CHALMERS

Optimization of Non-Local Means filtration for Cone Beam Computed Tomography

Master's Thesis in Engineering Mathematics and Computational Science

Lina Berneryd

Department of Mathemtics
Chalmers University of Technology
Gothenburg, Sweden 2015
Master's Thesis 2015:1

Abstract

Cone Beam Computed Tomography (CBCT) is a medical imaging technique used to visualize the internals of an object from a set of X-ray images. The X-ray images are taken with a cone shaped X-ray beam at many angles around the object being visualized. The images are then back projected through a volume forming a 3D representation of the object.

Non-local means (NLM) denoising was applied to the X-ray images before they were used in the back projection algorithm. NLM was implemented for the graphics card and runs in real time while the X-ray images are taken. Compared to current denoising methods the result was less noisy and had higher sharpness.

It was found that using NLM on the X-ray images led to streak artifacts and smearing of fine details in the reconstructed volume. Many versions of NLM were tested to reduce these effects, including variance reduction, total variation and using adjacent images in the scan. The different versions however gave very similar results. The state-of-the-art denoising algorithm BM3D was also tested for reference and the results were very similar to NLM.

NLM denoising in the reconstructed volume as a post-processing step was also tested which showed improved results over denoising the projection images. Denoising in 3D led to less artifacts and better preservation of details. The runtime for the 3D-denoising is only a few seconds on the graphics card, making it useful in practice.

3D-NLM denoising offers the possibility of lowering the radiation dose to the patient since the noise can be reduced while keeping edges and details intact. The noise can be reduced such that a low dose scan is comparable to a high dose scan denoised with current CBCT denoising methods. A comparison of spatial resolution and preservation of details indicates that the dose could be lowered by as much as half without significantly lowering the image quality.

Acknowledgements

I would like to thank my supervisor Markus Eriksson at Elekta for his guidance and for the opportunity to do this thesis.

Lina Berneryd, Stockholm, May 2015

Contents

1

Introduction

Elekta is currently developing a Cone Beam Computed Tomography (CBCT) system for the Gamma Knife. The purpose of the CBCT is to find the position of the patient relative the Gamma knife. A 3D volume is reconstructed from the CBCT images using a filtered back projection reconstruction algorithm. At a low radiation dose the resulting volume is very noisy, and currently used denoising methods are not edge preserving and thus reduce noise while reducing sharpness. Non-local means (NLM) is a newly proposed denoising algorithm [2], which is suitable for removing additive white noise while preserving edges. The drawback with NLM is that it is computationally intensive, but there are speed-ups available both algorithmically and using the fact that it is well suited for massively parallel computing on a GPU [13].

This thesis will focus on adapting and optimizing NLM for CBCT. NLM is not state-of-the-art in image denoising but comes quite close and has the advantage of being simple to implement and suitable for the GPU.

1.1 Objective

A previous thesis at Elekta [7] investigated NLM for CBCT, with the main focus of investigating NLM from a statistical point of view and compare it with more computationally intensive algorithms. The focus of this thesis will instead be on finding and implementing an algorithm that is useful in practice.

The NLM implementation should fulfill the time constraint of about 30 ms per CBCT image. The time constraint is such that the filtration can be performed in real time, while the X-ray images are taken. The algorithm will be implemented in `C++` and CUDA. Possible improvements special for CBCT will be investigated, for example using the fact that consecutive X-ray images are very similar.

The aim is to optimize NLM for CBCT such that the reconstructed 3D volume is of highest possible quality. The NLM implementation will be compared to the cur-

rent filtration in the Fourier domain, Wiener filtration, and state-of-the-art denoising algorithms.

1.2 Related work

NLM has been used for denoising CT images in several previous articles, both projection images and slices in the reconstructed volume. Below is a brief summary of NLM for CT and a review of image denoising in general.

1.2.1 NLM for CT

NLM has been applied to CBCT projection images in a previous thesis at Elekta [7]. The conclusion of that thesis was that NLM was slightly better than Gaussian blur, but for low dose images Gaussian blur gave better results. Applying a variance stabilizing transform before NLM denoising improved the results.

NLM has also been used on projection images with a different reconstruction algorithm [27] as a way of regularizing the images, here the NLM regularization managed to reduce the noise level and keep edges intact. Denoising using NLM has also been applied to the projection images with filtered back projection reconstruction [11] for CT. It was found that a half dose scan came close to a full dose in image quality as assessed by a radiologist. NLM was however also found to create structured noise which could compromise reconstruction quality.

NLM has been used in several articles for post processing 2D CT slices with improved noise reduction and detail preservation [15][5]. Using adjacent slices in the volume was found to greatly improve the results.

A temporal NLM method has been used to create an improved reconstruction algorithm for temporal 4D CBCT [14]. An iterative NLM version was used on the reconstructed volume as a post-processing step and artifacts caused by movement were reduced.

Previous studies have found NLM useful in combination with CT, allowing the noise level to be reduced while keeping edges and details intact, however in some cases at the cost of introducing artifacts. This thesis will further investigate using NLM for CBCT projection images, with the main focus on developing an implementation that is fast enough and gives improved reconstruction quality. The quality measures will mainly be performed in the reconstructed volume. Different improvements of NLM will be tested and CBCT specific improvements such as using adjacent projection images will be implemented. Artifacts introduced by NLM will be investigated, and if possible minimized.

1.2.2 Image denoising in general

The state-of-the-art denoising method is often considered to be BM3D [6], which compared to NLM is a more complex patch based algorithm. The main focus of this thesis

will be to investigate if NLM is useful for this application, since NLM is easier to implement and adjust, and BM3D will only be used for comparison. The results will show if NLM is useful for this application, and if newer and more complicated filtration methods could obtain even better results.

In [16], issues with NLM and BM3D that make them non-ideal filters are compared and it is found that both algorithms create structure in uniform noisy areas, and noise halos around edges remain after denoising.

A multitude of articles dealing with improvements upon NLM and different applications have been published since Buades et al. [2] introduced the algorithm in 2005. Some of the more promising suggestions will be implemented and evaluated based on the effect they have on the quality of the reconstructed volume. Examples of those versions include WAV-NLM [21], NLM combined with total variation [23] and NLM with other types of distance measures. The improvements often come with the drawback of being more computationally intensive than the original NLM and harder to perform in a massively parallel setting.

2

Non-local means

Non-local means relies on the fact that natural images contain repetitive patterns. The repetition of similar patches in images are used to estimate a noise-free image. A multitude of articles trying to improve NLM have been published since the introduction in 2005 by Buades et al. [2]. A few of the improvements that show promising results will be introduced here. The different improvements will be tested to see how adjusting the NLM algorithm affects the quality of the reconstructed volume.

2.1 Standard algorithm

A noisy image $v = \{v(i)|i \in I\}$ is filtered producing the denoised image $\hat{v} = \{\hat{v}(i)|i \in I\}$, where $I \subset \mathbb{N}^2$ is the set of all pixel indices in the image. The pixel values of $\hat{v}$ are computed as

$$\hat{v}(i) = \sum_{j \in S(i)} w(i,j)v(j), \tag{2.1}$$

where $w(i,j)$ is the weight between the patch centered at i compared to that centered at j and $S(i) \subseteq I$ is a square search window centered around pixel i. If the patches are similar the pixel value at j is considered a good guess for the pixel at i and thus the weight $w(i,j)$ is large.

There are many different possibilities for how the kernel w can be defined. The standard version is an exponentially decaying function in the euclidean distance between the patches. Let $P(i) \subseteq I$ be a square patch centered around pixel i, then the weight between pixel i and j is defined as

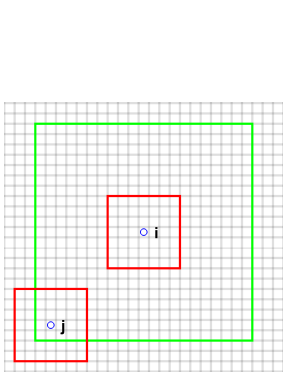
$$w(i,j) = \frac{1}{Z(i)} e^{\frac{-||v(P(i))-v(P(j))||^2}{h^2}}, \tag{2.2}$$

where $Z(i)$ is a normalizing constant,

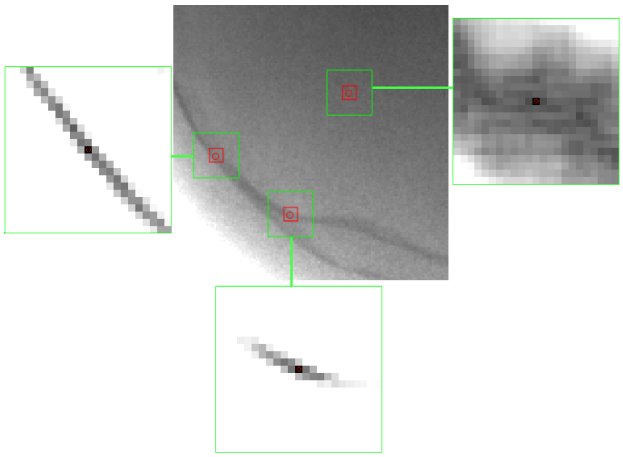
$$Z(i) = \sum_{j \in S(i)} e^{\frac{-||v(P(i)) - v(P(j))||^2}{h^2}} \tag{2.3}$$

such that $\sum_{j \in S(i)} w(i,j) = 1$

Here h is the kernel parameter determining the importance of the patch distance. A large h gives a more uniform distribution of weights. This parameter increases with the noise level in the image.



(a) Search window around i and patches around pixel i and j.



(b) Matching patches where $||v(P(i)) - v(P(j))|| < 0.2$ for three different pixels in an X-ray image. Edges and pixels near edges have fewer matching-ray patches than homogeneous areas.

Figure 2.1: Patch and search window.

The size of the search window $S(i)$ could be set to the whole image, but this has been shown to give similar results, or worse, than using a moderate window size. Typically, a window size around 21×21 is used. The size of the patch used for comparison is often set to around 7×7 pixels. A larger patch size tends to be more robust to noise, but worse at preserving fine details and structure.

There are a few identified issues with NLM that makes it non-ideal [16] such as introducing patterns in noisy homogeneous areas and noise halos around edges. The noise halos are mainly caused by the fact that there are very few matching patches around edges even though there is a lot of similarity in the neighborhood, see Figure 2.1. There are many suggested improvements on NLM which usually are more computationally intensive and/or less suitable for massively parallel implementations, since in most of these cases each pixel can not be treated independently.

2.2 Kernel

The kernel in the original NLM version is a Gaussian function, where the parameter depends on the global noise level in the image. Different kernels have been proposed,

(a) Exponential kernel. **(b)** Box kernel.

Figure 2.2: Shape of different kernels, the width is determined by the kernel parameter.

several articles point to advantages of using a kernel with compact support, e.g. an exponential function with two parameters; a weighting h determining the importance of the patch distance and a cutoff parameter putting small weights to zero. In [21], a box kernel is used, weighing all patches within some cutoff equally which according to the article performs equally well, or in some cases better, and only requires one parameter.

2.3 Weighted average patch re-projection

A different way of projecting from the patch space back to the denoised image has been proposed, called weighted average patch re-projection (WAV), that reduces the noise halos around edges [21]. A box kernel is used, and for each pair of patches where the distance is below the threshold, all pixels are updated in the patch, using values from the matching patch. This reduces the rare-patch effect since more similar pixels are available even for pixels near edges. More of the information in each patch is used, since WAV takes advantage of the fact that each pixel is actually part of 7×7 patches. This can be seen as a different way of projecting matches to the image, the usual being central re-projection, where only the central pixel is updated. The denoised image is in this case computed as,

$$\hat{v}(i) = \sum_{j \in S(i)} \sum_{\delta \in [-\frac{P-1}{2}, \frac{P-1}{2}]^2} w(i+\delta, j+\delta) v(j+\delta). \tag{2.4}$$

The δ here corresponds to a patch offset that determines the patch definition where $\delta = [0,0]$ corresponds to a patch centered at the current pixel, and $\delta = [-\frac{P-1}{2}, -\frac{P-1}{2}]$ to a patch where the pixel is in the upper left corner of the patch instead. The weights are normalized as $\sum_{j \in S(i)} \sum_{\delta \in [-\frac{P-1}{2}, \frac{P-1}{2}]^2} w(i-\delta, j-\delta) = 1$.

The difference between WAV and standard NLM is that equation 2.4 is the sum over all possible patch configurations and not just the central. For the box kernel this re-projection scheme has been shown to minimize the quadratic risk of the pixel estimator $\hat{v}(i)$ [21].

2.4 Total variation and NLM

Total variation NLM (TV-NLM) aims at reducing the rare patch effect by combining NLM with total variation (TV) [23]. After NLM has been applied to the image, a TV

minimization step is performed, starting in the denoised pixels from NLM and using the sum of the NLM weights as guidance. The total variation scheme results in the following optimization problem

$$\underset{u}{\operatorname{argmin}} \sum_{i \in I} Z(i)(u(i) - \hat{v}(i))^2 + TV(u). \tag{2.5}$$

where $Z(i)$ is the sum of the weights for pixel i from NLM, and $\hat{v}(i)$ is the denoised pixel value at pixel i.

This leads to a solution that is very close to the NLM solution for large Z_i and closer to a TV solution if Z_i is small. The weights for pixel i, Z_i, are small if few matches were found for that pixel. The idea is that the algorithms will work together such that TV will reduce the rare patch effect, and the noise halo around edges will be smaller.

Another way of using TV is by doing the opposite, instead of trying to reduce the noise halo using TV, the NLM weights can be used as a map of interesting areas that should be preserved. TV could then be used to reduce the artifacts left by NLM in homogeneous areas instead. In this case equation 2.5 would be changed into

$$\underset{u}{\operatorname{argmin}} \sum_{i \in I} (M - Z(i))(u(i) - \hat{v}(i))^2 + TV(u). \tag{2.6}$$

Where M is greater than the upper bound for the $Z(i)$ to avoid a zero term, M can then be calculated as $M = \max_i(Z(i)) + 1$. The weight of the NLM term in equation 2.6 is then inversely proportional to the number of matches found in NLM.

2.5 Neighborhood classification

Neighborhood classification NLM utilizes several patch distance measures to determine which pixels to use when creating the denoised image [17]. This NLM version is more selective, with the aim of not blurring fine details. Two measures are used in the pre-filtering step, the difference in the mean and the average gradient. These two measures are selected to make sure only patches that are actually similar are used in the denoising even when using a wide filter kernel. The method gives a mixture of heavily smoothed areas and parts where both noise and details are left intact.

2.6 Adjacent images

A possible denoising advantage for CBCT is that an X-ray scan contains adjacent images that are very similar. Using the neighboring images for denoising could give better results than only using similar patches in the current image. The search window will then be a stack of 2D search windows in neighboring images. Especially areas with few similar patches such as edges this could be an advantage. Edges often look similar and only move slightly between adjacent scan images.

In the following description x-NLM, where $x \in \mathbb{N}$ and $x \geq 0$, will denote a NLM version where in total $2x + 1$ images are used for each image denoised. For image

number i, in the sequence of X-ray images, the images in the span $[i-x, i+x]$ will be used to search for similar patches during denoising.

2.7 Contrast invariant distance measure

NLM has been applied to video denoising and using adjacent frames improved the results [19]. A contrast stretching transform was used between different video frames since it was found that changes in illumination between frames reduced the number of matching patches. Consecutive X-ray images in a CBCT scan might also suffer from similar variations.

A contrast stretching transform is proposed that only depends on a few values per search window that can be pre-calculated for each image with roughly the same runtime. The contrast stretching is calculated as follows; for a search window $S_t(i)$ around pixel i in frame t, let p_t and q_t denote the minimum and maximum in the search window in frame t. The transform is then given by

$$v_{out} = (v_{in} - \frac{q_{t+k} + p_{t+k}}{2}) \frac{q_t - p_t}{q_{t+k} - p_{t+k}} + \frac{q_t + p_t}{2}. \tag{2.7}$$

It was however found that a more computationally expensive contrast adjusting mapping based on histograms gave better results because the contrast stretching transform was sensitive to outliers. For runtime considerations the simpler form will be tested here.

3

Non-local means for CBCT

The aim is to adapt NLM specifically for CBCT, special consideration will be taken to what differentiates the X-ray images and reconstruction process from other general uses of NLM. There are several steps in the reconstruction algorithm for CBCT, the NLM denoising step is performed after the image correction steps related to the detector have been performed and a transform from Poisson to approximately Gaussian noise.

For CBCT applications it is important to find a denoising method that reduces noise but still preserves fine details, edges, and low contrast objects, otherwise important medical information could be lost.

3.1 Reconstruction algorithm

From the 2D X-ray images taken in a CBCT scan a 3D mesh of attenuation coefficients $f(x,y,z)$ is determined using the approximative filtered back projection algorithm (FDK) developed for CBCT by Feldkamp et al. in 1984 [10]. The attenuation coefficient depends on the material properties and is a measure of how X-rays propagates through the material. In a CBCT scan the object is radiated from a point source from which the X-rays form a cone beam. The source and detector rotate around the object taking X-ray images from a large set of angles. Each image is then separately filtered and back projected through the reconstruction grid of 3D voxels. The back projection of all the images in the scan then gives an estimate of the attenuation coefficient in each voxel in the volume.

In the description below β is the projection angle at which the X-ray image is taken in a circular trajectory around the objects z-axis. R is the source trajectory radii. A virtual detector is placed in the object, such that for coordinates (u,v) in the detector plane v coincides with the z-axis of the object. The data recorded at the detector is denoted $p(\beta, u, v)$. The FDK algorithm can be divided into three steps, pre-weighting, ramp filtering and backprojection [22].

Step 1 - Pre-weighting: The detector data $p(\beta, u, v)$ is weighted to account for the varying length the X-rays travel depending on the position in the detector plane. The length correction weight depends on the cone (κ) and fan angle (γ), and is given by $\frac{R}{\sqrt{R^2+u^2+v^2}} = \cos\kappa\cos\gamma$, for (u,v) in the detector plane. The weighted image data is then

$$p_W(\beta, u,v) = \frac{R}{\sqrt{R^2+u^2+v^2}} p(\beta, u, v). \tag{3.1}$$

Step 2 - 1D Line filtering: The weighted projection image is then convoluted with the ramp filter h per row in the image.

$$p_F(\beta, u,v) = p_W(\beta, u, v) * h(u) \tag{3.2}$$

The ramp filter h defined as

$$h(t) = \int_{-\omega_{max}}^{\omega_{max}} |\omega| e^{-j2\pi\omega t} d\omega, \tag{3.3}$$

is a high-pass filter which enhances high frequency content in the images.

Step 3 - Backprojection to the 3D volume: The weighted and filtered projection data is then backprojected through the 3D grid,

$$f(x,y,z) = \frac{1}{2}\int_0^{2\pi} \frac{R^2}{(R + x\sin\beta - y\cos\beta)^2} p_F(\beta, u', v') d\beta. \tag{3.4}$$

where $u' = R\frac{-x\sin\beta+y\cos\beta}{R+x\cos\beta+y\sin\beta}$ and $v' = \frac{zR}{R+x\sin\beta-y\cos\beta}$.

There are more intermediate steps in the algorithm, such as several detector related corrections and the ramp filtering step is often combined with a smoothing apodization filter which reduces noise.

3.2 Apodization filter

In the 1D line filtering stage of the reconstruction algorithm the X-ray projection data is row-wise filtered with a ramp filter. The ramp filter enhances high frequency components, such as edges and also noise. Usually a low-pass smoothing apodization filter, such as the Hamming filter, is used in combination with the ramp filter to reduce noise. An apodization filter is a filter that is zero outside some chosen interval, here such that high frequencies above some cutoff frequency are put to zero. The drawback with these kind of filters is that they are not edge preserving since smoothing and removing high frequencies blurs edges.

When using an edge preserving filtering procedure such as NLM in combination with FDK, the idea is to not use a smoothing filter in the line filtering stage to avoid blurring the image. The denoising algorithm should remove most of the noise so there is no need for denoising in the frequency spectrum.

3.3 Noise in X-ray images

The standard definition of NLM is suited for Gaussian noise, but CBCT images are subject to noise following a Poisson distribution with variance dependent on the image intensity. A pixel value g in the X-ray image can be modeled as

$$g = c_g X \tag{3.5}$$

where $c_g \in \mathbb{R}_+$ is the detector gain factor that varies from pixel to pixel, and X follows a Poisson distribution

$$Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}. \tag{3.6}$$

The noise level in the image depends on the strength of the signal. The standard deviation for Poisson noise is $\sqrt{\lambda}$, which gives a signal-to-noise ratio $SNR = \frac{\lambda}{\sqrt{\lambda}} = \sqrt{\lambda}$ which grows with the intensity λ. X-ray images taken with a higher dose therefore have a higher SNR.

Using a variance stabilizing transform (VST) the image noise can be transformed to approximately Gaussian noise with constant variance [18]. A common VST is simply $f(x) = \sqrt{x}$, where x is the original X-ray signal and $f(x)$ the stabilized signal.

There are extensions of NLM which deal with other types of noise. Given that the values of the pixels $i,j \in I$ follow a Poisson distribution, the distance measure is changed from the squared distance $f(i,j) = (v(i) - v(j))^2$ to

$$f(i,j) = v(i)\log(v(i)) + v(j)\log(v(j)) - (v(i) + v(j))\log(\frac{v(i) + v(j)}{2}). \tag{3.7}$$

which is the likelihood ratio for the hypothesis that pixels i and j have identical λ versus the hypothesis that their λ are independent [8]. This distance measure could then be an alternative to using a VST with standard NLM.

4

Quality measures

A way of measuring how the quality of the reconstructed volume is affected is needed to evaluate the performance of NLM and other types of filtering.

There are several factors of interest in medical imaging such as noise reduction, preservation of detail, contrast and sharpness of edges. These measures are often estimated from the Catphan phantom described below.

A head phantom modeling a human head is also used, mainly for visual comparison of results and artifacts. The head phantom has very detailed bone structure, teeth, and jaw. There are no soft tissue details since the head is filled with a homogeneous material.

4.1 Catphan

The Catphan phantom is a commonly used quality measurement phantom [25]. It is used to assess different quality aspects of a CT system. The phantom is a cylinder of a homogeneous material with several different modules containing objects aimed at measuring contrast, precision, spatial resolution etc.

One of the modules that will be used contains cylinders made of materials with different attenuation constants, and thin metal wires. This module is here used to estimate the sharpness of different denoising methods by measuring how sharp the edges of the cylinders are.

Another module considered contains a circle of high contrast line pair groups of decreasing width, that are used to estimate the spatial resolution of the CT-system. The resolution is determined by counting the number of resolvable line pair groups.

4.2 Sharpness-to-noise

The noise level is often measured as the standard deviation in a uniform volume when there is no noise-free volume available [12][20].

The sharpness of edges is estimated using the full-width at half maximum (FWHM). Following a similar procedure as in [9] and [20], a noise-resolution trade-off measure based on the FWHM is used.

The FWHM is the width of a function at half its maximum. For a Gaussian function, $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$, the FWHM is

$$FWHM = 2\sqrt{(2ln2)}\sigma. \tag{4.1}$$

An oversampled edge in an image can be used to estimate the FWHM. The forward difference of the edge is calculated and a Gaussian function is fitted in the least squares sense.

The sharpness is measured on reconstructed CBCT scans taken of the Catphan phantom. From the average of several slices of the highest resolution data available a good estimate of the cylinder centers are found. Then given the positions of the cylinders, and their radii the cylinder edge in each slice is estimated.

For each pixel in a square encapsulating the cylinder the distance from the center and the pixel intensity is recorded. The data is then re-binned based on the distance, and the mean is calculated in each bin. A Gaussian function is then fitted to the forward difference of the estimated circle edge. This procedure is applied to the cylindrical inserts with the sharpest contrast since these are less sensitive to noise.

Since the data is noisy the estimated circle edge is multiplied with a Hann window to reduce the influence of the noisy tails. The measurements are averaged over many slices and over several CBCT scans of the Catphan.

The cylinders in the Catphan are surrounded by a homogeneous material. The noise level is measured by taking the standard deviation in a volume surrounding the cylinders. This then gives an estimate of the relation between sharpness and noise in the volume. The above procedure will only work for high contrast circles since otherwise the data is so noisy that the forward difference of the edge does not resemble a Gaussian function.

4.3 Contrast-to-noise

The contrast of the cylinders in the Catphan are estimated using the contrast-to-noise ratio (CNR). The CNR is defined as the following ratio

$$CNR = \frac{2(C_c - C_b)^2}{\sigma_c^2 + \sigma_b^2} \tag{4.2}$$

[12]. Where C_c is the mean value of the cylinder and C_b the mean of the background. σ_c^2 and σ_b^2 are the respective variances, where σ_b^2 is calculated in a volume surrounding the cylinder. If this ratio increases after filtering the visibility of the object is enhanced in the denoising process.

4.4 PSNR and SSIM

A forward projector is used to simulate X-ray images from a known volume. Simulated noise is added to the images with an intensity that is estimated from real CBCT data for different dose intensities. Since the noise-less 3D volume is known measures such as Peak Signal to Noise (PSNR) can be used to compare different denoising methods [26]. PSNR is defined by

$$PSNR = 10 \log_{10}(\frac{peakval^2}{MSE}), \tag{4.3}$$

where *peakval* depends on the format of the data and is the maximum attainable value. MSE is the mean square error between the noise-free reference and the denoised volume.

Structural Similarity Index (SSIM) [26], is a newer measure of image quality that is meant to better measure changes in structural information. SSIM is based on local means and standard deviations of the two images being compared as well as the local cross-covariance between the images. The mean of the pixel-wise values is denoted the mSSIM.

Two types of volumes are available, real high resolution CT skull volumes which have been filtered such that they are free of noise and a 3D Shepp-Logan volume which is a simple model of the head consisting of two large ellipses containing several smaller ones with different attenuation constants.

5

Implementation

NLM was first implemented on the CPU. This was as expected not fast enough to adhere to the time constraint of 30 ms per X-ray image. An implementation was also written for the GPU in CUDA. The first CUDA version was also far from the time constraint, since it was a direct adaption of the fastest CPU version found and far from optimal in the GPU setting.

5.1 CPU

The naive implementation of NLM has time complexity of $O(MNQ^2R^2)$ where R is the patch width and Q is the search window width. A summary of the implementation can be seen in Algorithm 1, in this approach all pixels in the image are processed independently and for each pixel all the patch-wise distances in the search window are calculated.

Since the patch-wise distance calculations is a separable convolution it can be performed in linear time with respect to the convolution kernel size. In this specific case the separable convolution can be calculated using a sliding sum since the convolution kernel is flat, this means that the patch-wise distances can be calculated in constant time with respect to the patch size. The time complexity is then instead $O(MNQ^2)$, with a standard patch size of 7 this could reduce the runtime with a factor of 7×7. In Algorithm 2 the sliding sum separable convolution is calculated using a temporary difference image for each offset in the search window. The patch wise difference image is calculated for $M \times N$ pixels for in total $Q \times Q$ number of offsets.

Data: v noisy image, h filter parameter
Result: $\hat{v}$ denoised image, Z weights
$\hat{v}(x,y) = 0$, $\mathsf{Z}(x,y) = 0$ for all $(\mathsf{x}, \mathsf{y}) \in M \times N$
foreach $(x,y) \in M \times N$ **do**
    `//Calculate weights for each pixel in the search window.`
    foreach $(p,q) \in S(x,y)$ **do**
        $w = e^{-||v(P(x,y)) - v(P(x+q,y+p))||/h^2}$
        $\hat{v}(x,y) \mathrel{+}= w\mathsf{v}(x+q, y+p)$
        $\mathsf{Z}(x,y) \mathrel{+}= w$
    end
    $\hat{v}(x,y) \mathrel{/}= \mathsf{Z}(x,y)$
end

Algorithm 1: Implementation of NLM, that is $O(MNQ^2R^2)$ for an image of size $N \times M$. Here $P(x,y) \subseteq I$ and $S(x,y) \subseteq I$ are the square patch and search window centered around pixel (x,y), with size $Q \times Q$ and $R \times R$ respectively.

Data: v noisy image, h filter parameter
Result: $\hat{v}$ denoised image, Z weights
$\hat{v}(x,y) = 0$, $\mathsf{Z}(x,y) = 0$ for all $(\mathsf{x}, \mathsf{y}) \in M \times N$
foreach $(p,q) \in S(x,y)$ **do**
    `//Calculate all patch distances for this offset` (p,q).
    diffImage = `calculatePatchDistances(p,q)`
    `//Update image and weights.`
    foreach *pixel* $(x,y) \in M \times N$ **do**
        $w = e^{-\mathsf{diffImage}(\mathsf{x},\ \mathsf{y})/h^2}$
        $\hat{v}(x,y) \mathrel{+}= w\mathsf{v}(x+q, y+p)$
        $\mathsf{Z}(x,y) \mathrel{+}= w$
    end
end
`//Normalize the image using the weights in Z`
$\hat{v}$ = `normalize(`$\hat{v}$`, Z)`

Algorithm 2: $O(MNQ^2)$ implementation of NLM using the fact that the patch differences can be calculated as a separable convolution with a sliding sum making the **calculatePatchDistances** step independent of the patch size.

5.2 CUDA

The fastest CPU implementation, i.e Algorithm 2, was first translated to the GPU setting, which resulted in a runtime of about 300 ms far from the time limit. The bottleneck of this implementation became the read/writes from memory, since each difference image was written to global memory.

Then instead translating the naive $O(MNQ^2R^2)$ NLM implementation to the GPU setting resulted in a runtime of about 200 ms, the improved time was due to the reduced need for global memory accesses. The 200 ms time could be improved by letting threads in a GPU block work together loading and calculating differences, which gives an algorithm that is $O(MNQ^2R)$ in the patch size R. This algorithm performs a separable convolution locally in the GPU thread block but not using the sliding sum trick. With these changes the runtime came down to about 70 ms per image. Then doing some further improvements, reducing the need for synchronization by using more shared memory and unrolling the small patch difference calculation loops the resulting runtime came down to about 20 ms, which is below the 30 ms constraint.

5.3 WAV-NLM

The WAV-NLM version leads to a slightly different algorithm, the difference is shown in algorithm 3. When a match is found all pixels in the current patch are updated and the weights are increased.

```
//Update image.
if ||v(P(x,y)) − v(P(x + p,x + q)))|| < h then
    foreach (i,j) ∈ P(x,y) do
        //Update pixels in this patch.
        v̂(i,j) += v(i+p, j+q)
        Z (i,j)++
        //Update pixles in patch P(x+p,y+q).
        v̂(i+p,j+q) += v(i,j)
        Z (i+p, j+q)++
    end
end
```

Algorithm 3: The difference between WAV-NLM and original NLM. Here WAV-NLM uses a box kernel such that if the patch distance is smaller than h then the pixel is added to the denoised pixel sum. For each match update all pixels in patch based on other patch and vice versa.

This makes it much harder to parallelize since each thread cannot handle one pixels reads/updates independent of other threads. Thus the previous fast algorithm is slowed down significantly. The update step in the fast CUDA implementation was adjusted according to Algorithm 3 and using atomic updates to avoid race conditions. More work

is done in WAV-NLM since each set of matching patches leads to $R \times R$ updates.

5.4 TV-NLM

The TV step is performed after NLM denoising, fast parallel algorithms exist for TV that are suitable for a GPU. A basic solver for the ROF-model was implemented, Algorithm 4, using the primal-dual Chambolle-Pock algorithm applied to image denoising as in [4].

$\tau, \sigma > 0$
`//Start in the NLM result` $\hat{v}$
$u_0 = \hat{v}$
$y_0 = \nabla u_0$
$\bar{u} = u_0$
for *k iterations* **do**
  `//Per pixel do:`
  $y_{k+1} = \frac{y_k + \sigma \nabla \bar{u}_k}{max(1, |y_k + \sigma \nabla \bar{u}_k|)}$
  $u_{k+1} = \frac{u_k + \tau \nabla * y_{k+1} + \tau/\sigma^2 Z\hat{v}}{1 + 2\tau/\sigma^2 Z}$
  $\bar{u}_{k+1} = 2u_{k+1} - u_k$
end

Algorithm 4: The primal-dual Chambolle-Pock Algorithm for an image with Gaussian noise.

5.5 Runtime comparison

The aim was to have a version of NLM that allows the reconstruction to be completed in real time, which means about 30ms are available per 720×780 image. The time constraint was fulfilled for several NLM versions as seen in Table 5.1.

The improved NLM version that uses adjacent images is slightly slower than standard NLM, but it is still well below 30 ms per image. WAV-NLM is significantly slower since it is harder to parallelize due to the pixels no longer being independent. The implementation of WAV-NLM is a direct modification of the fastest NLM version that uses atomic add on every pixel when updating since several threads tries to update pixel values simultaneously. This could probably be improved significantly if WAV would show significantly better results. The TV algorithm is not the fastest available, but a faster version could be implemented if it proved better than standard NLM.

The Poisson version is slightly slower than the Euclidean measure since there are several logarithms that need to be calculated for each pixel-wise difference, and this is slower than multiplication.

Implementation	time (ms)
$O(MNQ^2R^2)$ CPU	2000
$O(MNQ^2R^2)$ CUDA	200
$O(MNQ^2)$ CUDA	300
$O(MNQ^2R)$ CUDA	70
$O(MNQ^2R)$ CUDA, unrolled loops, reduced synchronization	20

Other NLM versions	time (ms)
WAV-NLM	300
2-NLM	22
Poisson-NLM	22
TV-NLM	90

Table 5.1: Running time comparison for NLM implementations, with various patch and search sizes for 720×780 images. Patch size 7×7, and search window size 21×21 was used in all NLM versions, except WAV-NLM where a smaller window was used and the 2-NLM version uses a search window that is 9×9 with in total 5 images. Because of the memory conflicts that occur in WAV-NLM the runtime depends on the filter parameter, the time here was for a moderate degree of filtering. The denoising was performed on a GeForce GTX970 graphics card.

6

NLM denoising results

The NLM implementations were tested on several sets of CBCT scan images of different dose levels as well as on simulated data. The results are summarized for different NLM versions and different parameter settings. There is also a investigation of NLM induced artifacts in the reconstructed volume. The most prominent artifact are the streaks that appear between sharp edges in 3D volume, the possible causes for these are investigated. Motivated by the results found a 3D version of NLM was also tested.

6.1 Poisson noise

Two ways of handling Poisson noise were introduced in section 3.3, either using a VST to get approximately Gaussian noise or using a distance measure adapted for Poisson noise. These two approaches gives near identical results and since the VST approach is slightly faster that approach will be used.

6.2 Search window size

The search window affects the noise reduction, if the window is large more matches are found and the noise is reduced further. But if more matches are found that are of lower quality the result might be worse.

A set of simulated projection images were used to compare the effect of the search window. There seems to be no noticeable advantage in having a window size larger than 21×21, in terms of PSNR and mSSIM of the reconstructed volume. The PSNR and mSSIM of the ramp filtered images were also compared, showing similar results. The filter parameter was optimized according to the PSNR values for each search window size. These tests were performed with the standard NLM version having a Gaussian kernel. The same parameter optimization procedure was performed independently for

the projection images and the volume. For the volume it was found better to use a lower degree of filtering than for the projection images.

Window size	Number of images	2D PSNR	2D mSSIM	3D PSNR	3D mSSIM
11×11	1	39.309	0.928	45.096	0.984
21×21	1	39.240	0.933	44.310	0.983
31×31	1	39.170	0.932	44.116	0.982
9×9	5	**40.798**	0.943	**46.255**	0.987
11×11	5	40.586	0.943	45.823	**0.986**
21×21	5	39.949	0.937	45.125	0.986
9×9	7	40.714	**0.944**	45.924	0.986
11×11	7	40.548	0.943	45.670	0.986

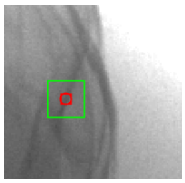
Table 6.1: The average 2D PSNR and mSSIM for ramp filtered images in a scan consisting of 330 simulated X-ray images. The best 3D PSNR and mSSIM in the reconstructed volume is also given. The filter parameter h was optimized for a fixed search window size and number of images used. The patch size used was 7×7.

The results in Table 6.1 indicate that there are more similar patches in a small neighborhood around the pixel being denoised, and that the adjacent images in a scan have similar neighborhoods. The overall high PSNR and mSSIM values are due to the fact that there are large homogeneous areas in the images where the denoising and structure preservation is almost ideal.

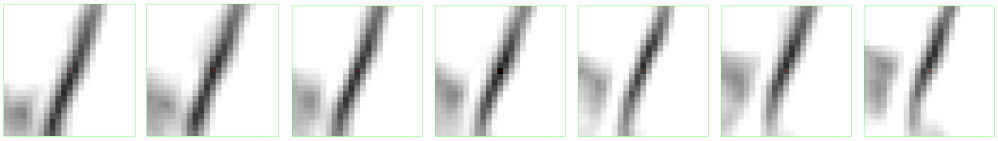
6.3 Adjacent images

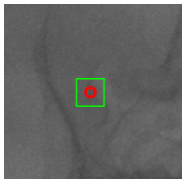
The distribution of similar patches for pixels in the X-ray images of the head phantom can be seen in Figure 6.1, many close matches are found on the edges in the adjacent images as well as in the current one. The increased number of similar patches found could have a significant impact on the denoising quality and maybe also reduce the rare patch effect around edges. Comparing the same reconstructed volume filtered using 0 or 9 adjacent images it is however hard to see any significant improvement.

The best PSNR and mSSIM for projection images was reached using several adjacent images, see Table 6.1. The best PSNR was found using 5 images in total with a small search window in each.



(a) Selected pixel on an edge in the image and the corresponding search and patch window.





(b) Selected pixel close to an edge and the corresponding search and patch window.

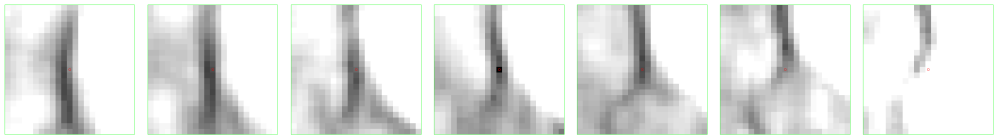


Figure 6.1: In (a) and (b) two selected pixels in the image that is being denoised are marked, below are the patch distance to some of the most similar patches in adjacent images within $||v(P(i)) - v(P(j))|| < 0.1$ where i is the selected pixel and j is a pixel in the search window. The scale goes from black (zero distance) to white (a distance greater than the cutoff 0.1). The patch distance image in the middle corresponds to the current image.

6.4 Patch size

Previous articles have found that a patch size close to 7×7 gives a robust similarity estimator, but a smaller patch size might be better at preserving small details. In Figure 6.2 the result can be seen for a few different parts of the head phantom. Varying the patch size seem to have a very small impact on the denoising result, when kept in a range close to 7. One difference that can be seen in some cases is that a smaller patch size leads to more patterns in homogeneous areas.

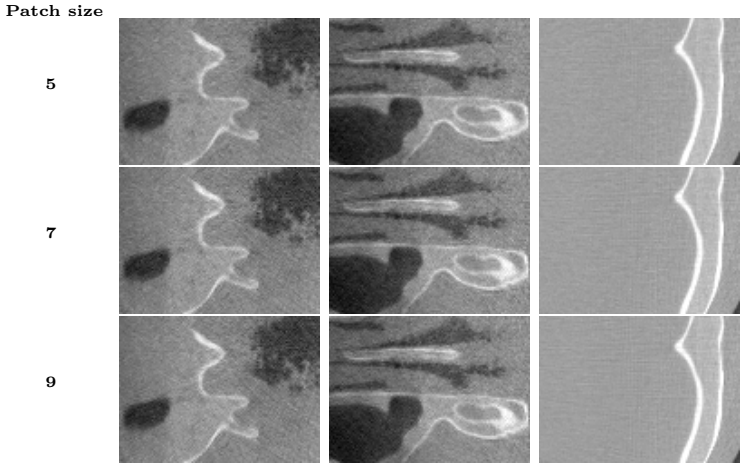


Figure 6.2: Comparison of NLM using different patch sizes. The volumes all look very similar, the patch size seems to only have a minor impact on the reconstructed volume.

Patch Size	PSNR	mSSIM
5×5	40.032	0.941
7×7	40.798	0.943
9×9	40.748	0.944
11×11	**40.848**	**0.947**
13×13	40.778	0.947

Table 6.2: The average PSNR and mSSIM for ramp filtered images in a scan consisting of 330 simulated X-ray images. The filter parameter h was optimized for different patch sizes. The number of images used was fixed to 5, and the search window size to 9×9.

The filter parameter was optimized for a fixed patch size using the previously found optimal search window settings the results are shown in Table 6.2. The size of the patch does not seem to affect the quality of the projection images to a great degree. It might be slightly better to use a bigger patch size but since this increases the runtime, the implementation is linear in the patch width, the standard patch size of 7×7 seems to be a good trade-off. The mSSIM indicates that structure is better preserved using a larger

patch size, however there is hard to see any difference in the reconstruction.

6.5 Kernel

The box and Gaussian kernel were tested, the results are very similar for the two, however there seem to be slightly less streak artifacts in the 3D volume when using the Gaussian kernel. The two kernels have similar denoising performance in terms of PSNR and mSSIM as seen in Table 6.3.

Kernel	PSNR	mSSIM
Gaussian	**40.798**	0.943
Box	40.760	**0.945**

Table 6.3: The average PSNR and mSSIM for ramp filtered images in a scan consisting of 330 simulated X-ray images. The filter parameter h was optimized for the different kernels. The number of images used was fixed to 5, the window size to 9×9 and the patch size used was 7×7.

6.6 Neighborhood classification

Neighborhood classification was tested to see if having a finer selection criterion when selecting pixels for denoising could improve the result. The difference in terms of noise reduction and sharpness was very small and this was not a significant improvement over regular NLM for this application.

6.7 Contrast invariant distance measure

A contrast invariant kernel using a contrast stretching transform between consecutive projection images was tested. The contrast stretching transform was applied with the previously found best parameter settings. The results were very similar to the original kernel and showed no significant improvement, it seems as the contrast is similar in neighboring regions in adjacent images.

6.8 TV-NLM

TV-NLM reduces the noise halos, however it also blurs edges and the overall shape of the edge is not improved. The TV step does not reduce the streak artifacts, setting the parameters such that mostly TV is used there are still streaks. TV-NLM does not significantly improve the result, in the projection images it is clear that the noise has been smoothened at edges but there is still a lot of variation in the shape of the edges and the streaks are still present in the reconstructed volume.

The inverse TV-NLM is able to remove patterns and noise left in smooth areas. The overall noise level in the volume, at a certain NLM filter parameter h can be significantly

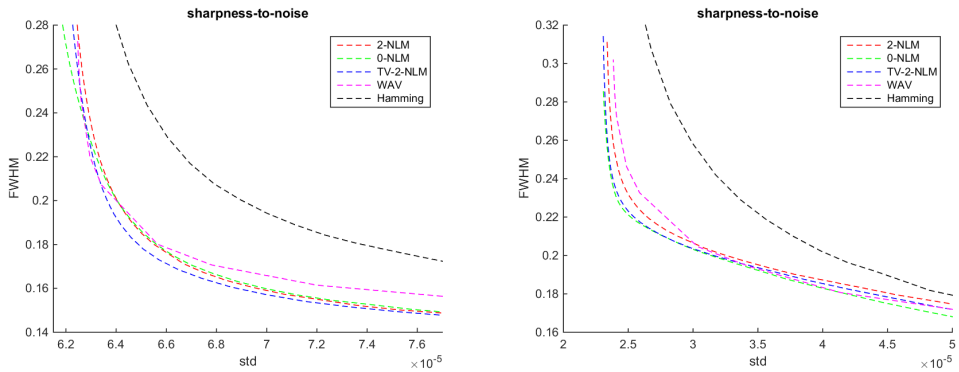
lowered without blurring edges. The TV step manages to remove the noise-only image artifacts left by NLM and the PSNR of the projection images are improved when using TV with NLM. The reconstruction quality is however not visibly improved, since artifacts become more visible in this case.

6.9 WAV-NLM

WAV-NLM seem to be a slight improvement on NLM for this application since the streak artifacts are reduced slightly using WAV. A smaller search size was used with WAV then with the other types of NLM for runtime considerations, and according to [21] a smaller search window is needed since more of the information in the neighborhood is used. The streak artifacts are slightly reduced but there are more patterns in homogeneous areas and the overall image quality is very similar to standard NLM.

6.10 Sharpness-to-noise

The procedure used for estimating the sharpness-to-noise relation using the Catphan phantom was introduced in section 4.2. The different NLM versions were tested on the same low dose Catphan X-ray dataset. The resulting sharpness-to-noise plot can be seen in Figure 6.3. The sharpness and noise values were recorded for a large range of filter parameter values. The sharpness ratio was calculated for the two high contrast cylinders 1 and 2 in Figure 6.4. In this plot an ideal filter would have a low constant FWHM value



(a) Circle nr 2. **(b)** Circle nr 1.

Figure 6.3: The relation between FWHM and the noise level in the reconstructed volume. The sharpness was measured on a cylinder edge in the Catphan phantom.

such that for greater noise reduction the sharpness would not be reduced. The non-edge preserving Hamming filter is here used as a reference, the normalized frequency cutoff that determines the degree of smoothing was varied in the span $[0.1, 1.0]$.

All the NLM versions here tested have a higher sharpness then the non-edge preserving Hamming filter for the same noise level. The variation among the different NLM

versions is quite small, probably too small to draw any definite conclusions about one being better than the other. The results also varies from cylinder to cylinder.

6.11 Contrast-to-noise

The contrast-to-noise relation is estimated using the Catphan phantom and following the procedure described in section 4.3. In Figure 6.4 the cylinders used to measure the CNR are labeled. The CNR per cylinder and for different levels of filtering are shown in 6.5. The CNR is as expected increased with filtering since NLM preserves the contrast and lowers the noise level in and around the cylinders. The CNR visibility limit is somewhere between cylinder number 5 which has the next lowest CNR and number 8 which can not be seen clearly at any denoising level.

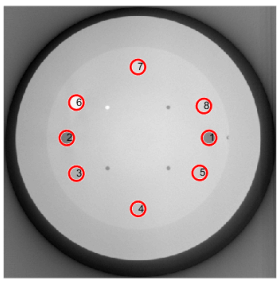
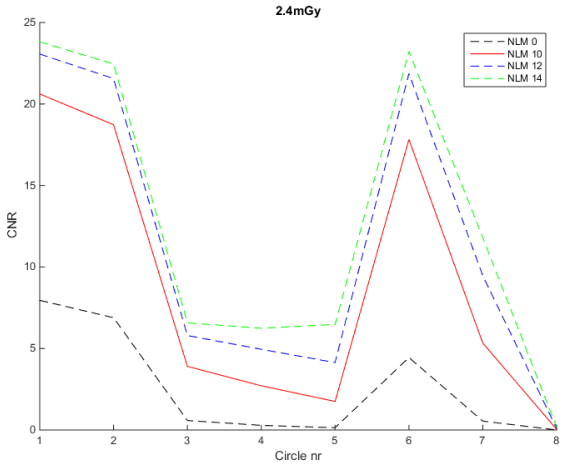
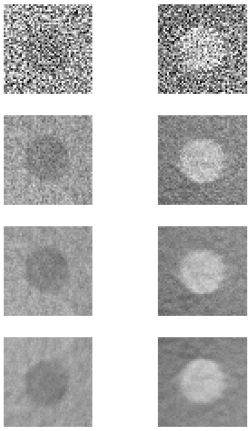


Figure 6.4: Cylindrical contrast inserts of varying density in the Catphan volume are marked and numbered. The inserts and a larger volume surrounding each of them respectively are used in the CNR calculations.



(a) CNR values at dose level 2.4mGy for the cylinders in the Catphan volume.

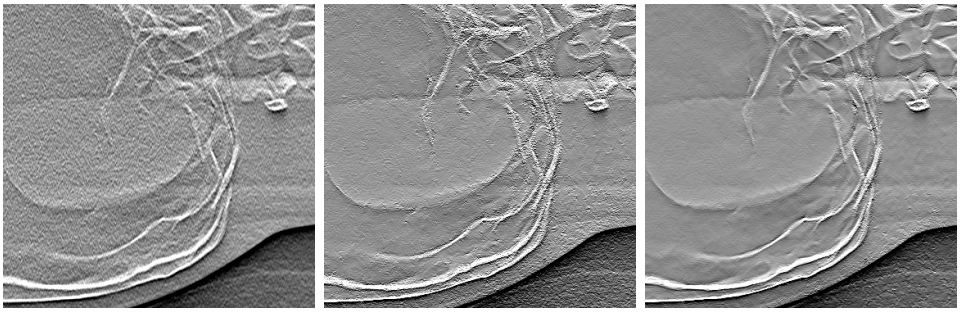


(b) Cylinder 5 **(c)** Cylinder 7

Figure 6.5: CNR plot for box-NLM letting the filter parameter range from 0 to 14, where 0 corresponds to no denoising.

6.12 Artifacts in Projection Images

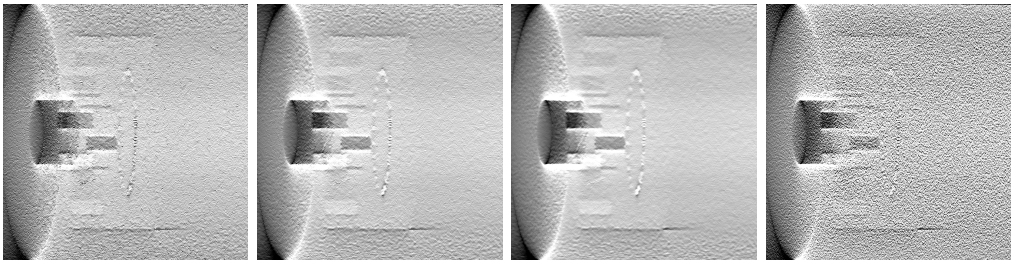
NLM is a non-ideal denoising filter since artifacts are introduced in denoised images [16], the most noticeable is the noise halos around edges and structure in homogeneous noise-only areas. The artifacts are not clearly visible in the unfiltered NLM output, but in the ramp filtered images they are enhanced as seen in Figure 6.6.



(a) Hamming, 0.63. **(b)** 2-NLM, $h = 3.2$. **(c)** 2-NLM, $h = 5$.

Figure 6.6: Comparison of part of a head phantom, with dose 2.4mGy, projection image for different degrees of denoising and for one of the NLM versions. NLM artifacts are enhanced by the ramp filtering step of the reconstruction process. Uneven smoothing in noisy areas and noise halos around edges are visible.

Along sharp edges in the Catphan projection images there is more and sharper noise remaining than in homogeneous areas, this is most noticeable for the edges of highest contrast. In Figure 6.7 the ramp filtered projection images for different levels of filtering are compared to a Hamming filtered projection image. There is more noise remaining in the Hamming case than for NLM filtered images, the noise left after using the Hamming filter is more evenly distributed over the whole image wheres in the NLM case the noise level depends on the number of similar patches found in the search window.



(a) 2-NLM $h = 3$. **(b)** 2-NLM $h = 5$. **(c)** 2-NLM $h = 10$. **(d)** Hamming filtered 0.63.

Figure 6.7: Comparison of Catphan projection images for different levels of filtering.

6.13 Artifacts in the reconstructed volume

When denoising the X-ray images using NLM streak artifacts appear for heavy filtering as seen in Figure 6.8 for two parameter settings in a slice of the Catphan volume. The streaks protruding from the cylinder edges are most prominent at the objects with the highest contrast.

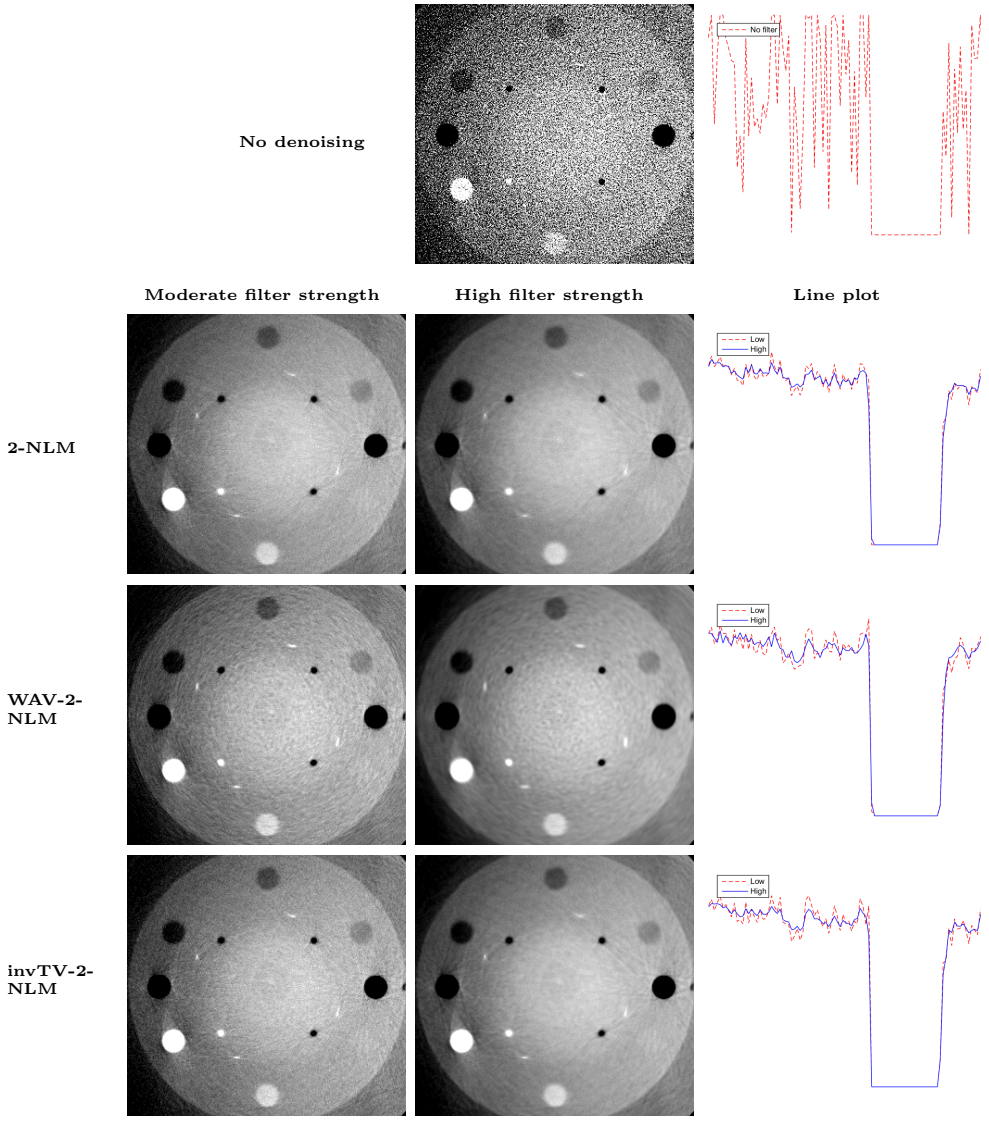


Figure 6.8: A slice of the Catphan 2.4mGy volume with different degree of filtering for several NLM versions and for comparison the same slice without any denoising. Per column each of the denoised volumes have similar noise level. Streak artifacts can be seen protruding from sharp edges. When the filter parameter is increased the streaks become more prominent. The line plot shows the second half of the center row in the image which contains the darkest cylinder.

In the head phantom, the streaks are more diffuse, a comparison for different NLM versions and filter parameters can be seen in Figure 6.9. The noise level is not uniform in the volume after filtering. There are fewer similar patches in areas with a lot of variation than in the homogeneous parts and thus the noise level is reduced less.

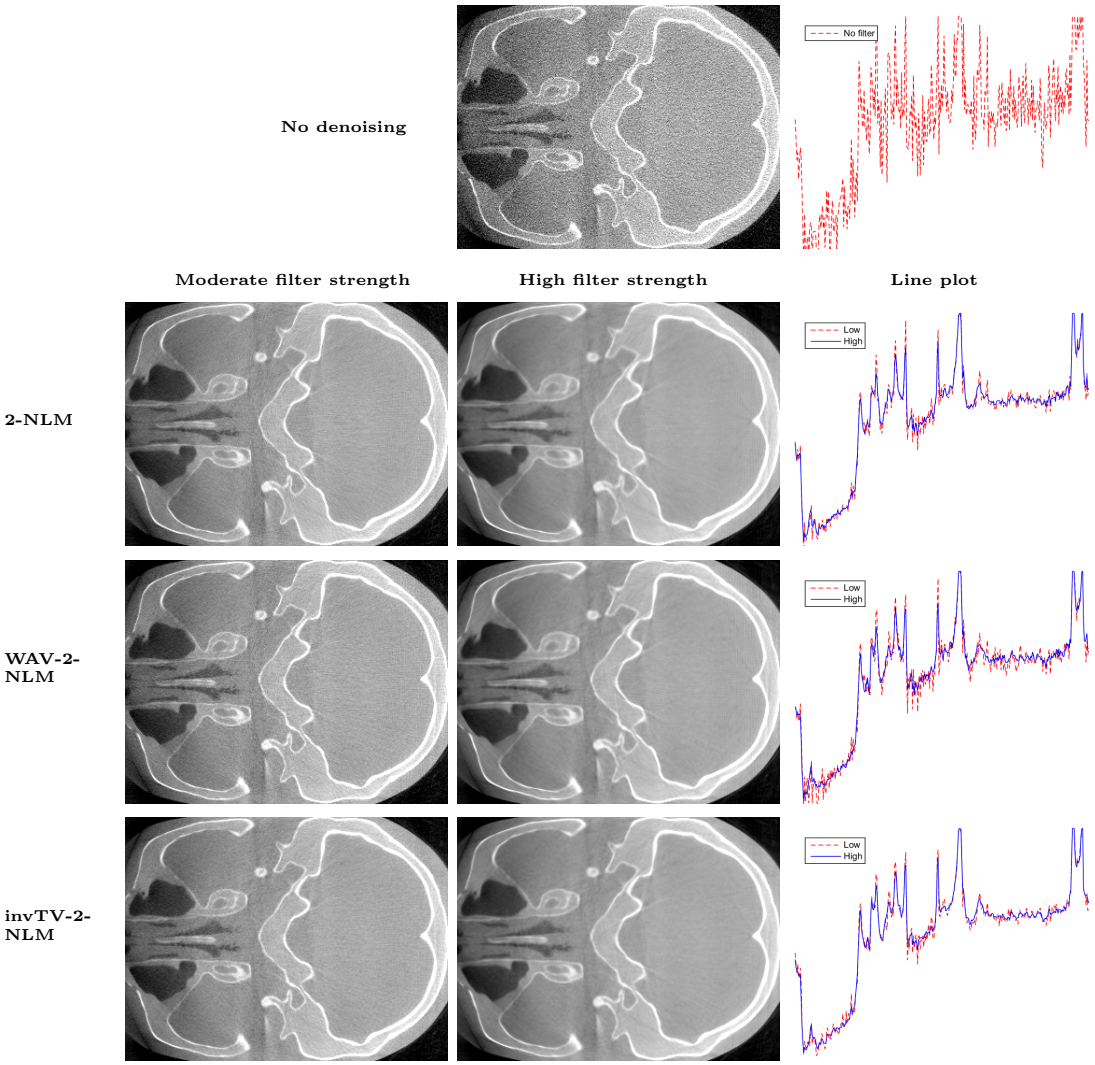


Figure 6.9: Comparison of artifacts and noise reduction using different versions of NLM for a 2.4mGy CBCT scan of the head phantom. Per row there are two degrees of denoising and a line plot of a row through the volume to clearly show how edges are affected.

The streaks that appear in the reconstruction are similar to the streaks that are caused by metal objects and motion. When there is metal present the very sharp edges of the metal object does not perfectly cancel when back-projected mainly because of the CBCT scan having finite resolution and cone beam artifacts. When the object being scanned moves sharp edges change position during the scan. In both these cases the

variations in the edges lead to streaks along the projection lines since the back projected edges does not cancel exactly [1].

With NLM edges are kept intact and noise in homogeneous areas is reduced to a greater degree, which leaves sharp distinct edges in the projection images. They are however affected by noise, and not completely regular in shape. The edge can move around a bit from image to image and there is noise remaining in its vicinity. When these edges are back-projected the scenario is quite similar to the metal and motion case.

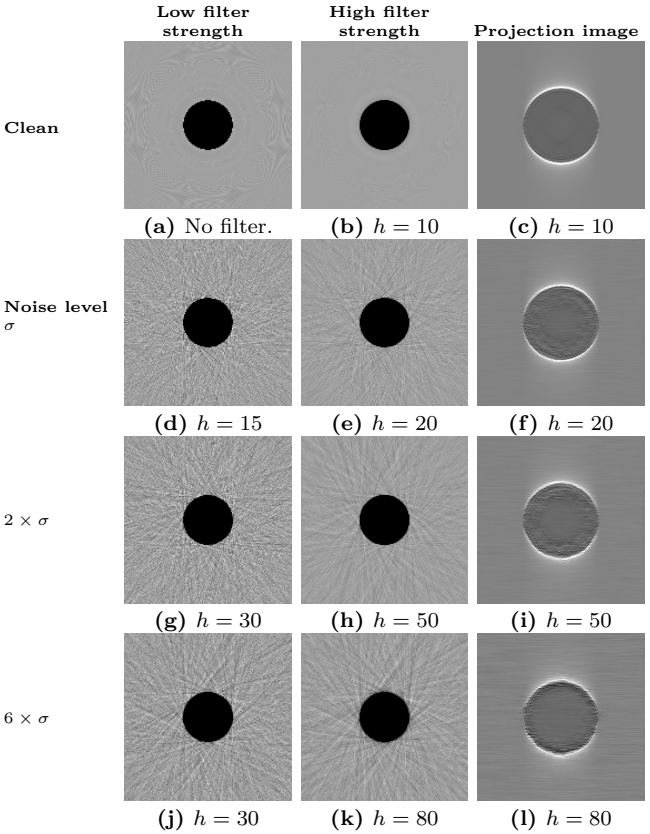


Figure 6.10: Box-2-NLM was used in all the simulations. Streak artifacts in a simulated ellipse volume that contains a sphere in the center, here only the sphere is visible. The magnitude of the streaks increase with the noise level. In the projection images the edges become more irregular with higher noise level. The parameter was selected such that the edges were smoothed to about the same degree, since the degree of smoothing probably also affects the intensity of the streaks.

The cause and degree of the streaks were investigated using an artificial 3D volume that consists of a sphere in the center of two large ellipses with different attenuation, simulated X-ray images were created from this volume and then reconstructed. If no noise is added and the images are NLM filtered, streaks do not appear and the edges of the ellipsoids are uniformly smoothed. The streaks seem to be dependent on the noise level, and they become more prominent as the noise is increased as can be seen in Figure 6.10. The noise at the sharpest edges is very dominant in the ramp filtered projection images where the overall noise level is very low. At lower levels of filtration the noise is more uniform over the whole image and the streaks in the reconstruction are less noticeable. This seems to be true for the real data as well, as seen in figure 6.8 for the Catphan phantom.

The streak artifacts are also dependent on the filter parameter used when denoising. In Figure 6.11 the denoising results are shown for the box kernel. For low levels of denoising the streaks are not clearly visible. If the denoising is increased the edges are kept sharp but most of the noise is removed, the streak artifacts are in this case clearly visible. Then for even higher degrees of filtering the edges are smoothed more and more uniformly. The streaks are then instead becoming less visible. For very high filtering the edge of the sphere is smoothed uniformly and here there are almost no streak artifacts present. In this case the sphere in the projection images are back-projected to a bigger sphere with smooth edges in the 3D volume.

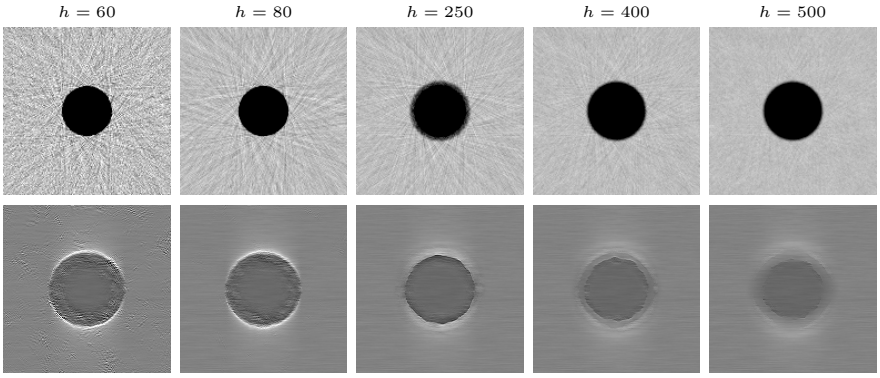
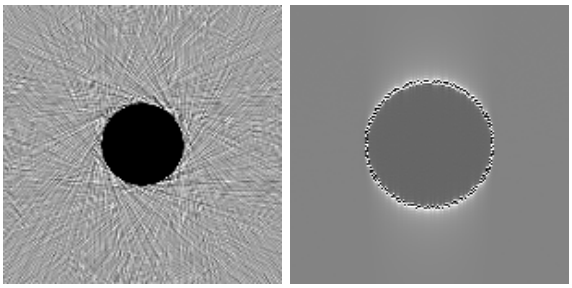


Figure 6.11: Streak artifacts in a volume that contains a sphere in the center denoised with box-2-NLM. The streaks depend on the filter parameter, here the filtering is increased with h.

The irregular and noisy shape of the edge can be simulated using the same volume as before and adding a noise halo manually to the edge of the sphere. When there is no other noise present than that added to the edge similar effects as seen with NLM are visible, see Figure 6.12.

There are also streak artifacts that appear without noise present if another object is added to the volume. Here a small ellipsoid was added close to the sphere in the center. These noise-independent streaks seem to appear when the scanned objects look very different from image to image in the projection image sequence. NLM will smear an object to a varying degree depending on the background and contrast. When using

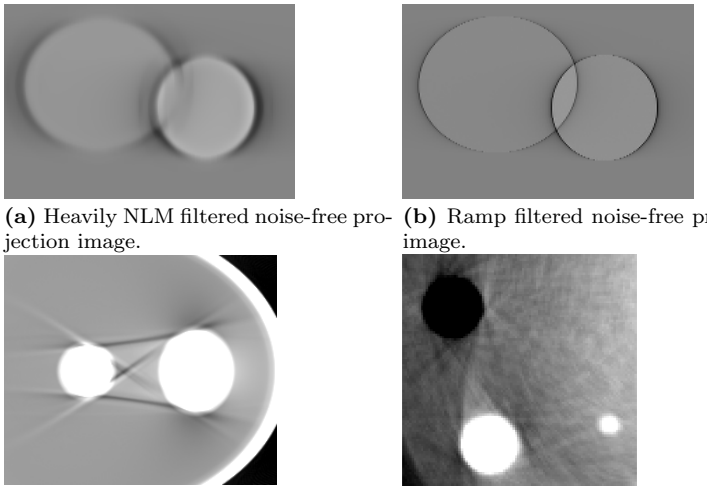


(a) Slice of volume.

(b) Ramp filtered simulated X-ray image.

Figure 6.12: Simulated NLM streak artifact. Noise was added to all projection images, but only on the border of the sphere in a segment 3 pixels wide. The resulting streaks protruding from the edges of the circle are very similar to the ones produced when NLM filtering a noisy image.

the volume above with just one sphere in the center the edges of the sphere look very much alike throughout the scan. The edges are denoised very similarly and thus with no noise present there are no streak artifacts. If another object is introduced that is on-top of the other object in some of the images, the edges have much lower contrast in those images. The edges are then in some of the images blurrier than in others. In Figure 6.13 the noise independent streaks can be seen between two spheres together with one of the projection images that is likely to cause this effect. In the figure there is also a part of the Catphan phantom where these types of streaks are present. If three independent scans are compared the streaks look very similar between the cylinders and are probably not dependent on the noise but rather the varying smoothing of the edge.



(a) Heavily NLM filtered noise-free projection image.

(b) Ramp filtered noise-free projection image.

(c) Streaks between the two ellipses in the reconstructed volume.

(d) Streaks between the three ellipses in the Catphan phantom.

Figure 6.13: NLM noise independent streaks at a very high degree of filtering of a volume with no noise added and a part of the Catphan phantom were these effects likely can be seen.

These streaks are also dependent on the filter parameter and if the filter parameter

is increased the effect after some peak value become less and less prominent. The patch wise distances becomes less and less important and the smearing of edges is more uniform throughout the sequence of images.

6.14 Denoising in 3D

Denoising the projection images introduces streak artifacts and smearing of fine details in the reconstructed volume. Performing the denoising in the 3D volume might be a better approach since in this case there will be no streaks introduced from the back projection step. 3D filtering however has the big drawback of not being able to run when the scan is taking place.

After the reconstruction, which was performed without any denoising, NLM filtering was applied to the 3D volume. The fastest GPU NLM algorithm was adapted for 3D by simply adding another direction of summation in the block-wise separable convolution kernel.

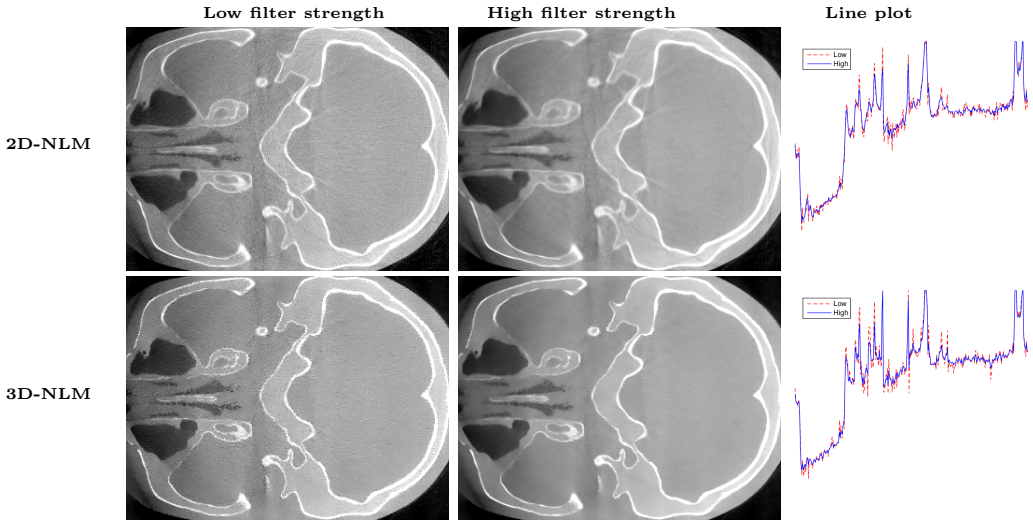


Figure 6.14: Comparison of artifacts and noise reduction using 2D projection NLM and 3D-NLM denoising. The NLM version used here was a Gaussian kernel with the best found settings for the search and patch window. 3D-NLM was used with a patch size of 3^3 and search window of 9^3.

In Figure 6.14 the difference between 2D-NLM and 3D-NLM can be seen for a slice of the head phantom, with 3D-NLM the sharp edges are well kept and there are almost no streaks. Around the edges in the 3D case some noise halos remains. The difference between the denoising methods is very evident for high degrees of denoising where a lot of streaks and blurring appear for the 2D-NLM case, wheres the 3D denoising method produces a smooth result with sharp details and edges.

In Figure 6.15 the difference between 3D and 2D filtering can be seen for a slice of the Catphan phantom. Here there is a very big difference in the preservation of the metal wires as they are smeared to a very low degree in the 3D-NLM case as compared

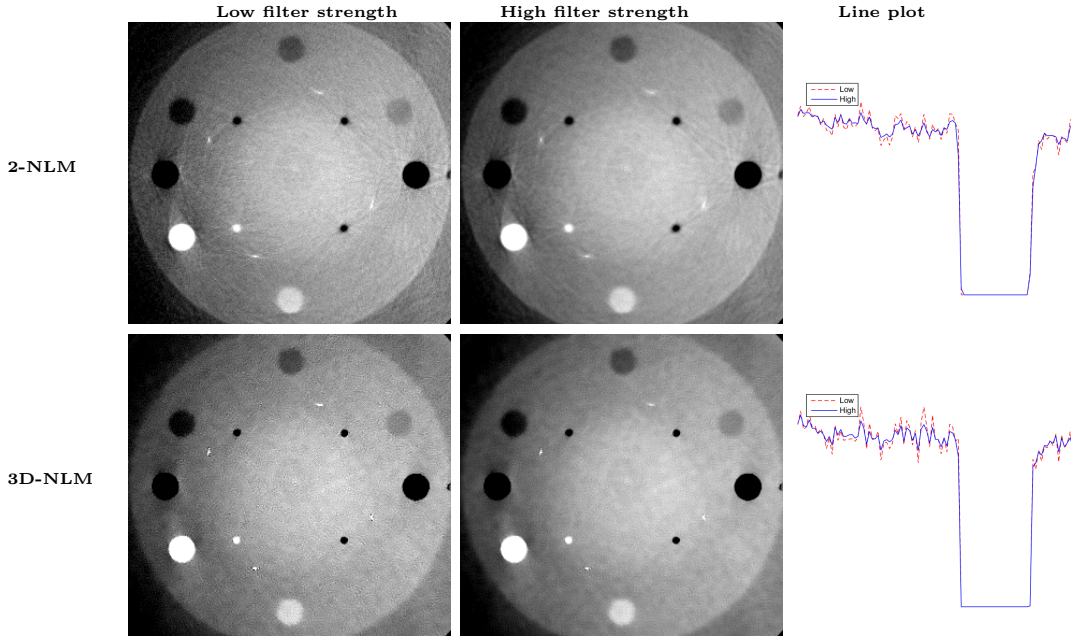
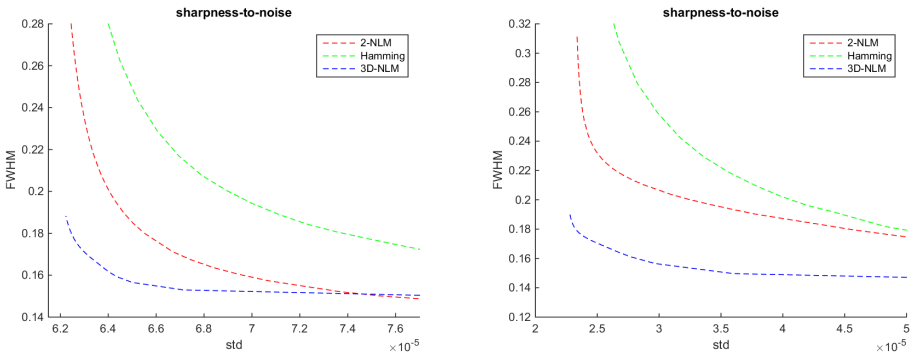


Figure 6.15: Comparison of artifacts and noise reduction in the Catphan phantom using 2D-NLM on the X-ray images and 3D-NLM.

to the 2D-case.



(a) Cylinder nr 2. **(b)** Cylinder nr 1.

Figure 6.16: Comparison of the relation between FWHM and the noise level in the reconstructed volume for 2-NLM which acts on the projection images and 3D-NLM.

The difference between 3D-NLM and 2D-NLM in the sharpness-to-noise ratio can be seen in Figure 6.16, for cylinder nr 1 the difference is very big even at low degrees of filtering this is probably due to the presence of streak artifacts around this cylinder in the 2D-NLM case. From the sharpness-to-noise plot it is clear that with 3D-NLM higher levels of noise reduction is possible while keeping edges intact.

3D-NLM seem to perform best with a small patch size, 3^3 was found to be the best

at preserving fine details and edges. Even with a small search window size such as 5^3 the result is better than 2D-NLM. A summary of timings for a few different search sizes can be seen in Table 6.4. When the search window size is increased noise halos around edges are decreased and fine details become more visible. There is a an advantage of a large search window but the difference becomes smaller for each increment in window size.

Search Size	Runtime (seconds)
5^3	1
7^3	3
9^3	5
11^3	11

Table 6.4: Runtimes for 3D-NLM with varying search window size. The patch size was fixed to 3^3 and the volume size was 448^3. The denoising was performed on a GeForce GTX970 graphics card.

Improved denoising results can be achieved with a runtime of only a few seconds which makes this algorithm useful in practice, at least at the currently used volume resolution of 448^3 voxels.

6.15 Summary NLM results

The NLM denoising algorithm lead to much sharper high contrast edges than the currently used frequency domain apodization filter. However, NLM in combination with FDK also introduces artifacts, the most notable being streaks protruding from sharp edges. Low contrast objects and fine details are smeared even at low degrees of filtering, and when increasing the filter strength these effects distort the image to a large degree.

The streaks seen in the volume are caused by the fact that denoised edges have an irregular shape and the fact that the same edge is denoised differently in different images in the X-ray image sequence. Non-local patch based methods depend on the neighborhood of a pixel when denoising it, in an X-ray scan edges will look different from different angles and will thus be denoised differently. This becomes a problem in the FDK backprojection since if the sharpness of the edge varies greatly the back projections of it will not cancel each other to form the object, instead streaks will remain. No solution has been found to completely remove the streaks when denoising the projection images.

Applying a 3D version of NLM to the reconstructed volume directly seem to offer the best results out of the methods tested here. Compared to what seems to be possible to do in the 2D domain, the 3D filtering offer a much higher reduction of noise, sharper edges, less artifacts, and better preserved details.

7

Comparison with current CBCT denoising methods

The purpose of this chapter is to briefly introduce current denoising methods that are used with CBCT and compare these with the best NLM version from the previous chapter. The aim is to see if NLM can compete with these denoising methods and how the results differ. The best NLM versions were selected as 2-NLM, NLM with a Gaussian kernel using 5 images in total, and 3D-NLM which filters the volume instead of the projection images.

7.1 Hamming filter

The Hamming filter is a low-pass filter in the frequency domain, that smooths high frequencies with a cutoff for all frequencies above some normalized frequency $f_c \in (0, 1.0)$. This filter is commonly used with FDK reconstruction algorithm to reduce noise. The Hamming filter multiplies the ramp filter by a Hamming window.

7.2 Wiener filter

Wiener filtration is also currently used for CBCT. The Wiener filter uses a neighborhood of the pixel being denoised from which the mean and variance is estimated, these statistics are then used to estimate the pixel value. The Wiener filter has a much lower computational complexity than NLM. The drawback with this filter is that it is not edge preserving. The wiener filter is, like the Hamming filter, a low-pass filter.

The Wiener filter was tested using the function **wiener2** in MATLAB. Since the filter works best for Gaussian noise a VST is used before the images are denoised. **wiener2** has two parameters namely the size of the filter and the estimated noise level.

The **wiener2** function estimates the mean $\mu(i)$ and variance $\sigma(i)^2$ in a $R \times R$ window around pixel i. These parameters are then used to estimate the noise-free pixel value in pixel i as

$$\hat{v}(i) = \mu(i) + \frac{\sigma(i)^2 - u^2}{\sigma(i)^2}(v(i) - \mu(i)). \tag{7.1}$$

Where $\hat{v}(i)$ is the estimated pixel, $v(i)$ the original pixel value at pixel i and u^2 is the noise variance of the image, if this is not known it is estimated as the average of all the local variances $\sigma(i)^2$ in the image. This filter has a low computational complexity and uses less information about the local neighborhood than NLM does.

7.3 Comparison projection images

In Figure 7.1 ramp filtered projection images are shown denoised with the current methods and NLM. The Hamming and Wiener filter produces a much smoother and uniform noisy image while NLM has smooth and noisy areas which is related to the number of matching patches found in each pixel.

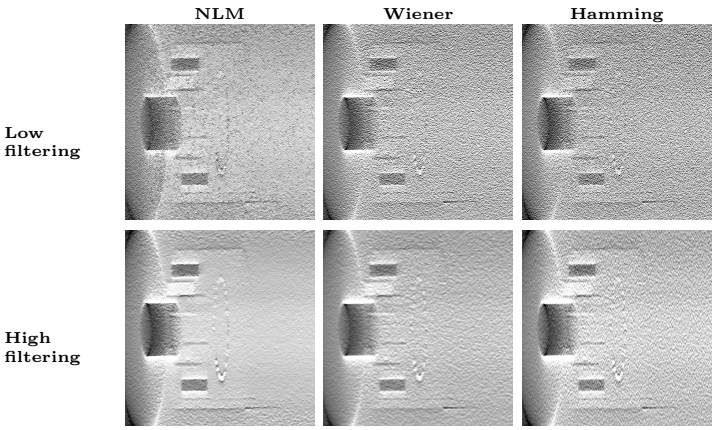


Figure 7.1: Comparison of ramp filtered projection images for two different levels of filtering for Hamming, Wiener and NLM denoising.

7.4 Comparison 3D reconstruction

A visual comparison of the sharpness-to-noise ratio for the different denoising methods can be seen in Figure 7.2, where a part of the skull bone is shown with varying filter strength. Here it is clearly seen that NLM preserves sharp edges when reducing the noise level to a higher degree than Wiener and Hamming does. Wiener is however a significant improvement over Hamming. When using the NLM filter the peaks that correspond to the skull bone are kept almost completely intact even though the noise is reduced by a large factor. The difference between 2D-NLM and 3D-NLM is in this case quite small,

but there are some streaks present in the 2D-NLM case which can not be seen in the 3D-NLM result.

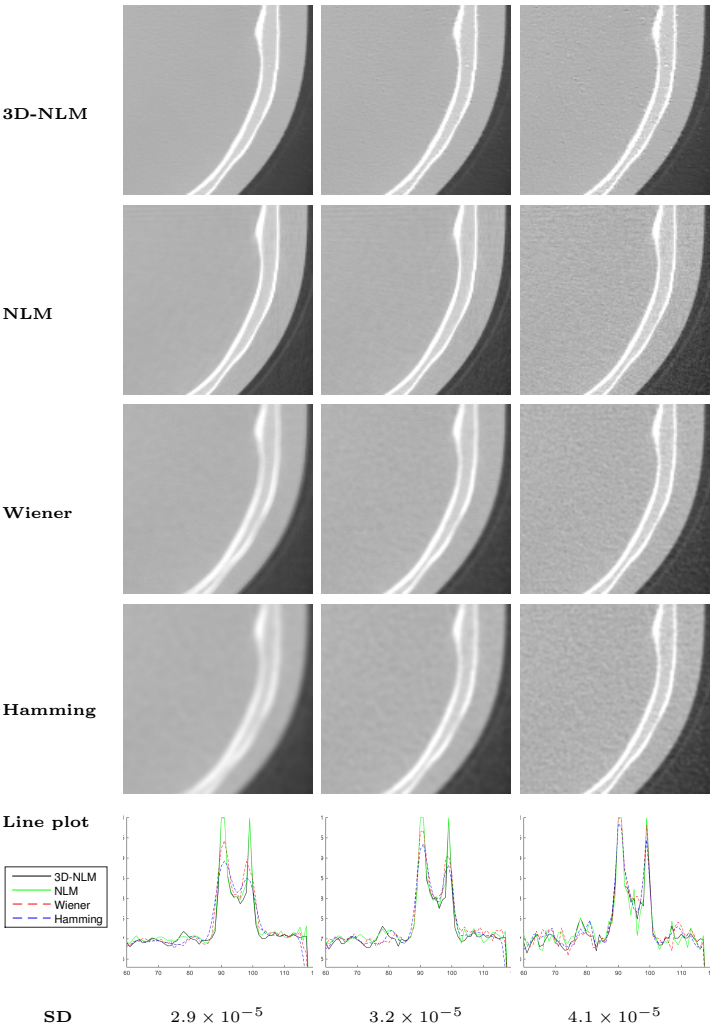
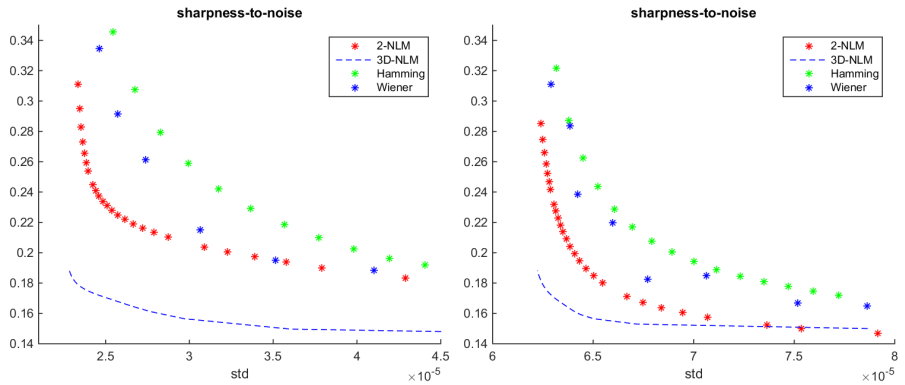


Figure 7.2: Comparison of part of the skull bone in the head phantom for different degrees of filtering. Per column all volumes have similar standard deviation measured in several homogeneous sub-volumes in the head. The line plot shows part of the first row in the image, the peaks corresponds to the skull bone.

In Figure 7.3 the sharpness-to-noise ratio for Hamming, Wiener and NLM is compared measuring the sharpness of a cylinder edge in the Catphan volume. Here both NLM versions perform better than Hamming and Wiener filtration.

Looking at a part inside the skull of the head phantom that contains bone and lower contrast objects, it is clear that NLM preserves smaller bone details better than the other two filters. Lower contrast details are also sharper as seen in Figure 7.4. The result using 3D-NLM is however significantly different in appearance compared to the result from the three different projection image filters, in this case sharp edges as well



(a) Circle 1, highest contrast circle. **(b)** Circle 2, next highest contrast.

Figure 7.3: Sharpness-to-noise plot showing the relation between FWHM and the noise level in the reconstructed volume for the best NLM versions and all other tested filtration methods. The values are estimated from a low-dose Catphan phantom.

as fine lower contrast objects are kept intact as the noise is reduced.

Using the Hamming or the Wiener filter does not give any streak artifacts in the Catphan phantom as seen in Figure 7.5, the volume is much more homogeneously denoised than in the 2D-NLM case where sharp edges are kept intact. 3D-NLM manages to keep the shape of all details intact, induces no streaks, and has the highest sharpness of all the denoising methods.

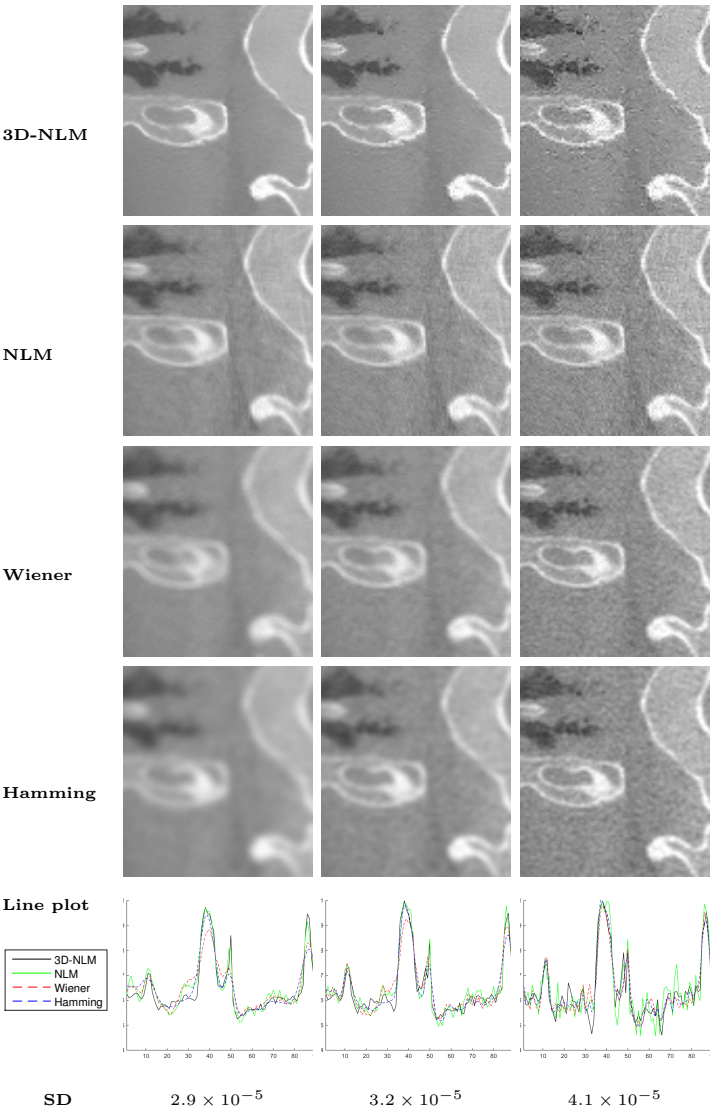


Figure 7.4: Comparison of an internal part of the volume in the head phantom for different degrees of filtering. Per column all volumes have the same variation measured as the standard deviation in several homogeneous sub-volumes in the head. The line plot shows the center row in the image.

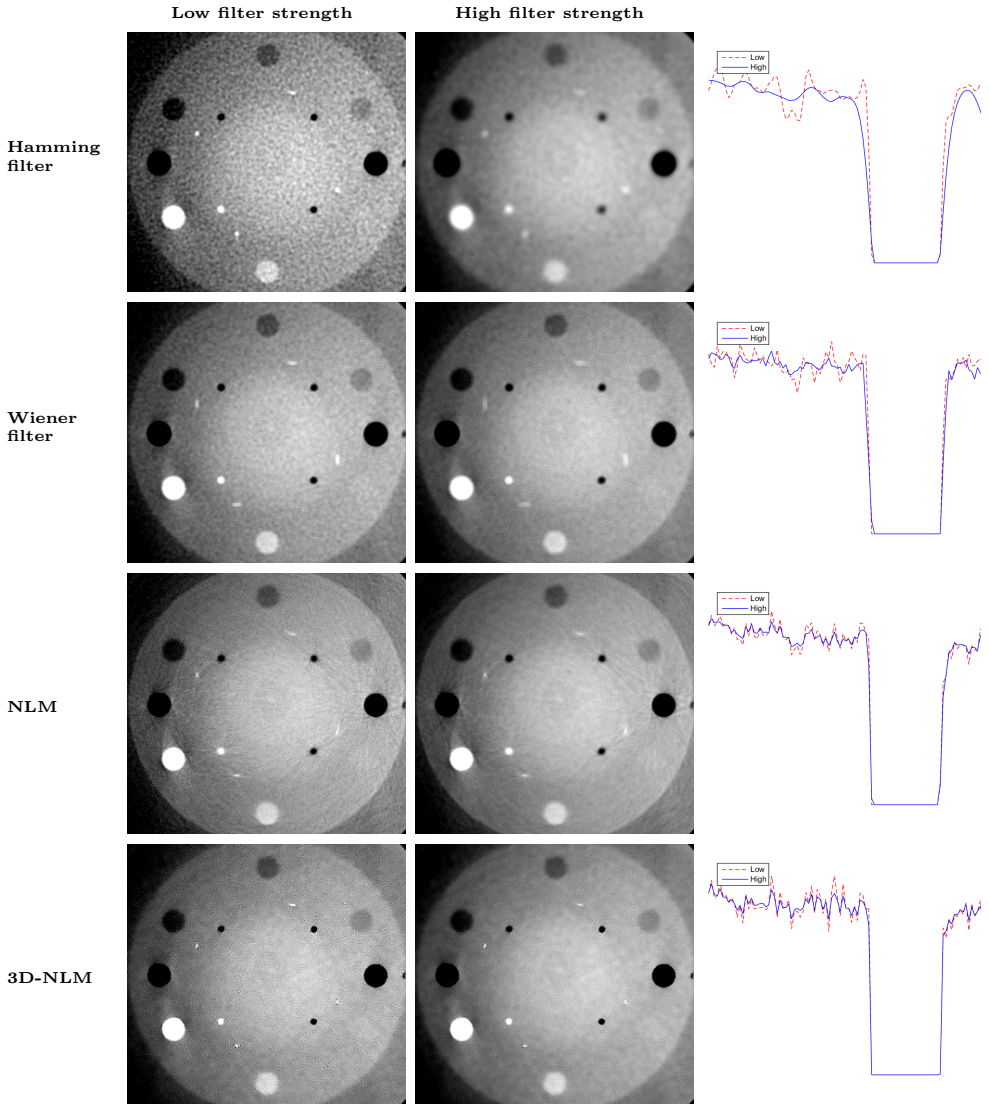


Figure 7.5: The difference between the Hamming, Wiener and NLM denoising results for the Catphan volume. The line plot shows a line trough one of the high contrast cylinders.

7.5 Quality comparison between low and high dose CBCT

Since it is preferable to use as low radiation dose as possible, it would be beneficial if a more advanced denoising method could offer the same image quality at a lower dose compared to standard filters. Here the comparison is between NLM, 3D-NLM, and the currently used Hamming filter.

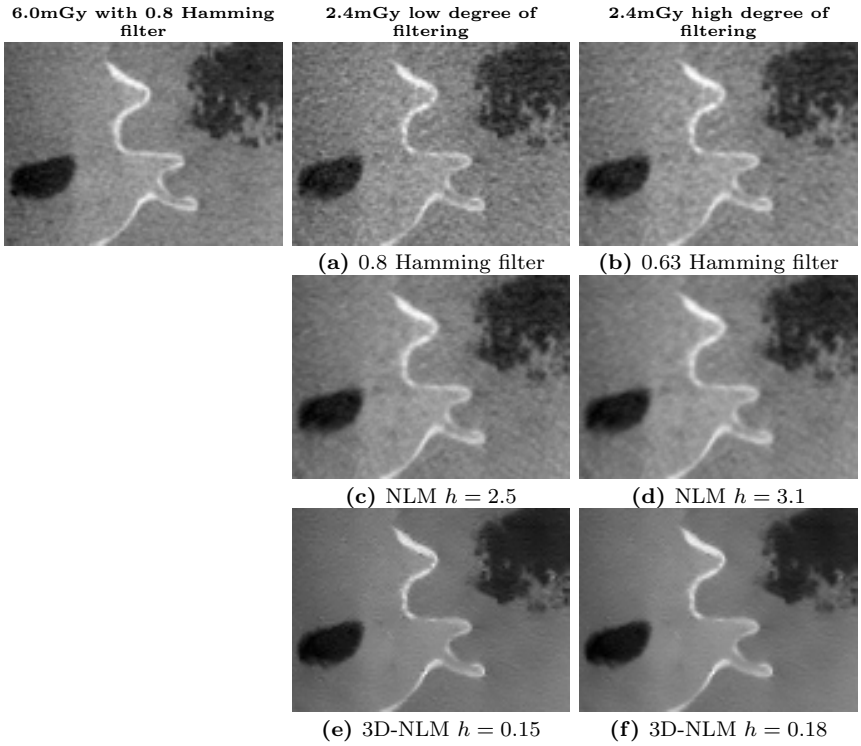
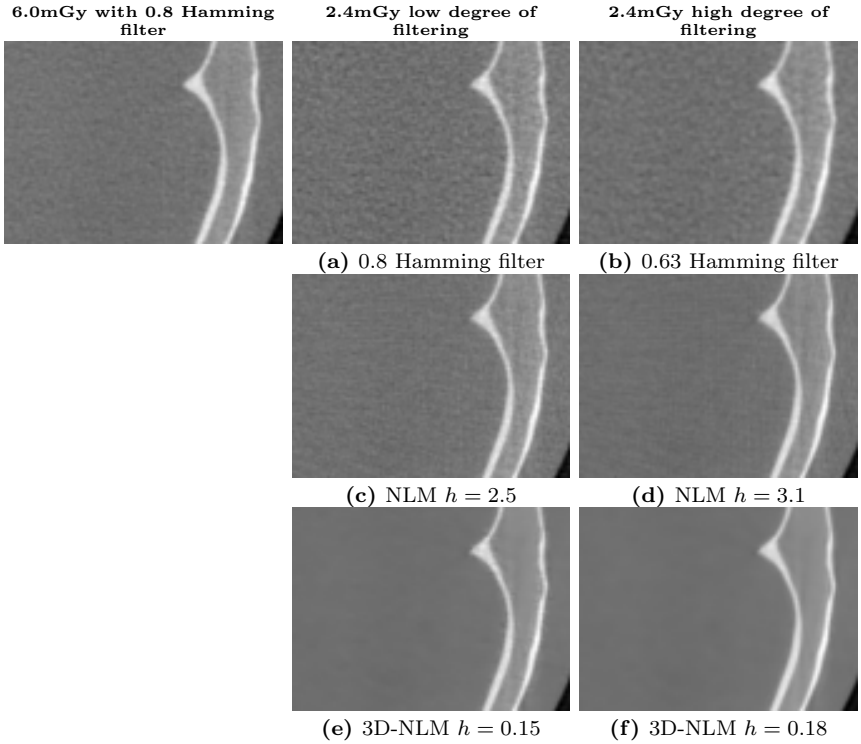


Figure 7.6: Comparison of a part of the head phantom using high versus low radiation dose. The high dose image is compared to low dose images with two different levels of Hamming filtering, NLM filtering and 3D-NLM filtering.

In Figure 7.6 a part of the head phantom from a 6.0mGy scan is compared to a 2.4mGy scan using different filters. For small details with low contrast the high dose image is slightly clearer, but the NLM filtered versions in Figure 7.6c and 7.6e comes quite close to the high dose counterpart. The 3D-NLM filtered version has a much lower noise level and is much smoother than the high dose volume, and about the same amount of fine details are visible.

High contrast objects such as skull bone is preserved well when filtering with NLM. It is possible to remove most of the noise and still keep the skull intact. The image quality in this case is comparable to the high dose image as seen Figure 7.7f. This is especially true for the 3D-NLM denoised volume where the noise can be reduced further than with 2D-NLM.

The spatial resolution of a CT system is often tested using the line pair module in



(a) 0.8 Hamming filter **(b)** 0.63 Hamming filter

(c) NLM $h = 2.5$ **(d)** NLM $h = 3.1$

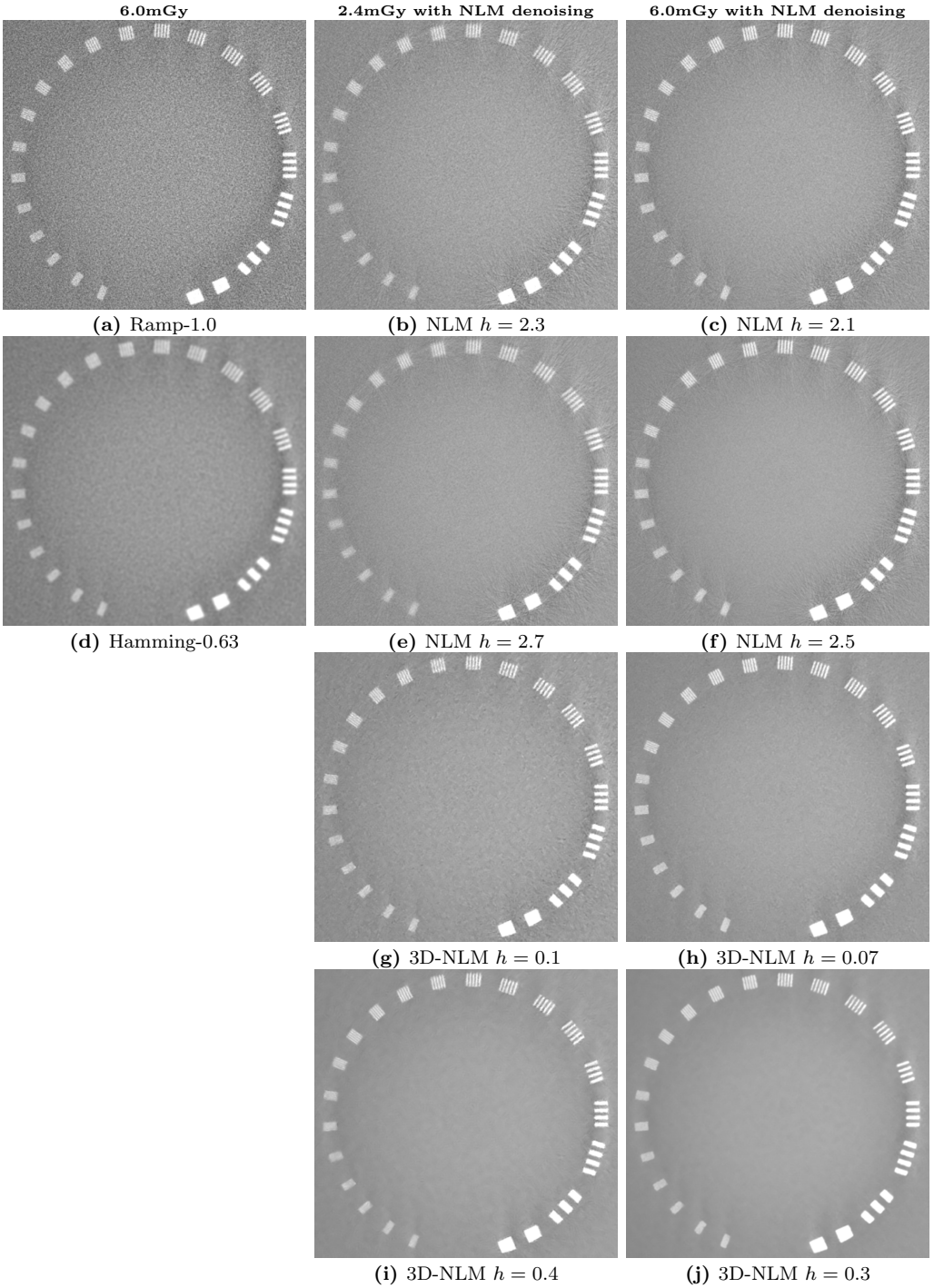
(e) 3D-NLM $h = 0.15$ **(f)** 3D-NLM $h = 0.18$

Figure 7.7: Comparison of a part of the skull bone using high versus low radiation dose.

the Catphan phantom. Higher spatial resolution means more separable line pairs are visible. A comparison between a high dose scan and a low dose one can be seen in Figure 7.8. The same number of line pairs are visible in the 2D-NLM filtered low dose and the unfiltered high dose image, but the pairs are blurrier and have lower contrast in the low dose case. Comparing the high dose scan with the 3D-NLM filtered volume instead the result is very similar. The smallest line pairs are however slightly clearer in the high dose scan.

When denoising the high dose scan with the currently used Hamming-0.63 filter there are more line pairs visible in all of the low dose volumes that are filtered using NLM. In Figure 7.8i the noise level is even lower than in the Hamming filtered high dose scan (Figure 7.8d) and more line pairs are separable. This indicates that by using 3D-NLM the dose to the patient could be lowered by as much as half while maintaining the same quality.

The 2D-NLM and 3D-NLM denoised volumes have similar spatial resolution but there is a large difference in the contrast of the line pairs and the presence of streak artifacts.



(a) Ramp-1.0 **(b)** NLM $h = 2.3$ **(c)** NLM $h = 2.1$

(d) Hamming-0.63 **(e)** NLM $h = 2.7$ **(f)** NLM $h = 2.5$

(g) 3D-NLM $h = 0.1$ **(h)** 3D-NLM $h = 0.07$

(i) 3D-NLM $h = 0.4$ **(j)** 3D-NLM $h = 0.3$

Figure 7.8: Comparison of spatial resolution using the line pair module in the Catphan volume. In the first column a high dose scan is shown with no denoising (ramp filter with no cutoff) and the currently used Hamming filter. The next column shows a low dose scan denoised using NLM and 3D-NLM at two different denoising strengths. In the third column the high dose scan is shown with NLM and 3D-NLM denoising.

8

Comparison with BM3D

The purpose of this section is to briefly introduce the state-of-the-art denoising algorithm BM3D and evaluate the performance of the best found NLM version in comparison to this algorithm. This comparison should give an idea of how much better results it is possible to achieve when denoising the projection images in a CBCT scan.

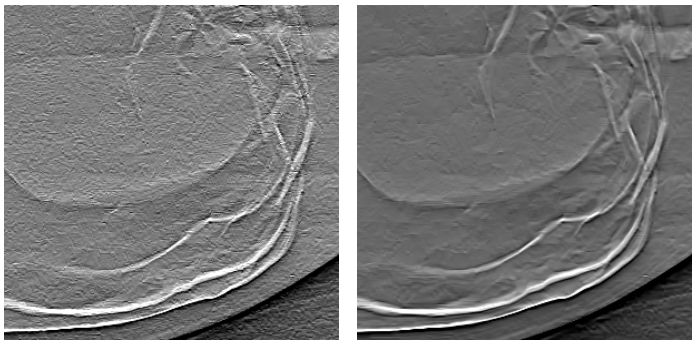
8.1 BM3D

Block matching 3 dimensional (BM3D) is a more advanced non-local denoising algorithm than NLM consisting of several steps. BM3D uses grouping of non-local patches with 3D transform domain filtering and Wiener filtering [6].

First similar patches are grouped together, by comparing patch-wise distances. The groups form 3D blocks which are hard-thresholded in the 3D transform domain. The filtered blocks are aggregated by a weighted averaging of the block-wise estimates. The result from the first step is then used as an estimate of the noise free image in a Wiener filtering step of 3D blocks. The resulting filtered 3D blocks are then aggregated by weighted average creating the final denoised image.

8.2 BM3D results

BM3D is suited for Gaussian noise therefore a VST transform is used before the images are denoised. There are many parameters and settings in the BM3D algorithm, here only the parameter σ that varies with the noise level is adjusted taking the standard settings for all other parameters as provided in the code from [24].



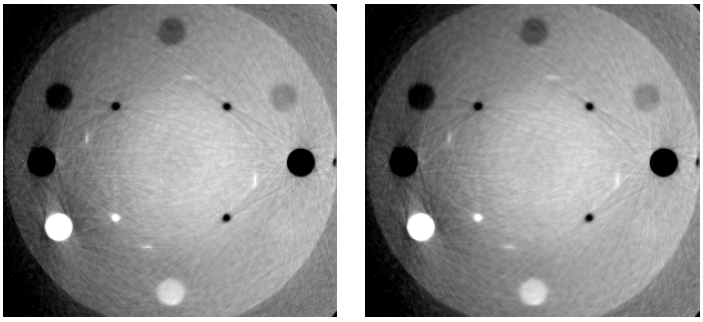
(a) BM3D with $\sigma = 0.5$ **(b)** BM3D with $\sigma = 1.0$

Figure 8.1: Part of a projection image of the head phantom at the dose 2.4mGy. The heavy filtered image is smooth and the less filtered has some patterns in homogeneous areas, similar to the NLM filtered projection images.

Performing the denoising and looking at the projection images BM3D shows similar artifacts as NLM such as patterns in noisy areas and noise halos around sharp edges. The patterns and noise left after BM3D filtering is however different in shape and structure.

Applying BM3D to the projection images gives a reconstructed result as seen in Figure 8.2, which is very similar to the NLM case. There are streak artifacts present from sharp edges and small details are smeared such as the metal wires The relation between noise reduction and presence of artifacts seem to be very similar to NLM.

BM3D



(a) $\sigma = 2.0$ **(b)** $\sigma = 3.0$

Figure 8.2: Slice of the reconstructed volume of a Catphan phantom for different levels of filtering, also for BM3D streak artifacts become more dominant with greater filtering.

Considering some parts of the skull, as seen in Figure 8.3 and 8.4, the results compared to NLM are also very similar. Sharp edges are kept intact and there remains very low levels of noise in homogeneous areas. Low contrast objects and small details are smeared.

8.3 Conclusion of comparison

NLM and BM3D give very similar results. BM3D suffer from the same types of artifacts as NLM does, the random and non-random streaks between objects. These streaks might

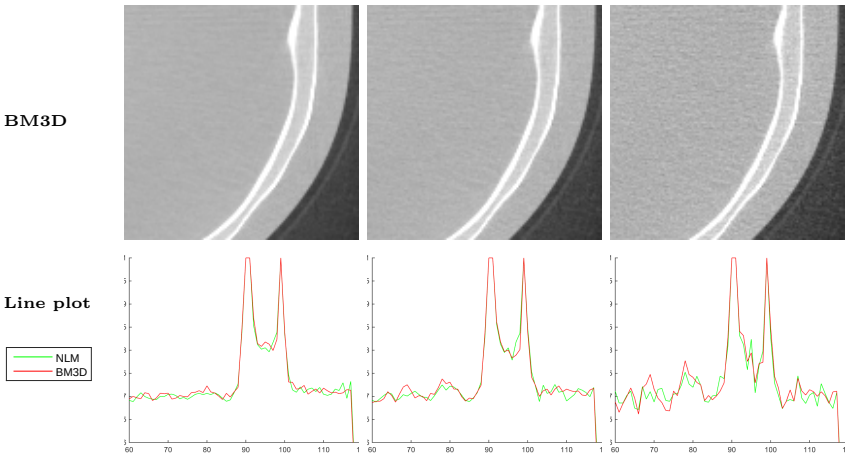


Figure 8.3: Comparison of denoising results for skull bone in the head phantom. In the line plot a row in the middle of the image is compared to a NLM denoised volume with the same standard deviation in uniform regions.

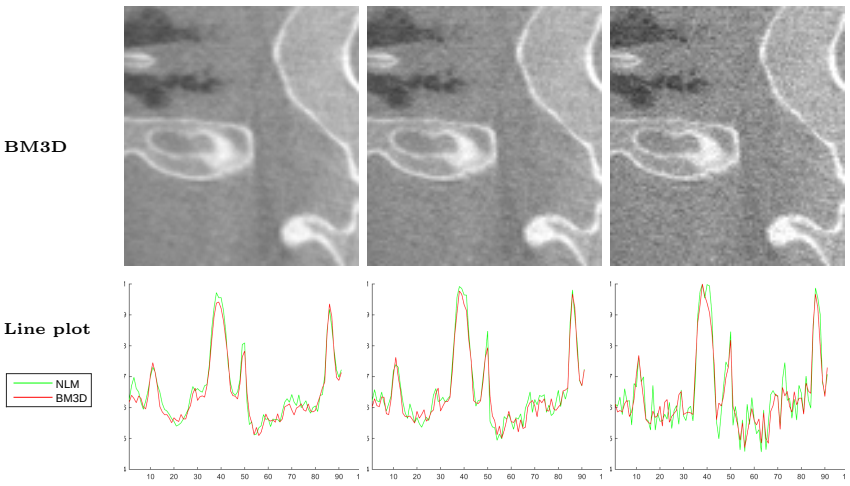


Figure 8.4: Comparison of denoising results for bone and low contrast details in the head phantom.

be even more prominent in the BM3D case. BM3D was not used with adjacent images which could have improved the results.

Just as there was very little variation among the different NLM versions, there is also a very small difference between NLM and BM3D. From this comparison there seem to be little gain, compared to NLM, in using the more advanced BM3D algorithm on the X-ray images.

9

Conclusion

NLM was found to be significantly better than current low-pass filters at reducing noise while preserving edges. There were also several issues identified, such as smearing of fine details and streak artifacts in the reconstructed volume. The performance of the different NLM versions tested only varied slightly, and no significant improvement upon the standard algorithm was found. The state-of-the-art denoising algorithm BM3D performs significantly better than NLM in terms of PSNR on natural images, but the reconstruction results were not noticeably improved. BM3D also introduces artifacts in the volume.

In general it seems as neighborhood dependent edge preserving denoising methods are not ideal to use with FDK reconstruction. FDK was found sensitive to the varying neighborhood dependent smoothing of the same edge throughout the X-ray image sequence.

Performing the NLM denoising in the reconstructed 3D volume instead leads to very different results compared to denoising the X-ray images. The resulting volume is sharper, details are better preserved, and no streak artifacts are introduced. The runtime for 3D-NLM is only a few seconds and makes the algorithm useful in practice. However, a big disadvantage with this approach when it comes to runtime is that it is not able to run while the X-ray scanning is taking place since it depends on the final result of the reconstruction.

In conclusion, the best denoising results were obtained with 3D-NLM applied to the reconstructed volume. 3D-NLM can reduce the noise such that a low dose scan is comparable to a high dose scan denoised with current CBCT denoising methods. A comparison of spatial resolution and preservation of details indicates that the dose could be lowered by as much as half without significantly lowering the quality.

9.1 Further work

There is probably not that much more that can be done with NLM for projection images since all the different approaches tried gave quite similar results. Even the significantly better denoising method BM3D did not improve the reconstructed volume quality significantly. It is likely that all patch-based denoising methods working on the projection images will have similar issues as BM3D and NLM, since the denoising will depend on the neighborhood which will change during the sequence of scan images and is then likely to cause streak artifacts.

The results from the 3D filtering indicates that there is an advantage of denoising the volume where all the information from the projection images is collected in the volume voxels. Denoising in this domain will not cause any FDK induced artifacts. Replacing the 3D-NLM denoising with a 3D version of a state-of-the-art denoising method would probably improve the results noticeably. This improvement in quality will however probably come at the cost of an increased runtime.

Bibliography

[1] F Edward Boas and Dominik Fleischmann. CT artifacts: causes and reduction techniques. *Imaging in Medicine*, 4(2):229–240, 2012.

[2] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

[3] Harold Christopher Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2392–2399. IEEE, 2012.

[4] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[5] Yang Chen, Wufan Chen, Xindao Yin, Xianghua Ye, Xudong Bao, Limin Luo, Qianjing Feng, Xiaoe Yu, et al. Improving low-dose abdominal CT images by weighted intensity averaging over large-scale neighborhoods. *European Journal of Radiology*, 80(2):e42–e49, 2011.

[6] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising with block-matching and 3D filtering. In *Electronic Imaging 2006*, pages 606414–606414. International Society for Optics and Photonics, 2006.

[7] Fredrik Dacke. Non-local means denoising of projection images in cone beam computed tomography. 2013.

[8] C-A Deledalle, Florence Tupin, and Loïc Denis. Poisson NL means: Unsupervised non local means for Poisson noise. In *Image processing (ICIP), 2010 17th IEEE international conference on*, pages 801–804. IEEE, 2010.

[9] Joshua D Evans, David G Politte, Bruce R Whiting, Joseph A O'Sullivan, and Jeffrey F Williamson. Noise-resolution tradeoffs in x-ray CT imaging: A comparison of

penalized alternating minimization and filtered backprojection algorithms. *Medical physics*, 38(3):1444–1458, 2011.

[10] LA Feldkamp, LC Davis, and JW Kress. Practical cone-beam algorithm. *JOSA A*, 1(6):612–619, 1984.

[11] JC Ramirez Giraldo, Zachary S Kelm, Luis S Guimaraes, Lifeng Yu, Joel G Fletcher, Bradley J Erickson, and Cynthia H McCollough. Comparative study of two image space noise reduction methods for computed tomography: bilateral filter and non-local means. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 3529–3532. IEEE, 2009.

[12] Kristine Gulliksrud, Caroline Stokke, and Anne Catrine Trægde Martinsen. How to measure CT image quality: Variations in CT-numbers, uniformity and low contrast resolution for a CT quality assurance phantom. *Physica Medica*, 30(4):521–526, 2014.

[13] Kuidong Huang, Dinghua Zhang, and Kai Wang. Non-local means denoising algorithm accelerated by GPU. In *Sixth International Symposium on Multispectral Image Processing and Pattern Recognition*, pages 749711–749711. International Society for Optics and Photonics, 2009.

[14] Xun Jia, Zhen Tian, Yifei Lou, Jan-Jakob Sonke, and Steve B Jiang. Four-dimensional cone beam CT reconstruction and enhancement using a temporal non-local means method. *Medical physics*, 39(9):5592–5602, 2012.

[15] Zachary S Kelm, Daniel Blezek, Brian Bartholmai, and Bradley James Erickson. Optimizing non-local means for denoising low dose CT. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*, pages 662–665. IEEE, 2009.

[16] M Lebrun, M Colom, Antoni Buades, and JM Morel. Secrets of image denoising cuisine. *Acta Numerica*, 21:475–576, 2012.

[17] Mona Mahmoudi and Guillermo Sapiro. Fast image and video denoising via nonlocal means of similar neighborhoods. *Signal Processing Letters, IEEE*, 12(12):839–842, 2005.

[18] VP Raj and T Venkateswarlu. Denoising of magnetic resonance and X-ray images using variance stabilization and patch based algorithms. *The International Journal of Multimedia & Its Applications (IJMA)*, 4(6):53–71, 2012.

[19] Jie Ren, Yue Zhuo, Jiaying Liu, and Zongming Guo. Illumination-invariant non-local means based video denoising. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 1185–1188. IEEE, 2012.

[20] Samuel Richard, Daniela B Husarik, Girijesh Yadava, Simon N Murphy, and Ehsan Samei. Towards task-based assessment of CT performance: system and object MTF across different reconstruction algorithms. *Medical physics*, 39(7):4115–4122, 2012.

[21] Joseph Salmon and Yann Strozecki. Patch reprojections for non-local methods. *Signal Processing*, 92(2):477–489, 2012.

[22] Chris C Shaw. *Cone Beam Computed Tomography*. Taylor & Francis Group, 2014.

[23] Camille Sutour, Charles Deledalle, and J Aujol. Adaptive regularization of the NL-means: Application to image and video denoising. 2014.

[24] Tampere University Of Technology. BM3D MATLAB software. `http://www.cs.tut.fi/~foi/GCF-BM3D/index.html#ref_software`. Accessed: 2015-05-02.

[25] The Phantom Laboratory. Catphan® 503 Manual. `http://www.phantomlab.com/library/pdf/catphan503manual.pdf`. Accessed: 2015-05-02.

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

[27] Hao Zhang, Jianhua Ma, Jing Wang, Yan Liu, Hongbing Lu, and Zhengrong Liang. Statistical image reconstruction for low-dose CT using nonlocal means-based regularization. *Computerized Medical Imaging and Graphics*, 38(6):423–435, 2014.