

Positron Emission Tomography, a step forward for diagnosing cancer

Thomas Boot¹

¹Department of Biomedical Engineering, Illinois Institute of Technology, Chicago, IL 60616, USA
tboot@hawk.iit.edu

Abstract: Positron Emission Tomography (PET) imaging is a medical method that provides more reliable information for diagnosing diseases in a noninvasive way compared to other scanning procedures. This technique requires the injection of a radiotracer, which is then monitored externally to generate PET data. Drastically evolved over the past years, the integration of two image modalities such as CT and PET provides a synergy that enhanced a growing appreciation of the image reconstruction technique providing advantages compared to other computing methods. Several mathematical methods have been adapted to solve inverse problem known as “image reconstruction based on projections”. This paper provides a brief introduction of the PET image reconstruction and how the data is collected in a 2D mode. The reconstruction methods are separated in two approaches, first the analytical approach and finally the iterative approach. Analytic methods provide a direct mathematical solution to a problem whether as iterative methods are based on a more realistic model in order to have a more accurate description of the imaging process resulting in a computationally more expensive algorithm, which require more steps to acquire the desired result. Current challenges exist nowadays, as all computation methods present their limitations. Advances are currently made in order to improve PET image reconstruction.

References and links

- [1] “Global Status Report on Noncommunicable Diseases”. World Health Organization, 2014.
- [2] "Surveillance, Epidemiology, and End Results ProgramTurning Cancer Data Into Discovery." *Cancer of the Brain and Other Nervous System*. National Cancer Institute, n.d.
- [3] Sorenson JA, Phelps ME. *Physics in Nuclear Medicine*. Grune & Stratton; FL, USA: 1987
- [4] Lewitt RM, Matej S. *Overview of methods for image reconstruction from projections in emission computed tomography*. Proc. IEEE. 2003;91(10):1588–1611. Overview of image reconstruction methods for emission tomography.
- [5] Kinahan P, Defrise M, Clackdoyle R. Analytic image reconstruction methods. *Emission Tomography: The Fundamentals of PET and SPECT*. Elsevier; London, UK: 2004. pp. 421–442. Discusses analytic image reconstruction in detail.
- [6] Jain AK. Image reconstruction from projections. In: Thomas Kailath, ed. *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall; 1989:431–475.
- [7] Lalush DS, Wernick MN. Iterative image reconstruction. *Emission Tomography: The Fundamentals of PET and SPECT*. Elsevier; London, UK: 2004. pp. 443–472. Comprehensive overview on iterative image reconstruction methods.
- [8] Statistical approaches in quantitative positron emission tomography. Stat. Comput. 2000;10(2):147–165. Survey on statistical image reconstruction methods.

- [9] Meikle SR, Hutton BF, Bailey D, et al. *Accelerated EM reconstruction for total-body PET: potential for improving tumor detectability*. Phys. Med. Biol. 1994;39(10):1689–1704.
- [10] Richard Gordon, *Algebraic Reconstruction Techniques*. IEEE, 1974
- [11] Gabor T.Herman, Michael Chan, Yair Censor, Emanuel Levitant, Robert M.Lewitt and T.K. Narayan, *Maximum A Posteriori Image Reconstruction from projections*, New York, 1996.
- [12] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. New York: IEEE Press, 1988.

1. Introduction

In order to evaluate the medical progress, for any disease that humans are treating, it is important to keep track of the number of new cases, deaths, and survival over time. This can help scientists understand whether progress is being made and where additional research is needed to address challenges, such as improving screening or finding better treatments. Noncommunicable diseases (NCD), also known as chronic diseases tend to be of long duration and are the result of a combination of genetic, physiological, environmental and behavior factors that kills nearly 40 million people each year. Which represents 70% of all deaths globally. Cardiovascular diseases account for most NCD deaths with 17.7 million people, followed by cancer with 8.8 million people, respiratory disease, 3.9 million, and diabetes, 1.6 million people. [1] In the United States, over 1,688,780 new cases of cancers are estimated for 2017. [2] The detection, screening and treatment of those cancers and palliative care are key components to respond to cancers.

One approach to diagnose cancer is Positron Emission Tomography (PET). PET is a medical imaging modality with proven clinical value for disease diagnosis and treatment monitoring. PET scanners use small amounts of radioactive materials that are injected into the human body, called radiotracers and are monitored externally to generate PET data [3]. The radioisotope that is used for tracer undergoes positron emission decay and emits a positron. The emitted positron travels in the tissues for a short distance (less than 1 mm but depends on the isotope), during which it loses kinetic energy and decelerates to a point where it can interact with an electron. The encounter between both electron and positron produces a pair of gamma photons that goes in opposite directions. Scintillators in the scanning ring device detect those gamma rays. Figure 1.

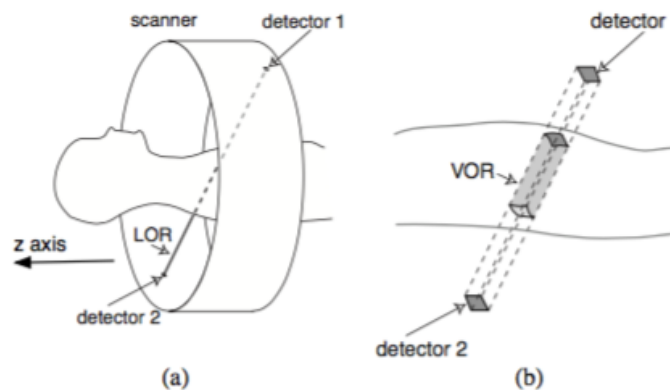


Fig 1. Detection by two detectors of a gamma ray on the LOR (Line of response) in (a). (b) represents the detailed view of the Volume of Response (VOR) scanned by the two rectangular detectors.

As the two photons are sensed almost at the same moment, it can be deduced that the decay happened on the line between the two involved detectors, i.e. on the line of response (LOR). Such an event is called a true coincidence, but other kinds of coincidences exist, which cause undesired effects on the reconstructed images, i.e. scattered coincidences, where a photon is deflected on the way to a detector, random coincidences, with two photons from different decays being randomly detected as a true event. Figure.2.

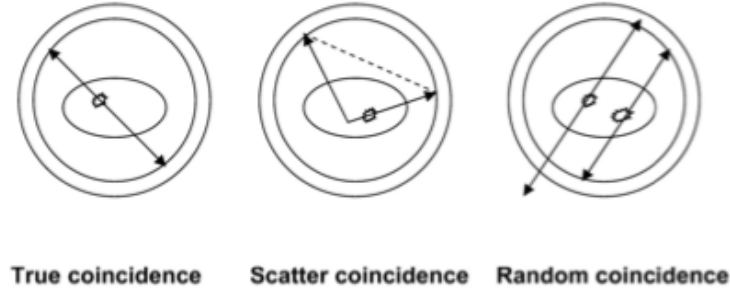


Fig 2. Different coincidence events.

The data acquired is a list of coincidence events. Those events are grouped into projection images for all angles and organized in a 2D case called a sinogram, in which the horizontal axis represents the count location on the detector, and the vertical axis corresponds to the angular position of the detector. The corresponding number of counts is assigned to each point of the sinogram.

PET imaging provides quantitative information of biological processes, and is widely combined with anatomical information from CT scans. Thus, the combination of PET and CT provides a synergy of two imaging modalities, which can be computed in two-dimensions or three-dimensions.

The image reconstruction can be computed thanks to mathematical/numerical algorithms that will be presented through this article.

2. Mathematical method

As PET images cannot be observed directly, they have to be reconstructed from the measured data via mathematical image reconstruction. To achieve this, we introduce a mathematical model of the reconstruction problem. In case of PET, we regard the problem as an inverse and ill-posed problem and is a statistically Poisson model.

2.1. Inverse Problem

The underlying image reconstruction problem can be considered as an inverse problem. For image reconstruction, the model is a linear relationship:

$$p = HF + n \quad (1)$$

where \mathbf{p} is the number of observations, \mathbf{H} is the known matrix system, \mathbf{F} is the unknown image and \mathbf{n} is the error of the reconstruction. The goal of reconstruction is to use the data values \mathbf{p} (projections through the unknown object) to find the image \mathbf{F} . There are two common approaches for dealing with PET data, the first one is to assume that the data is deterministic, so containing no statistical noise. In equation (1), \mathbf{n} would be a deterministic number and if known, the exact solution for the image could be found. Analytic reconstruction methods use

the inverse of the discrete Radon transform to solve this problem, offering a direct mathematical solution for the image \mathbf{f} from known projections \mathbf{p} . This deterministic approach simplifies the reconstruction allowing for a fast, direct mathematical solution. This approach nevertheless doesn't take into account the noise structure in the observations as it is based on an idealized model of the system. This leads to a low resolution and poor noise properties.

In reality, the data values are stochastic due to physical realities in PET imaging resolution including: the effects of attenuation and statistical variability in photon detection. The noise is more accurately representative as random noise, so the exact solution is impossible to find. It is modeled with as a Poisson random process. Therefore, we revert to 'estimation' techniques, which are solved iteratively. These 'estimation methods' lead to approximate solutions only through constraining the solution with some form of regularization. Thus, the second class of reconstruction methods is iterative reconstruction methods, leading to more accurate reconstruction images but it increases the complexity of the problem. Basically, iterative methods offer improvements over the analytical approach because they can account for the noise structure and therefore use a more realistic model of the system.

2.2. Statistical Modeling

In case of stochastic data, noise can be modeled as a Poisson random process on account of the radioactive decay of the tracer, as well as the counting of photon coincidences. The appearance of this noise is seen due to the statistical nature of electromagnetic waves such as X-rays, visible lights and gamma rays. The X-ray and gamma ray sources emitted number of photons per unit time. These rays are injected in patient's body from its source, in medical imaging systems. These sources are having random fluctuation of photons. The result gathered image has spatial and temporal randomness. This noise is also called as photon noise or shot noise. This noise obeys the Poisson distribution and is given as:

$$p(k) = \frac{\lambda^k}{k!} e^{-\lambda} \quad (2)$$

Here we have: k : the number of events in an interval, λ equal to the mean and the variance of the Poisson distribution. Figure 3 shows the Poisson distribution for several values of lambda.

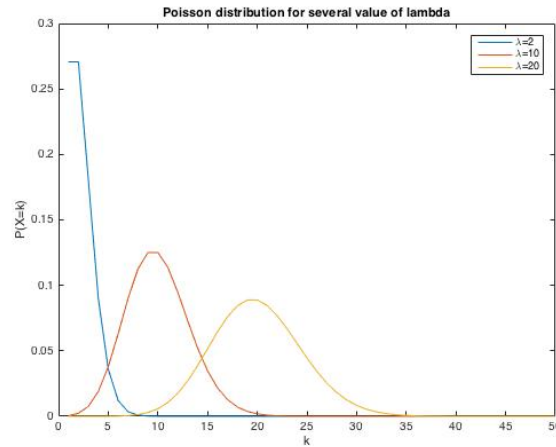


Fig 3. Poisson distribution for several values of lambda

In order to obtain the desired probability density $p(p|F)$, a conditional probability density of the noisy data p given an image F leads to the following assumptions:

- the measured values p_i are realizations of random variables P_i , because the random noise is caused by the detector system.
- the random variables P_i are Poisson distributed with expected values $(HF)_i$ and independent and identically distributed.

These assumptions lead to the following equation:

$$p_{Pi}(p|F) = \prod_{i=1} \frac{(HF)_i^{p_i}}{p_i!} e^{-HF_i} \quad (3)$$

This statistical modeling will further be used for the iterative image reconstruction methods.

3. Analytic two-dimensional image reconstruction

Analytic reconstruction methods assume that PET data is noise free, and attempt to find a direct mathematical solution for the image from known projections. A comprehensive review on the topic was given by Kinahan in [5].

3.1. Two-dimensional central-section theorem

The central-section theorem also known as Fourier-slice transform, is the foundation of analytic image reconstruction. This theorem is based on the Radon transformation $g(s, \theta)$, which is the line integral of the values of a function $f(x, y)$ along the line inclined at an angle θ from the x -axis at a distance s from the origin. This theorem is more explicitly explained in [6].

To illustrate this notion of projection, we use the following example, where the number of the slices are the pixels, and each point of measurement on the detector, for each projection angle, is called a bin. So, the number of bins equals the number of points of measurement multiplied by the number of angles. From a mathematic point of view, the sets of values in the sinogram and the reconstructed slice can be considered as matrices or vectors

Figure 4 presents an example of a discrete projection for a 3×3 image at 2 angles. The result is a sinogram with 2 rows (corresponding to 2 projection angles) and 3 columns (corresponding to 3 points of measurements on the detector), so there are 6 bins.

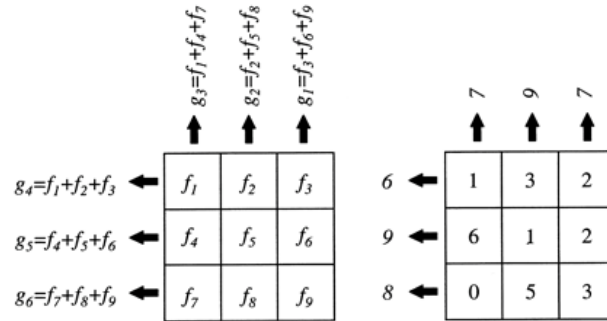


Fig 4. On the left, illustration of the principle of projection for one 3×3 slice at angle $\theta = 0$ and $\theta = 90^\circ$. Value in each bin is the sum of values of pixels that project onto that bin. On the right, for instance: $g_1 = f_3 + f_6 + f_9 = 2 + 2 + 3 = 7$. Result of projection is sinogram with 2 rows, whose values are (7, 9, 7) and (6, 9, 8).

The 2D central section theorem states that the 1D Fourier transform of a projection at angle θ is equivalent to a section at the same angle through the center of the 2D Fourier transform of the object:

$$P(Vs, \theta) = F(Vx, Vy) \quad (2)$$

With $Vx = Vr.\cos(\theta)$ and $Vy = Vr.\sin(\theta)$ in polar coordinates. $P(Vs, \theta)$ is the 1D Fourier transform of projection $p(s, \theta)$, $F(Vx, Vy)$ is the 2D Fourier transform of the object distribution $f(x, y)$ and Vx is the Fourier domain conjugate for x .

This theorem states that the Fourier takes 1D Fourier transform of each row in the sinogram (corresponding to one projection) and interpolate and sum the results on a 2D rectangular grid in Fourier domain. Then the inverse 2D Fourier transform is performed to obtain the image. The main difficulty in direct Fourier reconstruction is the interpolation involved. The reconstructed image strongly depends on the accuracy of interpolation, and is very sensitive to interpolation errors.

To illustrate this theorem through the Figure 5. It shows $F1\{p(s, \theta)\}$ as 1D Fourier transform of a projection and $F2\{f(x, y)\}$ is the 2D Fourier transform of the image. The central-section theorem indicates that if we know the projection $P(v_s, \theta)$ at all angles $0 \leq \theta < \pi$, then we can fill in values for $F(v_x, v_y)$. The inverse two dimensional Fourier transform of $F(v_x, v_y)$ will give us $f(x, y)$.

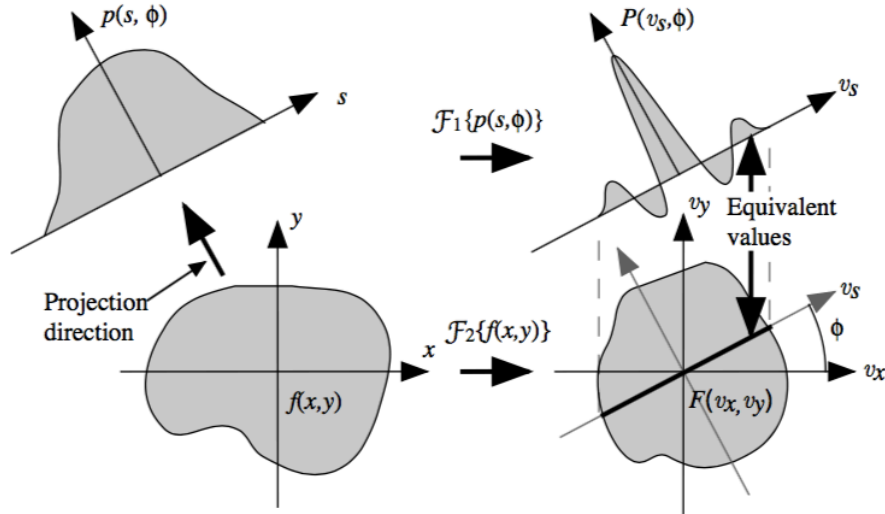


Fig 5. Illustration of the two-dimensional central-section theorem.

Regarding the Fourier slice theorem, while interpolation can be improved with different basis functions, direct Fourier methods are not as widely used as the filtered back projection method, which is described next.

3.2. Reconstruction by Filtered-Backprojection (FBP)

Filtered back projection is the most widely used algorithm for analytic reconstruction. It is also the most widely used method for CT image reconstruction. Its popularity arises from the combination of accuracy, speed of computation and simplicity of implementation.

An intuitive method of image reconstruction is back projection, which is the adjoint of the forward projection process of data acquisition. It simply runs the projections back through into

an image array along the LOR to obtain a rough approximation to the true activity distribution. Straight back projection returns an image approximation but it is a blurred version of the object.

A simplified example of backprojection can be seen in Figure 6.

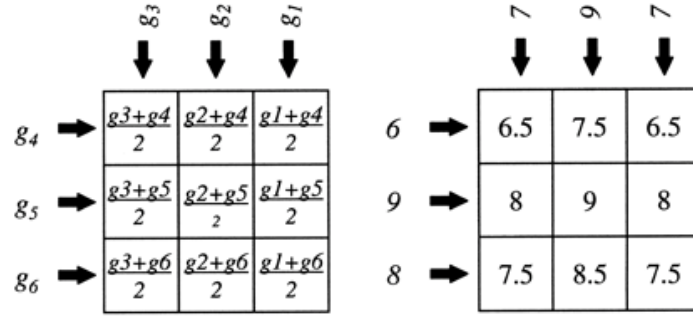


Fig 6. On the left, the principle of backprojection for one 2×3 sinogram. Value in each pixel is sum of values of bins that, given angle of detector, can receive photons from that pixel and is divided by number of rows of sinogram. On the right, we have for instance: $f_1 = (g_3 + g_4)/2 = (7 + 6)/2 = 6.5$.

Comparing this slice with that of Figure 4 and we see that after 1 projection and 1 backprojection, initial slice is not retrieved. The main point of this part is that the backprojection operation is not the inverse of the projection operation. This means that applying backprojection to the Radon $g(s, \theta)$ does not yield $f(x, y)$ but a blurred $f(x, y)$. This problem is due to the fact that during the backprojection process, each bin value is attributed to all pixels that project onto that bin and not only to the pixel the activity is coming from.

The idea is to use a ramp filter in order to eliminate blurring. The disadvantage of back projection filtering is the computation redundancy in the backprojection step (back projection needs to be computed on a larger image matrix than the final result). To avoid this, the filtering and back projection steps are interchanged, leading to the standard filtered-back projection (FBP) method. Figure 7 illustrates the difference between backprojection and filtered-backprojection.

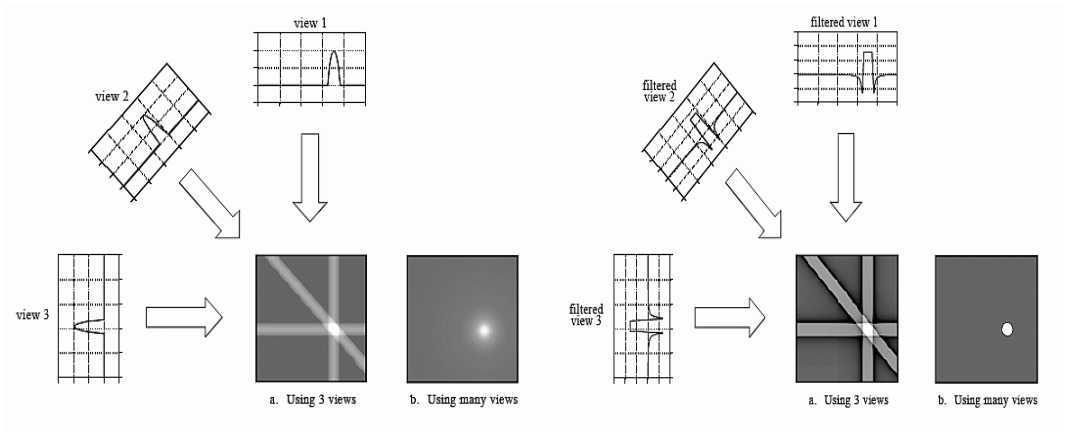


Fig 7. On the left, illustration of backprojection. Backprojection reconstructs an image by taking each view and smearing it along the respective LORs. The result is a blurry version of the original image. On the right we see, that each view is beforehand filtered before backprojected. The result is a mathematically exact reconstruction of the image.

As analytic reconstruction methods are fast and easy to implement, especially for the FBP algorithm. They don't represent the reality as it doesn't take into account the noise structure in the observations as it is based on an idealized model of the system.

Advances in computation speed and the development of efficient algorithms have permitted widespread clinical use of iterative image reconstruction methods [7-9].

4. Iterative two-dimensional image reconstruction

To recall, iterative reconstruction methods lead to more accurate reconstruction images but it increases the complexity of the problem, making it impossible to obtain direct analytic solutions. Therefore, the image reconstruction is solved iteratively. Overall, iterative methods offer improvements over the analytical approach because they can account for the noise structure and therefore use a more realistic model of the system.

The basic concept of iterative reconstruction is summarized here and illustrated in Figure 8. First, we make an initial estimate of the object activity distribution. Then we calculate the estimated projection by forward projecting the initial estimate. Based on the comparison between estimated and measured projections, the initial estimate is adjusted according to certain criterion. This 'forward project, compare, backproject, adjust' procedure is repeated till the estimated image reaches a desired solution.

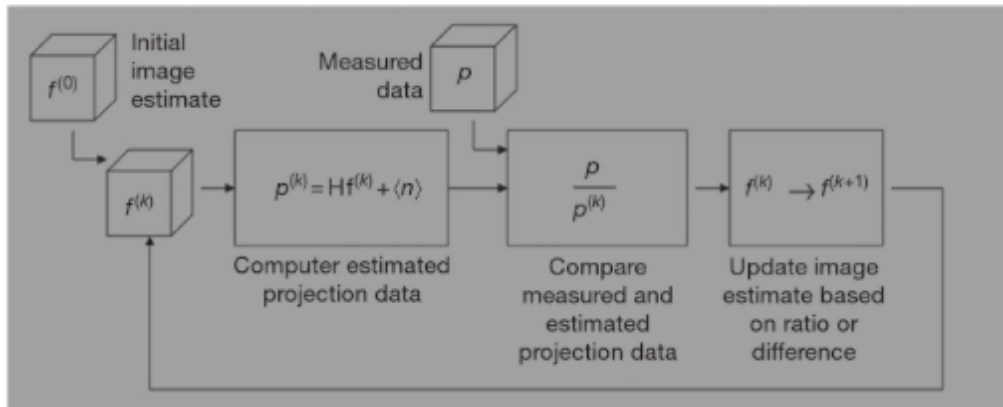


Fig 8. Block diagram illustrating the iterative image reconstruction process.

All iterative methods can be characterized with two key components: First, a criterion that defines the 'best' image. This criterion is represented as an objective function, which measures the similarity between the image estimate and the best image. The most widely used criterion is the maximum likelihood (ML) approach. Another common criterion is the least square principle, which measures the difference between measured and estimated projections using Euclidean distance. Second, all iterative methods require a numerical algorithm to determine how the image estimate should be updated at each iteration based on the criterion. The expectation-maximization (EM) algorithm is commonly used to find the ML image estimate.

4.1. Maximum Likelihood – Expectation Maximization

Maximum-likelihood estimation maximization (MLEM) is a standard statistical estimation method. It produces an estimate that maximizes the likelihood function. The Expectation Maximization (EM) algorithm is an efficient algorithm to find the ML estimate. MLEM image reconstruction adopts ML as the optimization criterion, and uses the EM algorithm to find the optimal solution. The EM algorithm is a numerical algorithm that solves the incomplete data

problems in statistics. If the image reconstruction deals with the Poisson likelihood, the MLEM can be expressed using the following equation:

$$\hat{f}_j^{n+1} = \frac{\hat{f}_j^n}{\sum_{i'=1}^N H_{i'j}} \sum_{i=1}^N \frac{H_{ij} \times p_i}{\sum_{k=1} H_{ik} \times \hat{f}_k^n} \quad (3)$$

With \hat{f}_j^n is the image estimate for the pixel j at iteration n.

The initial guess \hat{f}_j^0 (often a blank or uniform grayscale image) is forward projected into the projection domain. Then, the comparison between estimated and measured projections is determined by calculating their ratio. This ratio in projection domain is back projected to the image domain and properly weighted, providing a correction term. Finally, the current image estimate \hat{f}_j^n is multiplied by the correction term, generating the new estimate \hat{f}_j^{n+1} . MLEM shows two main drawbacks, the method yields very noisy images due ill-conditioning of the problem, indeed the MLEM reconstruction is stooped at early iterations due to the increasing noise level with iterations. The second one is its slow convergence.

Equation 3 has to be applied pixel by pixel but can be extended to the whole image and interpreted as:

$$Image^{n+1} = Image^n \times \text{Normalized Backprojection of } \left(\frac{\text{Measured projections}}{\text{Projections of image}^n} \right) \quad (4)$$

In order to improve the performance of the MLEM algorithm, other alternatives were created. One of them is the Ordered Subsets Expectation Maximization.

4.2. Ordered Subsets Expectation Maximization

Its popular variant, the ordered subsets EM algorithm (OSEM), uses ordered subsets to accelerate convergence. This algorithm partitions the projection data into B subsets and uses only one subset of data S_b for each update. It results in the following equation:

$$\hat{f}_j^{(n,b)} = \frac{\hat{f}_j^{(n,b-1)}}{\sum_{i' \in S_b} H_{i'j}} \sum_{i \in S_b} \frac{H_{ij} \times p_i}{\sum_{k \in S_b} H_{ik} \times \hat{f}_k^{(n,b-1)}} \quad (4)$$

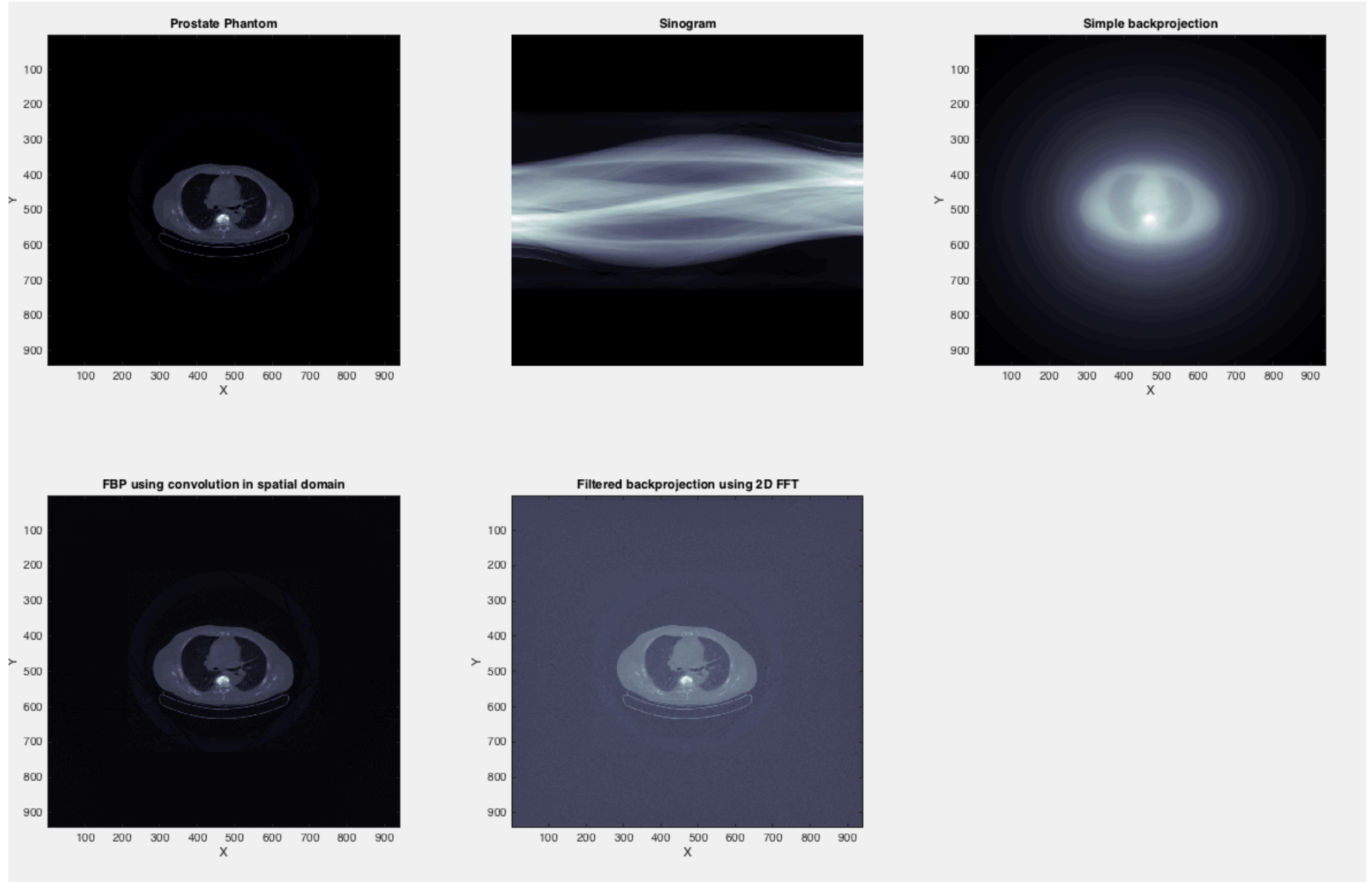
With b the index of each sub iteration, that changes with each subset of the data and $\hat{f}_j^{(n,0)} = \hat{f}_j^{n-1}$, $\hat{f}_j^{(n,B)} = \hat{f}_j^n$. This algorithm leads to a considerable acceleration compared to the MLEM algorithm. The number of subsets determine the degree of acceleration, for instance with 1 subset and 10 iterations we would have the same result as with 10 subsets and 1 iteration. MATLAB. This shows that the MLEM algorithm (1 subset) needs 10 times the computation time of the OLEM algorithm with 10 subsets to have a similar image reconstruction.

Typically ordered subsets EM reconstruction is stopped at early iterations and post smoothed to suppress noise. One drawback of OSEM is that it is not guaranteed to converge to the ML solution. Still this remains the most widely used iterative algorithm in PET reconstruction.

On the other hand, researches developed other types of iterative methods such as Algebraic Reconstruction Techniques (ART) [10] or Maximum a posteriori reconstruction (MAP)[11] in order to deal with the drawbacks of the MLEM algorithm.

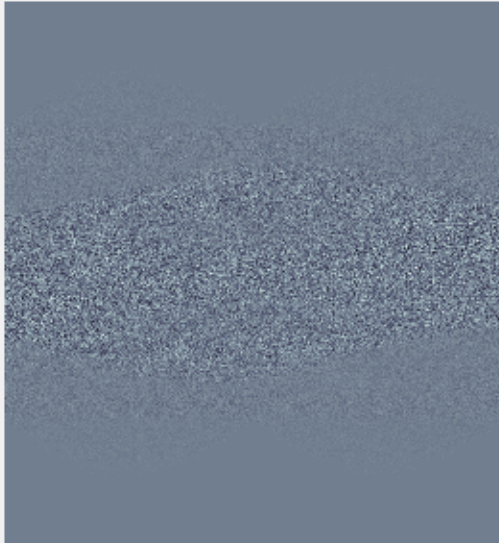
Simulations

Analytic Reconstruction

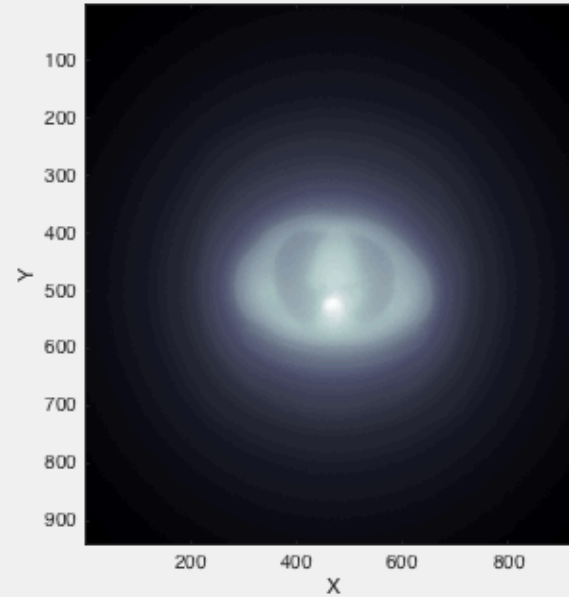


Iterative Reconstruction

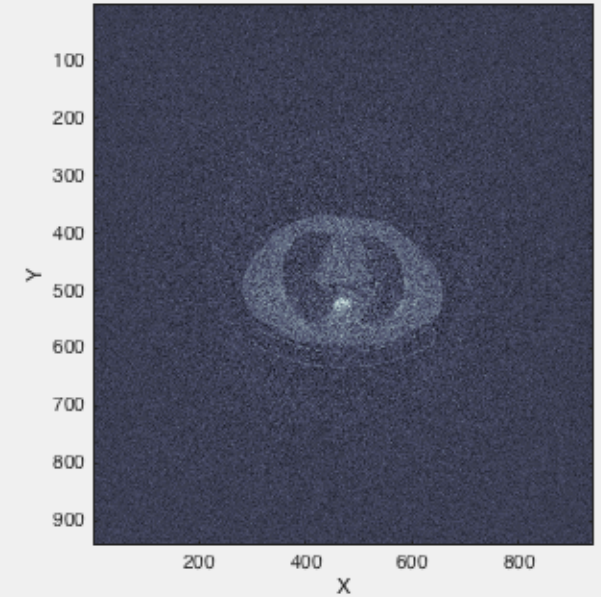
Difference between Sinogram and Sinogram with Poisson noise



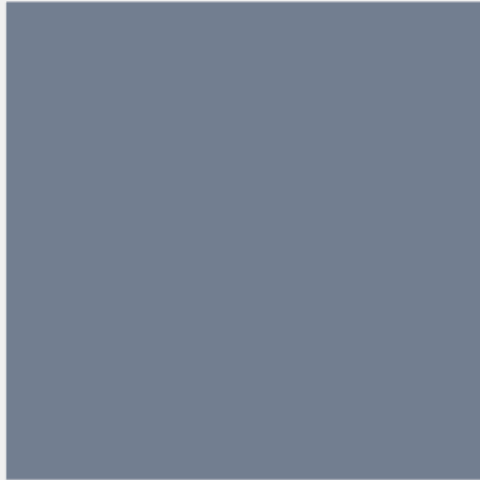
Simple backprojection with noise



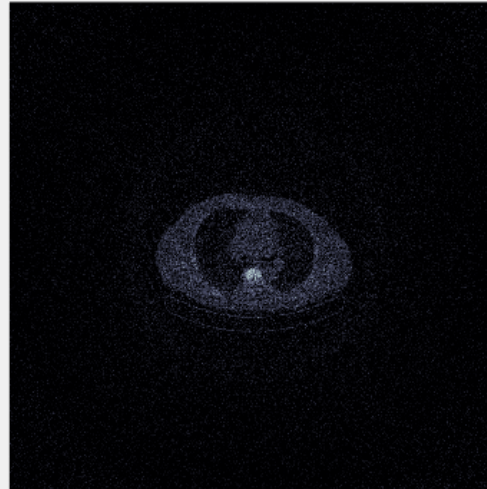
Filtered backprojection using 2D FFT



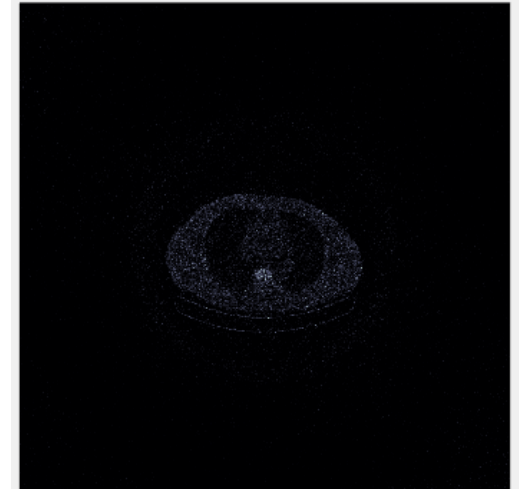
Iteration - 1



Iteration - 2



Iteration - 3



Discussion

These figures demonstrate that with analytic reconstruction we obtain a direct mathematical solution. This was the first logical step to establish before doing the iterative methods. Indeed, the first picture on the left represents the original image. The image on the right represents the sinogram of that image computed thanks to the Radon transformation. The image on the right on the first row, represents the backprojection in order to reconstruct the image, indeed we see that a simple backprojection isn't an accurate reproduction of the original image. As the simple backprojection is a blurry image, we proceed with a filtered backprojection, which is the image in the left corner. Here we used a RamLak filter but the reconstruction can also be done with a Shepp-Logan filter. We see that this solution provides a better reconstruction than a simple backprojection. The last image provides a FBP in 2D FFT reconstruction. In this case we see that the reconstructed image provides other information than the FBP in spatial domain, we notice here that the background has more a light greyscale shade, I haven't an explanation for that.

In order to create to a more realistic model, we apply some Poisson noise to the original sinogram. This part and the following part about iterative methods was slowed down due to coding bugs.

The poisson noise applied, the process of MLEM method can be tested. Indeed, we do a ratio between the noisy sinogram and the sinogram of the initial guess image thanks to the radon transformation. This ratio is then backprojected and using FBP to determine the estimated image. This estimated image is then reiterated in the ratio and so on. A simple case of MLEM is showed page 11. The first image is the guess image, initially a grey image, then we can see that the image is better estimated over the iterations. Still the Matlab code isn't perfect as we see that the image in the third iteration is deteriorated, it seems that over multiple iterations the image wouldn't be correctly estimated.

This was an interesting project, where I had to begin from scratch. Indeed, I didn't finish the section on iterative methods and the comparison between MLEM and OSEM. I would have liked to explore this with more time, in order to have a more accurate representation on how this process works. The theoretical aspect is known to me, but the implementation is Matlab isn't perfectly developed.

Conclusion

Positron Emission Tomography imaging has proven his medical importance in diagnosing cancer compared to other methods. PET image reconstruction is usually formulated as a linear inverse problem, a number of corrections are needed to produce quantitative reconstruction results. Analytic reconstruction methods provide a fast and direct solution, but they cannot model the physical effects of PET scanners or the statistical variability in photon detection. Filtered backprojection is the most widely used algorithm for analytic reconstruction. Iterative reconstruction methods can model statistical noise of PET data and physical effects, leading to more accurate reconstruction and increased complexity of problem. All iterative methods can be characterized with two key components: a set of mathematically defined criteria defining the 'best' image, and an algorithm to find the solution. Maximum-likelihood expectation-maximization image reconstruction uses maximum likelihood as the optimization criterion, and employs the expectation-maximization algorithm to find the maximum-likelihood solution. Its popular variant, the ordered subsets expectation-maximization algorithm, uses ordered subsets to accelerate convergence. Typically ordered subsets expectation-maximization reconstruction is stopped at early iterations and postsmoothed to suppress noise. Nowadays, current challenges in PET image reconstruction are to make the reconstruction more reliable, gain time in computation and decrease the complexity of the algorithm. PET reconstruction is on the way to become the most performing way to diagnose diseases in the medical field, and it has a great future ahead.

Acknowledgments

This work was supported by Mr Tichauer from Illinois Institute of Technology, Chicago.

Appendix Code

```
% Thomas Boot
}%{
This script reconstructs an image using Analytical Reconstruction and
Iterative Reconstruction.

Sources DICOM Files :
https://wiki.cancerimagingarchive.net/display/Public/NAF+Prostate
https://wiki.cancerimagingarchive.net/display/Public/NSCLC+Radiogenomics
}%}

clc; clear all; close all
%% Extraction of data from DICOM (Prostate)

cd '/Users/BootThomas/Desktop/IIT/Courses/1stSemester/522-MathMethods/Project/Matlab/DOI/NAF-PROSTATE-01-0001/Patient1/day1/'
pwd = '/Users/BootThomas/Desktop/IIT/Courses/1stSemester/522-MathMethods/Project/Matlab/DOI/NAF-PROSTATE-01-0001/Patient1/';

dicomlist = dir(fullfile(pwd,'day1','*.dcm'));
for count = 1 : numel(dicomlist)
    prostate(count) = dicomlist(count).name;
end
prostate = prostate(1,:);

%% Analytical Reconstruction
figure,
colormap bone,

[sampleimage,map] = dicomread(prostate{1,180});
% set min of image to zero
sampleimage = double(sampleimage-min(sampleimage(:)));
% pad the image with zeros
padDims = ceil(norm(size(sampleimage)) - size(sampleimage)) + 2;
P = padarray(sampleimage,padDims);

% Some parameters
freq = 3; % [1/degree]
theta = 0:1/freq:180-1/freq;

% Original image
subplot 231
imagesc(P);
title('Prostate Phantom')
xlabel('X')
ylabel('Y')

% Sinogram with Radon transformation
subplot 232
sinogram = radonTransformation(P,theta);
imagesc(sinogram);
title('Sinogram')
xlabel('\alpha'); axis off;
ylabel('Number of parallel projections')

% Simple backprojection
subplot 233
simpleBP = backProjection(sinogram,theta);
imagesc(simpleBP);
title('Simple backprojection')
xlabel('X')
ylabel('Y')
```

```

% FBP in spatial domain
% Ask the user the type of filter choose one 'Shepp-Logan' , 'Ram-Lak'
prompt = {'Select filter for back projection : ramLak or sheppLogan'};
dlg_title = 'Input';
num_lines = 1;
defaultans = {'ramLak'};
answer = inputdlg(prompt,dlg_title,num_lines,defaultans);
filterMode = answer{1};
subplot 234
RecoSpatialDomain1DFT = FBPSD(sinogram,theta,filterMode);
imagesc(RecoSpatialDomain1DFT);
title('FBP using convolution in spatial domain')
xlabel('X')
ylabel('Y')

% 2D fourier transformation reconstruction using BP of Sinogram
subplot 235
Reco2DFT = FBP2DFT(simpleBP);
imagesc(Reco2DFT);
title('Filtered backprojection using 2D FFT')
xlabel('X')
ylabel('Y')

```

With the corresponding functions

```

function [sinogram] = radonTransformation(image,thetas)
% Function that returns a sinogram with the radon transformation, with
% input parameters: sampleImage and the angles we want to project the image

nbAngularProjections = length(thetas);
nbParallelProjections = size(image,1);

sinogram = zeros(nbParallelProjections,nbAngularProjections);

for i = 1:length(thetas)
    tempImage = imrotate(image,-thetas(i),'bilinear','crop'); % Rotate image
    sinogram(:,i) = sum(tempImage,2);% fill in the sinogram
end

function BPI = backProjection(sinogram,thetas)
% Function that returns Backprojection image with
% input parameters: the previous sinogram and the angles

nbParallelProjections = size(sinogram,1);
nbAngularProjections = length(thetas);
thetas = (pi/180)*thetas;

% set up the backprojected image
BPI = zeros(nbParallelProjections,nbParallelProjections);
% find the middle index of the projections
midindex = floor(nbParallelProjections/2) + 1;
% set up the coords of the image
[x,y] = meshgrid(ceil(-nbParallelProjections/2):ceil(nbParallelProjections/2-1));

for i = 1:nbAngularProjections
    % figure out which projections to add
    rotCoords = round(midindex + x*sin(thetas(i)) + y*cos(thetas(i)));
    index = find((rotCoords > 0) & (rotCoords <= nbParallelProjections));
    newxy = rotCoords(index);
    % summation
    BPI(index) = BPI(index) + sinogram(newxy,i)./nbAngularProjections;
end

```

```

function BPI = FBPSD(sinogram,thetas,filterMode)
% Function that returns a FBP, with input parameters: sinogram, the
% angles and the proper filter
if nargin < 3
    filterMode = 'ramLak';
end
nbParallelProjections = size(sinogram,1);
nbAngularProjections = length(thetas);
thetas = (pi/180)*thetas;
% Initialization of back projected image
BPI = zeros(nbParallelProjections,nbParallelProjections);
mindex = floor(nbParallelProjections/2) + 1;

% Coordinates of the image in cartesian 2D grid
[x,y] = meshgrid(ceil((-nbParallelProjections/2):ceil(nbParallelProjections/2-1)));
if mod(nbParallelProjections,2) == 0
    halfFilter = floor(1 + nbParallelProjections);
else
    halfFilter = floor(nbParallelProjections);
end
if strcmp(filterMode,'ramLak')
    filter = zeros(1,halfFilter);
    filter(1:2:halfFilter) = -1./([1:2:halfFilter].^2 * pi^2);
    filter = [fliplr(filter) 1/4 filter];
elseif strcmp(filterMode,'sheppLogan')
    filter = -2./(pi^2 * (4 * (-halfFilter:halfFilter).^2 - 1) );
end
for i = 1:nbAngularProjections
    rotCoords = round(mindex + x*sin(thetas(i)) + y*cos(thetas(i)));
    index = find((rotCoords > 0) & (rotCoords <= nbParallelProjections));
    newxy = rotCoords(index);
    % filter
    filteredProfile = conv(sinogram(:,i),filter,'same');
    % summation
    BPI(index) = BPI(index) + filteredProfile(newxy)./nbAngularProjections;
    % visualization
    imagesc(BPI)
    drawnow
end

```

```

function Reco2DFT = FBP2DFT(backprojection)
% Function that returns a 2D FFT, with input parameters: backprojection

midindex = floor(size(backprojection,1)/2) + 1;
% RampFilter for frequency domain without normalization
[x,y] = meshgrid(1 - midindex:size(backprojection,1) - midindex);
rampFilter2D = sqrt(x.^2 + y.^2);

% 2 D Fourier transformation and centered
Reco2DFT = fftshift(fft2(backprojection));
% Filtering in Frequency Domain
Reco2DFT = Reco2DFT .* rampFilter2D;
% inverse 2 D fourier transformation
Reco2DFT = real(ifft2(ifftshift(Reco2DFT)));
end

```

Poisson noise to sinogram and then to the image

```
%% Add Poisson noise to image
% Add Poisson noise to sinogram
N0 = 1000; % Level of noise added to the image
sinogramRawScaled = 100*sinogram ./ mean(sinogram(:));
% We scale the sinogram, as the average was a high value. The poissrnd
% function on a sinogram with a high average doesn't add much noise.
% Therefore we drop the average of the sinogram to 100.
mean(mean(sinogramRawScaled))

sinogram_Noise = poissrnd(sinogramRawScaled);
figure,
colormap bone,
subplot 231, imagesc(sinogram_Noise-sinogramRawScaled);
title('Difference between Sinogram and Sinogram with Poisson noise')
axis off;
disp('SNR with sinogramScaled with mean =')
SNR1 = snr(sinogram,sinogramRawScaled);
disp(SNR1);

% BackProjection
simpleBP_noise = backProjection(sinogram_Noise,theta);
subplot 232,imagesc(simpleBP_noise);
title('Simple backprojection with noise')
xlabel('X')
ylabel('Y')
% FBP to 2D fourier transformation reconstruction
Reco2DFT_noise = FBP2DFT(simpleBP_noise);
subplot 233,imagesc(Reco2DFT_noise);
title('Filtered backprojection using 2D FFT')
xlabel('X')
ylabel('Y')
```

Iterative reconstruction – MLEM method

```
%% MLEM algorithm
[M N]= size(sinogram_Noise);
guessImage = ones(M,M);
size(guessImage);
Norimg = backProjection(ones(M,N),theta);

for iter = 1:3;
tic
figure; imagesc(guessImage); axis off; axis equal; colormap bone;
title(['Iteration - ',num2str(iter)]);
pause(0.1);
proj_ratio = sinogram_Noise./radonTransformation(guessImage,theta);
figure,imagesc(proj_ratio),colormap bone;title(['Iteration - ',num2str(iter)]);
proj_ratio(isnan(proj_ratio)) = 0;
proj_ratio(isinf(proj_ratio)) = 0;

img_ratio = backProjection(proj_ratio,theta);
img_ratio = FBP2DFT(img_ratio)./Norimg;
img_ratio(isnan(img_ratio)) = 0;
img_ratio(isinf(img_ratio)) = 0;

guessImage = max(guessImage.*img_ratio,0);
toc;
end
```