasonlítva. A kapcsolódó eredmények részletes bemutatása a 3 Fejezetben található. Az eredmények egy konferencia kiadványban [46] és egy folyóiratban kerültek publikálásra [47].

II/1. A fizikai folyamatokhoz hű adatbázist generáltam a neurális hálózatok tanításához és teszteléséhez a GATE szímulációs szoftver segítségével. Az adatbázis fókuszában a sugárkeményedés eredményeképpen megjelenő streak és cupping műtermékek voltak és az egyidejüleg jelentkező elektromos zaj.

II/2. Módosítottam az eredeti U-net struktúrát és a számítógépes tomográfiai képfeldolgozási folyamat különböző szakaszaiban alkalmaztam. Javasoltam egy módszert, ahol a rekonstrukciós lépés egy előfeldolgozó U-net után lett csatlakoztatva egy nem tanítható rétegként. Kísérletileg megmutattam a javasolt módszer jobb eredményeket ért el az összehasonlításban.

II/3. Két neurális hálózat struktúrát javasoltam, amelyek esetén a rekonstrukció nem tanuló rétegként van jelen residuális kapcsolatok formájában. Két adathalmazon értékeltem az eredményeket, amelyek alapján az új módszerek eredményesebbek voltak azon módszerekhez képest, ahol csak egy vagy egy rekonstrukciós réteg sem volt a neurális hálózatban.

Publications

Journal publications

[47] **Cs. Olasz**, L. G. Varga and A. Nagy. Novel U-net based deep neural networks for transmission tomography. *Journal of X-Ray Science and Technology*, VOL(30), 13-31, 2022.

Full papers in conference proceedings

[45] **Cs. Olasz**, L. G. Varga and A. Nagy. Evaluation of the Interpolation Errors of Tomographic Projection Models. In *14th International Symposium on Visual Computing*, Springer International Publishing, 394-406, 2019.

[46] **Cs. Olasz**, L. G. Varga and A. Nagy. Beam hardening artifact removal by the fusion of FBP and deep neural networks. In *Thirteenth International Conference on Digital Image Processing*, SPIE, 1187817, 2021.

[44] **Cs. Olasz** Addressing discretization artifacts in tomography by accessing and balancing pixel coverage of projections. In *22st International Workshop on Combinatorical Image Analysis*, Springer International Publishing, 2024. (**Accepted**)

Further related publications

[48] **Cs. Olasz**, L. G. Varga and A. Nagy. CT scanner calibration based on optimization. In *5th Winter School of PhD Students in Informatics and Mathematics.*, 2018.

[49] **Cs. Olasz**, L. G. Varga and A. Nagy. Tomográfiai modellek interpolációs hibájának vizsgálata. In *Képfeldolgozók és Alakfelismerők társaságának 13. konferenciája.*, 2021.

Acknowledgments

First of all, I would like to thank my supervisors, László Varga and Antal Nagy, for their guidance and work during my PhD years. I am especially grateful for their patience and that they shared their wisdom about life with me. I would also like to thank my colleagues and friends who helped me enjoy the period of my studies. Special thanks to Gábor Lékó, György Kalmár and István Megyeri for the discussions that were helpful in taking my research further. I also thank my colleagues at the Institute. Last but not least, I would like to thank my parents without whom my academic journey would not have been possible, my wife for her endurance, and my beautiful daughter for being a good baby.

We are grateful to the Nvidia Hardware Grant program for providing a GPU for the research. Ministry of Human Capacities, Hungary, grant 20391-3/2018/FEKUSTRAT is acknowledged. This research was supported by the project "Integrated program for training new generation of scientists in the fields of computer science", no EFOP-3.6.3-VEKOP-16-2017-0002. The project has been supported by the European Union and co-funded by the European Social Fund. This research was supported by grant TUDFO/47138-1/2019-ITM of the Ministry for Innovation and Technology, Hungary. This research was supported by the projects "Extending the activities of the HU-MATHS-IN Hungarian Industrial and Innovation Mathematical Service Network" EFOP-3.6.2-16-2017-00015. This research was supported by the project "Integrated program for training new generation of scientists in the fields of computer science", no. EFOP-3.6.3-VEKOP16-2017-00002. The research leading to these results has received funding from the national project TKP2021-NVA-09. Project no. TKP2021-NVA-09 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme.