EEE473-573 Medical Imaging Term Project Report

Image Noise Variance Analysis and Simulation of CT Imaging

Altan Akat [21602035] and Saltuk B. Çelik [21703311]

Abstract: In this report, we start by laying out the fundamentals of computational tomography (CT) imaging. We discuss the effect of Poisson nature of x-rays in CT and approximately calculate the relation of some CT scanning parameters with the image noise variance. To test these relations we perform realistic MATLAB simulations by adding noise to the projections of a 2D chest phantom. Then we apply filtered backprojection to get reconstructed images. We repeat the simulations 10 times for each parameter combination and calculate average variances of the reconstructed images. As a result, we observe that the image variance increases with bandwidth of the rectangular window and decreases with the number of projection angles, physical distance between detectors, photon count.

Video Link: https://www.youtube.com/watch?v=7nk1sMdnplA

1) Introduction

X-ray computed tomography (CT) is a medical imaging technique, in which fan shaped x-ray beam is passed through a patient's body. The intensity of the beam on a horizontal line is measured by an array of detectors. The information gathered by the scanner is like a 1-D projection of a 2-D axial cross section (or a "slice") of the patient's body. The scanner repeats the measurement at multiple angles, and the implementation of this depends on the generation of the scanner. For instance, a third-generation (3G) scanner rotates the source and detector around the body separated by 180°, and a 4G scanner rotates only the source while the entire field of view (FOV) is covered with a large ring of stationary detectors. The collection of the projections of one axial cross section is called its 2-D Radon transform. By applying inverse Radon transform, a 2-D image of the slice can be reconstructed. A standardized technique for this transform is called filtered back projection. Furthermore, it is possible stack 2-D images of the slices together to obtain a 3-D image of the body. In CT imaging, the values for the pixels in the image are calculated with reference to water's attenuation coefficient and are expressed in Hounsfield units (HU) [1], [2].

In real life, due to Poisson nature of x-rays, there is an intrinsic noise in measurement of beam intensity at the CT detector. The photons counted at the detector cells are also Poisson distributed. As a result, the values of pixels of reconstructed image are random variables. The mean and variance of the pixels depend on some parameters such as the bandwidth of the rectangular window of ramp filter, the detector width, the number of projections and the photon count [1]. In this report we investigate the noise in CT imaging. We simulate CT scans of phantom images and obtain experimental results for the effect of mentioned parameters on average variance of the pixels.

2) Methods

2.1) Theoretical calculation of variance due to noise

In this section mostly the calculations from the textbook are followed [1]. If a CT system is assumed to be monoenergetic, a measurement at ith detector and jth angle can be given as

$$g_{ij} = -\ln\left(\frac{N_{ij}}{N_0}\right)$$

where N_0 is the incident photon count, N_{ij} is the detected photon count at that point. N_{ij} and N_0 are related with the equation

$$N_{ij} = N_0 \exp\left(-\int_{L_{ij}} \mu(s)ds\right)$$

where L_{ij} is the ray-path. If we were to repeat a CT measurement on one object, we would get different results each time. This is due to the Poisson nature of the x-ray beams as discussed in the introduction. In reality, the knowledge of N_0 also comes from a CT measurement when the scanner is empty; therefore N_0 should be a Poisson random variable like N_{ij}. However the ratio of two Poisson random variables has no defined distribution and one should consider truncated Poisson distributions, which always takes on positive values [5]. Because the calculations of mean and variance of that ratio is exhaustive, we continue with the assumption that N_0 is constant in this section.

Consequently the measurement g_{ij} is also a random variable. Because N_0 is large, the mean and variance of g_{ij} can be approximated as

$$\bar{g}_{ij} \approx -\ln\left(\frac{\bar{N}_{ij}}{N_0}\right), \qquad var(g_{ij}) \approx \frac{1}{\bar{N}_{ij}}$$

We will assume parallel-ray geometry in reconstruction of the linear attenuation coefficients. The discrete approximation to the convolution backprojection is given by

$$\mu(x,y) = \frac{\pi T}{M}\sum_{j=1}^{M}\sum_{i=-N/2}^{N/2} g_{\theta_j}\tilde{c}(x\cos\theta_j + y\sin\theta_j - iT)$$

Where M is the number of projections and N is the number of ray-paths per projection. T is the physical spacing between detectors, $g_{\theta_j}(iT) = g_{ij}$ and $\tilde{c}(\cdot)$ is the realizable approximation of the ramp filter. We assume that g_{ij} are independent, and the variance is given by

$$\sigma^2(x,y) = var[\mu(x,y)] = \frac{\pi^2T^2}{M^2}\sum_{j=1}^{M}\sum_{i=-N/2}^{N/2} var[g_{ij}]\left[\tilde{c}\left(x\cos\theta_j + y\sin\theta_j - iT\right)\right]^2$$

$$= \frac{\pi^2T^2}{M^2}\sum_{j=1}^{M}\sum_{i=-N/2}^{N/2} \frac{1}{\bar{N}_{ij}}\left[\tilde{c}\left(x\cos\theta_j + y\sin\theta_j - iT\right)\right]^2$$

Then we make a significant assumption $\bar{N}_{ij} \approx \bar{N}$, that the mean of the number of detected photons is equal everywhere in the image. This is obviously wrong and probably will be the basis of the differences between the calculations and simulations.

$$\sigma_\mu^2 = \frac{\pi^2T^2}{M^2\bar{N}}\sum_{j=1}^{M}\sum_{i=-N/2}^{N/2}\left[\tilde{c}\left(x\cos\theta_j + y\sin\theta_j - iT\right)\right]^2$$

For large M and N we approximate

$$\frac{\pi}{M}\sum_{j=1}^{M} T \sum_{i=-N/2}^{N/2} \left[c\left(x\cos\theta_j + y\sin\theta_j - iT\right)\right]^2 \approx \int_0^{\pi}\int_{-\infty}^{\infty} \left[c\left(x\cos\theta_j + y\sin\theta_j - iT\right)\right]^2 dl\, d\theta$$

$$= \pi\int_{-\infty}^{\infty} [c(l)]^2\, dl = \pi\int_{-\infty}^{\infty} |C(\varrho)|^2\, dl$$

The last equality comes from Parseval's theorem. If a rectangular window with bandwidth ϱ_0 is applied

$$\pi\int_{-\infty}^{\infty} |C(\varrho)|^2\, dl = \pi\int_{-\varrho_0}^{\varrho_0} \varrho^2\, dl = \frac{2\pi\varrho_0^3}{3}$$

Finally the variance (or average variance) of the reconstructed image is

$$\sigma_\mu^2 = \frac{2\pi}{3}\varrho_0^3 T \frac{1}{M}\frac{1}{\overline{N}} \qquad (eq.1)$$

The takeaways from this equation are that the image variance increases with bandwidth of the rectangular window, physical distance between detectors, and decreases with number of projection angles, photon count.

2.2) MATLAB simulations of variance due to noise

The purpose of the simulations is to investigate whether the relations in equation 1 hold correct. Between the calculations and simulations, the main differences are

- We did not simulate the effect of physical distance between detectors because of its difficulty. Instead we incorporated the effect of physical width of the detectors.
- We did not assume equal variance all over the image. The variance is calculated among the simulations for each pixel, and average is taken.
- We took N_0 as a Poisson variable. We could not find a source on average photon count at a CT detector and we decided to give it a value in the range 10^4-10^6.

The simulation steps are shown here in continuous form, but in MATLAB, the built-in discrete functions are used. The first step is applying 2-D Radon transform to the phantom image P

$$g(l,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(x,y)\delta(x\cos\theta + y\sin\theta - l)dxdy$$

which is done in MATLAB by *radon* function for a fixed number of angles N_α. Then the Poisson variables, number of incident photons (N_0) and number of detected photons ($N(l,\theta)$), are randomly created. The mean of N_0 is given arbitrarily as $\overline{N}_0$. The mean of $N(l,\theta)$ is calculated as

$$\overline{N}(l,\theta) = N_0 e^{-g(l,\theta)}$$

The noisy measurement is calculated by taking the logarithm

$$g_{noisy}(l,\theta) = -\ln\left(\frac{N(l,\theta)}{N_0}\right)$$

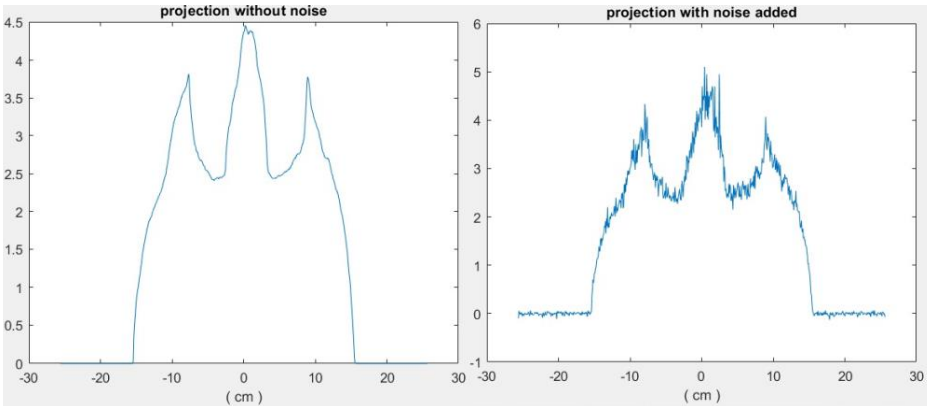

Figure 1: Projection at 0° angle with and without noise

Filtered backprojection method is applied for reconstruction. Firstly a windowed ramp filter with bandwidth ϱ_0 (cm^{-1}) is created in frequency domain.

$$F(\rho) = \begin{cases} |\rho| & |\rho| < \varrho_0 \\ 0 & otherwise \end{cases}$$

The detector effect is also created in frequency domain. The detector width is expressed as d (mm).

$$S(\rho) = dsinc(d\rho)$$

Then 1-D FFT is taken of the noisy measurement

$$G(\rho,\theta) = \int_{-\infty}^{\infty} g_{noisy}(l,\theta)e^{-j2\pi\rho l}dl$$

The filter and detector effects are applied

$$\hat{G}(\rho,\theta) = G(\rho,\theta)F(\rho)S(\rho)$$

The result is converted back to image domain

$$\hat{g}(l,\theta) = \int_{-\varrho_0}^{\varrho_0} \hat{G}(\rho,\theta)e^{j2\pi\rho l}d\rho$$

Lastly, inverse 2-D Radon transform (*iradon*) is applied without filter, because filter was added manually

$$\hat{P}(x,y) = \int_0^{\pi} \hat{g}(x\cos\theta + y\sin\theta\,,\theta)d\theta$$

Following the same procedure, k reconstructed images were created for each combination of $(\bar{N}_0, N_\alpha,\ \varrho_0, d)$ (average number of incident photons, number of projection angles, bandwidth of ramp filter, detector width). Then the variance of a pixel is calculated by

$$var[\hat{P}(x,y)] = \frac{1}{k-1}\sum_{i=1}^{k}\left(\hat{P}_k(x,y) - mean\left(\hat{P}(x,y)\right)\right)^2$$

$$mean\left(\hat{P}(x,y)\right) = \frac{1}{k}\sum_{i=1}^{k} \hat{P}_k(x,y)$$

where $\hat{P}_k(x,y)$ is the kth reconstructed image for the same combination of parameters.

2.3) Dataset

We used The Zubal Phantom data in simulations [3]. This dataset consists of segmented CT torso, head and MRI head slices of two living human males. The images are 512x512 matrices with 1 mm resolution. The segmentation was done on the reconstructed CT images by hand. Even though it is written that "original Hounsfield numbers are also known" in the paper [3], the elements of the reconstructed CT images have values between 0 and 3000. We assumed some kind of linear transformation is applied and tried to reverse it to get the original attenuation coefficients. Using the indices of air, kidney and muscle areas of segmented image of an abdomen slice, the mean values of these areas in the reconstructed CT image are calculated. Then *polyfit* function of MATLAB is used to find the best fit of these values to linear attenuation coefficients of the same tissues from 60 keV to 90 keV [4]. The best fit happened at 85 keV, therefore we assumed that the CT imaging was done at the effective energy of 85 keV. The calculated least-squares fit polynomial coefficients at 85 keV are applied to transform all the reconstructed CT images.

The reconstructed CT images already had blurring effects, so we decided to use a segmented slice as the phantom. The mean attenuation coefficients of the areas in transformed reconstructed CT images are copied to the same areas in segmented images. We decided use a chest slice as the phantom.

Region	Original mean value	Attenuation coeff. (cm^{-1})
outside phantom	59.34167621	0
skin	920.544067	0.156527486
spinal cord	1187.3	0.20500704
spine	1333.534559	0.231585927
rib cage & sternum	1245.188177	0.215528515
skeletal muscle	1057.605685	0.181434427
lungs	355.3499211	0.053799074
heart	935.8944563	0.159312781
esophagus	787.364532	0.132316699
blood pool	1050.165965	0.180082219
bone marrow	1300.352459	0.225554909

Figure 2: The original and transformed values (attenuation coefficients) of regions

The bone regions do not have high attenuation coefficients as expected ($2 \times \mu_{water} \approx 0.35\ cm^{-1}$ at 85 keV). It might be because the density is low, or segmentation was not done very accurately, or some kind of thresholding was applied to bring forward the areas with lower attenuation factor. However as it can be seen in the phantom image, there is a visible difference between gray scale values of those regions, which is enough for our purposes.

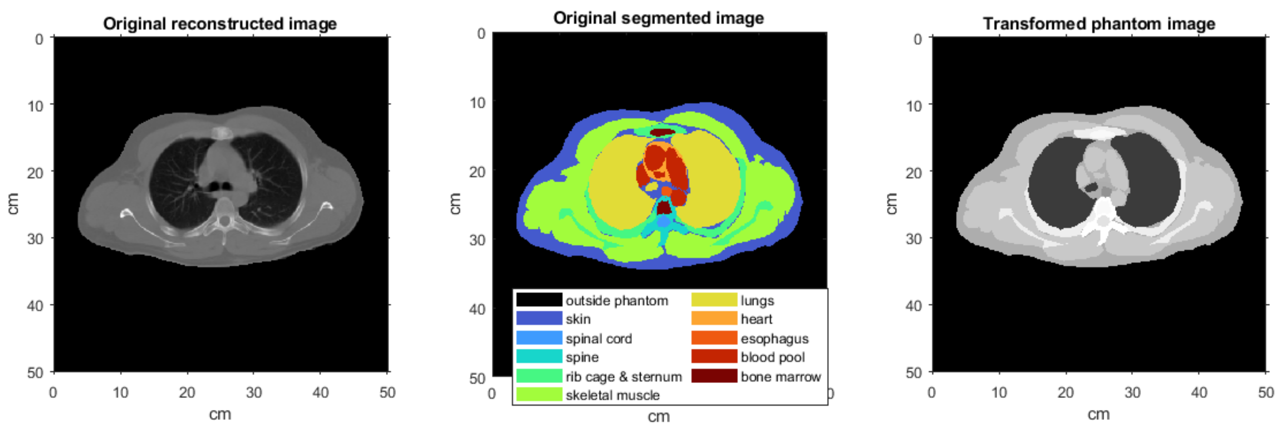

Figure 3: Comparison of the chest images

3) Results

The purpose of this project is to analyze the effects of rectangular window bandwidth, detector width, incident photon number and number of projection angles on the image noise variance in each case just one parameter is changed. Then the reconstructed image and its variance are plotted.

The average variances are calculated with 10 repetitions for each parameter analyses to observe the effect of the noise on the images accurately but without spending much time.

We have put two different images two observe the noise effect on the image. Also, to observe the effects in more detail we have plotted variance plot with different values of the effecting variables. This plot can reveal the information about the noise dependence degree on these factors. We can see whether these factors have linear or exponential effect on the image noise.

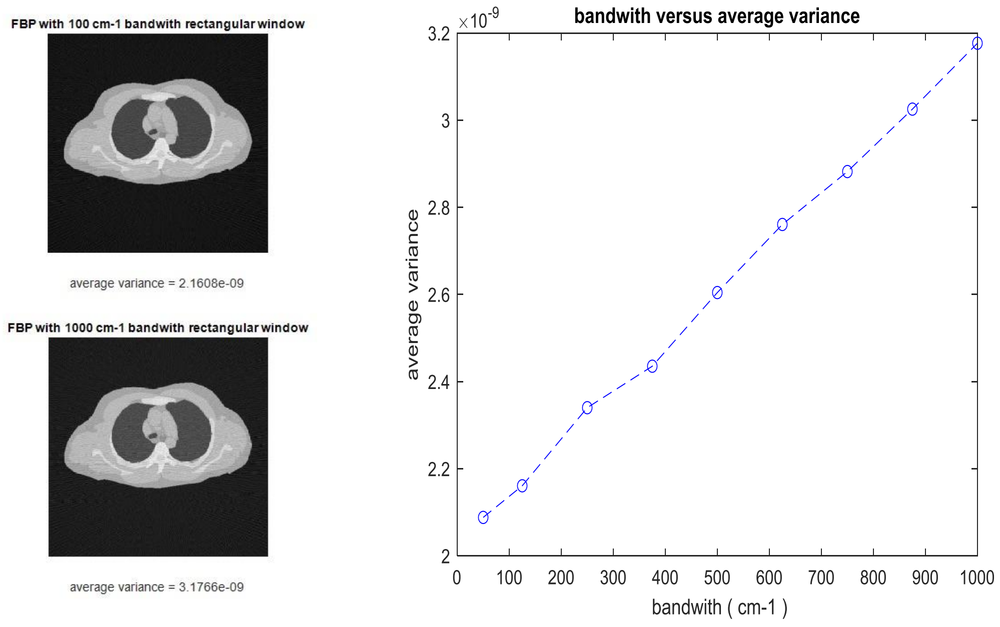

Figure 4: The effect of bandwidth on image quality and variance

In the reconstruction process (Inverse Radon Transform) the projections are filtered (FBP) with a rectangular window. The rectangular window is applied to the projections in the Fourier domain and changing the bandwidth of the rectangular window had a clear effect on the image noise variance. As it can be seen from Figure 4, increasing the bandwidth of the rectangular window increases the image noise variance. The reason is that noise has high frequency components and high bandwidth images have more noise components in real life environment. From the plot we can say that the bandwidth effect has a linear effect on the noise.

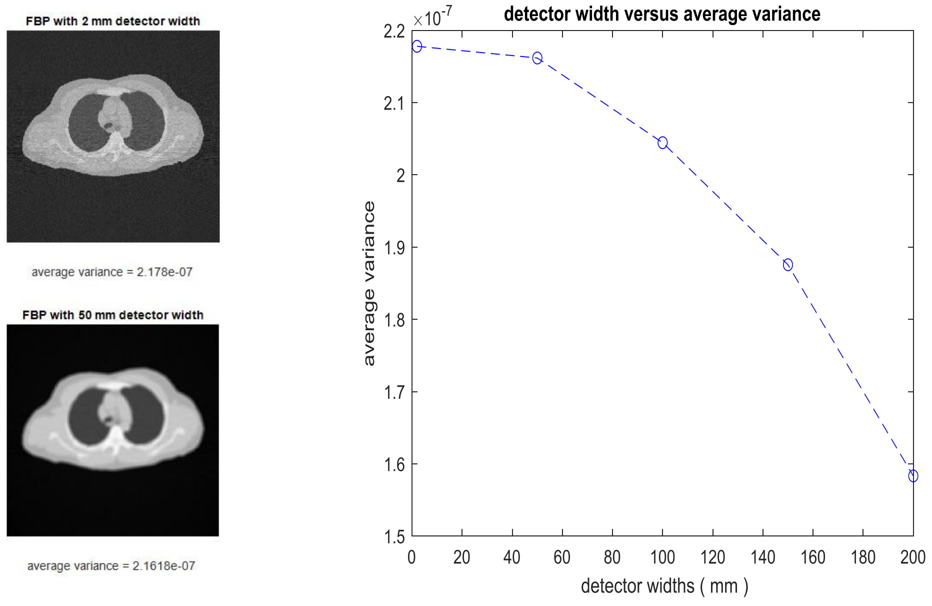

Figure 5: The effect of detector width on image quality and variance

In the simulations the detectors are modelled as small rectangular windows which sum the photons locally. The effect of the detector width is applied to the projections in the Fourier domain. As it can be observed from Figure 5, if the detector width increases the image noise variance decreases rapidly. Having smaller detector width is not good in terms of noise. Since the detector in our model sums the photons locally the noises are also summed and therefore the noise decreases. Also, the blurring effect of the 50mm detector width on the image can be observed clearly in Figure 5 images.

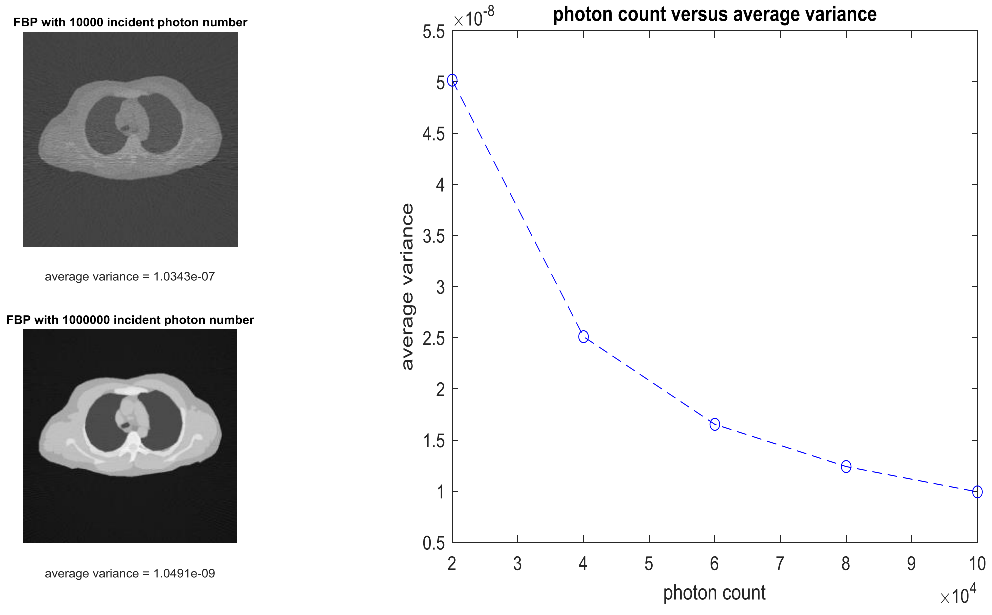

Figure 6: The effect of incident photon count on image quality and variance

The incident photon count is the number of photons which arrive when there is no object in scanner. This value depends on the photon source of the CT scanner. As it can be seen from the Figure 6, when the photon number increases the noise decreases. Due to Compton Effect and other factors some of the photons can't be detected by the detectors. Thus, having a CT beam with more photons is better in terms of noise. Also, from the image quality we can see that the contrast is worse when there is less number of photons. However, having too much photon decreases the contrast due to high brightness.

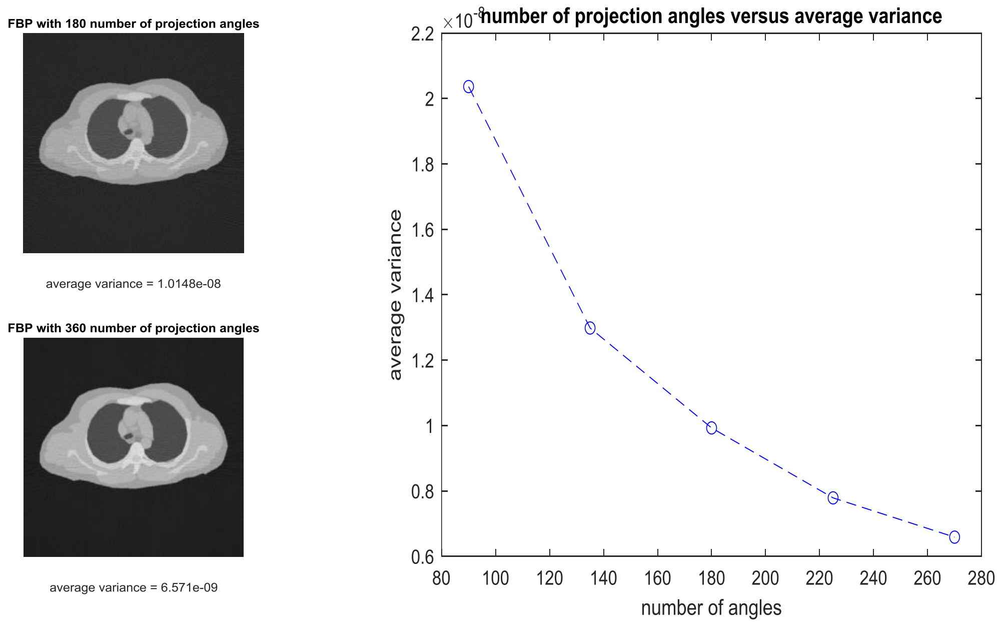

Figure 7: The effect of number of projection angles on image quality and variance

The number of projection angles also affects the noise in the image significantly. The greater number of angles the less noise variance is observed. Although at each projections angle there exist the noise, after reconstruction (Inverse Radon Transform) the effect of noise decreases. Also, we know that the more projection angles we have the more resolution we get. Thus, having enough projection angles is better in terms of noise and resolution.

From the results of the realistic (non-ideal) simulations of the CT image reconstruction process we have clearly observed the effect of rectangular window bandwidth, detector width, incident photon number and number of projection angles on the image noise variance. The results were parallel with theoretical calculations and expressions derived in the methods part. However, some of the effects had slightly less effect on the image noise according to the theoretical expression (eq.1). The effect of the bandwidth of the rectangular window applied to the ramp filter is third degree polynomial according to the theoretical expression (eq.1). However, we obtained a linear relation in the simulations.

As a result, to have a better image in terms of noise we should have the bandwidth of the rectangular window applied to the ramp filter as much as small while not affecting the image quality considering the image in Fourier domain. The width of the detector should be small. There should be sufficiently more number of photons in the CT beam and the number of projection angles should be sufficiently more.

4) Discussion

In this project, we were able to simulate a noisy environment and analyze the effects of rectangular window bandwidth, detector width, incident photon number and number of projection angles on the image noise variance clearly. The results were mostly parallel with the theoretical calculations and from the displayed images and plots we were able to observe the effects and analyze the relation between the non-ideal factors and image noise variance clearly. The only major difference between the simulations and the theoretical expression (eq.1) was the effect degree of the bandwidth of the rectangular window applied to the ramp filter. The reason might be the assumptions made while deriving the theoretical expression (eq.1). Due to the assumptions, the degree of the bandwidth effect might be different from the real-life noisy environment. Therefore, we don't think that our simulation results are totally wrong. However, the real-life noise effect on the images might be different than the Poisson distribution which we used in our simulations. Therefore, the observed noise effect of the factors on the images might be different depending on noise environment and measurement techniques.

References

[1] Prince, Jerry L., and Jonathan M. Links. Medical imaging signals and systems. Upper Saddle River: Pearson Prentice Hall, 2006.

[2] Budoff, Matthew J. "Computed tomography: Overview." *Cardiac CT Imaging*. Springer, London, 2010. 3-20.

[3] Zubal, I.G., Harrell, C.R, Smith, E.O, Rattner, Z., Gindi, G. and Hoffer, P.B., Computerized three-dimensional segmented human anatomy, Medical Physics, 21(2):299-302, 1994.

[4] Böke, Aysun. "Linear attenuation coefficients of tissues from 1 keV to 150 keV." Radiation Physics and Chemistry 102 (2014): 49-59.

[5] Griffin, Tralissa F. Distribution of the ratio of two poisson random variables. Diss. 1992.

Appendix

The MATLAB code

```matlab
clear;
close all;

% create the attenuation coefficient transform polynomials
p = mu_polyfit;
P_orig = createPhantom(p);

%% part 1: bandwith effect on image noise variance
bw = [500 1000 1500 2000 2500];
P_rec_noise1 = zeros(5,514,514);
av_var1 = zeros(1,5);
for i = 1:5
    [P_rec_noise1(i,:,:), av_var1(i)] = bandwith_effect(P_orig,bw(i));
end

figure;
subplot(2,1,1);imshow(P_rec_noise1(1,:,:), []);
title(sprintf('FBP with %.0f cm-1 bandwith rectangular window', bw(1)));
xlabel("average variance = " + av_var1(1));

subplot(2,1,2);imshow(P_rec_noise1(5,:,:), []);
title(sprintf('FBP with %.0f cm-1 bandwith rectangular window', bw(5)));
xlabel("average variance = " + av_var1(5));

%% part 2: detector width effect on image noise variance
d = [2 4 6 8 10];
P_rec_noise2 = zeros(5,514,514);
av_var2 = zeros(1,5);
for i = 1:5
    [P_rec_noise2(i,:,:), av_var2(i)] = detector_effect(P_orig,d(i));
end

figure;
subplot(2,1,1);imshow(P_rec_noise2(1,:,:), []);
title(sprintf('FBP with %.0f mm detector width', 10*d(1)));
xlabel("average variance = " + av_var2(1));

subplot(2,1,2);imshow(P_rec_noise2(5,:,:), []);
title(sprintf('FBP with %.0f mm detector width', 10*d(5)));
xlabel("average variance = " + av_var2(5));

%% part 3: incident photon count effect on image noise variance
N0 = [2 4 6 8 10]*10^5;
P_rec_noise3 = zeros(5,514,514);
av_var3 = zeros(1,5);
for i = 1:5
    [P_rec_noise3(i,:,:), av_var3(i)] = photon_effect(P_orig,N0(i));
end


figure;
subplot(2,1,1);imshow(P_rec_noise3(1,:,:), []);
title(sprintf('FBP with %.0f incident photon number', N0(1)));
xlabel("average variance = " + av_var3(1));
```

```
subplot(2,1,2);imshow(P_rec_noise3(5,:,:), []);
title(sprintf('FBP with %.0f incident photon number', N0(5)));
xlabel("average variance = " + av_var3(5));

%% part 4: number of projection angles effect on image noise variance
angle_num = [100 187 275 362 450];
P_rec_noise4 = zeros(5,514,514);
av_var4 = zeros(1,5);
for i = 1:5
    [P_rec_noise4(i,:,:), av_var4(i)] =
num_angle_effect(P_orig,angle_num(i));
end

figure;
subplot(2,1,1);imshow(P_rec_noise4(1,:,:), []);
title(sprintf('FBP with %.0f number of projection angles', angle_num(1)));
xlabel("average variance = " + av_var4(1));

subplot(2,1,2);imshow(P_rec_noise4(5,:,:), []);
title(sprintf('FBP with %.0f number of projection angles', angle_num(5)));
xlabel("average variance = " + av_var4(5));

%% FUNCTIONS

% part 1: bandwith effect on image noise variance
function [P_rec_noise, av_var] = bandwith_effect(P_orig,bandwith)

    % taking the radon transform of the image
    angles = 0:1:179;
    [g, xp] = radon(P_orig,angles);
    P_run = zeros(10,514,514);
    % 10 independent measurements
    for sa = 1:10
        % adding the poisson noise to the projections at each angle
        N0 = poissrnd(10^5);
        [m,n] = size(g);
        gN = zeros(m,n);
        for i = 1:n
            N = N0 * exp(-g(:,i));
            r = poissrnd(N);
            N = r;
            gN(:,i) = -log(0.0001 + N/N0);   % adding a very small term to
avoid infinite value
        end                                 % does not make a change since
to small

        % applying rectangular window to the ramp filter
        % creating a windowed ramp filter
        freqs=linspace(-1, 1, length(xp))' ;
        ramp_filter = abs(freqs);

        % applying the bandwith to the signals in the fourier domain
        FOV = 51.2;
        b = (length(xp)-1)/2;
        bw = (2*b^2)/FOV;
        index = round(((bw-bandwith)/bw)*183);
        ramp_filter(1:index) = 0;
        ramp_filter(length(xp)-index:length(xp)) = 0;
        ramp_filter = repmat(ramp_filter, [1 length(angles)]);
        % considering the detectors as having width d
        d = 0.1;
        V = sym(d * freqs);
        detector_filter = d * sinc(V) ;
        detector_filter = repmat(detector_filter, [1 length(angles)]);
        detector_filter = double(detector_filter);

        % doing FT domain filtering
        G = fftshift(fft(gN,[],1),1);
        G_filtered = G .* ramp_filter .* detector_filter;

        % taking inverse transform
        g_filtered = ifftshift(G_filtered,1);
        g_filtered = real(ifft(g_filtered,[],1));

        % FBP without any filter since we have applied own filter already
        P_rec_noise = iradon(g_filtered,angles,"linear", 'None');

        % calculating the image noise variance
        P_run(sa,:,:) = P_rec_noise;
    end
    % calculate the variance among 10 measurements, and the average
    % variance among the elements
    av_var = mean(mean(var(P_run)));
end

% part 2: detector width effect on image noise variance
function [P_rec_noise, av_var] = detector_effect(P_orig,detector_width)

    % taking the radon transform of the image
    angles = 0:1:179;
    [g, xp] = radon(P_orig,angles);
    P_run = zeros(10,514,514);

    % 10 independent measurements
    for sa = 1:10
        % adding the poisson noise to the projections at each angle
        N0 = poissrnd(10^5);
        [m,n] = size(g);
        gN = zeros(m,n);
        for i = 1:n
            N = N0 * exp(-g(:,i));
            r = poissrnd(N);
            N = r;
            gN(:,i) = -log(0.0001 + N/N0);    % adding a very small term to
avoid infinite value
        end                                  % does not make a change since
to small

        % applying rectangular window to the ramp filter
        % creating a windowed ramp filter
        freqs=linspace(-1, 1, length(xp))' ;
        ramp_filter = abs(freqs);
        ramp_filter = repmat(ramp_filter, [1 length(angles)]);

        % considering the detectors as having width d
        d = detector_width;
        V = sym(d * freqs);
        detector_filter = d * sinc(V) ;
        detector_filter = repmat(detector_filter, [1 length(angles)]);
        detector_filter = double(detector_filter);

        % doing FT domain filtering
        G = fftshift(fft(gN,[],1),1);
        G_filtered = G .* ramp_filter .* detector_filter;  % adding the
detector window effect
        g_filtered = ifftshift(G_filtered,1);
        g_filtered = real(ifft(g_filtered,[],1));

        % FBP without any filter since we have applied own filter already
        P_rec_noise = iradon(g_filtered,angles,"linear", 'None');

        % calculating the image noise variance
        P_run(sa,:,:) = P_rec_noise;
    end
    % calculate the variance among 10 measurements, and the average
    % variance among the elements
    av_var = mean(mean(var(P_run)));
end

% part 3: incident photon count effect on image noise variance
function [P_rec_noise, av_var] = photon_effect(P_orig,photon_count)
    % taking the radon transform of the image
    angles = 0:1:179;
    [g, xp] = radon(P_orig,angles);
    P_run = zeros(10,514,514);

    % 10 independent measurements
    for sa = 1:10
        % adding the poisson noise to the projections at each angle
        N0 = poissrnd(photon_count);
        [m,n] = size(g);
        gN = zeros(m,n);
        for i = 1:n
            N = N0 * exp(-g(:,i));
            r = poissrnd(N);
            N = r;
            gN(:,i) = -log(0.0001 + N/N0);   % adding a very small term to
avoid infinite value
        end                                 % does not make a change since
to small

        % applying rectangular window to the ramp filter
        % creating a windowed ramp filter
        freqs=linspace(-1, 1, length(xp))' ;
        ramp_filter = abs(freqs);
        ramp_filter = repmat(ramp_filter, [1 length(angles)]);

        % considering the detectors as having width d
        d = 0.1;
        V = sym(d * freqs);
        detector_filter = d * sinc(V) ;
        detector_filter = repmat(detector_filter, [1 length(angles)]);
        detector_filter = double(detector_filter);

        % doing FT domain filtering
        G = fftshift(fft(gN,[],1),1);
        G_filtered = G .* ramp_filter .* detector_filter;
        g_filtered = ifftshift(G_filtered,1);
        g_filtered = real(ifft(g_filtered,[],1));
        % FBP without any filter since we have applied own filter already
        P_rec_noise = iradon(g_filtered,angles,"linear", 'None');

        % calculating the image noise variance
        P_run(sa,:,:) = P_rec_noise;
    end
    % calculate the variance among 10 measurements, and the average
    % variance among the elements
    av_var = mean(mean(var(P_run)));
end

% part 4: number of projection angles effect on image noise variance
function [P_rec_noise, av_var] = num_angle_effect(P_orig,num_angle)

    % taking the radon transform of the image
    d_angle = 180 / num_angle;
    angles = 0:d_angle:179;
    [g, xp] = radon(P_orig,angles);

    % 10 independent measurements
    P_run = zeros(10,514,514);
    for sa = 1:10
        % adding the poisson noise to the projections at each angle
        N0 = poissrnd(10^5);
        [m,n] = size(g);
        gN = zeros(m,n);
        for i = 1:n
            N = N0 * exp(-g(:,i));
            r = poissrnd(N);
            N = r;
            gN(:,i) = -log(0.0001 + N/N0);   % adding a very small term to
avoid infinite value
        end                                % does not make a change since
to small

        % applying rectangular window to the ramp filter
        % creating a windowed ramp filter
        freqs=linspace(-1, 1, length(xp))' ;
        ramp_filter = abs(freqs);
        ramp_filter = repmat(ramp_filter, [1 length(angles)]);

        % considering the detectors as having width d
        d = 0.1;
        V = sym(d * freqs);
        detector_filter = d * sinc(V) ;
        detector_filter = repmat(detector_filter, [1 length(angles)]);
        detector_filter = double(detector_filter);

        % doing FT domain filtering
        G = fftshift(fft(gN,[],1),1);
        G_filtered = G .* ramp_filter .* detector_filter;
        g_filtered = ifftshift(G_filtered,1);
        g_filtered = real(ifft(g_filtered,[],1));

        % FBP without any filter since we have applied own filter already
        P_rec_noise = iradon(g_filtered,angles,"linear", 'None');

        % calculating the image noise variance
        P_run(sa,:,:) = P_rec_noise;
    end
    % calculate the variance among 10 measurements, and the average
    % variance among the elements
    av_var = mean(mean(var(P_run)));
end

% create and display the transformed phantom from chest CT
function [P] = createPhantom(p)
    body_tissues = {'outside phantom','adrenals','skin','fat','spinal
cord','blood pool','spine','gas (bowel)','rib cage & sternum','fluid
(bowel)','pelvis','bone marrow','long bones','lymph nodes','skeletal
muscle','thyroid','lungs','trachea','heart','cartilage','liver','spleen','g
all
bladder','urine','kidney','feces','pharynx','testes','esophagus','prostate'
,'stomach','rectum','small
bowel','diaphragm','colon','bladder','pancreas','lesion'};
    bt_ids =
[0,84,64,85,66,86,68,87,69,88,70,89,71,90,72,91,73,92,74,93,75,94,76,95,77,
96,78,97,79,98,80,100,81,102,82,103,83,126];
    head_tissues = {'outside phantom','medulla
oblongata','skin','fat','brain','blood pool','spinal','bone
marrow','skull','pons','spine','trachea','dens of axis','cartilage','jaw
bone','uncus (ear bones)','skeletal muscle','sinuses/mouth
cavity','lachrymal glands','optic nerve','spinal canal','cerebral
falx','hard palate','eye','cerebellum','lens','tongue','cerebral
aqueduct','pharynx','teeth','esophagus'};
    ht_ids =
[0,84,64,85,65,86,66,89,67,90,68,92,69,93,70,98,72,103,73,105,74,112,75,118
,76,120,77,121,78,124,79];

    % choose the phantom slice
    segment = 'BODY';
    slice = 15;

    % read color segmented image file convert
    fileID =
fopen(strcat('dataset\Color_slices\',segment,'\',segment,'_',num2str(slice,
'%03d'),'_C.DAT.1'));
    P_clr = fread(fileID,[512,512],'uint8=>double').';

    % read CT image file convert
    fileID =
fopen(strcat('dataset\CT_slices\',segment,'\',segment,'_',num2str(slice,'%0
3d'),'_I.DAT.1'));
    P_ct = fread(fileID,[512,512],'int16=>int16');
    P_ct = double(swapbytes(P_ct)).';
    P_ct_mu = P_ct*p(1) + p(2); % apply transform polynomials
    P_ct_mu(P_clr == 0 | P_ct_mu < 0) = 0; % mask the air regions and the
negative values

    % find the unique ids in the segmented image
    [Aval,~,indAval] = unique(P_clr);

    % create an array of the names of those ids
    if strcmp(segment,'BODY')
        [~,ia,~] = intersect(bt_ids,Aval);
        tissues = body_tissues(ia);
    else
        [~,ia,~] = intersect(ht_ids,Aval);
        tissues = head_tissues(ia);
    end

    % create a new segmented image with ids replaced with 1,2,3,...
    Avalnew = 1:length(Aval);
    Anew = Avalnew(indAval);
    Anew = reshape(Anew, size(P_clr));

    % plot the image with a distinct color for each region
    cmap = turbo(length(Aval));
    cmap(1,:) = [0,0,0];
    figure(1)
    subplot(1,3,2)
    image(Anew);
    colormap(cmap);
    hold on;
    for K = 1 : length(Aval); hidden_h(K) = surf(uint8(K-[1 1;1 1]),
'edgecolor', 'none'); end
    hold off
    uistack(hidden_h, 'bottom');
    legend(hidden_h, tissues,'NumColumns',2)
    clear hidden_h
    title('Original segmented image')
    xlabel('cm')
    ylabel('cm')
    axis on
    xticks(linspace(1,512,length(0:10:51.2)))
    xticklabels(0:10:51.2)
    yticks(linspace(1,512,length(0:10:51.2)))
    yticklabels(0:10:51.2)

    % create the phantom image by replacing the ids with mean values of
regions
    % of the transformed attenuation coefficient matrix
    P = zeros(size(P_ct));
    for i = 1:length(Aval)
        P(Anew==i) = mean(P_ct_mu(Anew==i));
    end
    subplot(1,3,3)
    imshow(P,[])
    title('Transformed phantom image')
    xlabel('cm')
    ylabel('cm')
    axis on
    xticks(linspace(1,512,length(0:10:51.2)))
    xticklabels(0:10:51.2)
    yticks(linspace(1,512,length(0:10:51.2)))
    yticklabels(0:10:51.2)
    subplot(1,3,1)
    imshow(P_ct_mu,[])
    title('Original reconstructed image')
    xlabel('cm')
    ylabel('cm')
    axis on
    xticks(linspace(1,512,length(0:10:51.2)))
    xticklabels(0:10:51.2)
    yticks(linspace(1,512,length(0:10:51.2)))
    yticklabels(0:10:51.2)

    % scale the phantom for (mm-1) unit
    P = P/10;
```

```
end

% create the attenuation coefficient transform polynomials from abdomen CT
function [p] = mu_polyfit
    % choose a slice which has air, kidney and muscle regions
    segment = 'BODY';
    slice = 40;

    % read CT image file convert
    fileID =
fopen(strcat('dataset\CT_slices\',segment,'\',segment,'_',num2str(slice,'%0
3d'),'_I.DAT.1'));
    P_ct = fread(fileID,[512,512],'int16=>int16');
    P_ct = double(swapbytes(P_ct)).';

    % read color segmented image file convert
    fileID =
fopen(strcat('dataset\Color_slices\',segment,'\',segment,'_',num2str(slice,
'%03d'),'_C.DAT.1'));
    P_clr = fread(fileID,[512,512],'uint8=>double').';

    % possible efficient energy levels
    keVs = [60,65,70,75,80,85,90];
    mu_air = [0,0,0,0,0,0,0];
    mu_kidney = [0.2127,0.2071,0.2013,0.1958,0.1908,0.1875,0.1842]; % these
values are from a paper
    mu_muscle = [0.2111,0.2052,0.1993,0.1935,0.1878,0.1844,0.1812]; % these
values are from a paper
    p = zeros(length(keVs),2); % fit polynomials
    s = zeros(1,length(keVs)); % fit errors
    for i = 1:length(keVs)
        % find the mean values in CT image of the regions of segmented
image
        x =
[mean(P_ct(P_clr==0)),mean(P_ct(P_clr==77)),mean(P_ct(P_clr==72))];
        y = [mu_air(i),mu_kidney(i),mu_muscle(i)];
        [p(i,:),S] = polyfit(x,y,1); % find the best fit
        s(i) = S.normr/mean(y);
        fprintf("%d keV, error: %d\n",keVs(i),s(i))
    end
    % the best fit with the least error is the efficient energy
    [~,I] = min(s);
    fprintf("\nThe best fit is at %d keV\n",keVs(I))
    % the corresponding polynomials are used in the transform
    p = p(I,:);
end
```