ABSTRACT

Title of Thesis: IMAGE RECONSTRUCTION TECHNIQUES and MEASURE OF QUALITY: CLASSICAL vs. MODERN APPROACHES

Lakshmi P Urimi, Master of Science, 2005

Thesis directed by: Professor Dennis M Healy
Department of Mathematics

Mathematical methods are of central importance in the new technologies of image reconstruction. Some of the most important procedures are classified as Back-projection, Filtered Back-projection and iterative reconstruction techniques. Back- projection played an important historical role but is no longer used because of sizable artifacts. Analytical methods like Filtered Back projection excel in speed and accuracy when a large number of projections are available. These are extensively used in x-ray imaging. Algebraic Reconstruction Technique (ART) is more attractive when the number of views is limited and when noise is significant. For these reasons, iterative methods are widely used in imaging. Two slight variants of ART are SIRT (Simultaneous Iterative reconstruction Techniques) and SART (Simultaneous ART).A modern method of image reconstruction technique is Fast Slant Slack(FSS). This method is rapidly computable, algebraically exact, geometrically faithful and invertible. A new software known as beamlab is used for FSS image reconstruction. All these reconstruction techniques are explored in this work. Also, various tasks are performed to measure the immunity to noise and quality of the images using PSNR, MSE and Universal Image Quality Index.

IMAGE RECONSTRUCTION TECHNIQUES and MEASURES OF QUALITY: CLASSICAL vs. MODERN APPROACHES

by

Lakshmi P Urimi

Thesis submitted to the Faculty of the Graduate School of the

University of Maryland, College Park in partial fulfillment

of the requirements for the degree of

Master of Science

2005

Advisory Committee:

Professor Dennis M Healy, Chair/Advisor

Professor John Benedetto

Professor Wolfgang Jank

CONTENTS

0.1 Introduction

The word Tomography refers to the cross-sectional imaging of an object from the data collected by passing a source (e.g. x-ray, ultra-sound, microwave) through the object from different directions. In 1917, Radon became the first one to invent a mathematical solution on how to reconstruct a function from its projections. This technique has a significant impact on areas which involve image processing. Nowadays, most of the imaging systems rely on the reconstruction of an image from its projection through the process of computed tomography (CT). For medical use, tomographic technique can reconstruct images of internal organs without making any physical contact to the patients bodies by doctors. This enables doctors having a thorough view of the patients internal conditions before making any surgical decision which, in some cases, are risky to the patient. The medical imaging system, shown in Fig(0.1), is used for tomographic imaging. The camera head rotates to get different angles of projections and images are reconstructed from these projections. That is projection should be inverted in some way to obtain the image. This is why tomography is considered as an important inverse problem. The two main issues involved in it are reconstruction accuracy and cost of inversion. This is the main idea of this work.

Fig. 0.1: CT Scanner

Apart from those medical applications, there are numerous non-medical applications for tomography. For example, the tomographic technology for mapping the underground resources becomes very popular in recent years. This is because there is no need for workers digging up the whole site as it is more efficient for undergoing researches. We can use this technique for detecting and monitoring contaminants in soils, groundwater and process effluents.

As seen from the above examples, tomography can be used in various aspects including both medical and non-medical. For different applications on different environments, we can simply change the emitting source or modify the reconstructing algorithms to achieve the results we want. Tomography is an imaging tool that has several mathematical algorithms involve in it. For example, the central slice theorem and algebraic reconstruction algorithm are two of the main methods for reconstructing images. Standard Tomography usually uses x-ray as the source for taking projections.CT scanner using for medicine is a good example.

0.2 Basic Definitions and Theorems

0.2.1 Radon transform

A line integral represents the integral of some parameter of an object along a line. A typical example is the attenuation of x-rays as they propagate through biological tissue. In this case the object is modeled as a two-dimensional distribution of the x-ray attenuation constant and a line integral represents the total attenuation suffered by a beam of x-rays as it travels in a straight line through the object. The coordinate system used is shown in Fig(0.2) to describe line integrals and projections. In this example the object is represented by a two-dimensional function $f(x, y)$ and each line integral by the (θ, t) parameters. The equation of line AB shown in Fig(0.2) is

$$x\cos\theta + y\sin\theta = t \tag{0.2.1}$$

The line integral is

$$Rf(t, \omega(\theta)) = \int_{(\theta,t)} f(x, y)\, ds \tag{0.2.2}$$

where $\omega(\theta) = (\cos(\theta), \sin(\theta))$ is the direction and t is the distance shown in the Fig(0.2).

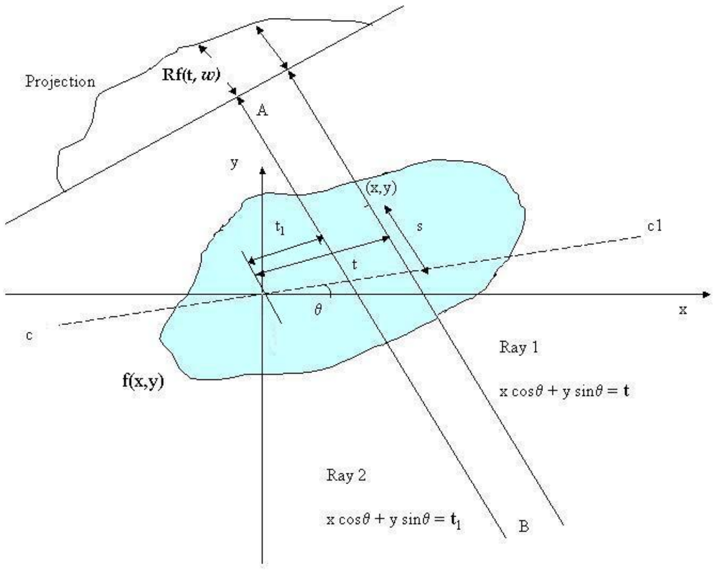


Fig. 0.2: Projection

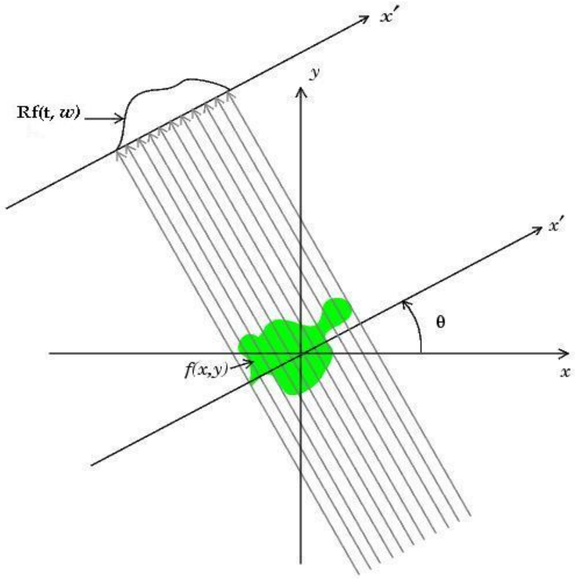


Fig. 0.3: Parallel Beam Projections

The function $Rf(t,\omega)$ is known as the Radon Transform of the function $f(x,y)$. A projection is formed by combining a set of line integrals. The simplest projection is a collection of parallel ray integrals as is given by $Rf(t,\omega)$ for a constant θ. This is known as parallel projection and is shown in the Fig(0.3). This projection is measured by moving an x-ray source and detector along parallel lines on opposite sides of an object. Another type of projection is possible if a single source is placed in a fixed position relative to a line

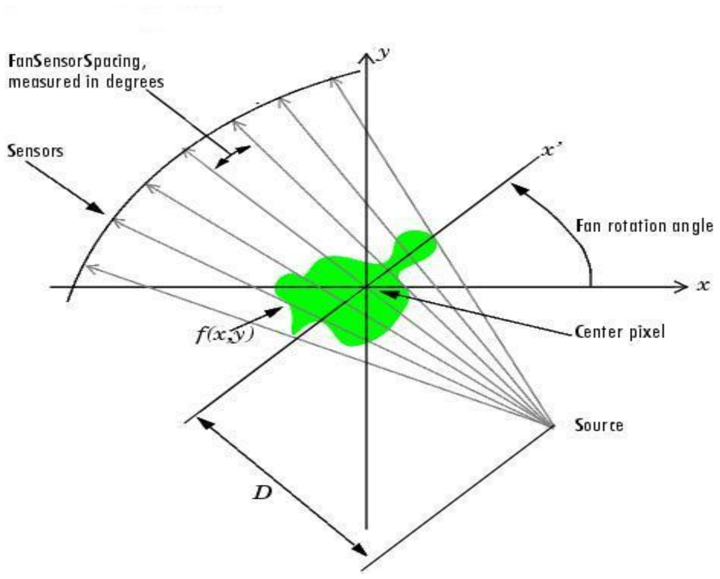


Fig. 0.4: Fan Beam Projections

of detectors. This is shown in the following Fig(0.4) and is known as fanbeam projection because the line integrals are measured along fans. Most of the computer simulation results in this work will be shown for

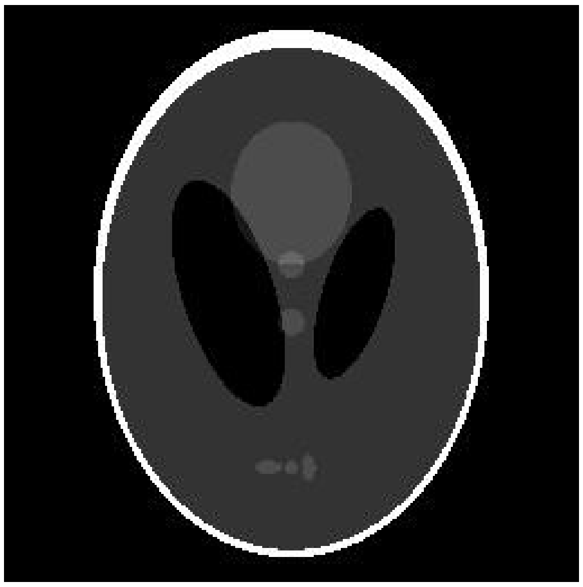


Fig. 0.5: Head Phantom

the image in Fig(0.5). This is the Shepp-Logan "Head Phantom".

0.2.2 Fourier Transform

The two-dimensional Fourier transform of the object function can be defined as

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}\,dxdy \tag{0.2.3}$$

where $f(x,y)$ is a function of two independent variables x and y. Similarly, a projection at an angle θ, $Rf(t,\omega)$, and its Fourier transform can be defined by

$$S_\theta(\omega) = \int_{-\infty}^{\infty} Rf(t,\omega)e^{-j2\pi\omega t}\,dt \tag{0.2.4}$$

A two-dimensional Fourier Inverse Transform can be given by

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{[j2\pi(ux+vy)]}\,du\,dv. \tag{0.2.5}$$

0.2.3 Central Slice Theorem

The one-dimensional Fourier transform of a parallel projection is equal to a slice of the two-dimensional Fourier transform of the original object. It follows that given the projection data, it should then be possible to estimate the object by simply performing a two-dimensional inverse Fourier transform. The simplest example of the Central Slice Theorem is given for a projection at $\theta = 0$. First, consider the Fourier transform of the object along the line in the frequency domain given by $v = 0$. The Fourier transform integral now simplifies to

$$F(u,0) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux)}\,dxdy \tag{0.2.6}$$

but because the factor, $e^{-j2\pi(ux)}$ is no longer dependent on y we can split the integral into two parts,

$$F(u,0) = \int_{-\infty}^{\infty}[\int_{-\infty}^{\infty} f(x,y)\,dy]e^{-j2\pi(ux)}\,dx \tag{0.2.7}$$

It is clear that the term in brackets is the equation for a projection along lines of constant x or

$$Rf_{(\theta=0)}(t,\omega) = \int_{-\infty}^{\infty} f(x,y)\,dy \tag{0.2.8}$$

Substituting this in (0.2.7) we find

$$F(u,0) = \int_{-\infty}^{\infty} Rf_{(\theta=0)}(t,\omega)e^{-j2\pi(ux)}\,dx \tag{0.2.9}$$

The right-hand side of this equation represents the one-dimensional Fourier transform of the radon transform $Rf_{(\theta=0)}(t,\omega)$; thus we have the following relationship between the vertical projection and the 2-D transform of the object function:

$$F(u,0) = S_{(\theta=0)}(u) \tag{0.2.10}$$

where $S_{\theta=0}(u)$ is the one-dimensional Fourier transform of the radon transform. This is the simplest form

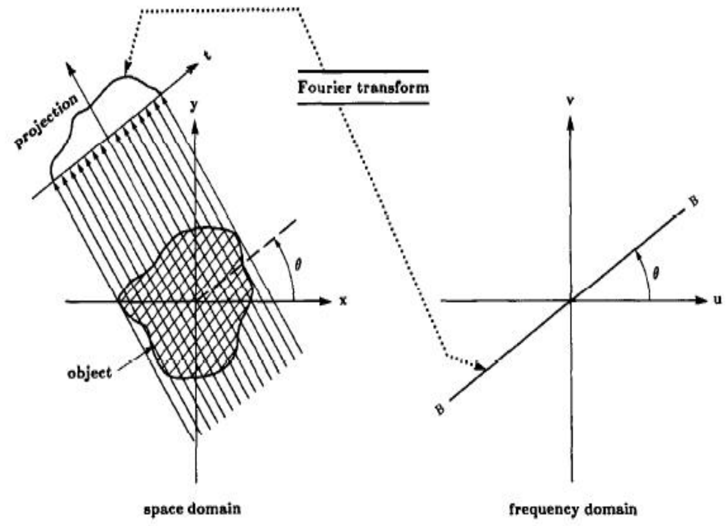


Fig. 0.6: Central Slice Theorem

of the Central Slice Theorem. Clearly this result is independent of the orientation between the object and the coordinate system. If, for example, as shown in Fig(0.6) the (t,s) coordinate system is rotated by an angle θ, the Fourier transform of the projection defined in equation(0.2.9) is equal to the two-dimensional Fourier transform of the object along a line rotated by θ. This leads to the Central Slice Theorem which is stated as: 'The Fourier transform of a parallel projection of an image $f(x,y)$ taken at angle θ gives a slice of the two-dimensional transform, $F(u,v)$, subtending an angle θ with the u-axis. In other words, the Fourier transform of $Rf(t,\omega)$ gives the values of $F(u,v)$ along line BB in Fig(0.6).

0.3 The Reconstruction Problem

0.3.1 Reconstruction Algorithm For Parallel Beam Machine

Obtaining Data for the Parallel Beam Machine

The structure of the reconstruction algorithm is determined by which samples of Rf are available. There are two types of x-ray CT machine, using two-dimensional slices: (a) Parallel beam scanner (b) Fan beam scanner. The former one is the simpler one because the data can be collected much faster. Most modern machines are fan beam scanners.

In a parallel beam scanner approximate samples of Rf are measured in a finite set of directions, $\{\omega(k\Delta\theta) \text{for } k = 0, \cdots, M\}$, where

$$\Delta\theta = \tfrac{\pi}{M+1} \text{ and } \omega(k\Delta\theta) = (\cos(k\Delta\theta), \sin(k\Delta\theta))$$

In terms of the angular variable, θ, the samples are equally spaced. The measurements made in a given direction are then samples of Rf at a set of equally spaced affine parameters,

$$\{jd + d_0 : j = -N, \cdots, N\}$$

where d is the sample spacing in the affine parameter and $N = Ld^{-1}$ where L is shown in the Fig(0.7) and d_0 is a fixed offset. Often d_0 is taken as 0.

Parallel beam data therefore consist of the samples

$$\{Rf(jd, \omega(k\Delta\theta) : j = -N, \cdots, N, k = 0, \cdots, M.)\}$$

Because of the symmetry of the radon transform,

$$Rf(-t, -\omega) = Rf(t, \omega),$$

measurements are only required of angles $[0, \pi)$. Sometimes it is useful to make measurements over a full $360°$; the extra measurements can then be averaged as a way to reduce the effects of noise and systematic errors. The individual measurements are called rays. A *view*, for a parallel beam machine consists of $Rf(t, \omega)$ for a fixed ω. These are the integrals of f along the collection of equally spaced parallel lines,

$$\{l_{jd_{\omega}(k\Delta\theta)} : j = -N, \cdots, N\}$$

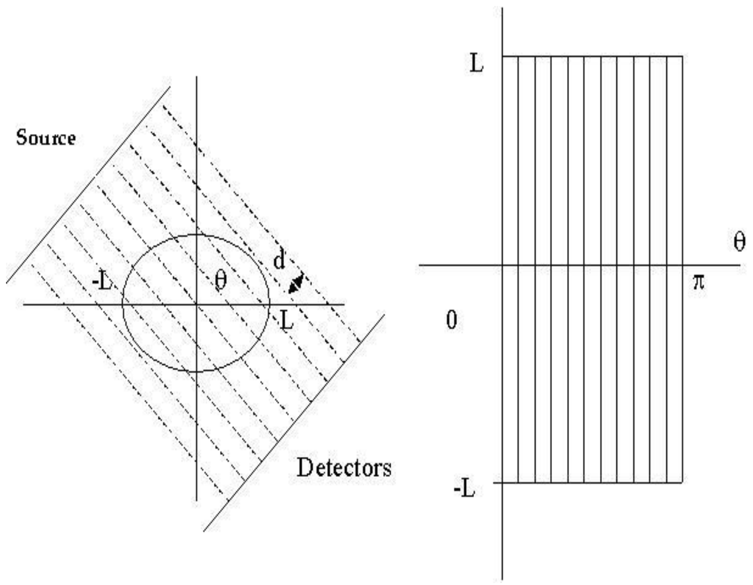


Fig. 0.7: (a)A paralel beam scanner (b) The parallel beam sample space

In (t, ω)-space, parallel beam data consists of equally spaced samples on the vertical lines shown in the Fig(0.7).

Filtered Back-Projection

Lets assume that we can measure all the data from a finite set of equally spaced angles. In this case the data would be

$$\{Rf(t, \omega(k\Delta\theta)) : k = 0, \cdots, M, t \in [-L, L]\}, \tag{0.3.1}$$

where $\Delta\theta = \frac{\pi}{M+1}$. With these data we can apply the central slice theorem to compute angular samples of the two-dimensional Fourier transform of f,

$$F(r\omega) = \int_{-\infty}^{\infty} Rf(t, \omega)e^{-irt}\, dt \tag{0.3.2}$$

where F represents the two dimensional fourier transform of f. For filtered backprojection, we use Fourier Inversion formulas written in polar coordinates.

$$\begin{aligned} f(x,y) &= \frac{1}{[2\pi]^2}\int_0^{2\pi}\int_0^{\infty} F(r\omega)e^{ir\langle (x,y),\omega\rangle} r\,dr\,d\omega && (0.3.3)\\ &= \frac{1}{[2\pi]^2}\int_0^{\pi}\int_{-\infty}^{\infty} F(r\omega)e^{ir\langle (x,y),\omega\rangle} |r|\,dr\,\omega && (0.3.4)\end{aligned}$$

Using the Central slice theorem we get,

$$f(x,y) = \frac{1}{[2\pi]^2}\int_0^{\pi}\int_{-\infty}^{\infty} \widetilde{Rf}(r,\omega)e^{ir\langle (x,y),\omega\rangle}|r|\,dr\,d\omega \qquad (0.3.5)$$

where $\widetilde{Rf}(r,\omega)$ is the one dimensional fourier transform of the radon transform. The above formula(0.3.5) is approximated discretely by the formula

$$\tilde{f}_\phi(x_m,y_l) = \frac{d}{2(M+1)}\sum_{k=0}^{M}\sum_{j=-N}^{N} Rf(jd,\omega(k\Delta\theta)\phi(\langle (x_m,y_l)\omega(k\Delta\theta)\rangle - jd) \qquad (0.3.6)$$

where $\tilde{f}$ is the approximate of f, ϕ is an approximate function to the $|r|$ and acts as a filter. We use this because $|r|$ is not integrable.

Filters

Noise reduction is one of the important tasks in image reconstruction. Various digital filters for reducing noise have been proposed. The problem of noise in computed tomography is usually handled with the application of low-pass digital filters. These filters are designed to suppress signals with high spatial frequencies. The noise at such high frequency will be suppressed with a chance of losing some useful signal as well. The filters that I used in this work are

- Ram-Lak
- Shepp-Logan
- Cosine
- Hamming
- Hann

These can be seen in Fig(0.8).

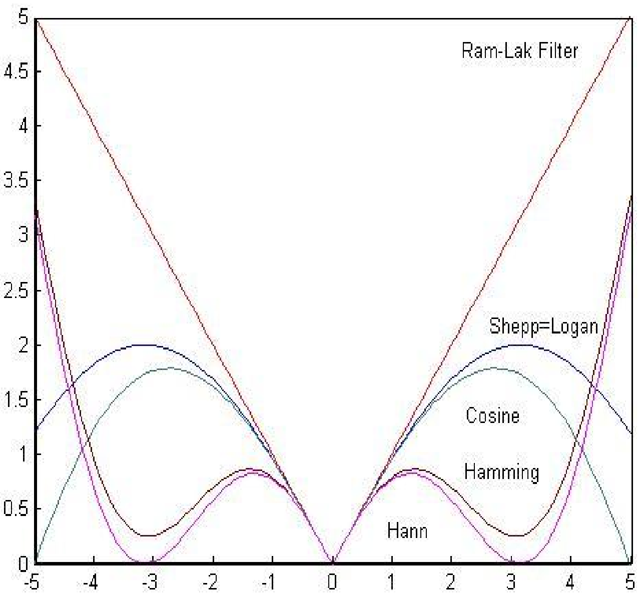


Fig. 0.8: Various Filters used in the reconstruction

0.3.2 Reconstruction Algorithm For Fan Beam Machine

Obtaining Data for the Fan Beam Machine

The other type of scanner is called a *fan beam scanner*. A point source of x-rays is moved around a circle centered on the object being measured. Measurements of Rf are collected for a finite family of lines passing through the source. In a machine of this type data are usually collected by detectors that are usually placed on a circular arc. There are two types of fan beam machines. In *third-generation machine*, the detectors are placed on a circular arc centerd on the source. The detectors and the source are rotated together. In *fourth-generation machine*, the detectors are fixed on a ring, centered on the object. only source is rotated, again around a circle centered on the object, within the ring of detectors. We consider the fourth generation machine to obtain the data. Let (t, θ) denote paramaters for the lines with oriented normal $\omega(\theta)$ at distance t from the origin. In Fig(0.9), S denotes the intersection point of the lines defining a view. It lies a distance D from the origin. The central ray makes an angle β with the positive y-axis. The other lines through S are parameterized by the angle, γ, they make with the central ray. These parameters define coordinates on

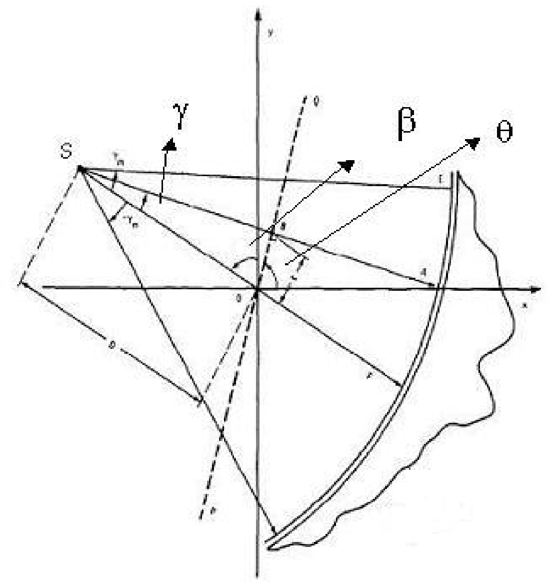


Fig. 0.9: Fan Beam Geometry

a subset of the space of oriented lines. They are related to the (t, θ) variables by

$$\theta = \gamma + \beta \text{ and } t = D\sin(\gamma);$$

This is seen in the Fig(0.9) above. Here the positive angles are measured counterclockwise. In this machine the source is placed at a finite number of equally spaced angles,

$$\beta_k \in \{\tfrac{2\pi k}{M+1} : k = 0, \cdots, M\},$$

and data collected along a set of lines through the source, equally spaced in the ϕ-parameter,

$$\gamma_j \in \{\tfrac{j\pi}{N} : j = -P, \cdots, P\}.$$

Therefore the fanbeam data are the set of samples

$$\{Rf(D\sin(\frac{j\pi}{N}), \omega(\frac{j\pi}{N} + \frac{2\pi k}{M+1} - \frac{\pi}{2})) : j = -P, \cdots, P, k = 0, \cdots, M\} \tag{0.3.7}$$

The maximum and minimum values $\pm\frac{P\pi}{N}$, for the angle ϕ are determined by the necessity of sampling all the lines that meet an object lying in D_L. The samples lie on the sine curve . A view for fan beam machine is the set of samples from the rays passing through a given source position.

Reconstructing f in the Fan Beam Algorithm

The continuous form of the approximate reconstruction formula used in fanbeam algorithms is an analouge of the formula:

$$f_\phi(x,y) = \frac{1}{2\pi}\int_0^\pi \int_{-L}^{L} Rf(t,\theta)\phi(x\cos(\theta) + y\sin(\theta) - t)\,dt\,d\theta \tag{0.3.8}$$

with ϕ the filter function, used for a parallel beam scanner. In (0.3.8) weighting of different lines in the filtering operation depends only on their distance from the point(x,y). Lets write this using the polar coordinates, (r,ϕ) in the reconstruction plane $(x,y) = (r\cos(\phi), r\sin(\phi))$. In fanbeam derivation, we use κ as the filter function. In polar coordinates the equation(0.3.8) becomes,

$$f_\kappa(r,\phi) = \frac{1}{2}\int_0^{2\pi} \int_{-L}^{L} Rf(t,\theta)\kappa(r\cos(\theta-\phi) - t)\,dt\,d\theta \tag{0.3.9}$$

This reconstruction formula should be reexpressed in fanbeam coordinates, as a filtered back projection. This is not possible because of the geometry of the fanbeam coordinates. So, we have a final formula which is weighted, filtered back-projection.

Using the expressions given above for θ and t to reexpress this integral in fanbeam coordinates we get

$$f_\kappa(r,\phi) = \frac{1}{2}\int_0^{2\pi} \int_{-L}^{L} Rf(D\sin\gamma, \beta+\gamma)\kappa(r\cos(\beta+\gamma-\phi) - D\sin\gamma)D\cos\gamma\,d\gamma\,d\beta \tag{0.3.10}$$

The function f is supported in the disk of radius L. The limits of integration in the γ-integral, $\pm\gamma_L$, are chosen so that the lines correspoding to the parameters

$$\{(\beta,\gamma) : \beta \in [0,2\pi), -\gamma_L \leq \gamma \leq \gamma_L\}$$

include all those intersecting D_L. The total angle $2\gamma_L$ is called the fanangle. The data actually collected with a fanbeam machine are an approximation to uniformly spaced samples in (β,γ)-coordinates. To simplify the notation we introduce

$$Pf(\beta,\gamma) = Rf(D\sin\gamma, \beta+\gamma). \tag{0.3.11}$$

In terms of these data, the formula reads

$$f_\kappa(r,\phi) = \frac{1}{2}\int_0^{2\pi} \int_{-L}^{L} Pf(\beta,\gamma)\kappa(r\cos(\beta+\gamma-\phi) - D\sin\gamma)D\cos\gamma\,d\gamma\,d\beta \tag{0.3.12}$$

After substituting some trigonometric identities, we get

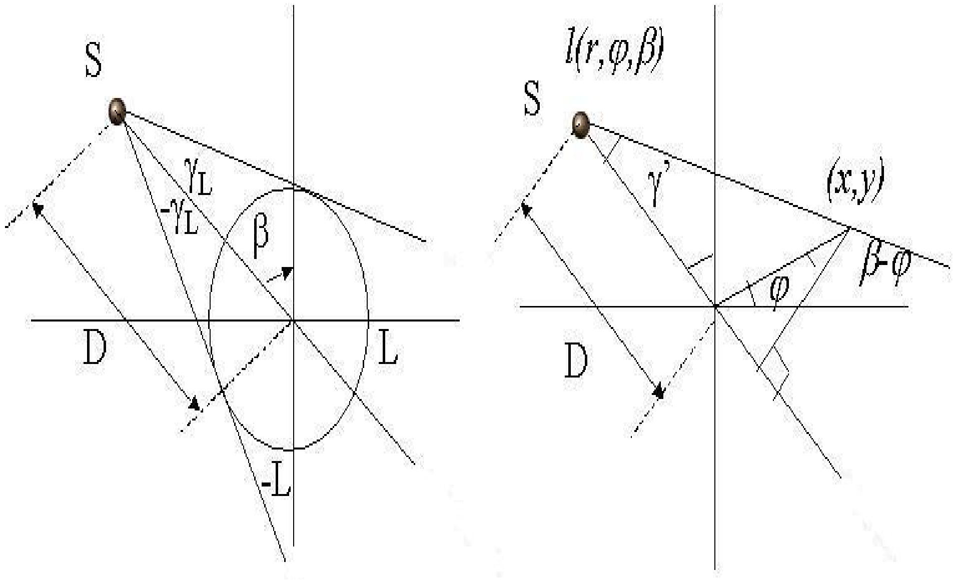


Fig. 0.10: (a)Physical parameters in a fan beam scanner (b)variables for the reconstruction formula

$$f_\kappa(r,\phi) = \frac{1}{2}\int_0^{2\pi}\int_{-L}^{L} Pf(\beta,\gamma)\kappa(l(r,\phi,\beta)\sin(\gamma'(r,\phi,\beta)-\gamma))D\cos\gamma\, d\gamma\, d\beta \tag{0.3.13}$$

where

$$l(r,\phi,\beta) = \sqrt{[D + r\sin(\beta-\phi)]^2 + [r\cos(\beta-\phi)]^2} \tag{0.3.14}$$

$$\gamma'((r,\phi,\beta)) = tan^{-1}\frac{r\cos(\beta-\phi)}{D + r\sin(\beta-\phi)} \tag{0.3.15}$$

So now the fanbeam data are $Pf(\beta_j, n\alpha)$, where $\beta_j = \frac{2\pi j}{M+1}, j = 0, \cdots, M$ and n takes the integer values. This reconstruction is given in 3 steps.

1) Replace the measurements by weighted measurements, ie. multiply by the factor $D\cos(n\alpha)$ to obtain

$$P'f(\beta_j, n\alpha) = Pf(\beta_j, n\alpha)D\cos(n\alpha) \tag{0.3.16}$$

2) Discretely convolve the weighted projection data $P'f(\beta_j, n\alpha)$ with $g(n\alpha)$ to generate the filtered projection at the sample points:

$$Q_g\tilde{f}(\beta_j, n\alpha) = \alpha[P'f(\beta_j, .) \star g](n\alpha), where \quad (0.3.17)$$

$$g(n\alpha) = \frac{1}{2}(\frac{n\alpha}{\sin(n\alpha)})^2\kappa(n\alpha) \quad (0.3.18)$$

The filter function κ should be real, even, and decay at infinity.

3) Perform a weighted back-projection of the filtered projection:

$$\tilde{f}_g(x_m, y_l) \approx \Delta\beta\sum_{k=0}^{M}\frac{1}{l^2(x_m, y_l, \beta_k)}Q_g f(\beta_k, \gamma'((x_m, y_l, \beta_k))) \quad (0.3.19)$$

0.3.3 Analysis of the Images reconstructed using Parallel beam and Fan beam techniques

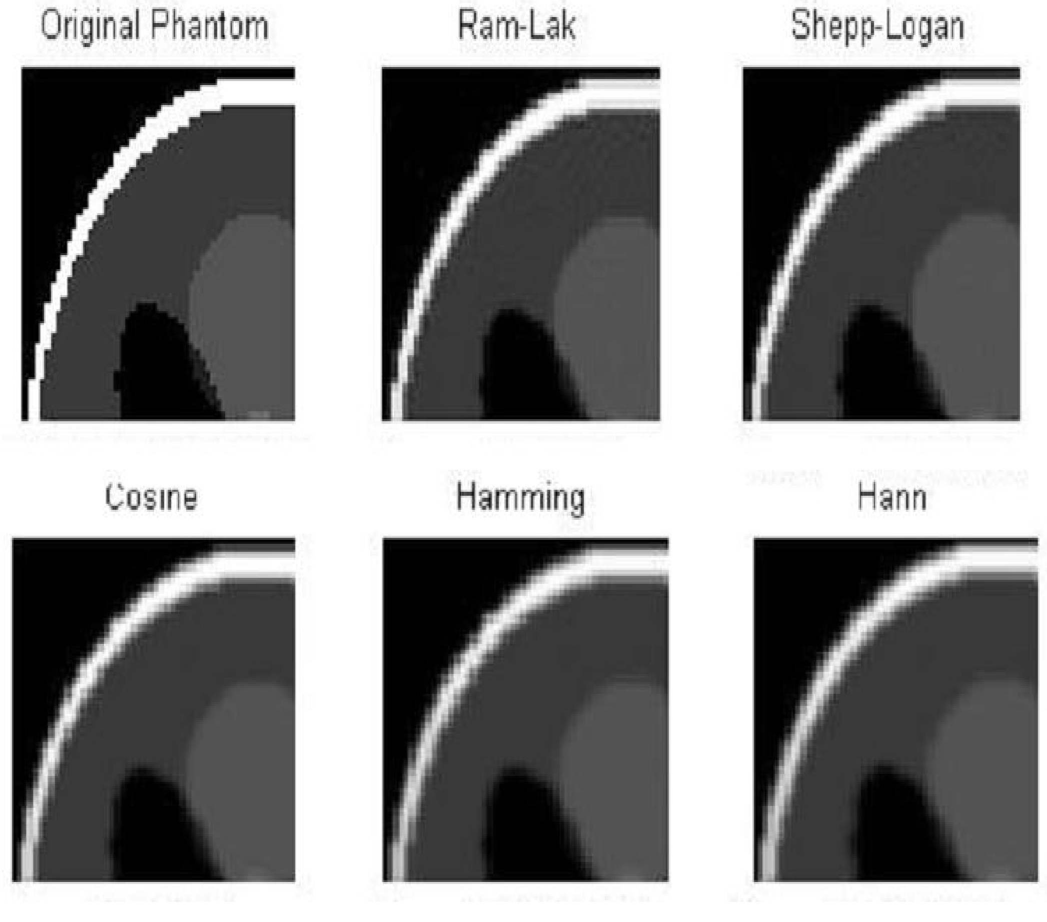


Fig. 0.11: Reconstructed Images Using Parallel beam data

From the Fig(0.11), we can observe that there are some artifacts present in the images reconstruted with Ram-Lak and Shepp-Logan filters and they are less seen in the images with other filters. But in the images reconstruted with Ram-Lak and Shepp-Logan filters, we can observe that the edges are sharp and the sharpening is decreased in the other images and some ringing artifacts are also seen towards the edges. So, basically, we can come to some conclusions observing the images but this is not a right way to measure the quality because, for one person, one image may seems to be better and for other, it may not. So, there should be some good measures of Quality of Image.

Measures of Quality of Images

Objective image quality measures play important roles in carious image processing applications. There are basically two classes of objective quality: first is mathematically defined measures such as the widely used mean squared error(MSE) and peak signal to noise ratio(PSNR), root mean squared error(RMSE) ,mean absolute error(MAE)and signal to noise ratio(SNR). The second class of measurement methods consider human visual system(HVS). The second class, complex compared to the first, did not show any clear advantage over the mathematical measures though. mathematical measures are attractive because of their low computation and also because they are independent of viewing conditions and individual observer. Few of the mathematical measures are given below:

MSE 'n' PSNR

Comparing reconstruction results requires a measure of image quality. Two commonly used measures are Mean-Squared Error and Peak Signal-to-Noise Ratio. The mean-squared error (MSE) between two images $g(x, y)$ and $\hat{g}(x, y)$ is given by

$$MSE = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} [g(m,n) - \hat{g}(m,n)]^2 \tag{0.3.20}$$

One problem with mean-squared error is that it depends strongly on the image intensity scaling. A mean-squared error of 100.0 for an image with pixel values in the range 0-255 looks dreadful; but a MSE of 100.0 for an image with pixel values in [0,1023] is barely noticeable.

Peak Signal-to-Noise Ratio (PSNR) avoids this problem by scaling the MSE according to the image range:

$$PSNR = 20\log_{10}\frac{S}{RMSE} \tag{0.3.21}$$

where S is the maximum pixel value and RMSE is the root mean squared error. PSNR is measured in decibels (dB). The PSNR measure is also not ideal, but is in common use. PSNR is a good measure for comparing image reconstruction results for the same image, but between-image comparisons of PSNR are meaningless. One image with 20 dB PSNR may look much better than another image with 30 dB PSNR.

Universal Image Quality Index

Universal image quality Index does not depend on the images being tested, the viewing conditions or the individual observers. It is applicable to various image processing applications and provide meaningful comparision across different types of noisy images.

Derivation:

Let $\mathbf{x} = \{\mathbf{x_i} : \mathbf{i} = \mathbf{1}, \mathbf{2}, \cdots, \mathbf{N}\}$ and $\mathbf{y} = \{\mathbf{y_i} : \mathbf{i} = \mathbf{1}, \mathbf{2}, \cdots, \mathbf{N}\}$ be the original and the test image respectively. Then

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2\sigma_y^2)[(\bar{x})^2 + (\bar{y})^2]} \tag{0.3.22}$$

where

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{0.3.23}$$

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{0.3.24}$$

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2 \tag{0.3.25}$$

$$\sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2 \tag{0.3.26}$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}) \tag{0.3.27}$$

The dynamic range of Q is $[-1, 1]$. The best value 1 is acheived if and only if $y_i = x_i$ for all $i = 1, 2, \cdots, N$. the lowest value -1 occurs when $y_i = 2\bar{x} - x_i$ for all $i = 1, 2 \cdots, N$. This quality index models any distortion as combination of three different factors: loss of correlation, luminance distortion and contrast distortion. We use this in the definition of Q and write it as

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \tag{0.3.28}$$

The first component is the correlation coefficient between $\mathbf{x}$ and $\mathbf{y}$, which measures the degree of linear correlation between $\mathbf{x}$ and $\mathbf{y}$ and its dynamic range is [-1,1]. The best value 1 is obtained when $y_i = ax_i + b$ for all $i = 1, 2, \cdots, N$, where a and b are constants and $a > 0$. Even if $\mathbf{x}$ and $\mathbf{y}$ are linearly related, there still might be relative distortions between them, which are evaluated in the second and third components. The second component with a value range of $[0, 1]$, meaures how close the mean luminance is between $\mathbf{x}$ and $\mathbf{y}$. It equals 1 if and only if $\bar{x} = \bar{y}$. σ_x and σ_y can be viewed as estimate of the contrast of $\mathbf{x}$ and $\mathbf{y}$, so the third component measures how similar the contrasts of the images are. Its range of values is also $[0, 1]$, where the best value 1 is achieved if and only if $\sigma_x = \sigma_y$

Image signals are generally non-stationary while image qulaity is quite often also sapce variant, although in practice it is usually desired to evaluate an entire image using a single overall quality value. Therefore, it is more appropriate to measure statistical features locally and then combine them together. Here the quality measurement method is applied the to local regions using a sliding window approach. Starting from the top-left corner of the image, a sliding window of size B*B moves pixel by pixel horizontally and verically through all the rows and columns of the image until the bottom right corner is reached. At the jth step, the local quality index Q_j is computed within the slidindg window. If there are a total of M steps then the overall quality index is given by

$$Q = \frac{1}{M} \sum_{j=1}^{M} Q_j. \tag{0.3.29}$$

I performed an anlysis of how the image quality changes with the size of the sliding window. This can be seen in the Fig(0.12) and Fig(0.13). Since from the size 32, all the values are starting to satbilize, I chose

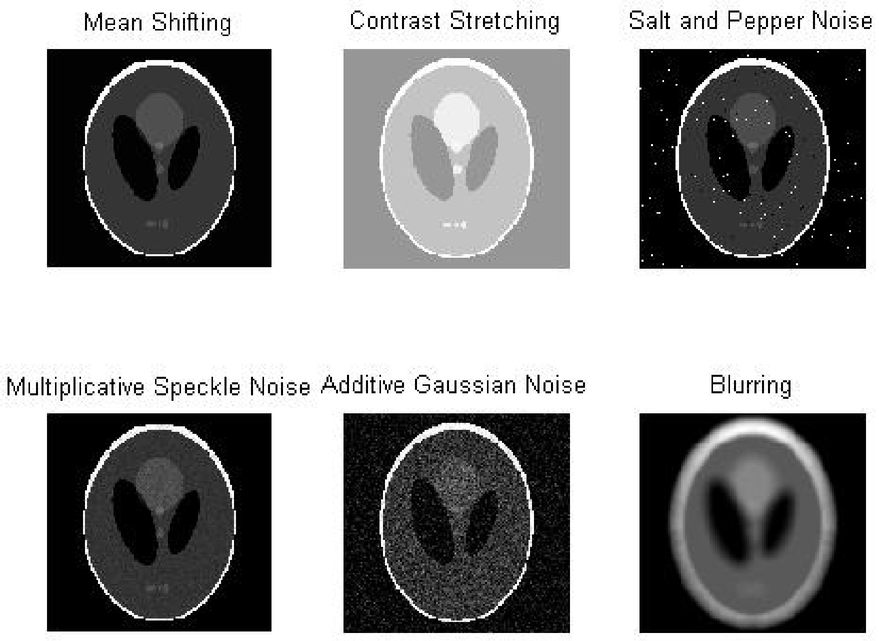


Fig. 0.12: The phantom with different types of distortion applied to it

the size of the sliding window to be 32*32 in this work.

The images with corresponding measures of quality values are seen in the fig(0.14).

From the above figure it is clear that the image quality index(QI) says that the image with Ram-Lak filter is better compared to all others. Its value is very close to the image with Shepp-Logan filter though. It is also imteresting to note that the three measures of quality are not consistent. The MSE and PSNR showed that the image with Cosine filter has better quality compared to others. The time taken for the reconstrution of each of image is 14-16 seconds.

It would be now interesting to apply these measures of quality to see what happens to these images if they are reconstructed with some noise added to the projections. This can seen in fig(0.15).

In the presence of noise, filtered-backprojection with parallel beam data performed as expected. As the standard deviation of noise is decreasing, the quality of the image is increasing. For a noise of standard deviation 0.005, the original image quality is almost obtained.

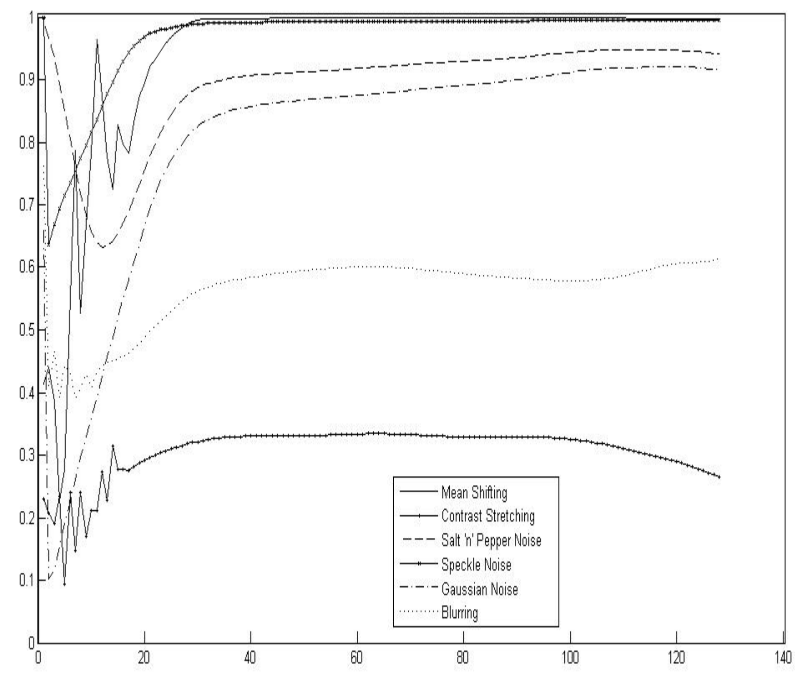


Fig. 0.13: The graph with blocksize of sliding window on x-axis and image quality on y-axis

Recontructed Images using Fan Beam data

In the fan beam reconstruction also, the image quality index(QI) says that the image with Ram-Lak filter is better compared to all others. Its value is very close to Shepp-Logan again. It is also imteresting to note that the three measures of quality are not consistent in this reconstruction as well. The MSE and PSNR showed that the image with Cosine filter has better quality compared to others. Finally we can say that the result for the fan beam reconstrution is very close to the one with parallel beam. The time taken for the reconstrution of each of image is 15-17 seconds

0.4 Iterative Reconstruction Techniques

Tomographic Image reconstruction can be approached in a different way. We may assume that the cross section consists of an array of unknowns, and then set up algebraic equations for the unknowns in terms

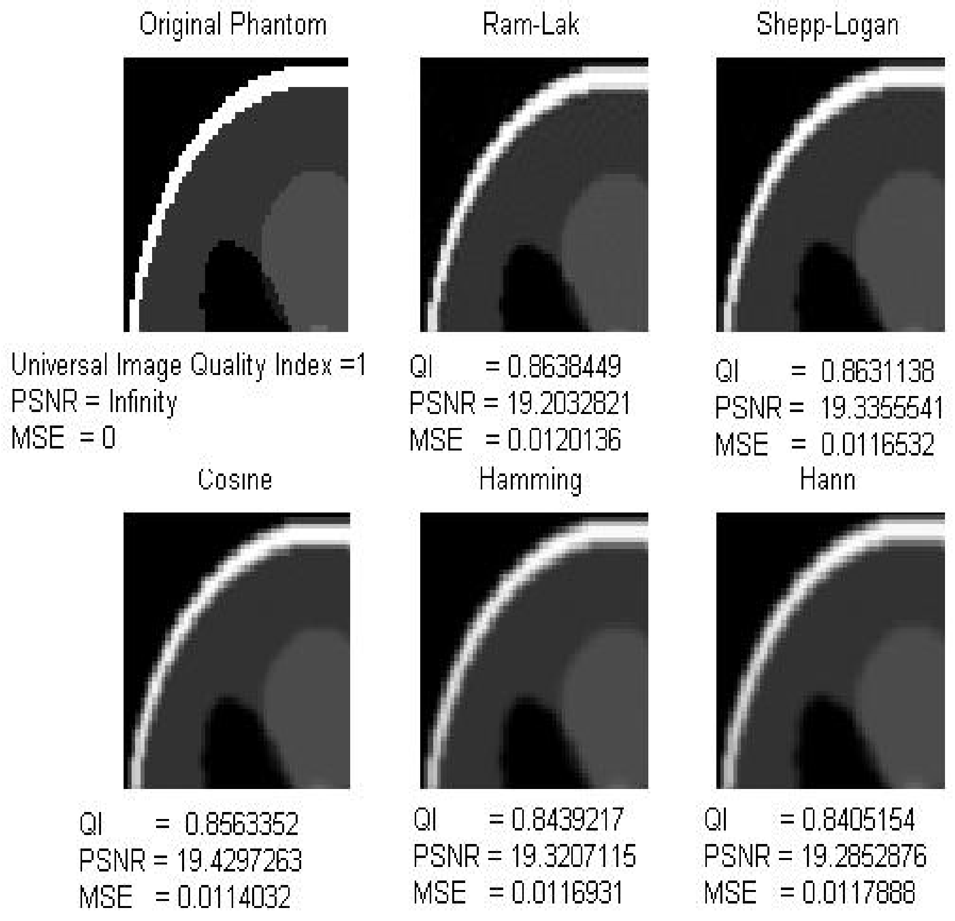


Fig. 0.14: Reconstrcted images in with parallel beam data with corresponding measures of quality values

of the measured projection data. This is conceptually a much simpler approach compared to the transform based models described above. It lacks the accuracy and speed of implementation though. This approach is useful in situations where measuring a large number of projections is not possible or measuring projections are not uniformly distributed over 180 or 360 both these conditions being necessary requirements for the transform based techniques to produce results with the accuracy desired in medical imaging. An example of such a situation is earth resources imaging using cross-borehole measurements.

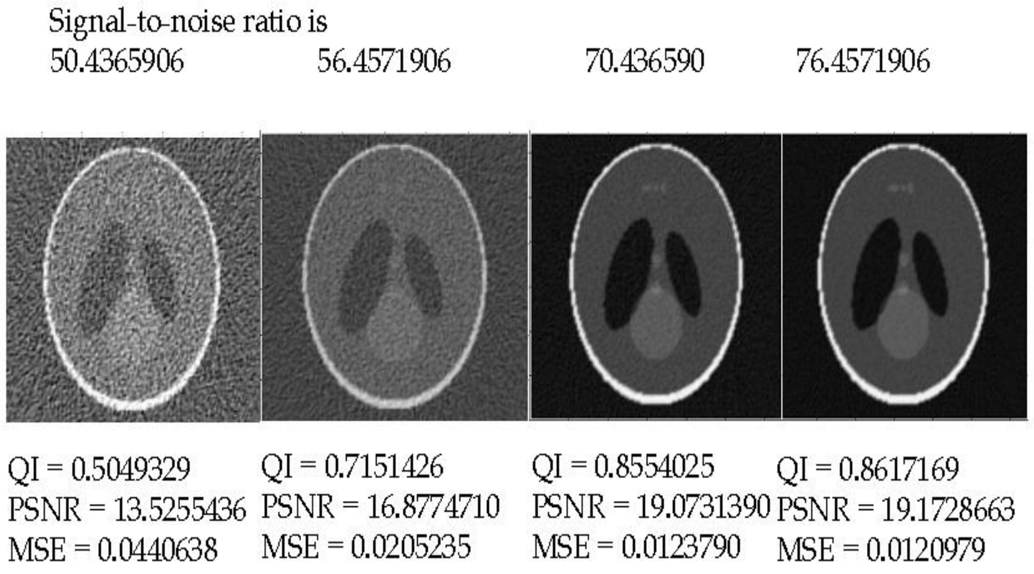


Fig. 0.15: Reconstructed Images using Parallel beam data with added noise of standard deviation 0.1, 0.05, 0.01, 0.005 respectively to the projections

0.4.1 Representation of Image and its Projection

We basically superimpose a square grid on the image $f(x, y)$; we will assume that in each cell the function $f(x, y)$ is constant. Let f_j denote this constant value in the jth cell, and let N be the total number of cells. For algebraic techniques a ray is defined somewhat differently. A ray is now a 'fat' line running through the (x, y) -plane. This can be seen as the ith ray(shaded) in Fig(0.18), where each ray is of width τ. In most cases the ray width is approximately equal to the image cell width. A line integral will now be called a ray-sum. Like the image, the projections will also be given a one-index representation. Let p_i be the ray-sum measured with the ith ray as shown in Fig(0.18). The relationship between the f_js and p_is may be expressed as

$$\sum_{j=1}^{N} w_{ij} f_j = p_i, \quad i = 1, 2, \cdots, M \tag{0.4.1}$$

where M is the total number of rays(in all the projections) and w_{ij} is the weighting factor that represents the contribution of the jth cell to the ith ray integral. The factor w_{ij} is equal to the fractional area of the

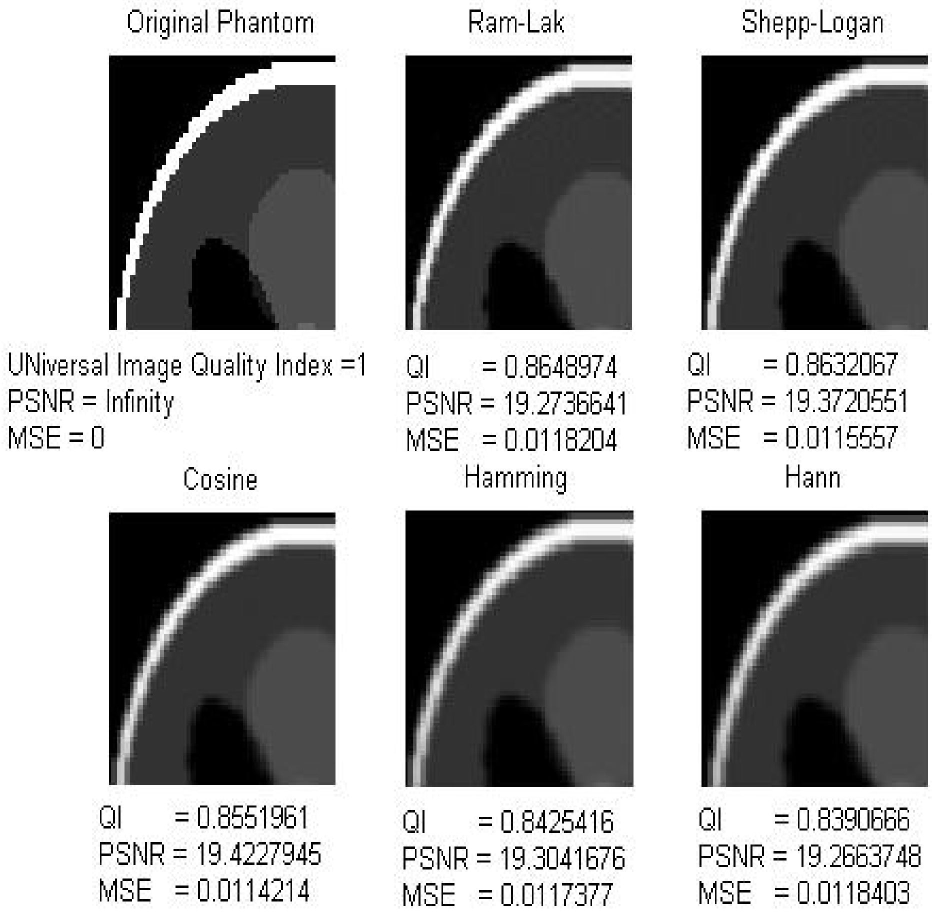


Fig. 0.16: Recontructed Images using Fan Beam data

jth image cell intercepted by the ith ray as shown for one of the cells in Fig(0.18). Most of the w_{ij}s are zero since only a small number of cells contribute to any given ray-sum. If M and N were small, we could use conventional matrix theory methods to invert the system of equation (0.4.1). However, in practice N may be as large as 65,000 (for 256 x 256 images), and, in most cases for images of this size, M will also have the same magnitude. For these values of M and N the size of the matrix $[w_{ij}]$ in equation(0.4.1) is $\approx$ 65,000 X 65,000 which precludes any possibility of direct matrix inversion. Of course, when noise is present in the measurement data and when $M < N$, even for small N it is not possible to use direct matrix inversion, and some least squares method may have to be used. When both M and N are large, such methods are also computationally impractical. For large values of M and N there exist very attractive iterative methods for

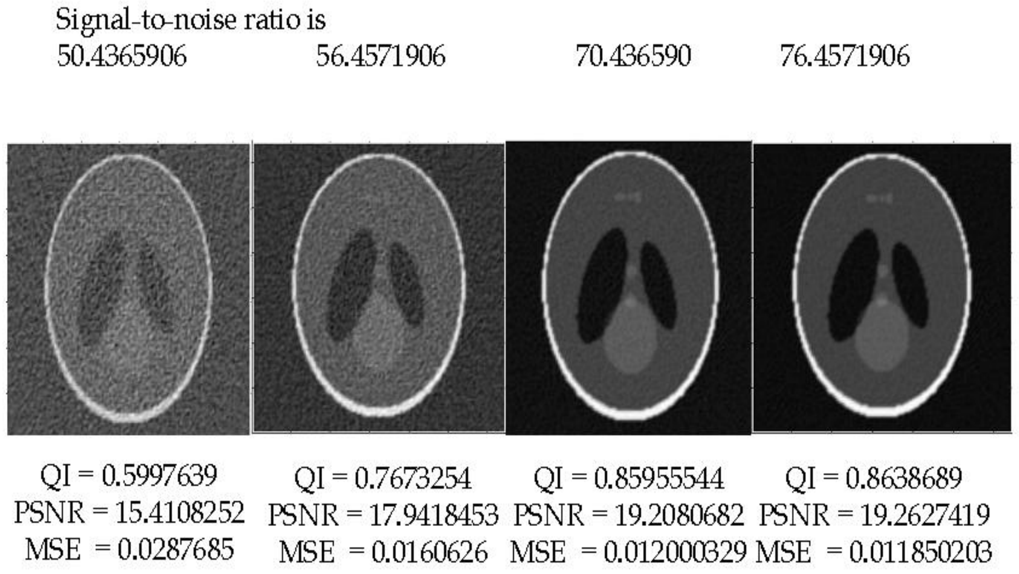


Fig. 0.17: Reconstructed Images using Fan beam data with added noise of standard deviation 0.1, 0.05, 0.01, 0.005 respectively to the projections

solving equation(0.4.1). These are based on the method of projections as first proposed by Kaczmarz. To explain the computational steps involved in these methods, we first write equation(0.4.1) in an expanded form:

$$w_{11}f_1 + w_{12}f_2 + w_{13}f_3 + \cdots + w_{1N}f_N = p_1 \tag{0.4.2}$$

$$w_{21}f_1 + w_{22}f_2 + w_{23}f_3 + \cdots + w_{1N}f_N = p_2 \tag{0.4.3}$$

$$\vdots \tag{0.4.4}$$

$$w_{M1}f_1 + w_{M2}f_2 + w_{M3}f_3 + \cdots + w_{MN}f_N = p_M \tag{0.4.5}$$

A grid representation with N cells gives an image N degrees of freedom. Therefore, an image, represented by $(f_1, f_2, ...f_N)$, may be considered to be a single point in an N-dimensional space. In this space each of the above equations represents a hyperplane. When a unique solution to these equations exists, the intersection of all these hyperplanes is a single point giving that solution. We will consider a simple example where $N = 2$. This can seen in Fig(0.19) where the two variables f_1, and f_2 satisfying the following equations:

$$w_{i1}f_1 + w_{i2}f_2 = p_i, \quad i = 1, 2 \tag{0.4.6}$$

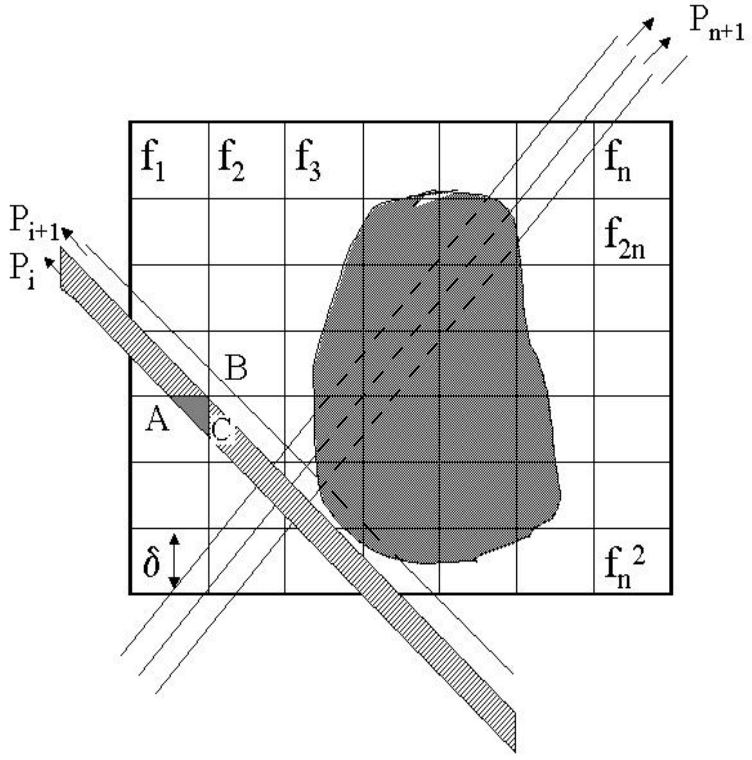


Fig. 0.18: Square grid is superimposed over an image. Image values are taken as constants

The computational procedure for locating the solution in Fig(0.19) consists of first starting with an initial guess, projecting this initial guess on the first line, reprojecting the resulting point on the second line, and then projecting back onto the first line, and so forth. If a unique solution exists, the iterations will always converge to that point. For the computer implementation of this method, we first make an initial guess at the solution. This guess, denoted by $f_1^{(0)}, f_2^{(0)}, f_3^{(0)}, ...f_n^{(0)}$, is represented vectorially by $\overrightarrow{f^{(0)}}$ in the N-dimensional space. In most cases, we simply assign a value of zero to all the f_is. This initial guess is projected on the hyperplane represented by the first equation(0.4.2) giving $\overrightarrow{f^{(0,1)}}$, as can be seen in Fig(0.19) for the two-dimensional case, $\overrightarrow{f^{(0,1)}}$ is projected on the hyperplane represented by the $i = 2$ equation(0.4.2) to yield $\overrightarrow{f^{(0,2)}}$ and so on. This iterate is equal to $\overrightarrow{f^{(1)}}$. So, covering all the hyperplanes constitutes one iteration. When $\overrightarrow{f^{(i-1)}}$ is projected on the hyperplane represented by the ith equation to yield $f^{\vec{(i)}}$ the process can be

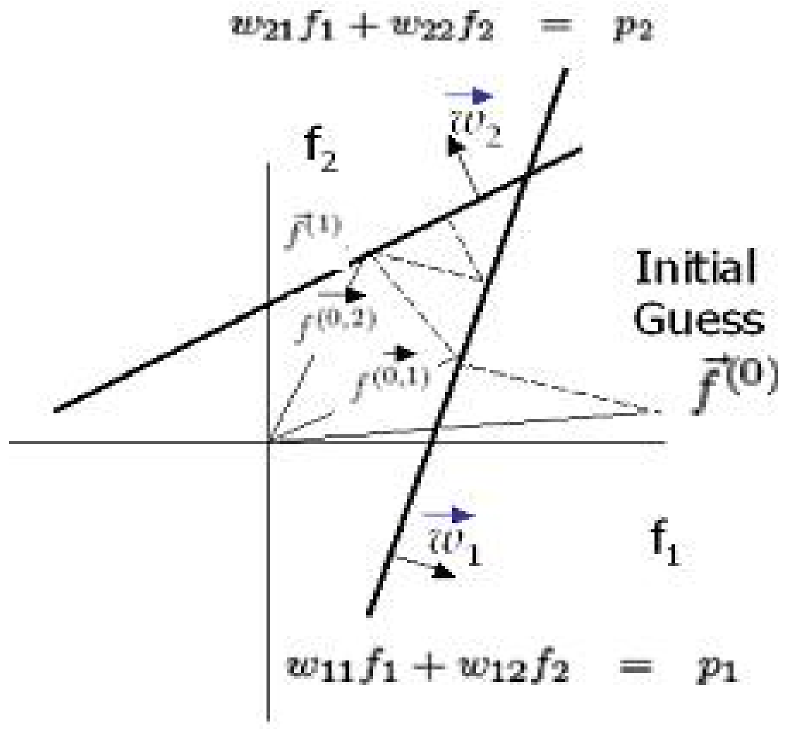


Fig. 0.19: The Kaczmarz method of solving algebraic equations

mathematically described by

$$\overrightarrow{f^{(j,i)}} = \overrightarrow{f^{(j,i-1)}} - \frac{(\overrightarrow{f^{(j,i-1)}}(\vec{w_i}) - p_i)}{(\vec{w_i}) \cdot (\vec{w_i})}(\vec{w_i}), j = 1 \cdots M \tag{0.4.7}$$

where $\vec{w_i} = (w_{i1}, w_{i2}...w_{iN})$, and $\vec{w_i} \cdot \vec{w_i}$ is the dot product of $\vec{w_i}$ with itself. To see how equation(0.4.7) comes about we write the first equation(0.4.1) as follows:

$$\vec{w_1} \cdot \vec{f_1} = p_1$$

For convenience, we now take the iterates as $f^{(1)}$ and $f^{(2)}$ instead of $\overrightarrow{f^{(0,1)}}$ and $\overrightarrow{f^{(0,2)}}$. The hyperplane represented by this equation is perpendicular to the vector $\vec{w_1}$. This is illustrated in Fig(0.20), where the vector $\overrightarrow{OD}$ represents $\vec{w_1}$. This equation simply says that the projection of a vector $\overrightarrow{OC}$(for any point C on the hyperplane) on the vector $\vec{w_1}$ is of constant length. The unit vector $\overrightarrow{OU}$ along $\vec{w_1}$, is given by

$$\overrightarrow{OU} = \frac{\vec{w_1}}{\sqrt{\vec{w_1} \cdot \vec{w_1}}} \tag{0.4.8}$$

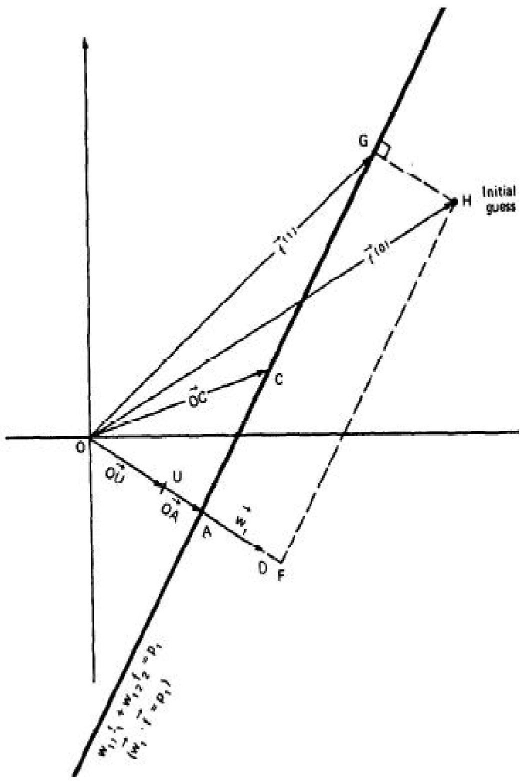


Fig. 0.20: The hyperplane represented by a line in this two-dimensional figure

and the perpendicular distance of the hyperplane from the origin, which is equal to the length of $\overrightarrow{OA}$ in Fig(0.20), is given by $\overrightarrow{OC} \cdot \overrightarrow{OU}$

$$
\begin{aligned}
|\overrightarrow{OA}| &= \overrightarrow{OU} \cdot \overrightarrow{OC} && (0.4.9)\\
&= \frac{\vec{w_1}}{\sqrt{\vec{w_1} \cdot \vec{w_1}}}(\vec{w_1} \cdot \overrightarrow{OC}) && (0.4.10)\\
&= \frac{\vec{w_1}}{\sqrt{\vec{w_1} \cdot \vec{w_1}}}(\vec{w_1} \cdot \vec{f}) && (0.4.11)\\
&= \frac{p_1}{\sqrt{\vec{w_1} \cdot \vec{w_1}}} && (0.4.12)
\end{aligned}
$$

Now to get $\overrightarrow{f^{(1)}}$ we have to subtract from $\overrightarrow{f^{(0)}}$ the vector $\overrightarrow{HG}$ i.e

$$\overrightarrow{f^{(1)}} = \overrightarrow{f^{(0)}} - \overrightarrow{HG} \tag{0.4.13}$$

where the length of the vector $\overrightarrow{HG}$ is given by

$$|\overrightarrow{HG}| = |\overrightarrow{OF}| - |\overrightarrow{OA}| \tag{0.4.14}$$

$$= \overrightarrow{f^{(0)}} \cdot \overrightarrow{OU} - \overrightarrow{OA} \tag{0.4.15}$$

Substituting equation(0.4.8) and equation(0.4.14) in this equation, we get

$$\overrightarrow{HG} = \frac{\overrightarrow{f^{(0)}} \cdot \vec{w_1} - p_1}{\sqrt{\vec{w_1} \cdot \vec{w_1}}} \tag{0.4.16}$$

Since the direction of $\overrightarrow{HG}$ is the same as that of the unit vector $\overrightarrow{OU}$, we can write

$$\overrightarrow{HG} = |\overrightarrow{HG}|\overrightarrow{OU} = \frac{\overrightarrow{f^{(0)}} \cdot \vec{w_1} - p_i}{\sqrt{\vec{w_1} \cdot \vec{w_1}}}\vec{w_1} \tag{0.4.17}$$

Substituting equation(0.4.17) in equation(0.4.13), we get equation(0.4.7). As mentioned before, the computational procedure for algebraic reconstruction consists of starting with an initial guess for the solution, taking successive projections on the hyperplanes represented by the equations in (0.4.1), eventually yielding $\overrightarrow{f^{(M)}}$. In the next iteration, $\overrightarrow{f^{(M)}}$ is projected on the hyperplane represented by the first equation in (0.4.1), and then successively onto the rest of the hyperplanes in equation(0.4.1), to yield $\overrightarrow{f^{(2M)}}$, and so on. It has been shown that if there exists a unique solution $\overrightarrow{f_s}$ to the system of equations(0.4.1) , then

$$\lim_{k \to \infty} \overrightarrow{f^{(kM)}} = \overrightarrow{f_s} \tag{0.4.18}$$

A few comments about the convergence of the algorithm are in order here. If in Fig(0.19), if the two hyperplanes are perpendicular to each other, this may be easily shown that given for an initial guess any point in the (f_1, f_2)-plane, it is possible to arrive at the correct solution in only two steps like equation(0.4.7). On the other hand, if the two hyperplanes have only a very small angle between them, it can take large number of iterations, depending upon the initial guess, before the correct solution is reached. Clearly the angles between the hyperplanes considerably influence the rate of convergence to the solution. If the M hyperplanes in equation(0.4.1) could be made orthogonal with respect to one another, the correct solution would be arrived at with only one pass through the M equations, assuming a unique solution does exist. This is not feasible computationally though. Full orthogonalization will also tend to enhance the effects of the ever present measurement noise in the final solution. A simpler technique, is to carefully choose the order in which the hyperplanes are considered. Since each hyperplane represents a distinct ray integral, it is quite likely that adjacent ray integrals (and thus hyperplanes) will be nearly parallel. By choosing hyperplanes representing widely separated ray integrals, it is possible to improve the rate of convergence of the Kaczmarz approach. A not uncommon situation in image reconstruction is that of an overdetermined system in the presence of

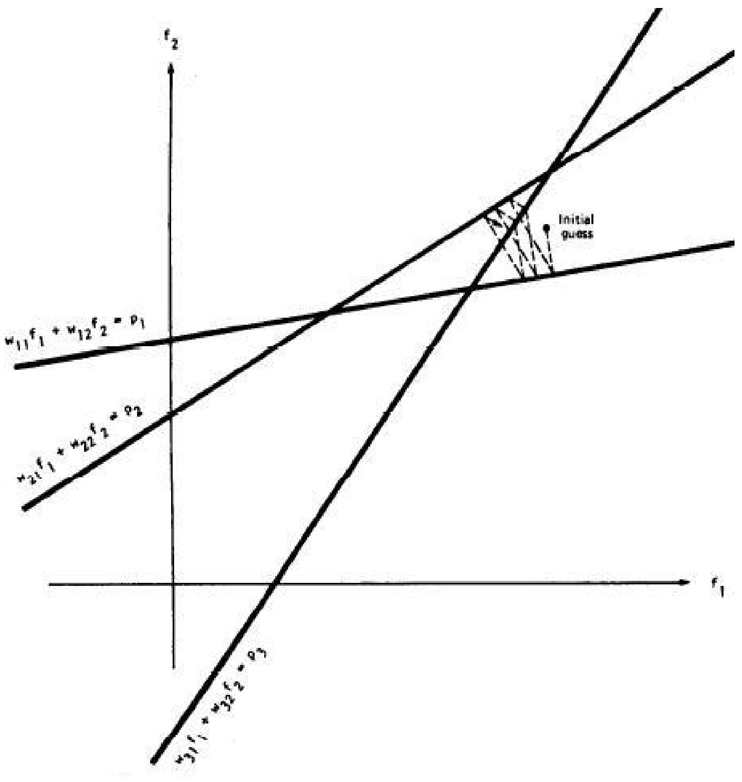

f_2
f_1
Initial guess
$w_{11}f_1 + w_{12}f_2 = p_1$
$w_{21}f_1 + w_{22}f_2 = p_2$
$w_{31}f_1 + w_{32}f_2 = p_3$

Fig. 0.21: Three noisy hyperplanes

measurement noise. That is, we may have $M > N$ in equation(0.4.1) and $p_1, p_2, ..., p_m$ corrupted by noise. No unique solution exists in this case. In Fig.(0.21) we have shown a two-variable system represented by three noisy hyperplanes. The broken line represents the course of the solution as we successively implement equation(0.4.7). Now the solution doesn't converge to a unique point, but will oscillate in the neighborhood of the intersections of the hyperplanes.

When $M < N$ a unique solution of the set of linear equations(0.4.1) doesn't exist, and, in fact, an infinite number of solutions are possible. For example, suppose we have only the first of the two equations in equations(0.4.6) to use for calculating the two unknowns f_1, and f_2; then the solution can be anywhere on the line corresponding to this equation. Given the initial guess $\overrightarrow{f^{(0)}}$ (seen in Fig(0.20), the best one could probably do under the circumstances would be to draw a projection from $\overrightarrow{f^{(0)}}$ on this line, and call the resulting $\overrightarrow{f^{(1)}}$ a solution. Here the solution obtained in this manner corresponds to that point on the line which is closest to the initial guess. This result has been shown that when $M < N$, the iterative approach described above converges to a solution, say $\overrightarrow{f'_s}$ such that $|\overrightarrow{f^{(0)}} - \overrightarrow{f'_s}|$ is minimized.

There are two attractive features of this iterative approach. One is its computational efficiency and another is that it is now possible to incorporate into the solution some types of a priori information about the image one is reconstructing. For example, if it is known a priori that the image $f(x, y)$ is nonnegative, then in each of the solutions $\overrightarrow{f^{(k)}}$, successively obtained by using equation(0.4.7), we may set the negative components equal to zero. We may similarly incorporate the information that $f(x, y)$ is zero outside a certain area, if this is known.

In some applications, we may require a large number of views and where large-sized reconstructions are made, the difficulty with using equation(0.4.7) can be in the calculation, storage, and fast retrieval of the weight coefficients w_{ij}s. Lets consider the case where we wish to reconstruct an image on a 100 x 100 grid from 100 projections with 150 rays in each projection. The total number of weights, w_{ij} needed in this case is 10^8, which is an enormous number and can pose problems in fast storage and retrieval in applications where reconstruction speed is important. This problem is somewhat eased by making approximations, such as considering w_{ij} to be only a function of the perpendicular distance between the center of the ith ray and the center of the jth cell. This perpendicular distance can then be computed at run time. To get around the implementation difficulties caused by the weight coefficients, a myriad of other algebraic approaches have also been suggested, many of which are approximations to equation(0.4.7). To discuss some of the more implementable approximations, we first see equation(0.4.7) in a slightly different form:

$$\overrightarrow{f^{(i)}} = \overrightarrow{f^{(i-1)}} + \frac{(p_i - q_i)}{\sum_{k=1}^{N} (\vec{w_{ik}^2})} (\vec{w_i}) \qquad (0.4.19)$$

where

$$
\begin{aligned}
q_i &= \overrightarrow{f^{(i-1)}} \cdot \overrightarrow{w_i} && (0.4.20)\\
&= \sum_{k=1}^{N} \overrightarrow{f_k^{(i-1)}} \cdot w_{ik} && (0.4.21)
\end{aligned}
$$

These equations say that when we project the $(i-1)$th solution onto the ith hyperplane (ith equation in (0.4.1)) the gray level of the jth element, whose current value is $f_j^{(i-1)}$ is obtained by correcting its current value by $\Delta f_j^{(i)}$, where

$$\Delta f_j^{(i)} = f_j^{(i)} - f_j^{(i-1)} = \frac{(p_i - q_i)}{\sum_{k=1}^{N} (w_{ik})^2} w_{ij} \qquad (0.4.22)$$

It is clear that while p_i is the measured ray-sum along the ith ray, q_i may be considered to be the computed ray-sum for the same ray based on the $(i-1)$th solution for the image gray levels. The correction Δf_j, to the jth cell is obtained by first calculating the difference between the measured ray-sum and the computed ray-sum, normalizing this difference by $\sum_{k=1}^{N} (w_{ik})^2$, and then assigning this value to all the image cells in the ith ray, each assignment being weighted by the corresponding w_{ij}. With the preliminaries presented above, we will now discuss three different computer implementations of algebraic algorithms. These are represented by the acronyms ART, SIRT, and SART.

0.4.2 Algebraic Reconstruction Techniques(ART)

In many ART implementations the w_{ik}s in equation(0.4.22) are simply replaced by 1s and 0s, depending upon whether the center of the kth image cell is within the ith ray. This makes the implementation easier because such a decision can easily be made at computer run time. In this case the denominator in equation(0.4.22) is given by $\sum_{k=1}^{N} w_{ik}^2 = N_i$ which is the number of image cells whose centers are within the ith ray. The correction to the jth image cell from the ith equation in (0.4.1) may now be written as

$$\Delta f_j^{(i)} = \frac{p_i - q_i}{N_i} \qquad (0.4.23)$$

for all the cells whose centers are within the ith ray. We are essentially smearing back the difference $(p_i - q_i)/Ni$ over these image cells. In equation(0.4.23), q_is are calculated using the expression in (0.4.20), except that one now uses the binary approximation for w_{ik}s. The approximation in equation(0.4.23), although easy to implement, often leads to artifacts in the reconstructed images, especially if N_i isn't a good approximation to the denominator. Superior reconstructions may be obtained if equation(0.4.23) is replaced by

$$\Delta f_j^{(i)} = \frac{p_i}{L_i} - \frac{q_i}{N_i} \tag{0.4.24}$$

where L_i is the length normalized by δ^2, i.e $L_i = \frac{areaofABC}{\delta^2}$ where δ is the length of side of each pixel. This can be seen in Fig(0.18) of the ith ray through the reconstruction region. ART reconstructions usually suffer from salt and pepper noise, which is caused by the inconsistencies introduced in the set of equations by the approximations commonly used for w_{ik}s. The result is that the computed ray sums in equation(0.4.20) are usually poor approximations to the corresponding measured ray-sums. The effect of such inconsistencies is exacerbated by the fact that as each equation corresponding to a ray in a projection is taken up, it changes some of the pixels just altered by the preceding equation in the same projection. The SIRT algorithm described briefly below also suffers from these inconsistencies in the forward process , but by eliminating the continual and competing pixel update as each new equation is taken up, it results in smoother reconstructions. It is possible to reduce the effects of this noise in ART reconstructions by relaxation, in which we update a pixel by $\alpha \cdot \Delta f_j^{(i)}$, where α is less than 1. In some cases, the relaxation parameter α is made a function of the iteration number; that is, it becomes progressively smaller with increase in the number of iterations. The resulting improvements in the quality of reconstruction are usually at the expense of convergence.

0.4.3 Simultaneous Iterative Reconstruction Techniques (SIRT)

This approach, at the expense of slower convergence usually leads to better looking images than those produced by ART. In this we again use equation(0.4.23) or equation(0.4.24) to compute the change $\Delta f_j^{(i)}$ in the jth pixel caused by the ith equation in (0.4.1). However, the value of the jth cell isn't changed at this time. Before making any changes, we go through all the equations, and then only at the end of each iteration are the cell values changed, the change for each cell being the average value of all the computed changes for that cell. This constitutes one iteration of the algorithm. In the second iteration, we go back to the first equation in (0.4.23) and the process is repeated.

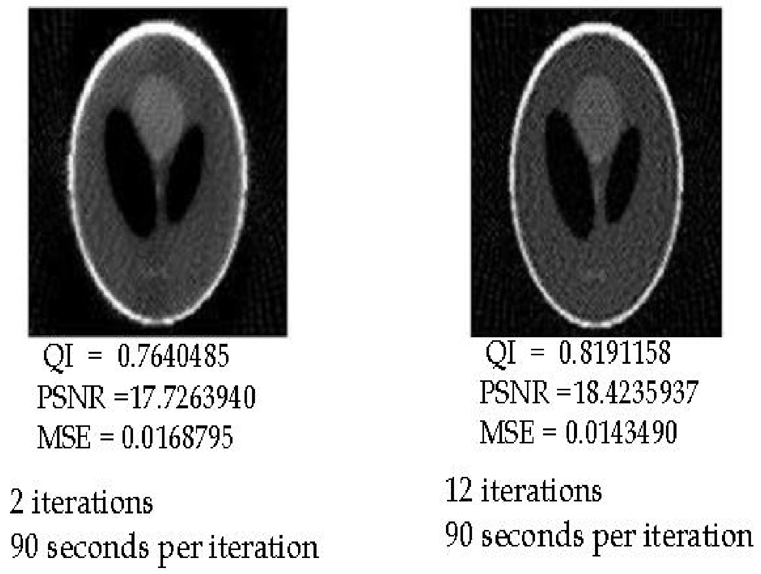


Fig. 0.22: Reconstructed Images using Algebraic Reconstruction Techniques

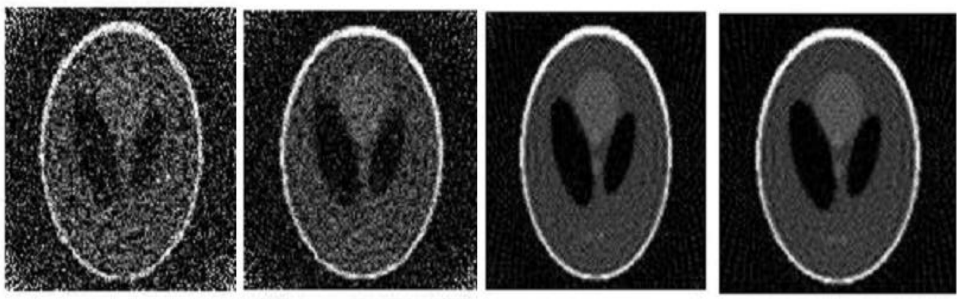
QI says that the ART did not perform any better than Filtered back-projection. Even the PSNR and MSE are less compared to the Filtered back-projection. Bascially the artifacts are seen in the inner region and outer region as well. As the number of iterations increase it did not give any better result after ceratin number of iterations. Even the time taken is more when compared to the Filtered back-projection. In the presence of noise, ART performed similar to the filtered-backprojection. As the standard deviation of noise is decrease, the quality of the image is increasing. For a noise of standard deviation 0.005, the original image quality is almost obtained. SIRT gave a good result compared to ART in terms of QI, PSNR and MSE. But if we compare the times taken, it is worst comapred to the normal Filtered back-projection and ART. It is taking some hours to reconstruct an image.

Main Draw back of Iterative Reocnstruction Techniques

Though Iterative Reconstruction Techniques are simpler when compared to the Filtered back-projection, it did not give the good quality of the image. In Iterative Reocnstruction Techniques, convergence is proved in theory but in practice since the condition number is huge. Here is a table which shows the condition number

Signal-to-noise ratio is

50.4365906 56.4571906 70.436590 76.4571906



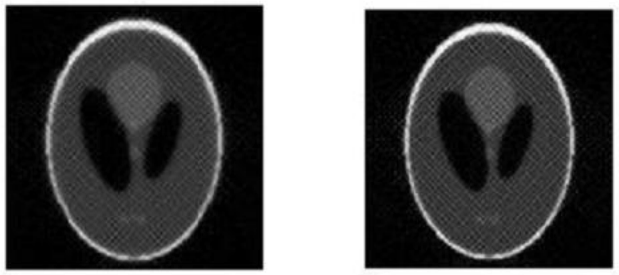
QI = 0.4701038
PSNR =11.0462363
MSE = 0.0785916

QI=0.6651730
PSNR=15.1193539
MSE=0.0307655

QI = 0.8119189
PSNR = 18.2337723
MSE = 0.0150183

QI= 0.8166475
PSNR = 18.3693961
MSE = 0.0145566

Fig. 0.23: Reconstructed Images using ART in the presence of noise of standard deviation 0.1 ,0.05, 0.01, 0.005 respectively



QI = 0.7942563
PSNR = 18.6115284
MSE = 0.0137672

QI = 0.8030332
PSNR = 18.6003567
MSE = 0.0138027

50 iterations

4 hours

75 iterations

5.5 hours

Fig. 0.24: Images reconstructed using SIRT

as the image size increases.

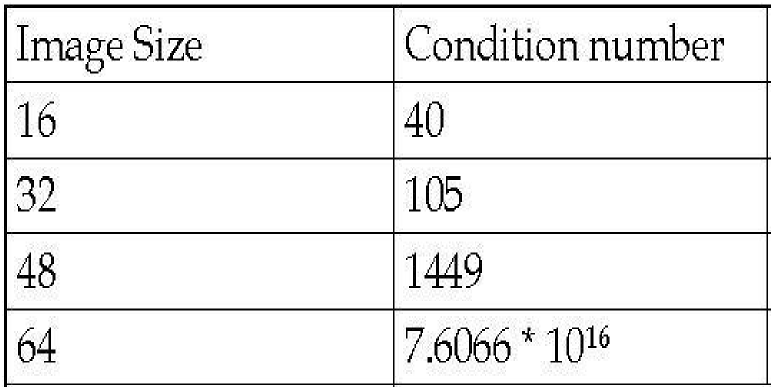


Image Size	Condition number
16	40
32	105
48	1449
64	$7.6066 * 10^{16}$

Fig. 0.25: Condition numbers of Weight matrices corresponding to the size of the image

0.5 Fast Slant Stack

In this method of reconstruction of an image, Radon Transform is intrinsically defined on an n-by-n grid rather than through the approximation of continuum integers. This basically is the summation along lines of absolute slope less than 1 as a function either of x or of y, with the values at non-Cartesian locations which are defined using trigonometric interpolation. The definition is geometrically faithful since the lines exhibit no wraparound effects. These lines in one projection are equispaced in slope. For these, an exact algorithm is described, which uses $O(N \log N)$ flops, where $N = n^2$ is the number of pixels. This is based on a discrete central-slice theorem, relating this Radon transform and the Pseudopolar Fourier transform. The Pseudopolar FT evaluates the $2-D$ Fourier transform on a non-Cartesian point set, on the so called pseudopolar grid. This Radon transform is invertible on its range since it is one-to-one; it is rapidly invertible to any degree of desired accuracy using a preconditioned conjugate gradient solver. The numerical conditioning is very good and three iterations of the conjugate gradient solver typically suffice for 6 digit accuracy.

0.5.1 Definition:

In this new notion, Radon transform is thought of an object that assigns a numerical value to each member of a family of lines. The lines are parameterized somewhat differently than in the traditional way of Radon transforms, using slopes and intercepts rather than angles and offsets. This parameterization is natural for dealing with data on Cartesian grids. These lines are of two types in this definition, a *basically horizontal line* of the form $y = sx + z$, with the slope $|s| \leq 1$; a *basically vertical line* of the form $x = sy + z$, with the slope $|s| \leq 1$. This separation of lines into two classes is required to maintain two separate but related data structures, based on interchange of roles of x and y. Given an array $I(u, v)$, a slope s with $|s| \leq 1$, and an offset z, the Radon transform associated with the basically horizontal line $y = sx + z$ is

$$Radon(\{y = sx + z\}, I) = \textstyle\sum_u \tilde{I^1}(u, su + z)$$

Thus, n values $(u, su + z)$ are assumed along the line $y = sx + z$. The values that are being summed come from a trigonometric interpolant $\tilde{I^1}(u, y)$ of the original image I. The interpolation in y is performed as follows. The Dirichlet kernel of order m is

$$D_m(t) = \frac{\sin(\pi t)}{m \sin(\pi t/m)}$$

Letting $m = 2n$, the trigonometric interpolant is given by

$$\tilde{I^1}(u, v) = \sum_{v=-n/2}^{n/2-1} I(u, v) D_m(y - v)$$

We note that this is an interpolating kernel, so that

$$I(u, v) = \tilde{I^1}(u, v), -n/2 \leq u, v < n/2.$$

The Radon transform for basically vertical lines is a similar case interchanging roles of x & y:

$$Radon(\{x = sy + z\}, I) = \sum_v \tilde{I^2}(sv + z, v)$$

with the interpolant defined analogously:

$$\tilde{I^2}(x, v) = \sum_{u=-n/2}^{n/2-1} I(u, v) D_m(x - u)$$

Let θ represent the angle associated to the slope s. This gives a definition, for $\theta \in [-\pi/4, \pi/4)$,

$$(RI)(t, \theta) = Radon(\{y = \tan(\theta)x + t\}, I) \tag{0.5.1}$$

and, for $\theta \in [-\pi/4, 3\pi/4)$,

$$(RI)(t, \theta) = Radon(\{x = \cot(\theta)y + t\}, I) \tag{0.5.2}$$

Lets consider the lines having an intercept $-n \leq t < n$, and let T_n denote this set of intercept values. Because the array $I(u,v)$ has only $N = n^2$ entries, N pieces of Radon information characterize I. The choice of the angles $\theta_l^1 = \arctan(2l/n), -n/2 \leq l < n/2$ and $\theta_l^2 = \pi/4 + \arctan(2l/n)$ is fixed. These are not equispaced in angle but instead in slope, having $s = 2l/n, -n/2 \leq l < n/2$. Let Θ_n denote this set of angles. This set of angles has some very special properties. The first sign of this is that basically horizontal lines make an integer vertical displacement as they traverse from right to left of the image. Similarly, the corresponding basically vertical lines exhibit an integer horizontal displacement as they traverse the image from top to bottom. Therefore, these angles are called "grid-friendly". The object $RI = (RI(t,\theta) : t \in T_n, \theta \in \Theta_n)$, defined with the grid-friendly angles in Θ_n and intercepts in T_n, may be viewed as a result of mapping from the space of $n - by - n$ arrays I to the space of $2n - by - 2n$ arrays RI. This mapping is called the Radon transform and is represented by R.

0.5.2 Properties of Radon Transform(Slant stack):

The properties of this Radon transform are listed below

1) Discrete Central-Slice Theorem: $This\ shows\ that\ the\ 1-D\ Fourier\ transform\ of\ R_\theta I\ gives\ the\ values\ in\ a\ radial\ slice\ of\ the\ 2-D\ Fourier\ transform\ of\ I.$ ie

$$\widehat{R_\theta I}(k) = \hat{I}(\pi \tfrac{k}{n} \tan(\theta), \pi \tfrac{k}{n})$$

From the central-slice theorem, one can obtain simultaneously a number of values of R starting from Fourier domain information. If one wants R at all values $t \in T_n$, $\theta \in \Theta_n$, one needs to know the Fourier transform, $\hat{I}$ at all values in a certain non-Cartesian pointset, so called the Pseudopolar grid. This grid is illustrated in Fig(0.26) for the case $n = 8$. This is not the usual Cartesian grid for which the fast Fourier transform is well-known. Some ideas provide FFT for this grid, operating in $O(Nlog(N))$ flops.

Here, few remarks should be made about the interpolation method. The choice $m = 2n$ means that, in the case of I^1, the interpolation scheme is algebraically identical to embedding the image I in an array that is m tall and n wide, with zero padding by $n/2$ rows of zeros both above and below the array, and using trigonometric interpolation of degree m within each column of the array. Second, is that summing along a

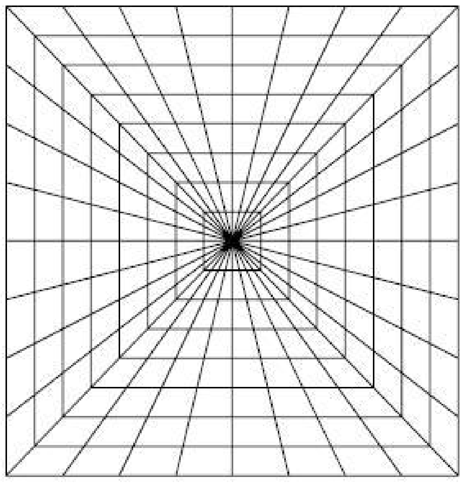


Fig. 0.26: The Pseudopolar Grid for $n = 8$

line $y = sx + z$ is equivalent to first shifting each column vertically by an amount $-su - z$ using trigonometric interpolation, and then summing along $y = 0$.

Definition as Slant Stack From the above observations, a different definition of R can be formed. Let E^1 be the operator of extension, padding an array n wide and n tall to be n wide and m tall by adding extra rows($n/2$) of zeros symmetrically above and below the input argument. Let E^2 be the operator of extension, padding an array n wide and n tall to be m wide and n tall by adding extra columns of zeros symmetrically left and right of the input array. Let $\tilde{I^1} = E^1 I$ and $\tilde{I^2} = E^2 I$. Let T_τ denote the operator of τ translation, taking an m vector $\alpha = (\alpha_t : -n \leq t < n)$ into a vector $T_\tau \alpha$ of m elements indexed by $(-n \leq u < n)$ in which the position of elements is shifted by τ according to

$$(T_\tau \alpha)_u = \textstyle\sum_{v=-n}^{n-1} \alpha_v D_m(v - u - \tau)$$

Here τ is not necessarily integer; the formula performs trigonometric interpolation when necessary. For $-\pi/4 \leq \theta < \pi/4$, let $\tau(\theta, u; m) = \tan(\theta) \cdot u$; this is a shift which varies systematically with u following a line of slope $\tan(\theta)$ and for $\pi/4 \leq \theta < 3\pi/4$ let $\tau(\theta, u; m) = \cot(\theta) \cdot u$. For $-\pi/4 \leq \theta < \pi/4$, let S^1_θ denote the operator of shearing the array I so that the line at slope $\tan(\theta)$ is moved to become a horizontal line. This takes an array of size $n - by - n$ and produces an array of size $n - by - m$ with

$$(S_\theta^1 I)(u,.) = T_{-\tau(\theta,u)} I(u,.).$$

Here the translation is applied in the v coordinate, with a different translation at each different value of u, i.e. in each column. For $\pi/4 \le \theta < 3\pi/4$, S_θ^2 is defined analogously with roles of v and u reversed, and with the names 'column' and 'row' reversed.

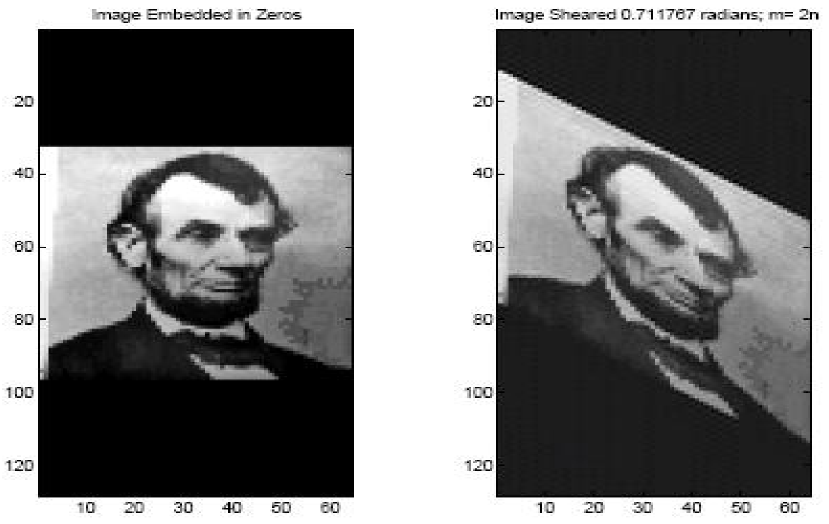


Fig. 0.27: Shearing of an image, $m = 2n$.

For each θ, *summing along lines as in the Radon Transform produces the same result as shearing the array and summing the sheared array, either horizontally or vertically as the case may be*:

$$(RI)(t,\theta) = \begin{cases} \sum_u (S_\theta^1 \tilde{I}^1(u,t)), & -\pi/4 \le \theta < \pi/4, -n \le t < n \\ \sum_u (S_\theta^2 \tilde{I}^2(t,v)), & \pi/4 \le \theta < 3\pi/4, -n \le t < n \end{cases}$$

Indeed, the shearing has simply transported the values along certain specific basically horizontal lines to be exactly horizontal, (or else the values along basically vertical lines to be exactly vertical), and so simple summation across values $I(u,t)$ evaluates the same sum that earlier was across $I(u, su + t)$. One can think of this as performing what seismologists call a 'slant stack'. For seismologists, stacking is the operation of summing an array with two subscripts $A(u,v)$ to be an array of one subscript $\sum_u A(u,v)$, which is exactly the operation performed here. However, the operation is being performed on a slanted version of the original image; hence it is indeed a slant stack.

This formal equivalence can be seen in the following Fig(0.28) for a graphical illustration.

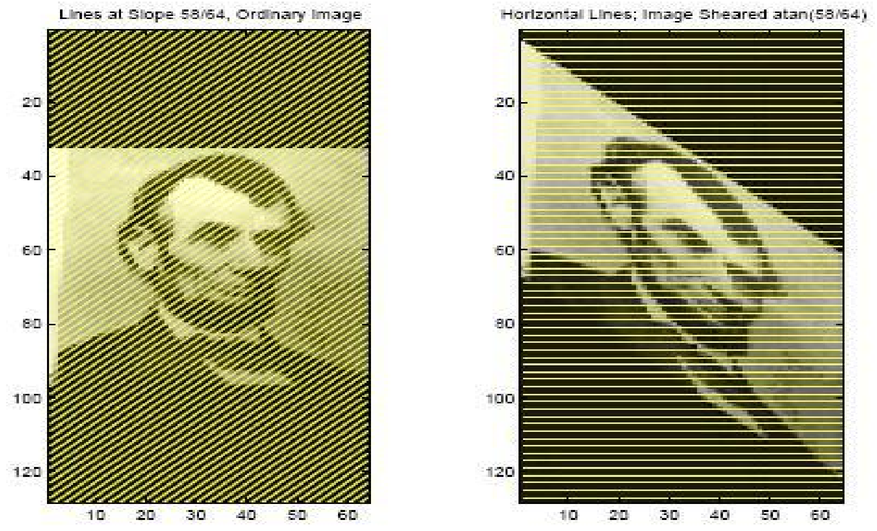


Fig. 0.28: Summing unsheared image along slanted lines is the same as summing a sheared image along horizontal lines.

2) *The* $4n^2$ *values* $RI(t,\theta) : t \in T_n, \theta \in \Theta_n$ *can be calculated in order* $O(N\log(N))$ *flops, where* $N = n^2$ *is the number of samples in the array* I.

The key point here is the special nature of the angles $\theta_k^s \in \Theta_n$ i.e grid-friendliness of the set of angles. Taking the Radon transform at these angles, we get, according to the discrete Central-Slice Theorem a one-one connection with the values of the 2-D Fourier transform at an associated set of frequencies Ξ_n. We label these frequencies $\xi_{l,k}^s$; they are given by

$$\xi_{l,k}^1 = \left(\frac{2\pi}{n}lk\frac{2}{n}, \frac{2\pi l}{n}\right), -n \le k < n, \frac{-n}{2} \le l < \frac{n}{2} \tag{0.5.3}$$

$$\xi_{l,k}^2 = \left(\frac{2\pi l}{n}, \frac{2\pi}{n}lk\frac{2}{n}\right), -n \le k < n, \frac{-n}{2} \le l < \frac{n}{2} \tag{0.5.4}$$

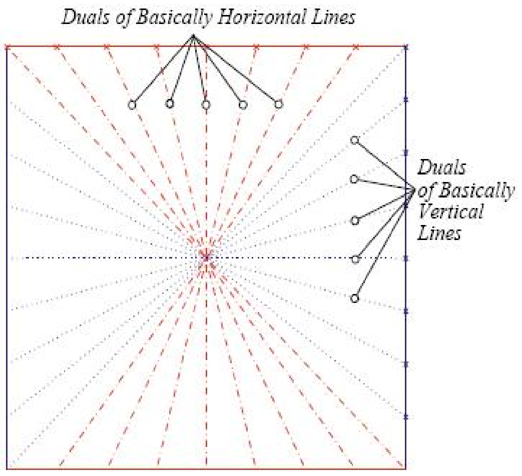

Duals of Basically Horizontal Lines
Duals of Basically Vertical Lines

Fig. 0.29: Lines in frequency space corresponding to pseudopolar angles

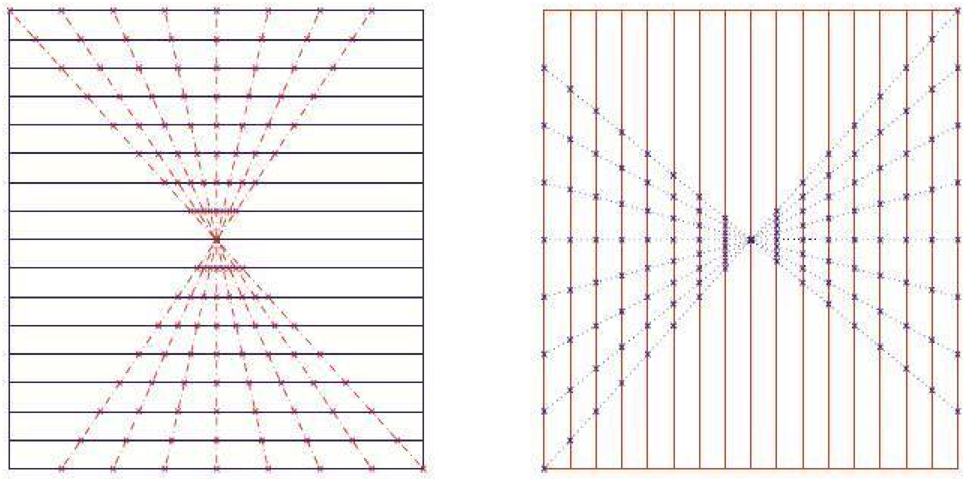


Fig. 0.30: Definition of pseudopolar grid points. Left: Panel $s = 1$, using duals of basically horizontal lines; Right: Panel $s = 2$, using duals of basically vertical lines.

This is a special non-Cartesian pointset in frequency space $[\pi, \pi)^2$ which we earlier called the Pseudopolar grid, and illustrated in Fig(0.26). A geometric construction can be seen in Fig(0.29) and Fig(0.30). First, we define a special collection of $2n$ lines in the continuum square, by markingout an equispaced sequence of n points on the upper boundary of the square and a comparable sequence of n points on the right-hand boundary. Connecting these points to the center of the square, we define the $2n$ lines. This is clearly seen in Fig(0.29). In the continuous Fourier analysis, a radial line in the Fourier domain is associated, by duality, with a line in the spatial domain oriented at 90 degrees. The same thing is true for the central-slice theorem proved earlier. Hence the basically vertical lines in the frequency domain are actually duals of the basically horizontal lines in the ordinary spatial domain, so we associate them to $s = 1$; the basically horizontal lines we associate to $s = 2$. The grid points corresponding to thesse lines are illustrated in Fig(0.30). $\xi^1_{l,k}$ is the intersection of the l-th line with the k-th horizontal line. $\xi^2_{l,k}$ is the intersection of the l-th line with the k-th vertical line. In this pseudopolar representation, $|k|$ indexes (pseudo-) radius and l indexes angle. The level sets of (pseudo-) radius are squares; so this grid may be called a Concentric Squares Grid. We can obtain the values of the Fourier transform from the following definition:

Definition: *The Pseudopolar Fourier Transform P is the linear transformation from data* $\{I(u,v) : -n/2 \le u, v < n/2\}$ to data $\{\hat{I}(\xi^s_{l,k}), s = 1, 2, -n \le k < n, -n/2 \le l < n/2\}$, where $\hat{I}$ is the trigonometric sum

$$\hat{I}(\xi_1, \xi_2) = \sum I(u,v) exp\{-i(\xi_1 u + \xi_2 v)\} \tag{0.5.5}$$

By the Central-Slice Theorem, we have the decomposition

$$R = F_1^{-1} \circ P, \tag{0.5.6}$$

where F_1^{-1} denotes the $1 - D$ inverse discrete Fourier transform.

Comlexity of Pseudopolar FT: *Pseudopolar FT can be computed in* $O(n^2 \log(n))$ *flops.*

This allows us to compute R rapidly. We know R is the composition of P with F_1^{-1}. Now P can be computed in $O(n^2 \log(n))$ flops and F_1^{-1} requires a series of $2n$ $1 - D$ FFTs, for $O(n^2 \log(n))$ flops. In terms of the $N = n^2$ values in the array, the whole procedure takes $O(N \log(N))$. The Pseudopolar FFT algorithm follows two stages. First, we calculate the usual $2 - D$ FFT, which rapidly gives values of $\hat{I}$ on the Cartesian grid $(\frac{2\pi}{m}k_1, \frac{2\pi}{m}k_2)$ for $-n \le k_1, k_2 < n$; Second, we perform Cartesian-to-Pseudopolar conversion, calculating values of $\hat{I}$ at the pseudopolar grid points. Both stages cost only $O(N \log(N))$ flops. Figures (0.31) and

(0.32) illustrate the structural features of Cartesian to Pseudopolar conversion. To obtain pseudopolar values in the Panel $s = 1$, we work one row at a time. The Cartesian values in a single row of the array are used to calculate the pseudopolar values in the same row. This is applied across all rows of the array, obtaining thereby all the pseudopolar values in panel $s = 1$. The approach for Panel $s = 2$ is similar, with the role of rows replaced by columns. At the heart of the algorithm is the interpolation of $m = 2n$ equispaced points in the k-th row to produce n points with special spacing $\alpha = 2k/n$.

Given m values of a trigonometric polynomial T of degree $m = 2n$ and period m

$$T(l) = \sum_{u=-n}^{n-1} c_u e^{i\frac{2\pi}{m}lu}, -n \leq l < n$$

it is possible, for any α, to find an equispaced set of m values $T(l\alpha)$ $l = -n, ..., n-1$ in order $O(n \log n)$ time. Conceptually, an operator $G_{n,k}$ can be defined which takes m values at the Cartesian grid points $-n \leq k < n$, obtains the unique trigonometric polynomial generating those values, and delivers n values of that polynomial at more finely spaced points $-\alpha n/2 \leq l\alpha < \alpha n/2$.

3) *Let I denote the vector space of $n-by-n$ arrays and R denote the vector space of $2n-by-2n$ arrays. The transform $R : I \mapsto R$ is $one-to-one$. There is a bounded operator $R\dagger : R \mapsto I$ so that $R\dagger R = Id$.* Although fast exact algorithm for the inverse transform is not known, a fast iterative approximation algorithm is known. This is based on two ingredients. First is a fast exact algorithm for the adjoint transform $adjR$. The second ingredient is a simple useful preconditioner, again Fourier-based.

4) *The adjoint mapping $adjR : R \mapsto I$ taking $2n-by-2n$ arrays into $n-by-n$ arrays can be computed in order $O(N\log(N))$ flops, where $N = n^2$ is the number of samples in the array I.*

From eq(0.5.6), we get

$$adjR = adjP \circ F_1$$

and so rapid computation of adj R reduces to rapid computation of adj P. Conceptually, we have the decomposition $P = S \circ F_2$ where F_2 denotes $2-D$ FFT, and S is the operator of resampling data on certain lines from the original Cartesian grid to the pseudopolar gridpoints. Now S is a block matrix, using only data on associated to a certain line to compute values at pseudopolar grid points on that line.

5) *Lets define a convolution operator* Δ *which acts one dimensionally on each constant* $\theta - slice$ *of the array* $\{RI(t,\theta) : -n \leq t < n\}$ *giving*

$$\widetilde{RI}(t,\theta) = \sum_u RI(u,\theta)\Delta_{t-u} \tag{0.5.7}$$

the operator is characterized by a $frequency - domain$ *representation*

$$\hat{\Delta}_k = \begin{cases} \sqrt{|k|/2}/n & k \neq 0 \\ \sqrt{1/8}/\sqrt{n} & \text{k} = 0 \end{cases}$$

The resulting array $\{\widetilde{RI}(t,\theta) : -n \leq t < n, \theta \in \Theta_n\}$ *is a* $near - isometry$ *with* I; *the mapping* $\tilde{R} : I \mapsto R$ *has* n^2 *nonzero singular values with bounded ratios.*

In this result, the behavior of the preconditioner weights at $k \neq 0$ can be motivated geometrically; the behavior at $k = 0$, while crucial, does not have a geometric explanation. Because of its Fourier domain representation, the convolution preconditioner can be computed rapidly for each fixed θ, using $O(n\log(n))$ flops; the preconditioner can therefore be applied for all $2n$ values of $\theta \in \Theta_n$ using $2n$ $1-D$ FFTs of length $2n$, for total work $O(N\log(N))$ flops. Because application of the preconditioned transform is so efficient, and because it has bounded condition number, it follows that traditional methods of iterative linear algebra (conjugate gradients) can efficiently yield approximate solutions of the equation

$$Y = RX$$

for X, given Y . Formally, we have

The generalized inverse $R\dagger$ *applied to an array* $Y \in R$ *can be computed approximately within an* l^2 *error* ϵ *in no more than* $C_\epsilon N\log(N)$ *flops, where* $C_\epsilon = O(\log(\epsilon^{-1})))$.

In practice, the behavior of the iterative algorithm is even more favorable than one might expect based on the above formal results. Empirically, the maximum ratio of the singular values of R is not greater than 1.1. Typically, three iterations of a CG solver are adequate for six-digit accurate reconstructions. In short, this notion of Radon transform possesses properties i.e it is algebraically exact, geometrically faithful, rapidly calculable, and invertible on its range, with rapidly calculable approximate inverse.

Ill-Conditioning: Like the continuous Radon transformation, the discrete Radon transformation is one-

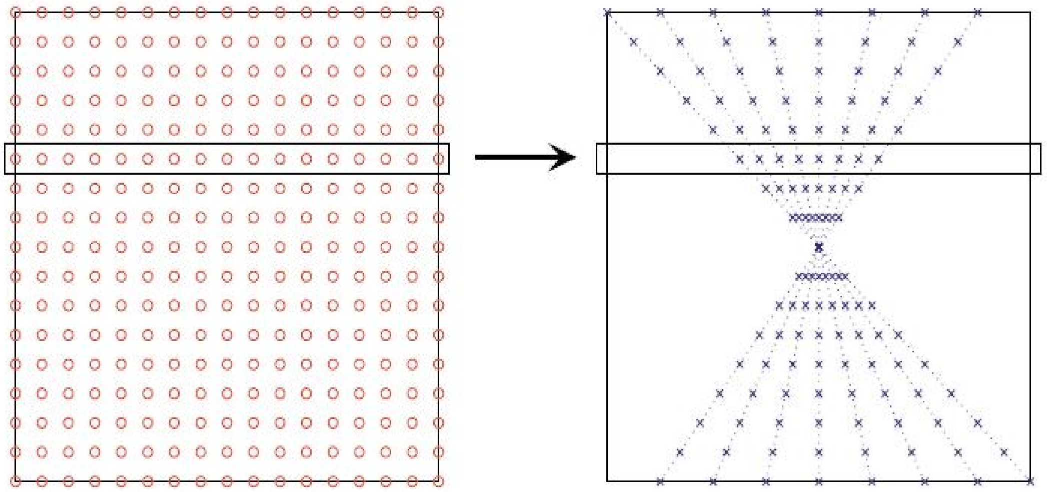


Fig. 0.31: Converting from Cartesian Grid to Panel $s = 1$ of Pseudopolar Grid.

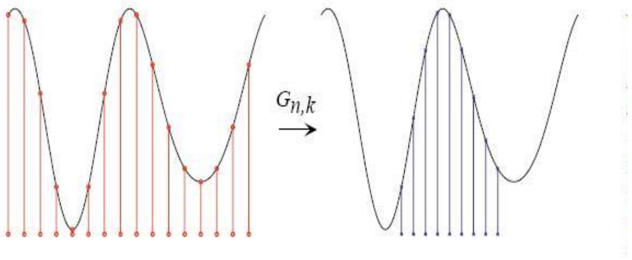


Fig. 0.32: Operator $G_{n,k}$. Cartesian to Pseudopolar Resampling within a Single Row

to-one, but the problem of recovering an image I from a noisy version of RI is ill-conditioned; there are objects I_1, I_2 of equal norm where RI_1 is small, but RI_2 has a large norm. This is easily understood using the pseudopolar FT. By the cenral-slice theorem, we have the isometry $||RI||_2 = ||PI||_2$. Hence for equal norm objects I_1, I_2 to have very different norms under mapping by R, they must also have very different norms under mapping by P. This issue is understandable in terms of the oversampling of the Fourier domain that is carried out by P. We note that the pseudopolar grid samples points at pseudo radius $k = n$ at a rate of exactly one pseudopolar sample per Cartesian sample. On the other hand, at pseudoradius zero, the grid samples points at a rate of n pseudopolar samples per Cartesian sample. This is important, because we know that there is a Parseval relation for the 2-D FFT, which implies that the l^2 norm of Cartesian samples of the Fourier transform is always identical to the normalized l^2 norm of the original object I. Combining these remarks, it is therefore clear that objects I which are concentrated in the Frequency domain at high pseudo-radius $k \approx n$ will have much smaller values of $||PI||_2$ than objects of equal norm concentrated at low pseudo-radius $k \approx 0$.

Preconditioning Operators: We now define a pre-conditioning operator for P; it normalizes pseudopolar samples by the sampling rate relative to the Cartesian samples; the appropriate weight is

$$D^s_{l,k} = \begin{cases} \sqrt{|k|/2}/n & k \neq 0 \\ \sqrt{1/8}/\sqrt{n} & \mathrm{k} = 0 \end{cases}$$

The preconditioned pseudopolar FT is then defined by

$$\tilde{P} = D \circ P.$$

Here D is a purely diagonal operator in the pseudopolar domain, defined by $(D \circ X)^s_{l,k} = D^s_{l,k} \cdot X^s_{l,k}$. We note that the normalization at $k \neq 0$ is rather natural based on the idea that the samples $(T(l))_l$ of a trigonometric polynomial of degree m and period m at unit sampling rate gives the same mean square as the normalized samples $\sqrt{(}\alpha)(T(\alpha l))_l$, while that at $k = 0$ is motivated by the $4n$-fold sampling of the point zero ($2n$ times in panel $s = 1$ and $2n$ times in panel $s = 2$), although an additional factor $1/\sqrt{(}2)$ has been inserted which cannot be explained in this way.

It has been shown that for some constants C_0 and C_1, we have

$$C_0||I|| \leq ||\tilde{P}I|| \leq C_1||I||.$$

which will imply that we have an effective preconditioner for P and consequently also for R.

0.5.3 Reconstruction Algorithm

Suppose we are given Radon data $Y = RI$. To reconstruct I, we pursue the following steps:

1. Define $F = F_1 Y$, i.e. take the standard 1-dimensional Fourier transform of length m along each constant-θ slice.

2. Define $\tilde{F} = DF$, i.e. apply the preconditioning operator to the data F.

3. By the discrete central slice theorem, we have $F = PI$ where $P = S \circ F_2$ where F_2 is 2D FFT.

4. for the above set of equations, we can use the method of normal equations in least squares and reexpress them as $(adj\tilde{P})PI = (adj\tilde{P})F$

5. Solve iteratively, by conjugate gradients, the system of equations $GI = \tilde{I}$.

where G is the Gram operator $G = (\text{adj } \tilde{P})\tilde{P}$ and $\tilde{I} = (adj\tilde{P})F$. This requires a series of applications of G to various vectors, and a few vector operations per iteration.

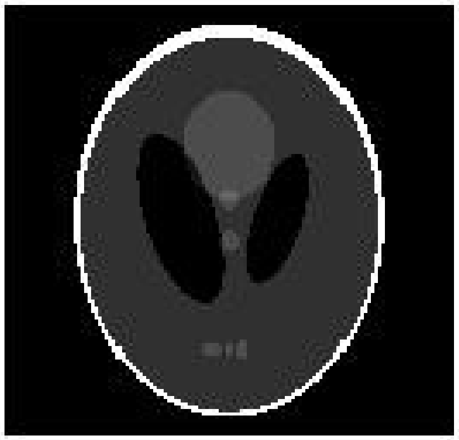
This algorithm solves the problem given of the introduction, of iteratively inverting the Radon transform, and doing so rapdily. The first three steps are all accomplished exactly in exact arithmetic, and rapidly in order $O(N \log N)$ flops or less. The only step which is not exact and not in closed form is the final step solution of a Hermitian system by conjugate gradients. Solving Hermitian systems by conjugate gradients is, of course, a central part of modern scientific computing. Defining the condition number κ of the Gram operator $G = (adj\tilde{P}) \circ \tilde{P}$ to be the ratio of largest to smallest eigenvalues of G, then the error of the k-th iterations approximate solution I_k is bounded by

$$||I_k - I||_G \leq 2 \cdot ||I_0 - I||_G \cdot [\frac{\sqrt(\kappa) - 1}{\sqrt(\kappa) + 1}]^k \tag{0.5.8}$$

where $||v||_G^2 = v^H G v$. In practice this is an extremely pessimistic bound. Indeed, we know that κ is finite from the previous section, and therefore we can get an error tending geometrically to zero as a function of the number of iterations k. The cost per iteration is essentially the cost of P and *adj* P, each one being $O(N \log(N))$.

0.5.4 Analysis of Images Reconstructed using Fast Slant Stack

The FSS method gave a very good quality of the image. This can be seen in Fig(0.33) The value of QI is 1 which is the best it can attain. The PSNR $\approx$ 181 which is way higher than the values obatined in other techniques. And the MSE is of the order of 10^{-19} which is very very less. And the time taken to reconstruct this image is 5 seconds. So, on the whole, the FSS method is the best compared to all other in accuracy as



```
QI     = 1
PSNR = 181.160968
MSE   = 7.6542591e-019
```

Fig. 0.33: Images Reconstructed using Fast Slant Stack

well as cost.

0.6 Conclusion

In the first three methods, there is a trade off between the quality of the image, cost and simplicity. Filtered back projction gave a good result for the quality of the iamge. But it needs some uniform number of projections to give that result. ART and SIRT on the other hand produced somewhat lesser quality images when compared to filtered back projction but in general, ART and SIRT can be performed even when the data is not uniform and also when the data is limited. EM did not do well compared to any of the above in terms of cost as well as accuracy. The drawback of the ART is that, it has a very large condition number for the weight matrix which is the cause for the loss of quality. The FSS method overcame this drawback by using preconditioners. So, Condition number plays a very important role in inverting the radon transform.

BIBLIOGRAPHY

[1] Charles L.Epstein, *Introduction To The Mathematics of Medical Imaging*

[2] Avinash C.Kak, Malcolm Slaney, *Principles of Computational Tomographic Imaging*

[3] L.A.Shepp ,Y.Vardi, *Maximum Likeliood Reconstruction for Emission Tomography*

[4] A. Averbuch, R.R. Coifman, D.L. Donoho, M. Israeli, J. Walden *Fast Slant Stack: A notion of Radon Transform for Data in a Cartesian Grid which is Rapidly Computible, Algebraically Exact, Geometrically Faithful and Invertible*

[5] Zhou Wang, Alan C. Bovik, *A Universal Image Quality Index*