2005

Iterative Methods for the Reconstruction of Tomographic Images with Unconventional Source-detector Configurations

Abey Mukkananchery
Virginia Commonwealth University

To Mummy, Daddy, Sofy, Joby & Jessica

ITERATIVE METHODS FOR THE RECONSTRUCTION OF TOMOGRAPHIC IMAGES WITH UNCONVENTIONAL SOURCE-DETECTOR CONFIGURATIONS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University.

by

ABEY MUKKANANCHERY
Bachelor of Engineering, Mumbai University, India, 2001
Master of Science, Virginia Commonwealth University, 2005

Director: Dr. Paul A. Wetzel
Associate Professor, Department of Biomedical Engineering

Co-Director: Dr. Alen Docef
Assistant Professor, Department of Electrical Engineering

Virginia Commonwealth University
Richmond, Virginia
June, 2005

Acknowledgements

Before all I thank my parents, Annamma Varghese and M.P. Varghese to whom, humanly speaking, I owe everything. I am greatly indebted to my sister, Sofy, and brother in-law, Joby, for their support and encouragement. I would also like to thank my relatives for their care and well wishes.

I would like to thank the Departments of Biomedical Engineering and Electrical Engineering of Virginia Commonwealth University for offering me an opportunity to pursue my graduate study and for supporting me in the completion of this thesis work for the Master of Science degree.

Special thanks to my advisor, Dr. Alen Docef, for his support, guidance and willingness to answer my questions throughout the project. Thanks to my academic advisors, Dr. Paul Wetzel and Dr. Peter Lum, for their good advice on research. Thank you, Dr. James Ames for being a part of my thesis committee. I would also like to thank all the faculties and staff associated with the two departments for their support.

I would especially like to thank all my friends, far and near, who encouraged me to pursue my Masters, been there for me during my good and bad times, and have cared and helped me in every aspect of my research. Efath, it has been amazingly wonderful knowing you and having you as my friend. Thanks for being there whenever I needed you.

Table of Contents

List of Tables

List of Figures

Page

List of Abbreviations

1-D	One Dimensional
2-D	Two Dimensional
ART	Algebraic Reconstruction Technique
CGM	Conjugate Gradient Method
CRT	Cathode Ray Tube
CT	Computed Tomography
DICOM	Digital Imaging and Communications in Medicine
FBP	Filtered Back Projection
ICU	Intensive Care Unit
ILST	Iterative Least-Square Technique
LCD	Liquid Crystal Display
LTI	Linear Time Invariant
MRI	Magnetic Resonance Imaging
MSE	Mean Square Error
PD	Probability Distribution
SIRT	Simultaneous iterative reconstruction technique
SNR	Signal to Noise Ratio

Abstract

Iterative methods for the reconstruction of tomographic images with unconventional source-detector configurations

By Abey Mukkananchery, M.S.

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University.

Virginia Commonwealth University, August 2005

Major Director: Dr. Paul A. Wetzel
Associate Professor, Biomedical Engineering

Co-director: Dr. Alen Docef, Ph.D.
Assistant Professor, Department of Electrical and Computer Engineering

X-ray computed tomography (CT) holds a critical role in current medical practice for the evaluation of patients, particularly in the emergency department and intensive care units. Expensive high resolution stationary scanners are available in radiology departments of most hospitals. In many situations however, a small, inexpensive, portable CT unit would be of significant value. Several mobile or miniature CT scanners are available, but none of these systems have the range, flexibility or overall physical characteristics of a truly portable device. The main challenge is the design of a geometry that optimally trades image quality for system size. The goal of this work has been to develop analysis tools to help simulate and evaluate novel system geometries. To test the tools we have developed, three geometries have been considered in the thesis, namely, parallel projections, clam-shell and parallel plate geometries. The parallel projections

geometry is commonly used in reconstruction of images by filtered backprojection technique. A clam-shell structure consists of two semi-cylindrical braces that fold together over the patient's body and connect at the top. A parallel plate structure uses two fixed flat or curved plates on either side of the patient's body and image from fixed sources/detectors that are gated on and off so as to step the X-ray field through the body. The parallel plate geometry has been found to be the least reliable of the three geometries investigated, with the parallel projections geometry being the most reliable. For the targeted application, the clam-shell geometry seems to be the solution with more chances to succeed in the short term. We implemented the Van Cittert iterative technique for the reconstruction of images from projections. The thesis discuses a number of variations on the algorithm, such as the use of the Conjugate Gradient Method, several choices for the initial guess, and the incorporation of a priori information to handle the reconstruction of images with metal inserts.

1 Chapter One – Introduction

1.1 Tomography

Tomography, derived from the Greek word *tomos*, is an X-ray photography technique by which a single plane is photographed while outline of structure in other planes are eliminated. In conventional radiography the three-dimensional volume of the body is compressed along the direction of the X-ray, to a two-dimensional image, which results in significant reduction in visibility of the object of interest. The spatial resolution is excellent but the image suffers from poor low-contrast resolution [1].

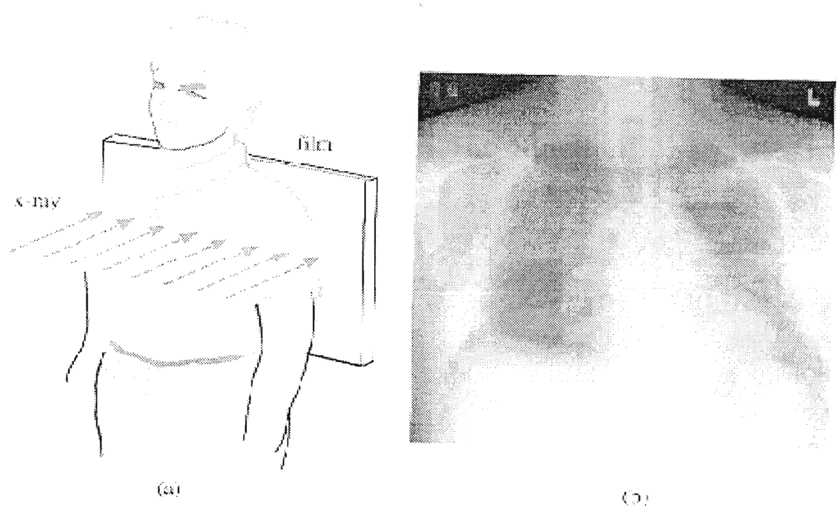


Figure 1.1 Illustration of conventional X-ray. (a) Acquisition setup, and (b) an example of a chest X-ray image. (From [1])

X-ray computed tomography holds a critical role in current medical practice for the evaluation of patients, particularly in the emergency department and intensive care units. This is because the technique provides rapid and reliable imaging of three-dimensional anatomy and is less dependent on patient cooperation for adequate image quality. It is particularly sensitive for detecting acute intracranial blood, bony abnormalities of the head, spine and torso, and a variety of other pathologic processes associated with emergent or urgent medical conditions of the head, chest and abdomen. For these reasons CT is a mainstay for the evaluation of patients with acute or recent head trauma, providing critical information for the management of this prevalent health risk. It is the first diagnostic modality of choice for patients suspected of cerebral infarction (stroke), patients with new neurologic deficits. In addition, CT is commonly used in the emergency setting to assess for internal thoracic and abdominal injuries, appendicitis and Cholecystitis. Because of its value in demonstrating pathologic processes not evident on plain chest X-ray imaging, CT of the chest has seen increasing utilization in the intensive care setting for this purpose as well [2].

1.2 History of Computer Tomography

The development of the first clinical CT began in 1967 with Godfrey N. Hounsfield at the Central Research Laboratories of EMI, Ltd. England [3]. He deduced that X-ray measurements taken through a body from different directions would allow for the reconstruction of its internal structure. The first clinically available CT device was installed at the Atkinson-Morley hospital in 1971 after further refinement of the data acquisition and reconstruction algorithms.

Since the introduction of the first clinical scanner, tremendous advancements have been made in CT technology. Improvements in spatial and low-contrast resolution are evident. Scan time per slice has been reduced by a factor of 1.34 per year over the last 30 years. A wide range of gantry speeds, reconstruction times, and tube power settings are now available.

1.3 Generations of CT Scanners

In a first generation CT scanner (Figure 1.2), only one pencil beam is measured at a time. The X-ray tube and the detector translate linearly to cover the entire object. The entire apparatus then rotates by one degree to repeat the scan [1].

For a second generation CT scanner (Figure 1.3), at each instant, measurements from 6 different angles are collected. Although the X-ray source and the detector still need to be linearly translated, the X-ray tube and detector can rotate every 6 degrees [1].

In the case of a third generation CT scanner (Figure 1.4), at any instant, the entire object is irradiated by the X-ray source. The X-ray tube and detector are stationary with respect to each other while the entire apparatus rotates about the patient. Several technology challenges associated with this configuration, such as detector stability and aliasing, led to the investigation of the fourth generation concept [1].

For a fourth generation CT scanner, at any instant, the X-ray source irradiates the detector in a fan-shaped X-ray beam, as shown by the solid lines in Figure 1.5 [1]. A projection is formed, with the collection of measurement samples of a single detector over time.

The electron beam scanner, also known as a fifth generation scanner, shown in Figure 1.6, was built for cardiac applications. The motion of the electron beam provides the rotation of the source [1]. To freeze cardiac motion, a complete set of projections is collected within 20 to 50 milliseconds. A high-speed electron beam is focused and deflected by carefully designed coils to sweep along the target ring, similar to cathode ray tube. Fan shaped X-ray beams are produced and collimated to a set of detectors, represented by the top arc of 216 degrees. Since there is no mechanical moving part in the system, the scan time is around 50 milliseconds [1].

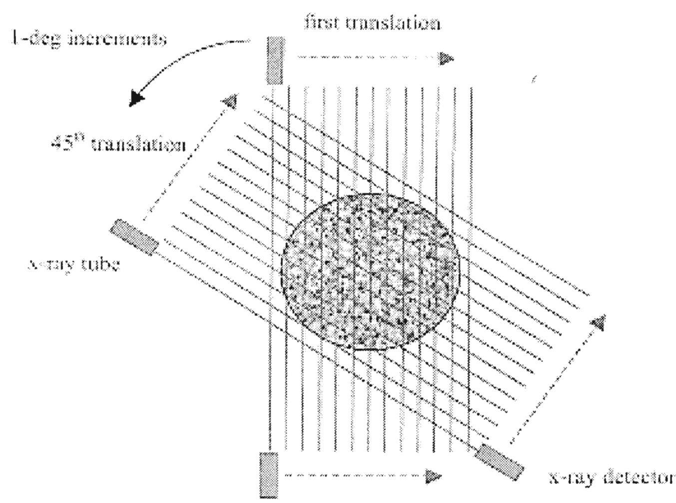


Figure 1.2 First generation CT scanners. (From [1])

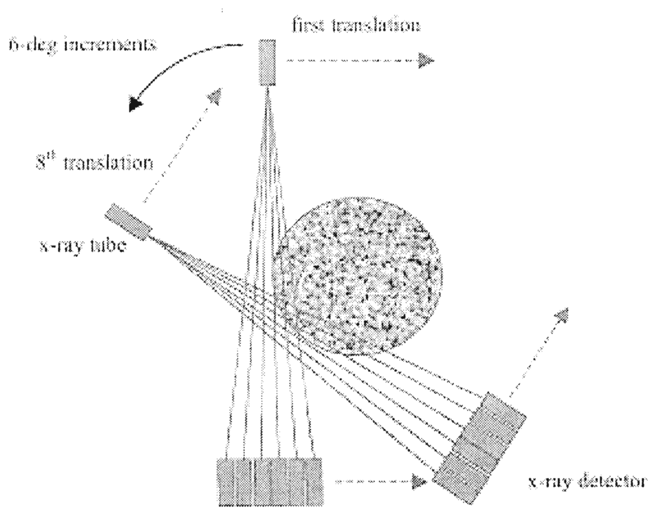


Figure 1.3 Second generation CT scanner. (From [1]).

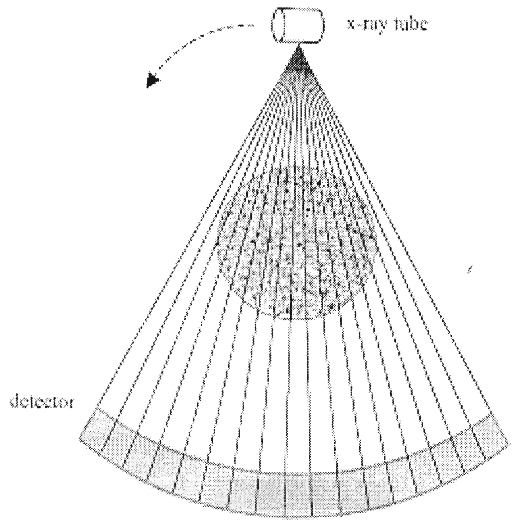


Figure 1.4 Third generation CT scanner. (From [1])

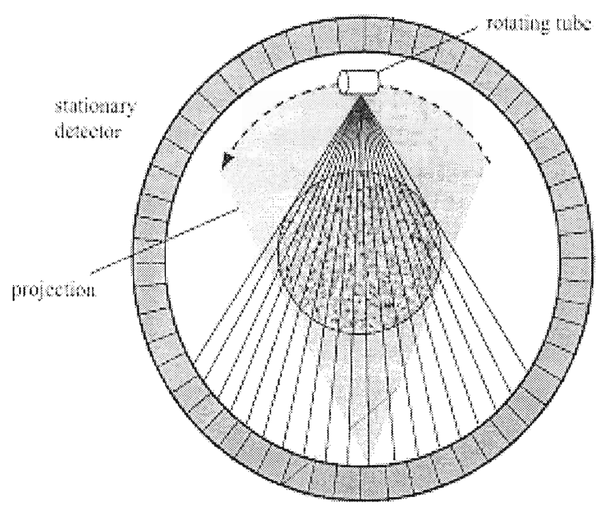


Figure 1.5 Fourth generation CT scanner. (From [1])

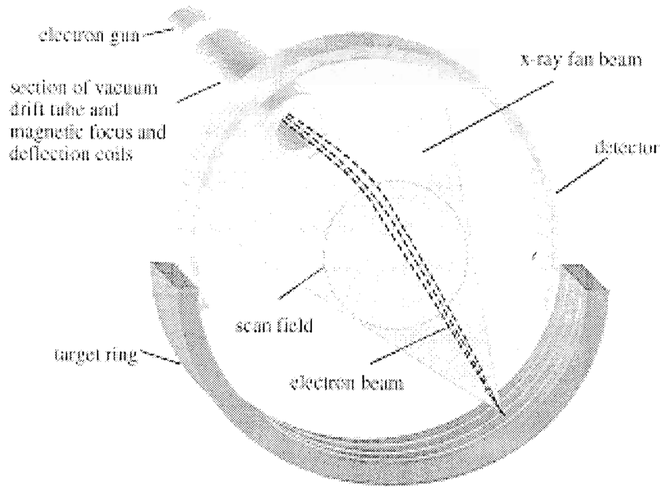


Figure 1.6 Geometry of electron beam scanner. (From [1])

1.4 The process of computed tomography

The formation of images in a CT scanner involves three processes [3], as shown in Figure 1.7. The first process, data collection, refers to the collection of X-ray transmission measurements from the patient. Once X-rays pass through the body they fall into special detectors that measure the transmission values. Once enough transmission measurements have been collected they are sent to the computer for processing. The computer uses special mathematical techniques known as reconstruction algorithms to reconstruct the CT image. After the image has been reconstructed, it can be displayed and stored as well for future the images are usually displayed on a cathode ray tube (CRT) although other technologies have become available. Image manipulation has become rather popular now and various software packages are now available to manipulate the images to make them more useful for observation.

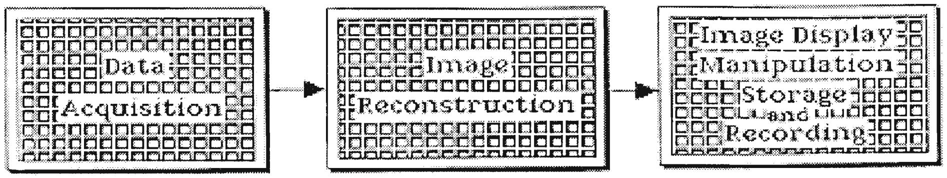


Figure 1.7 Steps in the production of a CT image (From [3])

A critical component of a CT system, image reconstruction from projections produces sharp, clear images of cross-sectional anatomy. Radon, an Austrian mathematician, proved that it is possible to reconstruct an image of a two-dimensional or three-dimensional object from a large number of its projections from different directions. Similarly, images of the human body can be reconstructed by using large numbers of projections from different locations. Consider an object divided into slices. Radiation passes through each of the slices and is projected into the detector that sends signals to the computer, which generates images of the slices. A more definite definition of this technique is given by Herman (1980), who stated, "Image Reconstruction from projections is the process of producing an image of a two-dimensional distribution from estimates of its line integrals along a finite number of lines of known locations" [3].

1.5 A Miniaturized CT Scanner

Computer Tomography is one of the most widely used modalities in the imaging of both hard and soft tissue organs. High-resolution stationary scanners are available in radiology departments of most hospitals. Typically, a patient walks or is wheeled to the radiology department and into the CT room, and a set of tomographic images are collected. In some situations however, a small mobile CT unit [3] would be of significant value. Here are several possible scenarios:

- In the emergency room and the intensive care unit, a small mobile CT unit could help the attending physician make a quick diagnosis, reducing the time before adequate care is administered.
- A similar situation may occur in a trauma vehicle, helicopter, or battlefield hospital, where a large conventional CT scanner may not be available.
- In the operating room, a small mobile CT unit with tableside control and display could be used to guide surgery in reduced visibility procedures, such as needle positioning in brain tumor treatment.
- In home care or remote care situations, a small mobile CT unit could be used when bringing the patient to a hospital is not economically feasible. A nurse or radiology technician could collect the images to be transmitted to a radiologist or other specialist via telemedicine networks.

A miniature mobile CT imaging system can be designed to meet the requirements for emergency room and operating room use [3]. This system is capable of simple transportation, rapid application for imaging acquisition with a configuration that allows real-time interventional/surgical techniques. Several mobile or miniature CT scanners have been proposed. They address some of the requirements for such a unit, but fall short to provide the range, flexibility or overall physical characteristics to meet the ER-based requirements for a truly portable device.

2 Chapter Two – Background

2.1 The Role of Computed Tomography

X-ray computed tomography holds a critical role in current medical practice for the evaluation of patients, particularly in the emergency department and intensive care units (Chen et al.1996; Huda et al.1997; Lee and Chew, 1998; Rizzo et al.1995; Rothrock et al.1997). This is because the technique provides rapid and reliable imaging of three-dimensional anatomy and is less dependent on patient cooperation for adequate image quality. It is particularly sensitive for detecting acute intracranial blood, bony abnormalities of the head, spine and torso, and a variety of other pathologic processes associated with emergent or urgent medical conditions of the head, chest and abdomen. For these reasons CT is a mainstay for the evaluation of patients with acute or recent head trauma, providing critical information for the management of this prevalent health risk (Bagley, 1999; Bullock et al.1996). It is the first diagnostic modality of choice for patients suspected of cerebral infarction (stroke), and patients with new neurologic deficits. In addition, CT is commonly used in the emergency setting to assess internal thoracic and abdominal injuries, appendicitis and cholecystitis (Pena et al. 1999; Feliciano and Rozycki, 1999; Novelline et al. 1999).

Because of its value in demonstrating pathologic processes not evident on plain chest X-ray imaging, CT of the chest has seen increasing utilization in the intensive care setting for this purpose as well (McCunn et al. 2000; Gross and Spizarny, 1994; Ivatury and Sugerman, 2000).

Another important use of CT scanning is for addressing questions of pathology when conventional X-rays are unable to provide adequate information. Particular examples of this are in evaluations for intrathoracic pathology (Gross and Spizarny, 1994; Ivatury and Sugermean, 2000; Novelline et al. 1999; White et al. 1999), and in evaluations of the cervical spine in trauma patients. In the latter case, the lower cervical spine is often difficult to evaluate due to the impediment of the shoulder structures on lateral views in heavy-set individuals. CT imaging of the lower cervical spine in such case quickly provides the information necessary to determine the presence or absence of structural injuries to this area, facilitating the management of these patients (Tan et al.1999; Tehranzadeh et al. 1994). Again however, this requires transporting the patient for CT imaging to acquire these studies [2].

2.2 Current State of the Art and Technical Limitations

One of the major drawbacks of CT scanning has been the fact that these devices are typically large structures requiring dedicated space for their installation. This has been due to the requirement for X-ray shielding and for climate-controlled housing of high-speed data and graphics processing computer equipment. The placement of CT devices in Radiology suites thus requires that patients travel from the emergency department, ICU or hospital ward to have the imaging study performed. Placing additional CT scanners in the emergency department has helped to alleviate this problem in many larger institutions, but then these scanners are less available for use in other clinical applications. This has even resulted in patients sometimes being transported from ICU or hospital ward to the emergency department for CT scans (the reverse logistical problem). The transport process places a burden on medical and nursing staff and adds risk for patients who may be clinically unstable or deteriorating. Not infrequently, CT scans for patients are deferred or delayed because they are too unstable for transport, delaying the diagnostic process, which could potentially provide critical information as to how to manage them [2].

The potential for using the CT scanner for image guidance during interventional or operative procedures is also limited by similar logistical issues. Aside from installing a CT scanner in an operating room, which carries issues of potential under-utilization of a useful and expensive imaging device, such procedures must be carried out in the radiology-imaging suite. Because these suites are rarely set up properly for such procedures and because such procedures usually consume considerable time, safe performance of the procedure, efficient use of the imaging suite and maintenance of elective imaging schedules become problematic. The availability of a portable CT unit has demonstrated the value of addressing these problems with a more compact device, but this device remains bulky and unwieldy, severely limiting its portability. Nevertheless, its development and production serve as strong support for the value of increasing the portability of CT imaging. Indeed, the use of conventional portable X-ray machines and C-arm fluoroscopy units in hospitals, operating rooms and emergency departments, serves to highlight the kinds of applications that could potentially be improved by the availability of an easily portable miniature CT device.

Reducing the size of a CT unit such that it could be made mobile and portable has been a goal of ongoing research, and the medical imaging industry has made significant strides in this area, like the "Light Speed" CT family of scanners introduced by General Electric Medical Systems in 1998.[4]

2.3 Potential Benefits and Applications of a Miniature CT Scanner

The most obvious potential applications for a portable miniature CT scanner would be in the emergency room, intensive care unit and operating room. A high-quality portable miniature CT would allow rapid acquisition of head CT images on trauma victims while their evaluation and management by trauma nursing and medical specialists could continue nearly uninterrupted. Many cases that are routinely carried out with fluoroscopic guidance could be more accurately carried out with intra-operative CT guidance. Similarly, many cranial neurosurgical procedures currently relying on intra-operative navigation with pre-operative images or on intra-operative MR imaging would be well suited to performance using an intra-operative miniature CT scanner. A variety of other scenarios can also be envisioned in which a miniature CT scanner would be advantageous or ideal. These include any settings in which space or portability of the unit would be of value. Potential scenarios of this sort include transportable imaging units (panel truck mobile scanners as opposed to tractor-trailer units), hospital ship installations, and a variety of defense- and space-related applications [2].

2.4 Tomography: a Theoretical Background

The foundations of computer tomography are in the theory of image reconstruction from projections. In a typical system geometry, a set of projection signals is acquired, each with an associated angular rotation, as shown in Figure 2.1. Using signal processing algorithms, these angular projections can be converted into 2-D cross-sectional slices.

In the simplest model, the X-ray source irradiates the object of interest $x(t_1, t_2)$, resulting in a projection signal at the detector. A set of these projection signals is collected from the detector at different angles. A projection is a line integral through the object. The projection signal at a given angle θ is denoted $p(\theta,\tau)$ and is given by

$$p(\theta,\tau) = \int_{-\infty}^{\infty} x(\tau\cos\theta - \xi\sin\theta, \tau\sin\theta + \xi\cos\theta)d\xi \tag{2-1}$$

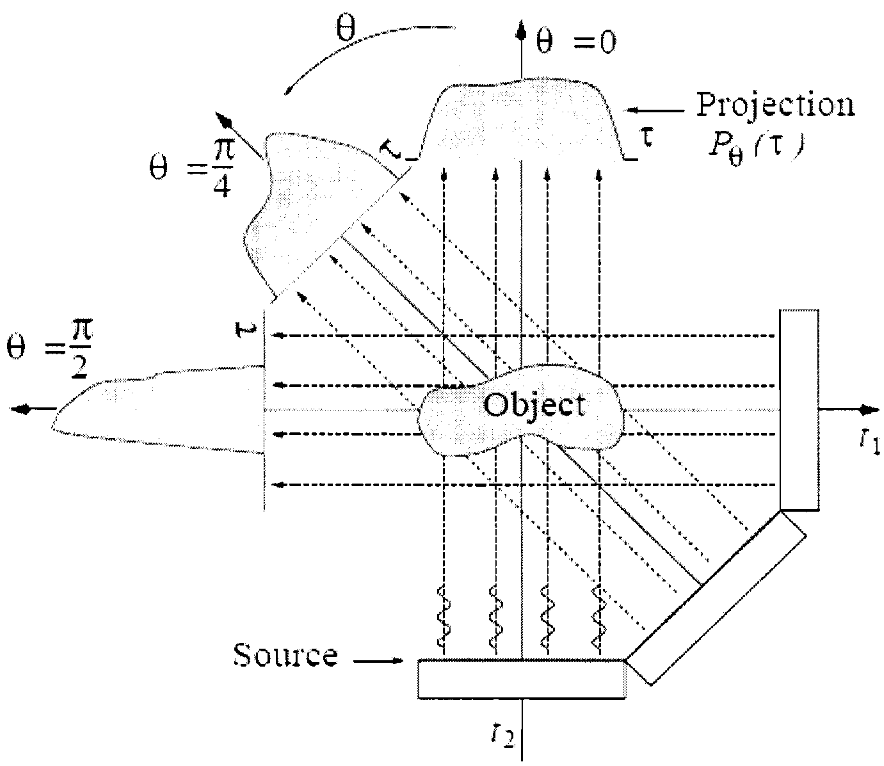


Figure 2.1 Illustration of a computed tomography system [2]

This relationship between an image and its set of projections is called the Radon transform. The Radon transform is invertible, and applying the inverse transform allows for the reconstruction of the image from its projections:

$$x(t_1, t_2) = \frac{1}{2\pi}\int_0^{\pi}\frac{p(\theta,\gamma)}{\tau-\gamma}\,d\gamma d\theta, \text{ where } \tau = t_1\cos\theta + t_2\sin\theta \tag{2-2}$$

In practice, several assumptions made in the simple theoretical model above are not true:

- The angular variable θ is discrete, that is, a finite number of projections is captured.
- Reconstruction is performed on a computer using a sampled data representation. That is, p(θ,τ) is sampled along the projection axis τ.
- The inverse Radon transform is computed using discrete approximations of the differentiation and integration operations.
- The source of radiation consists of a finite number of tubes, possibly moving along a circular support mechanism.
- Each X-ray tube may produce a cone beam rather than a parallel beam.
- The projection capture subsystem consists of a finite number of sensor devices.

These issues have been addressed in various image processing literatures and modified versions of the Radon equations have been proposed [2].

2.5 Reconstruction algorithms

Several algorithms have been developed for calculating the image from a set of projection data. These include backprojection, iterative methods and analytic methods.

Backprojection: Back projection [3] is illustrated in Figure 2.2. Consider four beams of X-rays passing through an unknown object to produce projections, P1, P2, P3 and P4. The problem involves using these projection data to reconstruct the image of the hidden image in the box. The projected data are back projected to form corresponding images BP1, BP, BP3, and BP4. Each of these images is consistent with its corresponding projection and is constant along the projection direction. The reconstruction involves averaging all the backprojected images to form an image of the object. Although intuitively straightforward, this method does not produce an exact reconstruction of the image, regardless of the number of projections, and therefore is not used in practical CT systems.

Filtered Backprojection: Filtered backprojection is similar to the backprojection technique. The difference between the two techniques is that the projections are filtered prior to backprojection in order to remove the "star-like" blurring artifacts characteristic of the simple backprojection technique. Filtering is accomplished by convolving each projection with a one-dimensional Hilbert filter, as described in the inverse Radon formula shown in the previous section. It has been shown that, as the number of projections increases, the reconstructed image converges towards the exact cross-section image [3].

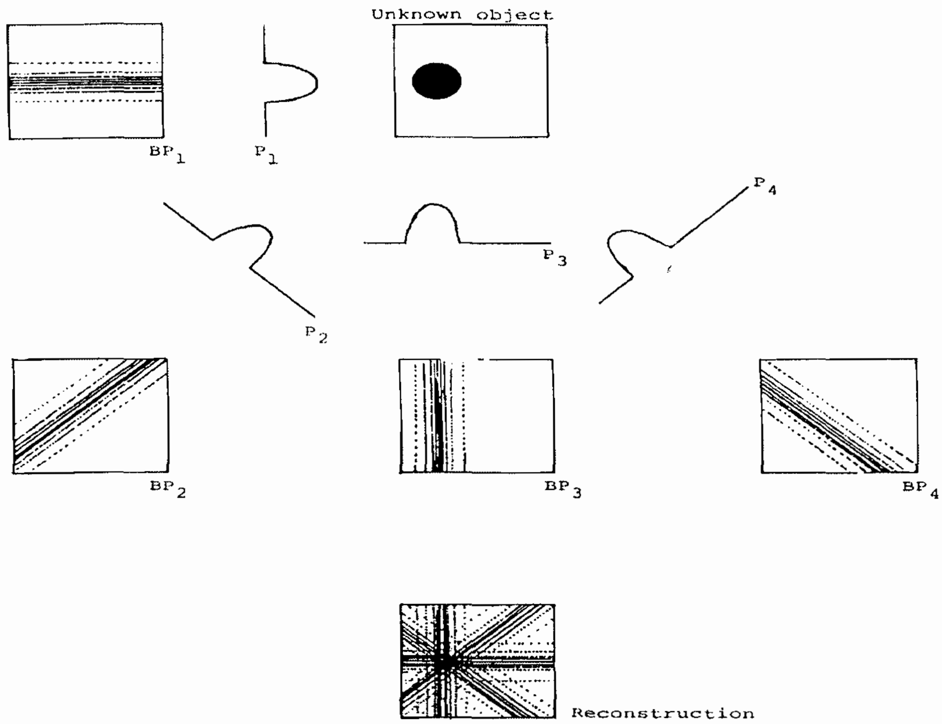


Figure 2.2 The backprojection reconstruction technique (from [3])

Fourier Reconstruction: Like the filtered backprojection method, Fourier reconstruction produces an exact reconstruction of the image. It is based on the Fourier slice-projection theorem, which states that the 1-D Fourier transform of a projection is equal to a slice through the 2-D Fourier transform of the image. Therefore, given enough projections, the 2-D Fourier transform of the image can be reconstructed with sufficient accuracy. The image is then obtained by performing an inverse Fourier transformation. The challenge posed by this method is due to the fact that the image and its transform are represented on a rectangular grid, while the slice-projection theorem assumes a polar grid. Therefore, interpolation formulas must be used to convert the data from one sampling pattern to the other, thus adding to the complexity of the reconstruction algorithm.

Iterative Algorithms: An iterative reconstruction algorithm [3] starts with an initial guess and iteratively applies corrections to the current image estimate. The correction term is typically a measure of the mismatch between the known projection data and the projections computed from the current estimate. The iterative process continues until this mismatch is zero or within acceptable limits. Some of the popular techniques in this category are the Simultaneous Iterative Reconstruction Technique (SIRT), the Iterative Least-Square Technique (ILST) and the Algebraic Reconstruction Technique (ART). These techniques differ in the way in which the corrections are applied to the subsequent iterations. Our reconstruction algorithms, described in the next chapter, are all adaptations of the ILST.

As an example of a simple iterative reconstruction, consider the illustration in Figure 2.3, where projections of a 2-by-2 matrix are computed as row sums and column sums. The goal is to reconstruct the matrix elements from these sums. Let us choose an initial guess that is a constant matrix where each element is equal to the average of the four elements in the unknown matrix. This average can be computed as (3+7+4+6)/2/4 = 2.5. In a first correction step, the matrix columns are modified so that the row sums are correct while the column sums are unchanged. This is achieved by adding to the columns a correction vector equal to the original horizontal ray sums minus the new horizontal ray sums divided by two, that is,

$$\frac{1}{2}\left(\begin{bmatrix}3\\7\end{bmatrix}-\begin{bmatrix}5\\5\end{bmatrix}\right)=\begin{bmatrix}-1\\1\end{bmatrix}.$$

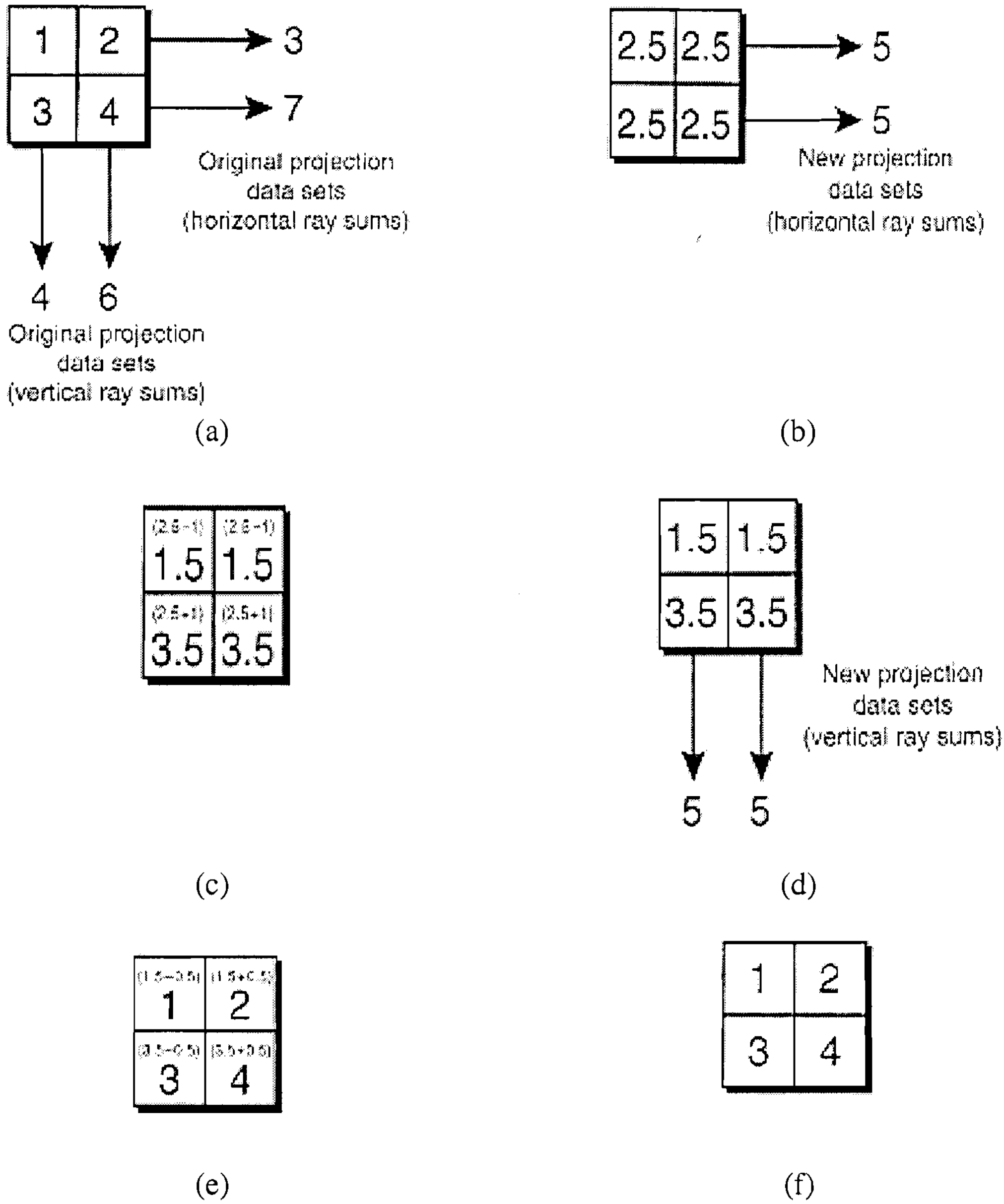


Figure 2.3 Illustration of an iterative algorithm

This results in the matrix shown in Figure 2.3(c). In a second correction step, the same procedure is applied to the matrix rows, resulting in the matrix in Figure 2.3(e). As we can see, this second estimate is actually equal to the original matrix. The algorithm converges in two steps. Practical tomography reconstruction problems are more complex than this simple example because projections are not only computed in the vertical and horizontal directions. As a consequence, these algorithms rarely end with a perfect estimate.

2.6 Comparison of Distance Metrics

To measure the accuracy of an image reconstruction process, or the mismatch between actual and estimated projection sets, several distance metrics can be used. The choice of the distance metric will be based on analytic tractability (especially the ease of computing partial derivatives), computational complexity, and suitability for the particular type of data (both images and projections take positive values only).

Euclidean Distance: The Euclidean distance function [14], also known as L_2-distance, measures the 'as-the-crow-flies' distance. The formula for this distance between a vector $X = (x_1, x_2, \dots x_n)$ and a vector $Y = (y_1, y_2, \dots y_n)$ is:

$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\ . \tag{2-3}$$

The Euclidean distance between two data points involves computing the square root of the sum of the squares of the differences between corresponding values. Sometimes, this distance is computed in its normalized form, known as the Mean Square Error (MSE), equal to the Euclidean distance divided by the number of elements n in each vector.

Manhattan Distance Metric: We can define the Manhattan distance [14], also known as the L_1-distance, as the distance between two points measured along axes at right angles. The Manhattan distance between two items is the sum of the differences of their corresponding components. The distance between a vector X = (x_1, x_2, ... x_n) and a vector Y = (y_1, y_2, ... y_n) is:

$$d(X,Y) = \sum_{i=1}^{n} |x_i - y_i| . \tag{2-4}$$

The difference between the Euclidean distance and the Manhattan distance is illustrated in Figure 2.4.

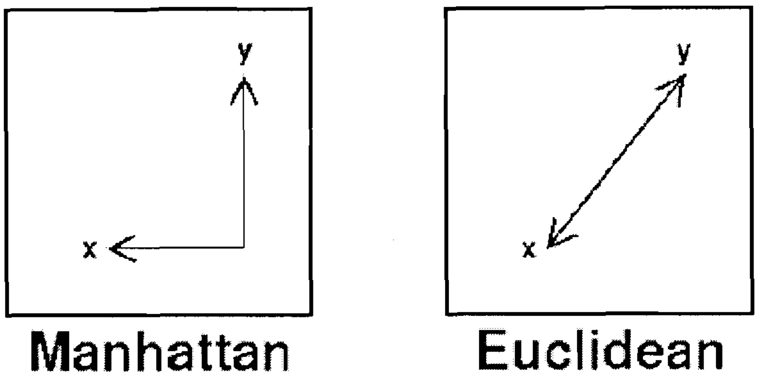


Figure 2.4 Illustration of the difference between the Manhattan distance and the Euclidean distance [13].

Infinity metric: The infinity distance, also known as the L_∞–distance or maximum distance [14], is given by

$$d(X,Y) = \max(|x_1 - y_1|, \ldots\ldots, |x_n - y_n|) . \tag{2-5}$$

The advantage of this metric is that it provides a measure of the maximum deviation between an ideal image and an approximate one. Unfortunately, this metric is very difficult to treat analytically. Derivatives of functions involving the max(,) operator are difficult to compute.

I-Divergence: The I-divergence [17] also known as information divergence or relative entropy or Kullback-Leibler distance, was first introduced to compare two probability distributions P = (p_1, p_2, ... p_n) and Q = (q_1, q_2, ... q_n) using the formula

$$d(P,Q) = \sum_i p_i \ln \frac{p_i}{q_i}. \tag{2-6}$$

This formula applies to probability distributions, which have positive values that add up to one. For a more general case, when the sum of the elements is not one but the elements are positive, the metric becomes

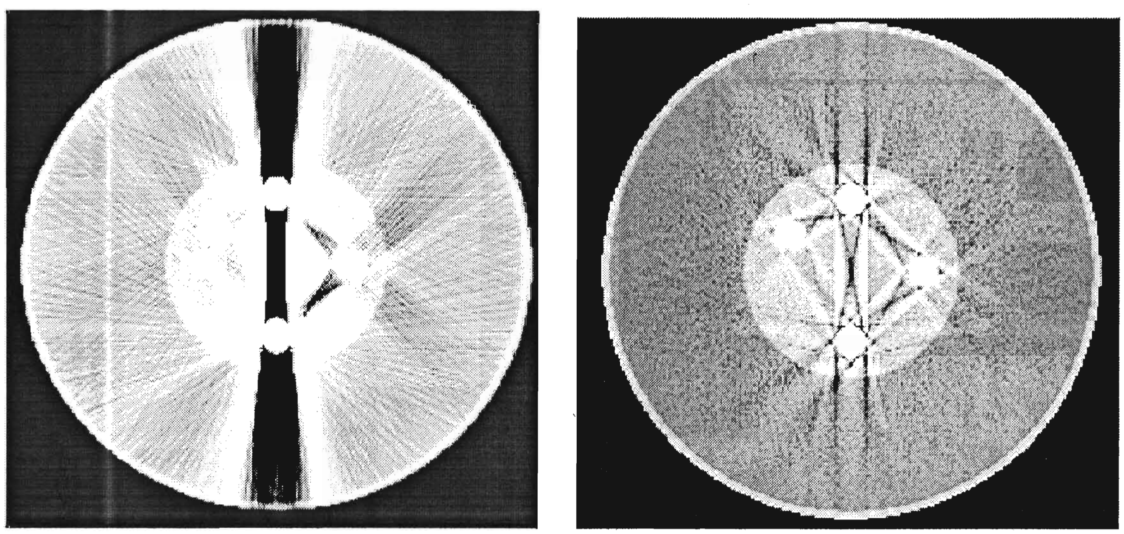
$$d(P,Q) = \sum_i p_i \ln \frac{p_i}{q_i} - p_i + q_i. \tag{2-7}$$

The I-divergence is a non-symmetric information theoretic measure of the distance between P and Q. A key property is that D(P,Q) $\geq$ 0, with equality if and only if P = Q. This quantity, while not a true metric due to its non-symmetric nature, is useful for problems where the data are naturally positive-valued, including problems in optical and hyper-spectral imaging, and in approximation of joint probabilities. It has several properties analog to the Euclidean distance, such as the Pythagorean Theorem and the cosine theorem [14].

Both the MSE and the I-divergence metrics are analytically tractable, so we will focus on them. Moreover, the I-divergence assumes positive data values, which fits the characteristics of CT images and projections. The effect of this fact on reconstruction accuracy is assessed in the Results chapter.

2.7 High Impedance Artifacts

A major challenge with X-ray computed tomography (CT) is metal artifact reduction. Because of the higher atomic number a metal object attenuates X-rays in the diagnostic energy range much more than soft tissues and bones, and much fewer photons can reach the detectors. It is well known that when metal is present, pronounced dark and bright streaks are produced in reconstruction with conventional filtered back projection as seen in Figure 2.5. These artifacts seriously degrade image quality, particularly near the metal region [18].



(a) (b)

Figure 2.5 High Impedance Artifacts in Image using (a) Filtered Backprojection and (b) the Van Cittert technique

The effects of metallic objects on X-ray CT scanning have been studied extensively, and the following three effects have been found to be most important:

- Beam hardening, due to prostheses attenuating X-rays differently in the diagnostic energy range, introduces a nonlinear effect in recording the projection data and leads to low-frequency comet-tail artifacts around prostheses, especially between prostheses and other high density objects (such as bones).
- Projection data noise is mostly caused by low photon counts and produces high frequency streaking artifacts. Decreasing X-ray attenuation by using less attenuating materials or by increasing X-ray energies to facilitate photon

penetration is effective for metal artifact reduction. However these approaches are not clinically practical, because prosthetic materials are not selected based on the X-ray attenuation characteristic, and photon energy increment is limited by the permissible radiation dose [19].

- Motion interference enhanced due to the high contrast between metals and neighboring anatomical structures.

Conventional Filtered Backprojection algorithms are computationally efficient but produce poor images when complete and precise projection data are unavailable, such is the case when metal objects are present. Iterative reconstructive algorithms have been successfully applied with incomplete/noisy projections data (see Figure 2.5).

3 Chapter Three - Methods, Materials & Systems

In this chapter, we present the requirements for the design of a miniaturized CT scanner, we introduce two possible system geometries, and briefly discuss some of the tools used in the evaluation of these geometries.

3.1 Usability Requirements

In designing a miniaturized portable CT scanner, the first questions that need to be answered are related to its usefulness in the targeted applications. The relevant system parameters are discussed briefly [3].

Portability: The ideal portable unit will be light and small enough to be carried in a suitcase. In a less desirable embodiment, the system will be supported by wheels and will have portability features roughly similar to that of a fluoroscopy C arm device or portable x-ray machine. The physical dimensions should be similar to the size of a suitcase (70 cm x 60 cm x 20 cm).

Weight: Portability requires that the component count, packaging and power supply system be constructed so as to keep the total weight within 30 kg.

Size: Portability requires that the system, in its inactive (packed) state, fit into a reasonable size case. In the active state, the system should allow to be easily deployed in emergency rooms, ambulances, operating rooms, and to be easily moved through hospital doorways.

System Configuration: One important goal in the configuration of this system will be that it will be intervention-friendly. That is, open access to the patient during the scanning process will be possible, positioning of the patient within the device for image acquisition will be feasible in the operating room, emergency room, mobile medical facilities and radiology suite environments. Given the intended use of this device virtually at the patient's bed side or in isolated environments, the configuration will be mechanically simple and reliable. Furthermore, the system will be characterized by easily cleaned surfaces which can be readily draped with standard (polyethylene film) techniques for carrying out sterile procedures.

Image Quality: The final product of any medical imaging system is an image of the organ or tissue being examined. A useful imaging system produces an image that is a very close representation of the physical reality. For digital imaging, two measures of image quality are popular. Image resolution reflects the ability of the imaging system to accurately display small details and it is measured in pixels per unit of length (inch, mm). Image bit depth reflects the ability to distinguish between areas with very close properties. For the particular case of CT imaging, this means tissues with very similar density, such as soft tissues within the brain. Bit depth is measured in bits per pixel. Objective measures are based on computing a norm of the difference between the ideal and actual images. The most popular measure is the mean square error (MSE) and the related signal-to-noise ratio (SNR). It is anticipated that initial designs will involve some sacrifice in image quality and performance with respect to resolution and scanning time in order to enable portable, robust imaging. This trade-off is analogous to that between conventional high resolution MRI and open magnet MRI. The latter is more comfortable for many patients, but requires a sacrifice in image resolution.

Human Interface: The image display is an important component. An external display (not part of the portable system) is preferable. For other situations, a small built-in LCD display will be available. Possible input devices are: ballpoint or equivalent point-and-click device, small keyboard, or other custom console.

Computer Interface: The system is to be physically compatible with the medical environment and other devices which are commonly used, as well as to provide image data output which will meet the demands of evolving medical communications and information processing standards (DICOM, for example). It should provide wireless connection to acquisition/image processing computer, display and keyboard and a telemedicine radio link to hospital (so a surgeon can have paramedic adjust positions, etc).

3.2 Implementation Requirements

Several implementation issues [3], both technical and economical, will affect the design of the miniaturized system.

Geometry: The geometry of the radiation source and capture subsystems will determine the image quality. Within the constraints imposed by usability and safety requirements, the following parameters need to be optimized:

- The arc size necessary to capture full head (or body, in general)
- The number and geometry of sources of radiation
- The number and geometry of radiation detectors
- Mobility of sources/detectors: should they be static or mobile?

Source/Detector Parameters:

- Radiation source type, power consumption, size
- Radiation detector type, power consumption, size
- Radiation beam angle, angular energy distribution

Power Requirements: The device will be designed to utilize standard available electrical power (110 volts AC in the US), drawing no more than 10 A. The device should feature a low power mode (for example, for field applications) where a 12 V backup/emergency/alternate source is used. Different requirements may apply to military applications.

Computational Requirements: The reconstruction from projections algorithm is computationally intensive. Usability requirements set limits to both scanning speed and computational subsystem size. The hardware and software components of this subsystem need to be designed accordingly.

3.3 System Geometries

In conventional CT systems, the gantry houses the imaging technology that acquires the CT data. The image capture system rapidly rotates around the patient and table. This implies a complex and heavy mechanical design. In our miniaturized design, alternate solutions are proposed that alleviate these limitations. Two proposed system geometries are shown in Figure 3.1 and Figure 3.2. Their features are discussed below, and their advantages and disadvantages are summarized in Table 1. Software tools have been developed to simulate the projection process for both structures, as well as the image reconstruction process.

3.3.1 The Clam-Shell Structure

This system [3], shown in Figure 3.1, consists of two semi-cylindrical braces that fold together over the patient's body and connect at the top, thus allowing almost 360 degrees traversal/rotation of the imaging system. This seems to be the solution with more chances to succeed in the short term. The diameter of the braces should be chosen to accommodate the largest patient size, and the width and thickness are dictated by the size of the sources and detectors and their mechanical enclosing. The structure can be built sufficiently rigid so that the sources and detectors are in known fixed relative positions. The reconstruction algorithm for this structure is very similar to conventional, cone-beam CT scanners.

There are two challenges associated with this structure. One is packing the source and detectors sufficiently close to fit on the braces. This may limit the resolution of the projections, and therefore the reconstructed image quality. The other concern has to do with the top and bottom connections, which will introduce gaps in the source and detector arrays. This means that there will be missing projections in the data set. This problem is similar to the problem of removing streaking artifacts from CT scans with high-density insert.

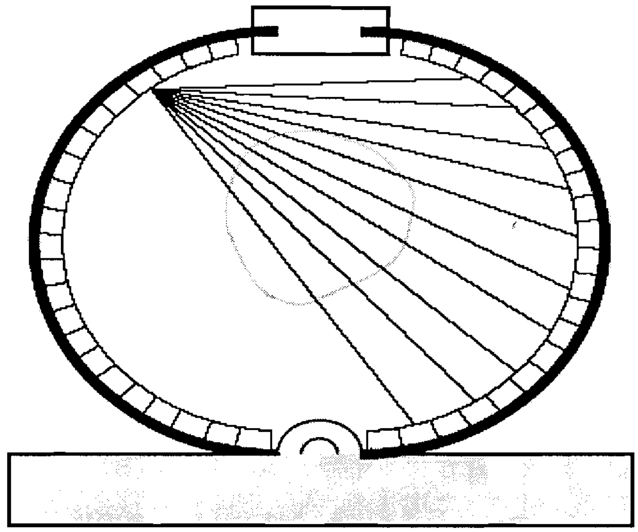


Figure 3.1 The clam-shell structure

3.3.2 The Parallel Plates Structure

This system [3], shown in Figure 3.2, uses two fixed flat or curved plates on either side of the patient's head and image from fixed sources/detectors that are gated on and off so as to step the X-ray field through the head. The advantage of this solution is easy access to the investigated body part, making it useful in CT-assisted surgery.

The disadvantage is that it may require a higher radiation dose than the clam-shell structure. The plates need to be sufficiently far from each other to allow access to the patient. The height of the plates determines the width of the field of view. The width (depth in the image) can be designed to be sufficiently large to allow for multiple slices to be reconstructed. This is beyond the scope of our work. Conventional reconstruction algorithms cannot be applied directly to this geometry.

The main challenges associated with this geometry are handling the very irregular field of view and mechanically placing sources and detectors at fixed positions.

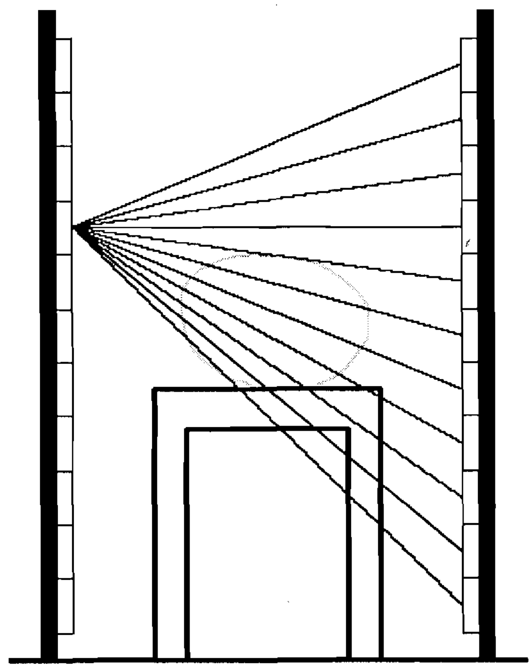


Figure 3.2 The parallel-plates structure

Structure	Advantages	Disadvantages
Clam-Shell	Small size Allows almost 360 degrees field of view Simpler reconstructions algorithm, similar to cone-beam Rigid geometry	Must be opened and closed More intrusive in an operating room 3-D imaging difficult
Parallel Plates	Allows 3-D imaging Less intrusive Allows access to the imaged organ	Larger size Narrower field of view More complex reconstruction algorithms Flexible structure may introduce errors More radiation dose.

Table 1 Comparison of the two structures [3].

3.4 Phantoms

In the early days of imaging, imaging instruments were calibrated by making measurements of composite objects with known attenuation coefficients. These objects are called phantoms [20]. The problem with this approach was that it mixed artifacts caused by physical measurement errors with those caused by algorithmic errors. In order to replace the algorithmic errors, Larry Shepp [21] developed a mathematical phantom. He suggested one give a mathematical description of a phantom to create simulated and controlled data. This way the algorithmic errors could be separated from mechanical errors. A mathematical phantom is created in the following way:

- A simplified model of a slice of the human head is described as an arrangement of ellipses and polygons.
- Each region is assigned a density or attenuation coefficient.
- The continuous model is digitized by superimposing a regular grid and replacing the piecewise continuous densities by their averaged values over the squares that make up the grid.
- Measurements are simulated by integrating the digitized model over a collection of strips, arranged to model a particular measurement apparatus.

The robustness of an algorithm to different measurement errors such as beam hardening, noise, patient motion and miscallibration can be tested by incorporating these errors into the simulated measurements. A priori, it is known exactly what is being measured and hence one can easily compare the reconstructed image to the known model.

Mathematical phantoms are especially useful in the study of artifacts caused by sampling errors and noise as well as for comparison of different algorithms.

The Shepp-Logan phantom, shown in Figure 3.3, is one of the most popular CT phantoms, perhaps because its software description is distributed with Matlab. It contains elliptical shapes of size and attenuation factors typical for head CT scans.

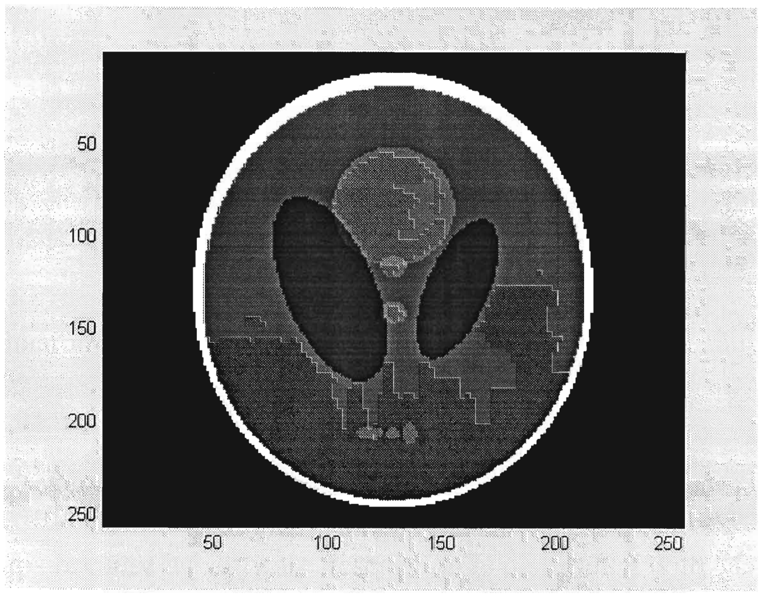
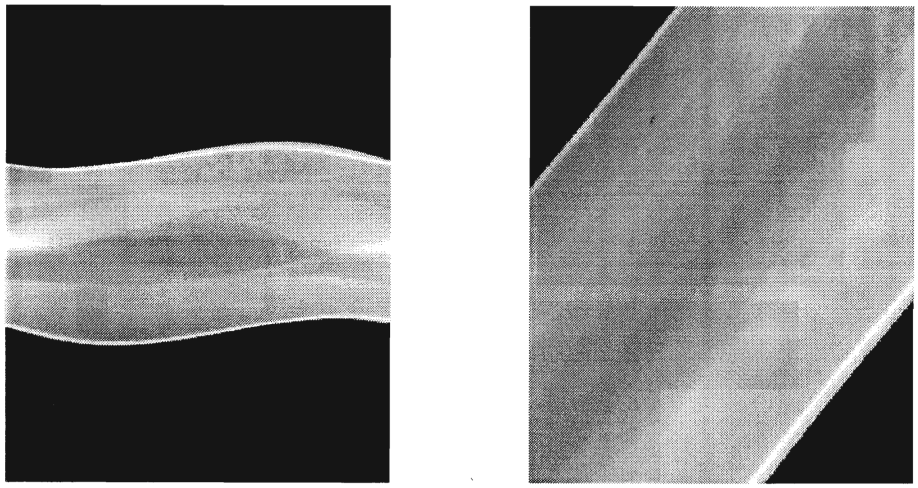


Figure 3.3 Shepp-Logan phantom generated using Matlab (256x256 pixels)

3.5 Sinograms

For CT imaging, the X-ray source irradiates the object of interest resulting in a projection signal at the detector. A set of these projection signals is collected from the detector at different angles. A projection is a line integral through the object. The set of projections are sometimes called a sinogram, because of the sinusoidal structures visible in the two-dimensional array [3]. A sinogram is a gray scale plot where the density of the image is a monotone function of the measured values. Sinograms are generally difficult to interpret directly. As they contain all the information available in the data set, they may be preferable for machine-based assessments [20]. Figure 3.4 shows typical sinograms generated by using the clam-shell and parallel plate geometry.



(a) (b)

Figure 3.4 Computed sinograms for (a) the clam-shell geometry and (b) the parallel plate geometry.

3.6 Compound Materials and Their Densities

A narrow beam of mono-energetic photons with an incident intensity I_o, penetrating a layer of material with mass thickness x and linear attenuation coefficient μ, emerges with intensity I given by the Lambert-Beers equation

$$I = I_o e^{-\mu x}. \qquad (3\text{-}1)$$

The attenuation coefficient μ is a basic quantity used to calculate the penetration and the energy deposition by photons in biological, shielding and other materials. It is tabulated for various materials usually as the mass attenuation coefficient μ/ρ, where ρ is the material's density. Reference [22] provides a list of densities and coefficient values for a variety of relevant materials.

4 Chapter Four – Discussion

In this chapter, we describe in detail the algorithms used to evaluate image reconstruction for the two proposed system geometries and their parameters

4.1 Matrix Formulation of the Sinogram Calculation

A more useful and more compact formulation of the Radon transformation is obtained if the image and the sinogram are represented as vectors. If x is the vector of pixel values in the image (in raster order) and y is the vector of projection values in the sinogram, the relation between these vectors is given by

$$y = Mx, \tag{4-1}$$

where M is the projection operator matrix. The element m_{ij} of the matrix M represents the weight (or contribution) of pixel j to the i^{th} sinogram value. Therefore, the matrix M has a number of rows equal to the number of pixels in the image and a number of columns equal to the number of source-detector pairs. To compute the element m_{ij} of the matrix M, we determine if the corresponding line integral intersects the rectangular pixel i; if it does, then m_{ij} is equal to the length of the intersection; otherwise it is zero.

To accurately compute the matrix M, we developed a precise projector [3] that computes exact projections for images composed of uniform elliptical and rectangular attenuators, such as rectangular pixels. Let us consider an X-ray beam traveling from the source located at (xA, yA) to the detector located at (xB, yB), through a rectangular object, as shown in Figure 4.1. To determine the contribution of the rectangle to the projection, we need to first decide whether the beam intersects the rectangle, and, if it does, to compute the length of the intersection. The following algorithm achieves these goals.

1. Translate and rotate the coordinates so that the rectangle is centered at the origin and its sides are aligned with the x-y axes.

$$\begin{aligned} x' &= \cos(\varphi)(x - xc) + \sin(\varphi)(y - yc) \\ y' &= -\sin(\varphi)(x - xc) + \cos(\varphi)(y - yc) \end{aligned} \tag{4-2}$$

2. Compute the intersection length l using the following algorithm:

- If the line is horizontal (y'A = y'B) then

$$l = \begin{cases} 2b, & |y'_A| \leq b \\ 0, & \text{else} \end{cases} \tag{4-3}$$

- If the line is vertical (x'A = x'B) then

$$l = \begin{cases} 2a, & |x'_A| \leq a \\ 0, & \text{else} \end{cases} \tag{4-4}$$

- Compute the coordinates of the intersection points

$$\begin{aligned} m &= \frac{y_B - y_A}{x_B - x_A} \\ xc1 &= xA + (-b - yA)/m \\ xc2 &= xA + (b - yA)/m \\ yc1 &= yA + (-a - xA)/m \\ yc2 &= yA + (a - xA)/m \end{aligned} \tag{4-5}$$

- If the line does not intersect the rectangle, that is,

$$\min(|xc1|, |xc2|) > a, \text{ then } l = 0 \tag{4-6}$$

- If the line intersects the rectangle, then

$$\begin{aligned} xs &= \text{sort}(-a, a, xc1, xc2) \\ ys &= \text{sort}(-b, b, yc1, yc2) \\ l &= \sqrt{(x_s(2) - x_s(3))^2 + (y_s(2)^2 - y_s(3))^2} \end{aligned} \tag{4-7}$$

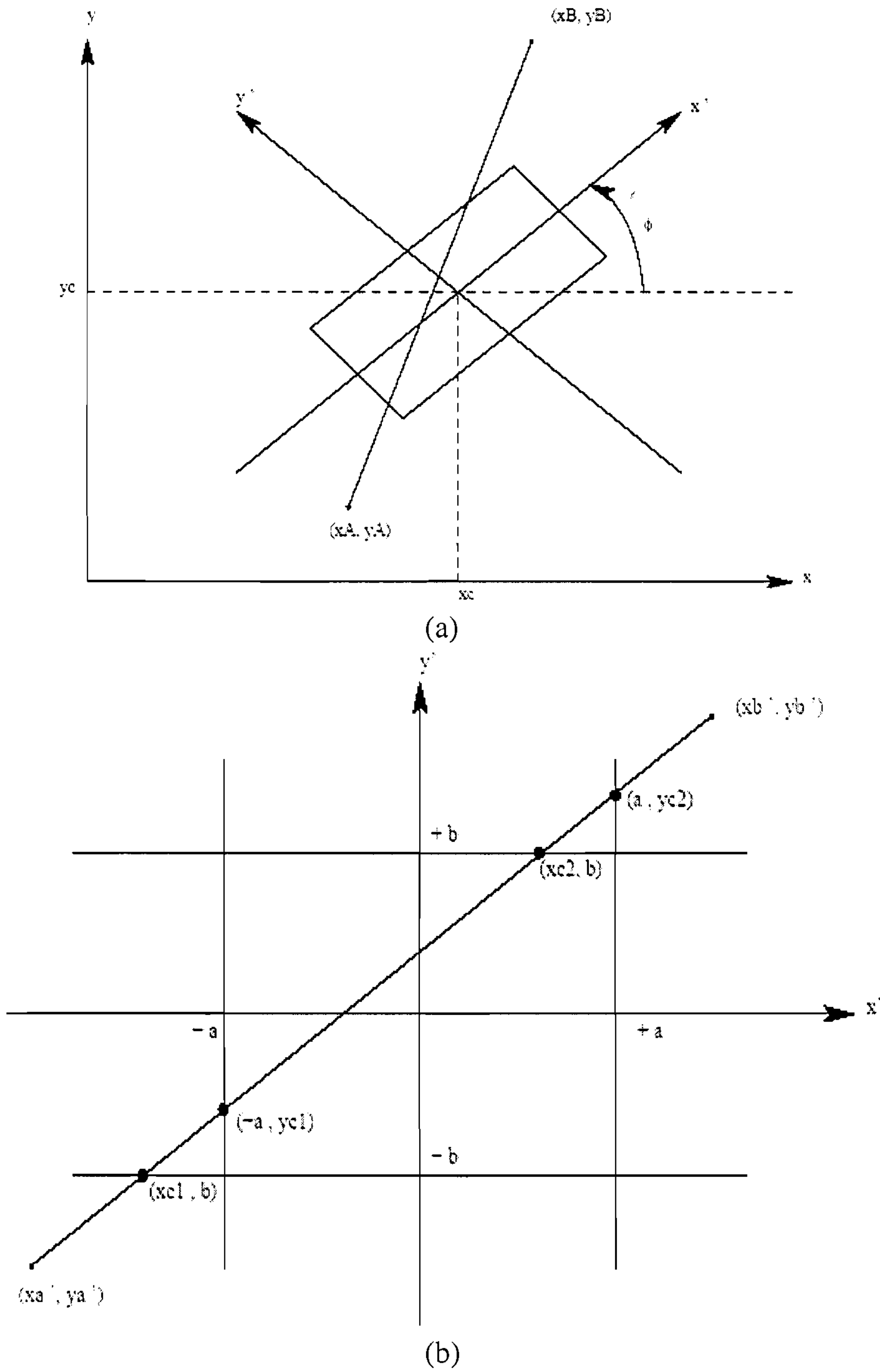


Figure 4.1 (a) Geometry of a rectangular attenuator; (b) Calculation of line integral through the rectangular attenuator.

Given the matrix formulation, the reconstruction from projections problem can be seen as a matrix inverse problem, with solution

$$x = M^{-1}y. \tag{4-8}$$

This is not a feasible solution, because M is, in general, not a square matrix. Therefore, alternate solutions must be used.

4.2 The Van Cittert Algorithm

A number of physical problems can be modeled by equations of the form

$$y(n_1,n_2) = D[x(n_1,n_2)] \tag{4-9}$$

where D[.] is a distortion operator that acts on the input sequence $x(n_1,n_2)$ to produce the output sequence $y(n_1,n_2)$. In many cases it will be very difficult to determine the inverse operator D^{-1} such that

$$x(n_1,n_2) = D^{-1}[y(n_1,n_2)]. \tag{4-10}$$

Even if D^{-1} is approximated and implemented, its application to $y(n_1,n_2)$ could produce large errors if $y(n_1,n_2)$ is known imprecisely due to such uncertainties as measurement or quantization noise. One alternative for finding the inverse operator is by the application of the method of successive approximations. It is based on starting with an initial guess and then iteratively updating the estimate using an equation of the form

$$x_{k+1}(n_1, n2) = F\,[x_k(n_1,n_2)], \tag{4-11}$$

where F is an update operator that varies from one iterative algorithm to another. In all cases though, the exact solution is a fixed point of this operator, that is, for the exact solution $x(n_1,n_2)$,

$$x(n_1, n2) = F\,[x(n_1,n_2)]. \tag{4-12}$$

The iterative methods provide a convenient way to incorporate prior knowledge of the properties of $x(n_1,n_2)$. Assume, for example, that all the pixels in the image $x(n_1,n_2)$

are known to have values in a certain interval. Then after each image update the pixel values can be forced to be within this interval by "clipping" at the interval boundaries. Using a priori information generally increases convergence speed and reduces the probability that the process converge to a local minimum.

The Van Cittert algorithm uses an iterative update given by

$$x_{k+1}(n_1,n_2) = x_k(n_1,n_2) + \lambda(y(n_1,n_2)-D[x(n_1,n_2)]) \tag{4-13}$$

where λ is a constant that determines the convergence speed and stability of the process. A value too small will lead to slow convergence, while a very large value may turn the algorithm unstable. For a few particular distortion operators, formulas have been derived for the optimum value of this parameter. In our research, we have manually adjusted this parameter to obtain an optimal convergence speed.

The general equation above can be rewritten for the matrix formulation of the projection operator as

$$x_{k+1} = x_k + \lambda(y-Mx). \tag{4-14}$$

The terms in this equation have inconsistent sizes, as the matrix M is not square (in other words, the image and the sinogram have different sizes). A version of this algorithm, called the "reblurred" Van Cittert algorithm has been proposed to address this issue. The reblurred iteration is

$$x_{k+1} = x_k + \lambda M'(y-Mx), \tag{4-15}$$

where M' is the transposed matrix M.

One advantage of using the iterative procedure is that it can be stopped after finite number of iterations, at which point the output may be subjectively preferable to the actual inverse filter output.

4.3 The Conjugate Gradient Algorithm

The Van Cittert method is an optimization algorithm that searches for the optimum along the direction of the gradient of the function to be minimized (in our case, the distance d(y,Mx)). This approach can lead to very slow convergence when the function to be minimized is highly asymmetrical in its variables in the neighborhood of the minimum. Faster convergence is achieved by the conjugate gradient method, which uses a different way to determine the direction of search.

The steepest gradient method minimization starts at a point p_o and as many times required, we move from point p_i to the point p_{i+1} by minimizing along the line from p_i, in the direction of the local gradient $-\nabla f(pi)$. The method will perform many steps in going down a long, narrow valley, even if the valley is a perfect quadratic form. This is because the new gradient at the minimum point of any line minimization is perpendicular to the direction traversed. Therefore with the steepest method, right angle has to be taken even if it doesn't lead to the minimum. The conjugate gradient method proceeds not down the new gradient but rather in a direction that is perpendicular to the old gradient, and to all previous directions traversed [24].

Conjugate gradient methods [24, 25] are the most popular methods for solving large systems of linear equation. CG methods provide a suitable method for calculation of N x N linear systems.

$$Ax = b \tag{4-16}$$

The minimization is carried out by generating succession of search directions p_k and improved minimizers x_k. At each stage a quantity αk is found that minimizes $f(x_k+$

$\alpha_k p_k)$ and x_k+1 is set to the new point $x_k + \alpha_k p_k$. The p_k and x_k are built such that x_k+1 is also the minimizer of f over the whole vector space already taken, $\{p_1, p_2, \ldots, p_k\}$.

4.4 The Initial Guess

Estimating an initial guess is very important in the reconstruction of images. The more accurate the estimation is, the better the convergence of the final image to the actual image. There are different methods for the generation of the initial guess. The techniques employed in our research were: zero initial guess, the pseudo-inverse solution, and the backprojection solution.

Zero initial guess: The zero matrix [27] is generated by using the Matlab function zero(). The size of the matrix is equal to the size of the actual image.

$$x = \mathrm{zero}(\mathrm{size}(x)) \tag{4-17}$$

This initial guess does not use any a priori information.

Pseudo-inverse solution: A pseudo-inverse function [27] in available in Matlab that generates, for an input matrix M, a pseudoinverse matrix pinv(M) with the property that

$$M\ \mathrm{pinv}(M)\ M = M. \tag{4-18}$$

Our initial guess is

$$x = \mathrm{pinv}(M)\ y. \tag{4-19}$$

This initial guess uses the projection data as a priori information, but the pseudoinverse calculation is highly sensitive to noise in the data.

Generalized backprojection: A generalized backprojection function, similar to the backprojection technique applied to parallel projections geometry and described in the previous chapter, has been generalized for arbitrary source-detector geometries. The generalized backprojection function takes the matrix and the sinogram as the input and outputs the initial guess. Sinogram values are backprojected along the corresponding line integrals. Pixels not covered by line integrals are interpolated from those who are. As usual, the backprojected images are finally averaged to obtain the initial guess.

4.5 Use of a Priori Information

Despite the great progress in CT, artifacts due to metallic objects remain a major problem in several important clinical applications, including orthopedic, dental, and spine imaging. These artifacts severely impair visualization and quantification of anatomic and/or pathologic features. Several methods have been proposed but none have of them produces results in a robust and efficient manner [19]. Incorporating the known attenuation map and location of high density structures as constraints substantially improves image quality [28]. When prior information of high density objects locations is ignored, significant streaking artifacts remain even when the detector model and the algorithm are perfectly matched.

For reconstruction of an image with a high density object, the reconstruction algorithm has been modified to account for a priori information available about the object's location and density. The most convenient way to incorporate a priori information is through the use of mask image.

Masks [29] are grayscale images that are used to control other images. They may be opened, edited and otherwise changed as we desire. What makes a grayscale scale image a "mask" is simply how we use it. In our case, we use two masks: the high density mask and the circular mask. They are both binary masks, that is, each pixel is a zero or a one.

The high density object mask or metal mask is equal to one for pixels corresponding to the metal object, thus helping to distinguish between the tissues and metal inserts that may be present in the image. We use this mask by forcing all the pixels selected by the mask to the known density of the metal object.

The circular mask is a useful mask if the shape of the region of interest is less than the entire image. It assumes that all pixel values outside a circular region are zero, and is therefore equal to zero for these pixels. We use this mask by forcing to zero all pixels not selected by the mask.

Depending on how the mask has been computed, a mask can be known exactly or it can be determined from the projection data. For the purpose of testing, an exact mask is always available because we created the original image, therefore both the region of interest and the location and shape of the metal object are known. In a practical setting, all that is available is the projection data. Therefore, masks must be computed from this data. The steps we used to compute the metal and circular mask are described in the next chapter.

4.6 Matlab functions

The purpose of this thesis was to develop iterative reconstruction algorithms for CT images with unconventional source-detector configurations using a suitable software package. The algorithms were written and implemented in Matlab. Most of the Matlab functions were developed specifically for the thesis research. However certain functions incorporated in the research were adapted from Matlab code developed by Martin King [26]. The following table lists the Matlab functions used for the thesis research.

User defined Matlab Functions	
backproj	Employs the backprojection technique similar to one used in parallel projections geometry. It takes in the projection matrix M and the sinogram values and generates an initial guess
clam_image32	Reconstructs a CT image for clam-shell structure. The program also generates the phantom image and the sinogram.
clam_testing	Generates the matrix M, sinogram and reconstructs the phantom image from a suitable initial guess.
high_imp	Reconstructs CT image in the presence of metals, when the location of the metal is provided.
line_disp32	Determines if the line intersects the pixel and if it does, measures the length of intersection.
mask_test	Similar to high_imp except that it is used when the location of the metal is not known.
pplate_testing	Generates the matrix M, sinogram and reconstructs the phantom image from a suitable initial guess for parallel plate structure.
test_iter	Reconstructs the CT image for parallel projections.
test_mtx	Generates the matrix M for parallel projections.
Matlab functions referenced from Numerical Recipes in C. [25]	
cg	Finds a vector that gives the minimum of a function.
dfunc	Finds the first derivative of the function to be minimized.
func	Finds the minimum of a function
func_golden	Finds the minimum with tolerance and returns independent variable which is scalar and function value as golden.
func_mnbrak	Brackets the minimum.
Matlab Defined Standard Functions	
phantom	Used to generate a phantom.
pinv	Pseudo-inverse function that takes in the product of the matrix m and sinogram and generates an image similar to the actual image
Zero	Generates an image

Table 2 Matlab functions used in this research

5 Chapter Five – Results

5.1 Image Reconstruction for Parallel Projections

Although the parallel projections geometry is not novel and has been thoroughly studied, we use it to test the functionality of the tools we developed. It is a useful benchmark.

In all experiments, the sinogram is generated by multiplying the matrix with the actual image (i.e. the phantom or the actual CT image). A zero initial guess is used in all cases, except where a pseudo-inverse or generalized backprojection initial guess is explicitly specified. The reblurred Van-Cittert technique is applied to the initial guess and the operation is performed until the resulting image converges to the actual image. Convergence is determined by the amount of iterations required for the reconstructed image to stop improving visually.

Phantom Image: For reconstruction using parallel projections, a 32x32 phantom image was generated using the Matlab function *phantom ()*. The Matlab function *test_mtx()* generates an 1167x1024 projection matrix M. The size of the sinogram is 32x32 corresponding to 32 sources and 32 detectors. The reconstructed images are shown in Figure 5.1. It takes around 10000 iterations for the image to converge. It is observed that as the number of sources and detectors are increased the rate of convergence becomes faster. It takes half the number of iterations for the image to converge if the sources and detector are doubled. Reconstruction of the phantom image is the fastest in parallel projections requiring minimum number of iterations and time for the image to converge.

Actual CT Image: The parallel projections geometry was tested to reconstruct a CT image of the human head. Due to computational complexity limitations (mostly the amount of memory needed to store the matrix M), an original 512x512 CT image was downsampled vertically and horizontally by a factor of 16 to obtain a 32x32-pixel test original. The function *test_mtx* generates an 1167x1024 matrix. The size of the sinogram is 32x32 corresponding to 32 sources and 32 detectors. The reconstructed images are shown in Figure 5.2. It takes roughly 25000 iterations for the image to converge. As the number of sources and detectors are increased the rate of convergence becomes faster. Reconstruction of the CT image is the fastest in parallel projections requiring least number of iterations and time to converge. The parallel projections geometry works very well on CT images even with minimum number of sources and detectors. This was expected, as this is the most regular of all geometries.

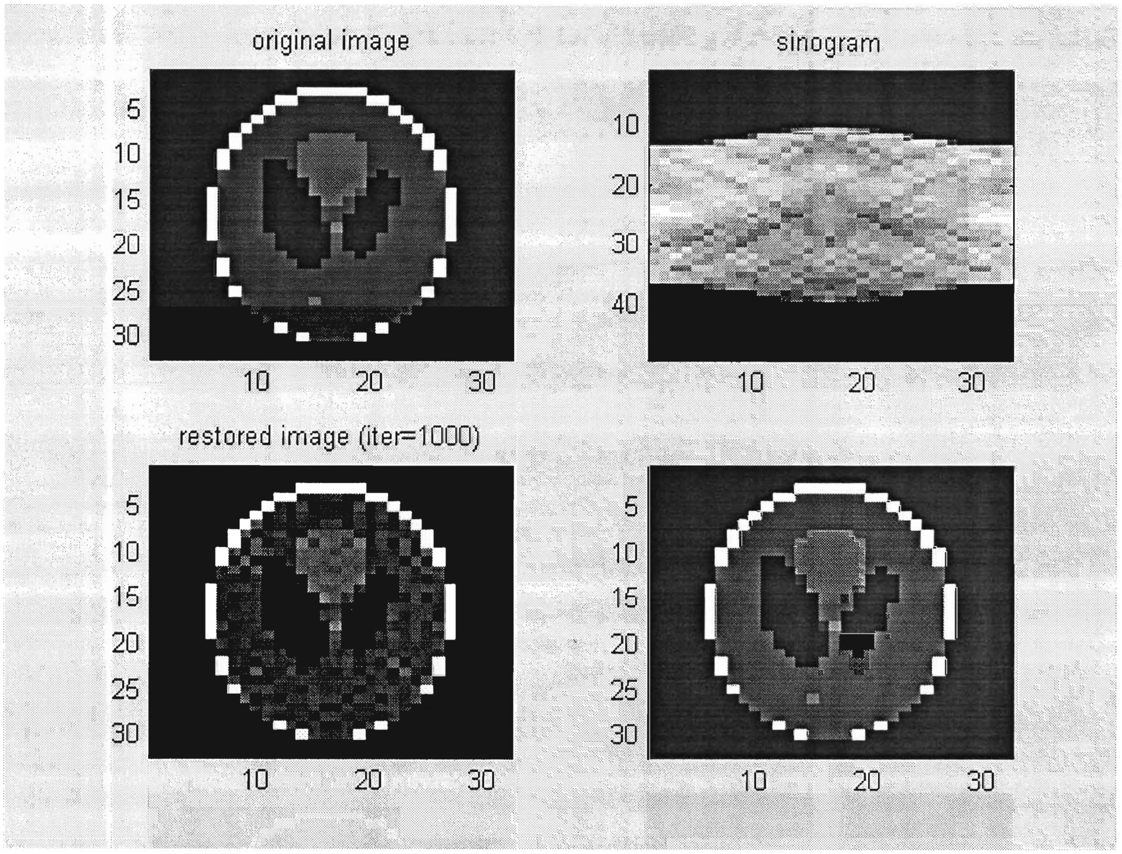


Figure 5.1 Reconstruction of phantom image using the parallel projections geometry (32 sources, 32 detectors, 10000 iterations, lambda=0.5).

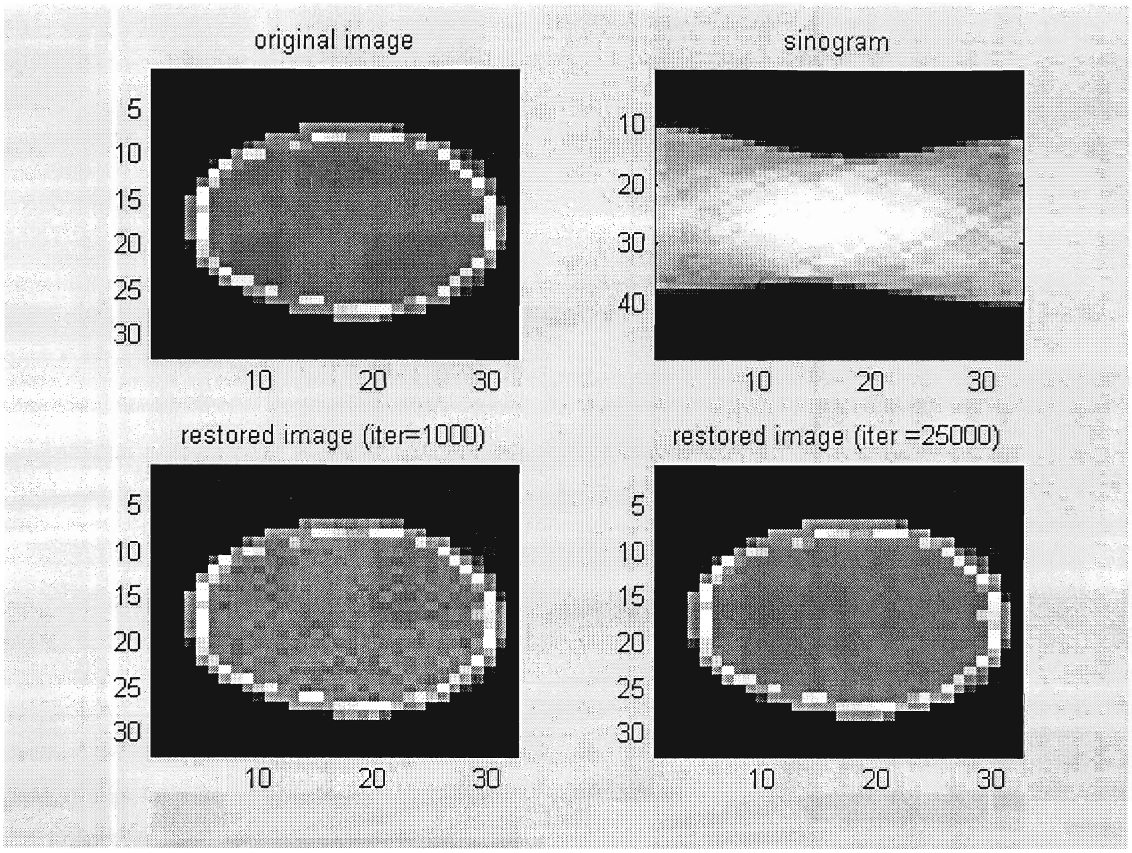


Figure 5.2 Reconstruction of CT image using the parallel projections geometry (32 sources, 32 detectors, 25000 iterations, lambda=0.5).

5.2 Image Reconstruction Using the Clam-Shell Geometry

Phantom Image: Unlike the parallel projections, we require more sources and detectors to reconstruct a 32x32-pixel image. For reconstruction using the clam-shell geometry, a 32x32 phantom image was generated. For our experiment we considered 128 sources and 128 detectors. The function *test_mtx* generates a 16384x1024 matrix. The size of the sinogram is 128x128 corresponding to 128 sources and 128 detectors. It takes roughly 40000 iterations for the image to converge, as shown in Figure 5.3. Reconstruction of the phantom image is slower with clam-shell as compared to parallel projections.

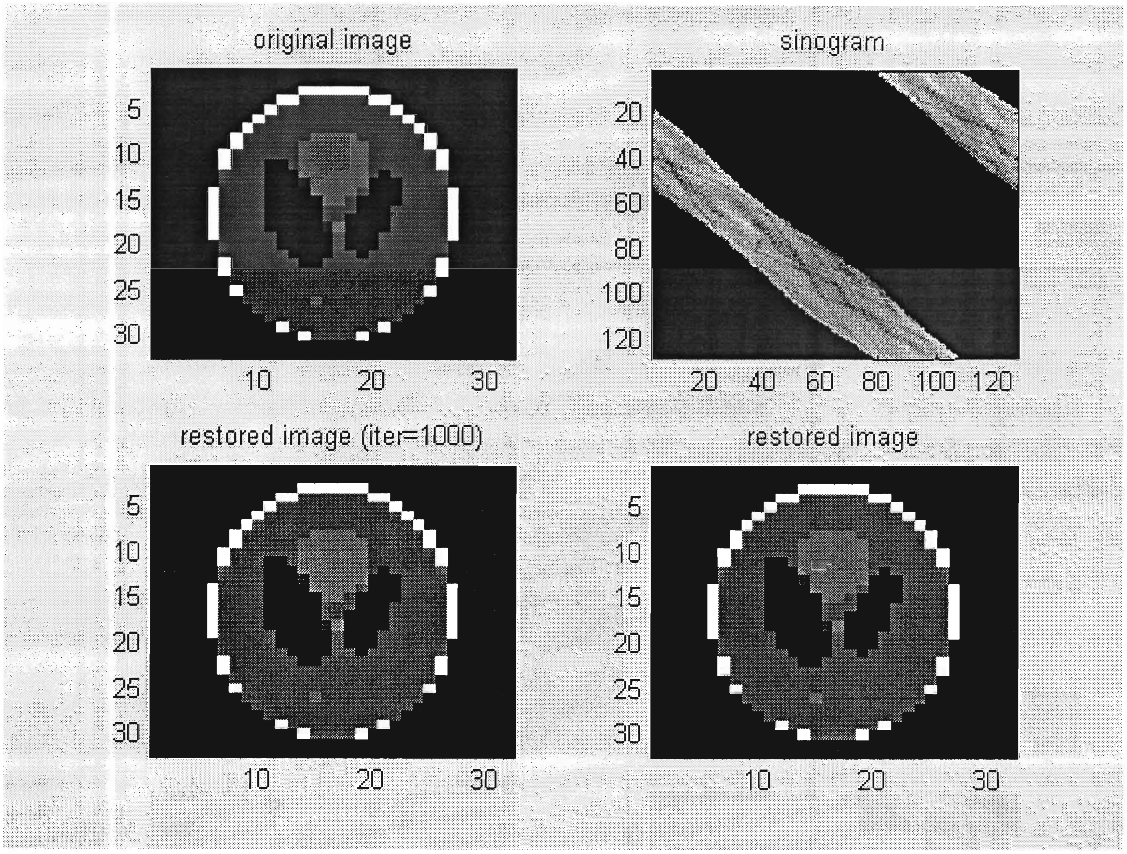


Figure 5.3 Reconstruction of phantom image using the clam-shell geometry (128 sources, 128 detectors, 40000 iterations, lambda=0.5).

CT Image: In the case of reconstruction of CT images using the clam-shell geometry, we considered two different cases: human brain image for the imaging of bone structures human brain image for the imaging of soft tissue.

In the case of reconstruction of the CT image with bone structures, we had to consider 128 sources and 128 detectors. Consider Figure 5.4: a 32x32 phantom image was generated. The function *test_mtx* generates a 16384x1024 matrix. The size of the sinogram is128x128 corresponding to 128 sources and 128 detectors. It takes roughly

45000 iterations for the image to converge. Reconstruction of the phantom image is slower with the clam-shell structure as compared to the parallel projections geometry.

In the case of human brain with tissue visible, we again considered 128 sources and 128 detectors (Figure 5.5) and the same procedure was followed. However more iterations are required for convergence to occur as compared to earlier cases.

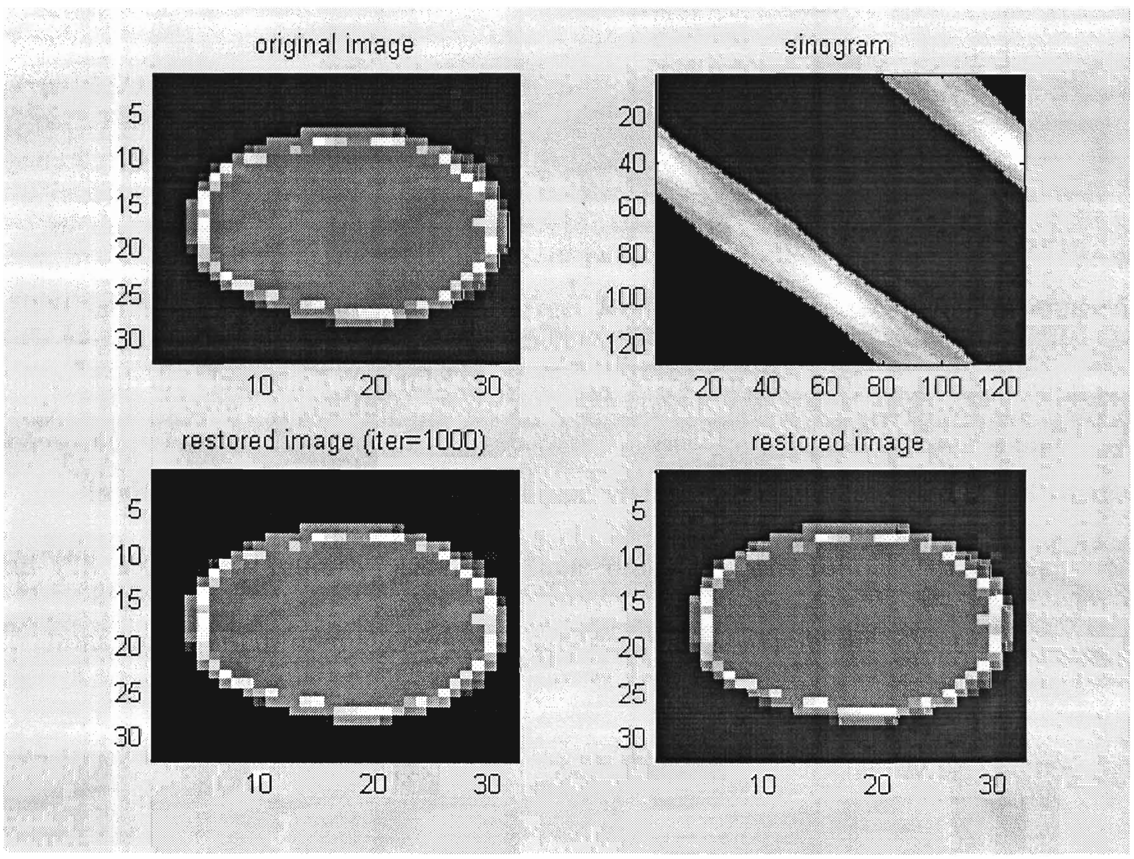


Figure 5.4 Reconstruction of CT image (bone visible) using the clam-shell geometry (128 sources, 128 detectors, 45000 iterations, lambda=0.5).

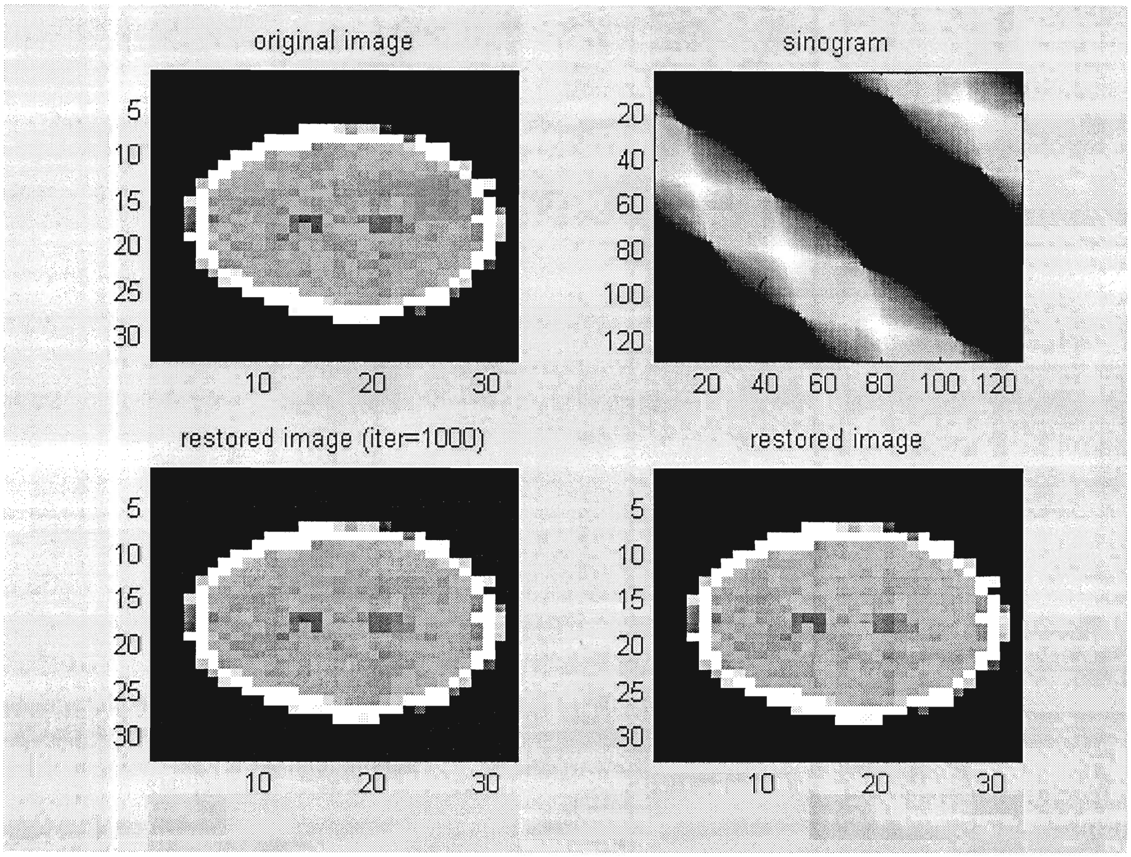


Figure 5.5 Reconstruction of CT image (visible tissue) using the parallel projections geometry (128 sources, 128 detectors, 45000 iterations, lambda=0.5).

5.3 Image Reconstruction for Parallel Plate Geometry

For reconstruction using the parallel plate geometry, a 32x32 phantom image was generated. The Matlab function *test_mtx* generates an 1167x1024 matrix (see Figure 5.6). The size of the sinogram is 32x32 corresponding to 32 sources and 32 detectors. Unlike the parallel projections and clam-shell geometry, reconstruction using parallel plate geometry results in a very poor output. The resultant is blurred and it is difficult to distinguish between the different layers in the image. This is because, unlike the clam-shell and parallel projections which are more or less circular in structure with X-ray beams being detected in all directions, there are no X-rays passing through the image vertically in a parallel plate structure as it is open at the top and the bottom. The solution to this problem could be to increase the length of the plates or to use slightly curved plates.

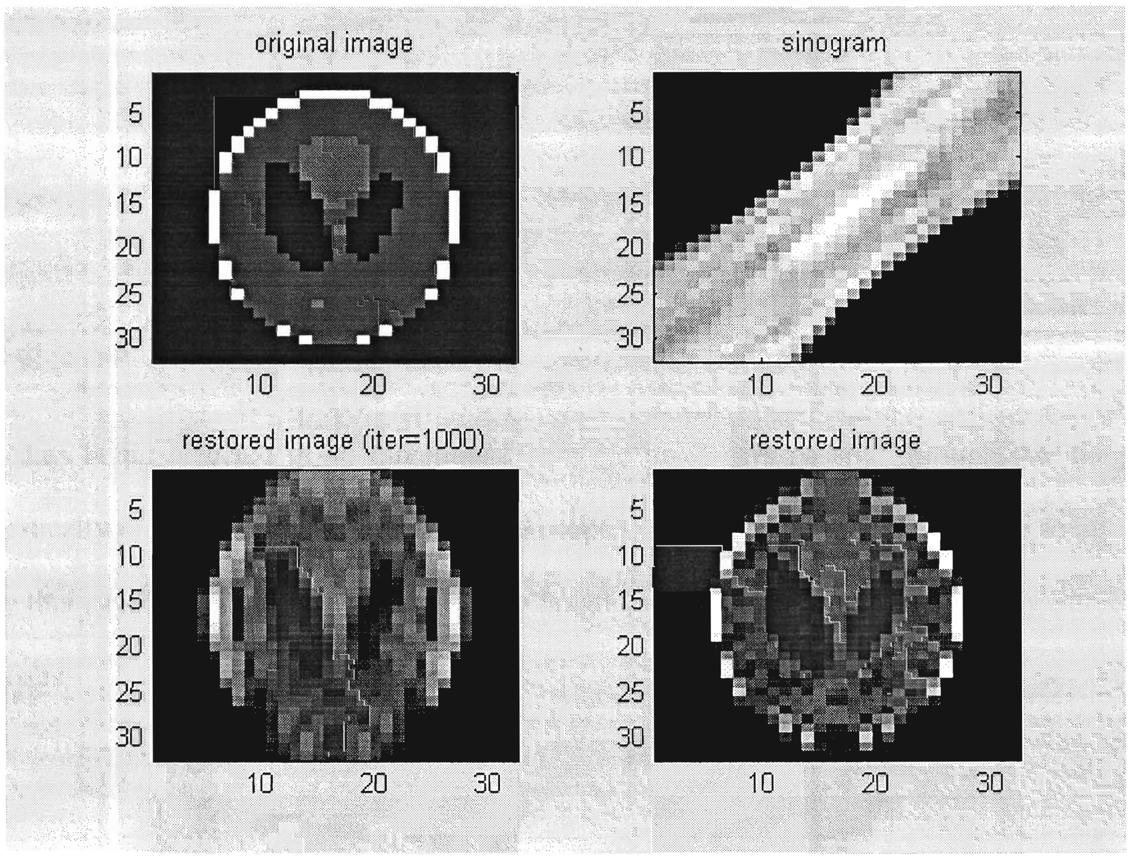


Figure 5.6 Reconstruction of phantom image using the parallel plate geometry (32 sources, 32 detectors, 60000 iterations, lambda=0.05).

5.4 Comparison of the Parallel, Clam-Shell and Parallel Plate Configurations

The Mean Square Error (MSE) for each configuration was plotted for different number of sources and detectors. The number of sources and detectors considered are 16, 32 and 64. It can be observed from the Figure 5.7 that as the number of sources and detectors increase for each configuration there is a corresponding decrease in the mean square error. The Mean Square Error is lowest for parallel projection configuration as compared to that for the parallel plate and clam-shell configurations. Increasing the number of sources and detectors results in increase in the number of projections generated at the detector. Hence more data is available for processing information and a clearer image is obtained.

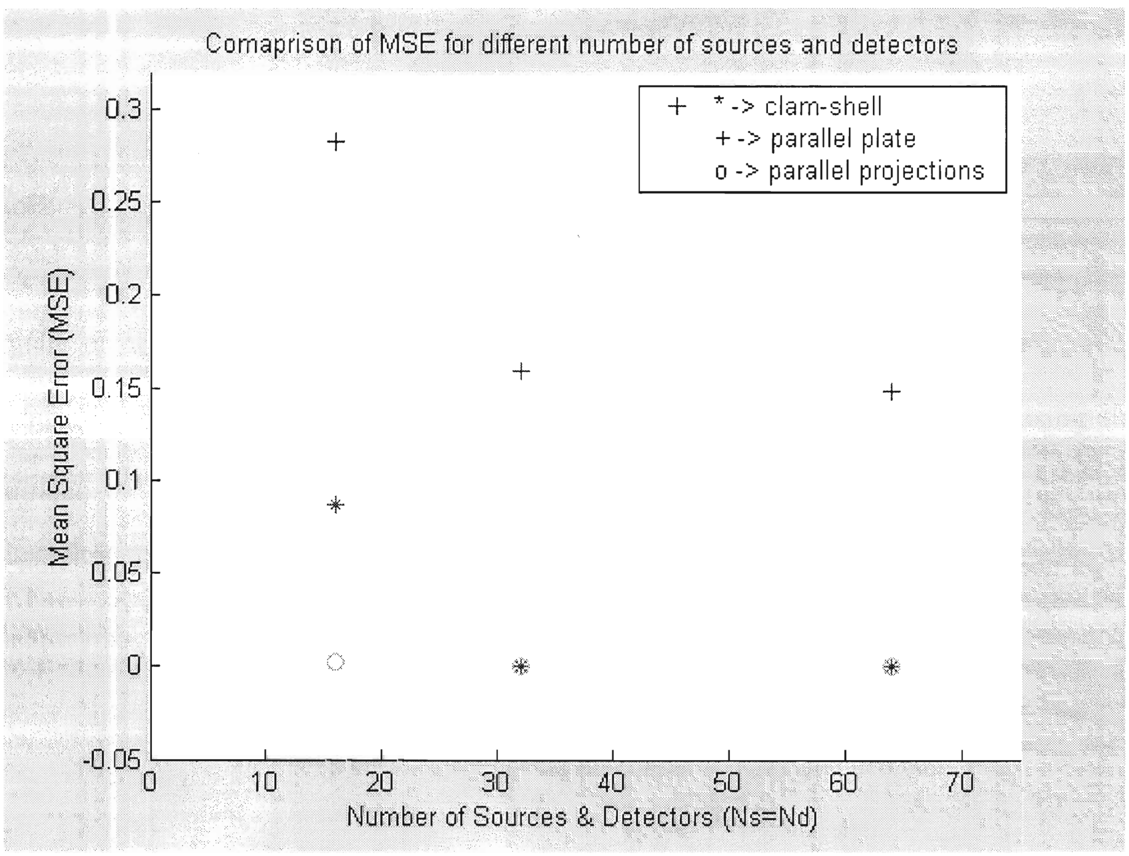


Figure 5.7 Comparison of MSE for different number of sources and detector

5.5 Comparison of the Three Configurations using Plain Gradient and Conjugate Gradient Methods

Parallel Projections: Figure 5.8 was plotted for the parallel projections configuration with 32 detectors and 32 sources and a square image of size 32x32. The initial guess was considered as set of zeros of size 32x32. The Mean Square Error was plotted against the number of iterations for the two gradient methods. The Mean Square Error when obtained using the plain gradient methods is initially very high but as the iterations increase, there is a sharp decrease from 0.16 units to zero where it remains constant. For the conjugate gradient method, MSE remains almost constant around 0.038 units throughout except for a small decrease initially. Note that the units used for the MSE are the same units used for the pixel values. Since all our images are normalized, the pixel values are between 0.0 and 1.0. So a MSE value of 0.038 represents 3.8% of the dynamic range of the image.

Clam-shell Geometry: The plot of MSE for the clam-shell geometry is similar to the parallel projections configuration. Figure 5.9 was obtained using a 32x32 square image, 32 detectors and 32 sources and initial guess as set of zeros of size 32x32. The plain gradient method gives a better result for clam-shell as compared to conjugate gradient method over all iterations.

Parallel Plate Geometry: The parallel plate configuration generates a better result with conjugate gradient method as compared to plain gradient technique. The error is an exponentially rising curve initially and then becomes constant for plain gradient and conjugate gradient methods. Figure 5.10 is obtained by considering a 32x32 square image with 32 sources and 32 detectors. Since neither algorithm actually converges, it is irrelevant which yields the smaller MSE.

It can be observed from the three plots that the Mean Square Error (MSE) is the least in case of parallel projections as compared to the clam-shell and parallel plate geometries. The MSE decreases to zero in the case of parallel projection and the clam-shell however, it is the opposite in case of parallel plate. The MSE decrease is steeper in the case of parallel projections as compared to the clam-shell geometry.

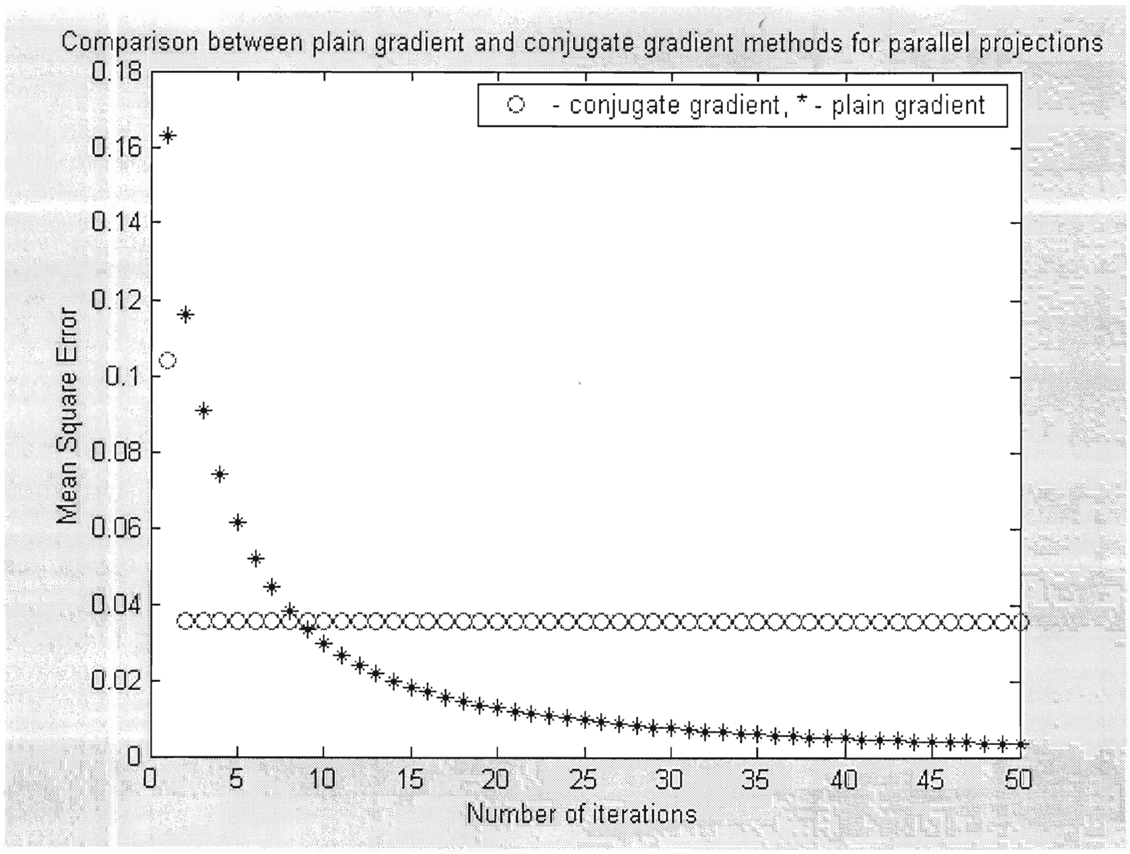


Figure 5.8 Comparison between the plain gradient and the conjugate gradient methods for parallel projections.

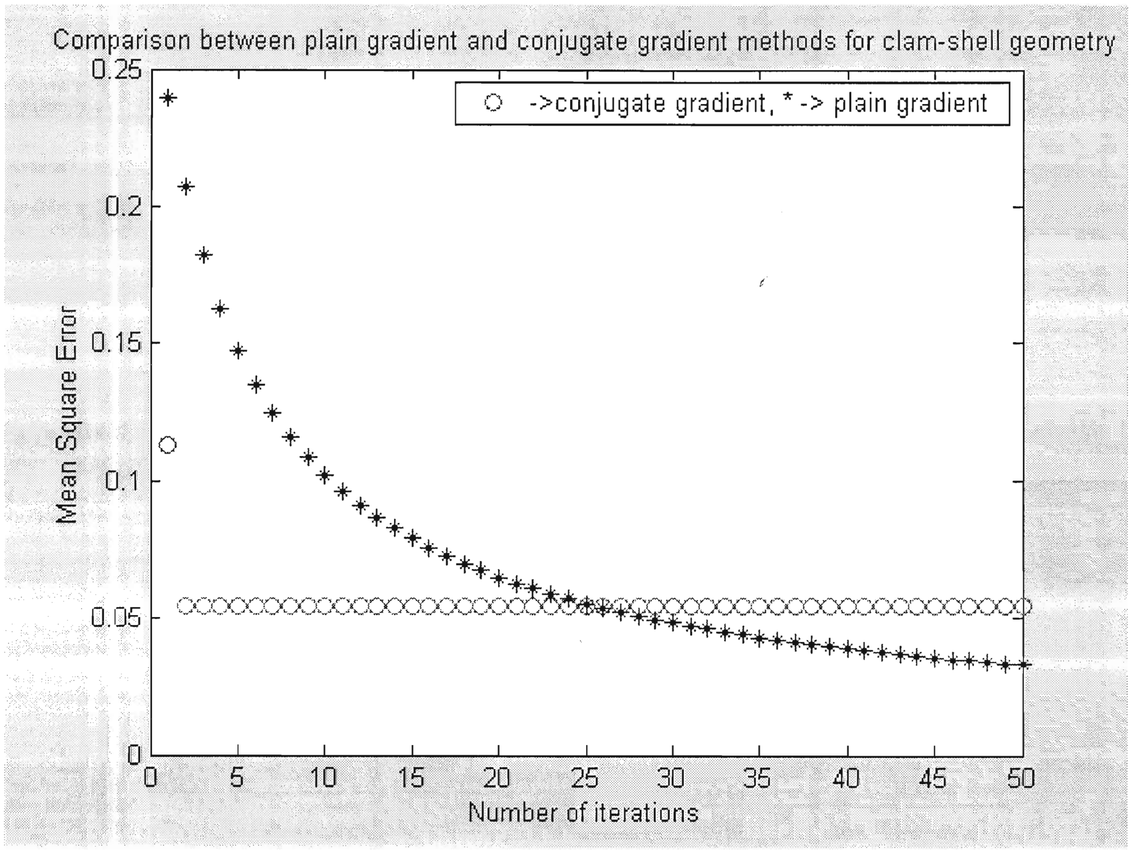


Figure 5.9 Comparison between the plain gradient and the conjugate gradient methods for clam-shell geometry.

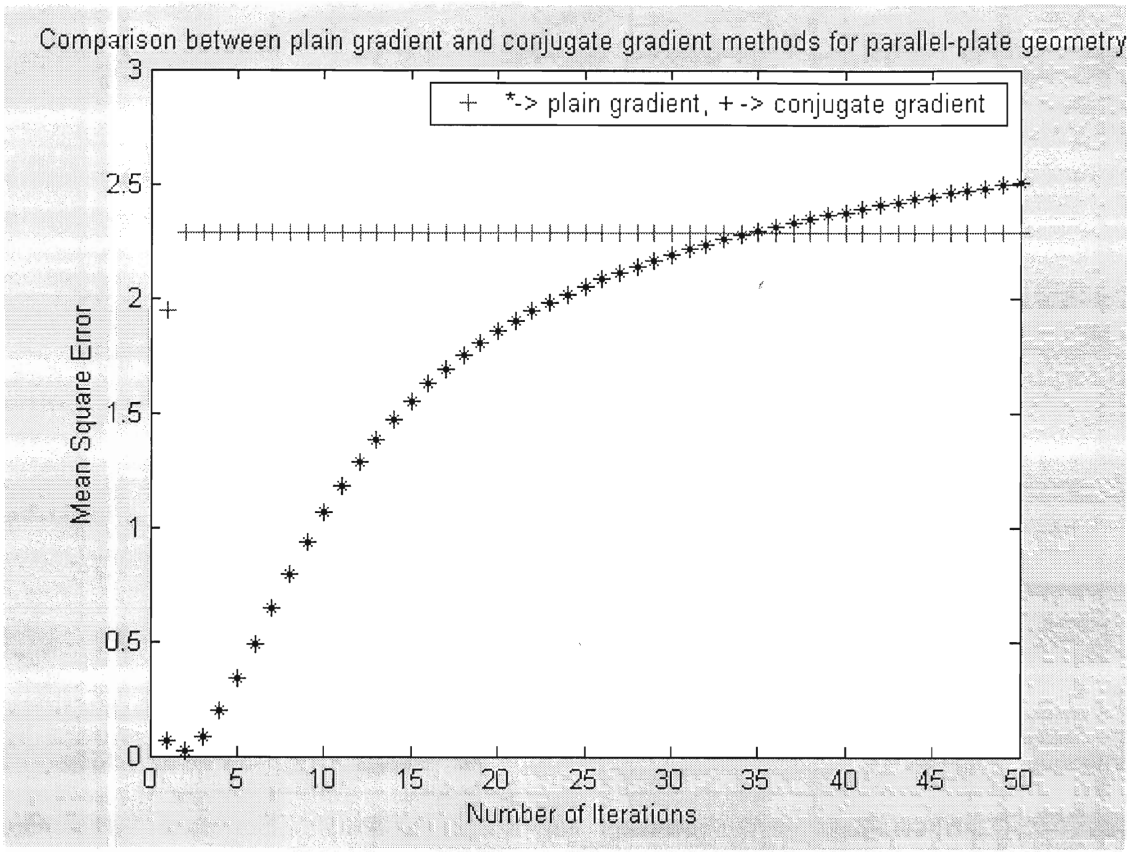


Figure 5.10 Comparison between the plain gradient and the conjugate gradient methods for parallel plate geometry.

5.6 Comparison of Different Initial Guess Choices

A comparison of Mean Square Error was performed by considering three different initial guesses: the zero solution, the pseudoinverse solution, and the generalized backprojection solution. The comparison was performed by considering a CT image of human head of size 32x32 pixels. The experiment was performed by application of plain gradient iterative reconstructive algorithm for the clam-shell geometry. From the graph of MSE versus the number of iterations in Figure 5.11, it can be observed that the time and number of iterations required for convergence to occur with the backprojection solution function is much smaller as compared to that required with the other two guesses.

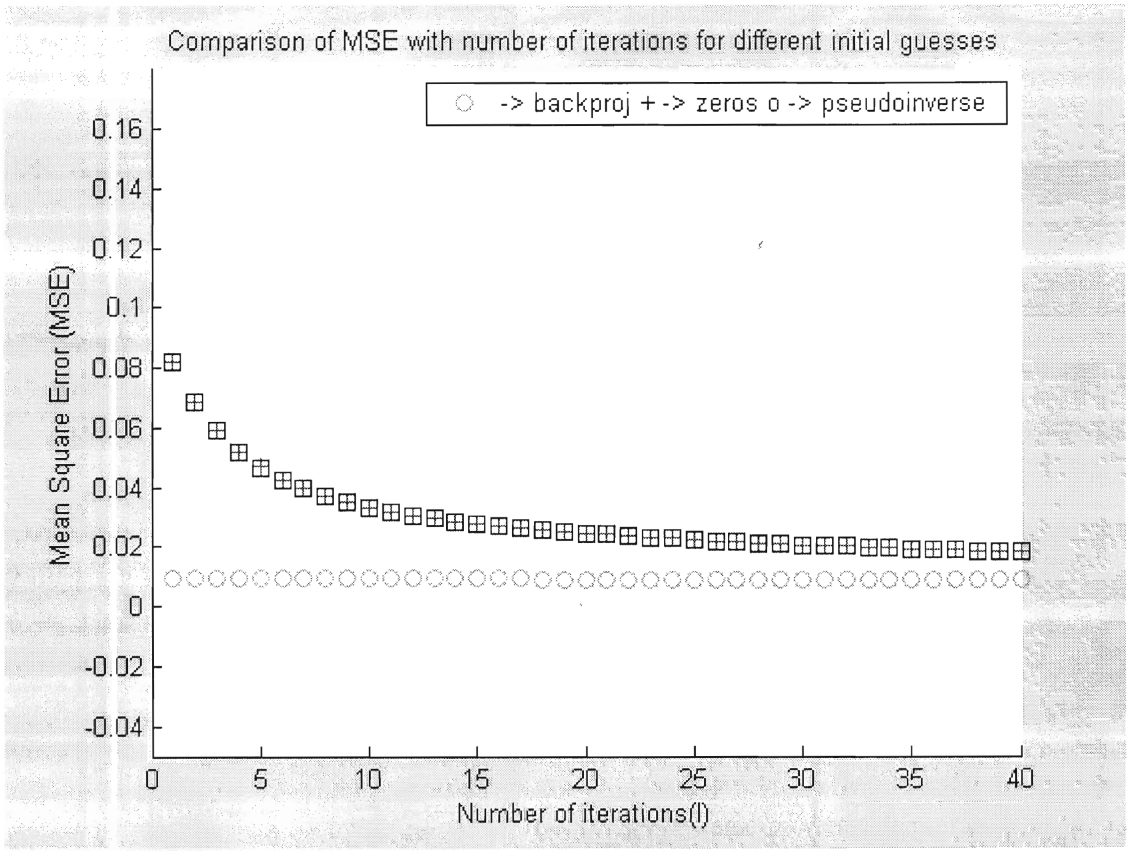


Figure 5.11 MSE Comparison for different initial guesses.

5.7 Handling of High Impedance Objects

It is well known that when metal is present in the region of interest, pronounced dark and bright streaks are produced in reconstruction with conventional filtered back projection. These artifacts seriously degrade image quality, particularly near the metal region [18]. Iterative reconstructive algorithms have been successfully applied with incomplete/noisy projections data. We will consider two cases: the location and shape of the metal object is known, and they are not known. In both cases, the density of the metal is known. In both cases, a high-density circular object was introduced in the phantom image. To simulate photon starvation, the line integrals passing through the metal object have been set to zero, as shown in Figure 5.12.

Location of High Density Object Known

We implemented a reblurred Van Cittert iterative technique and applied it to the parallel projections, clam shell and parallel plate geometries. To model photon starvation, All the rows of the matrix M corresponding to sinogram values falling in the shadow created by the metal object have been set to zero. The reconstruction algorithm was executed with this modified matrix. Prior information was available in the form of a metal mask and a circular mask.

The results of the investigations are presented in Figure 5.13. Experiments for parallel projections were initially performed with 32 sources and 32 detectors and then with 32 sources and 129 detectors. With 32 sources and detectors the image is blurred with considerable streaking. However on increasing the number of detectors to 129, when the prior information is provided and a mask is applied the restored image is clear, except for few light streaks observed around the object. If the a priori is not used, a black spot can be seen at the location of the object as shown in Figure 5.12, although the effect is not too damaging to the image quality.

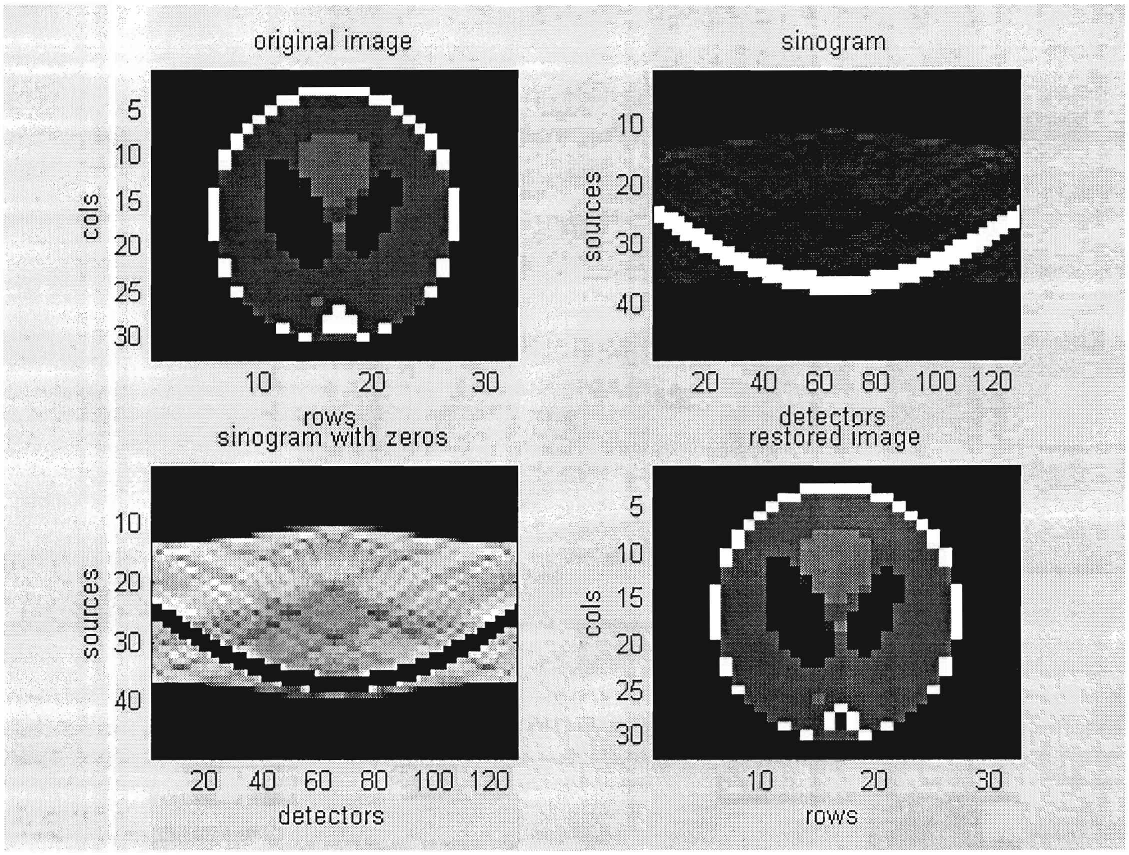
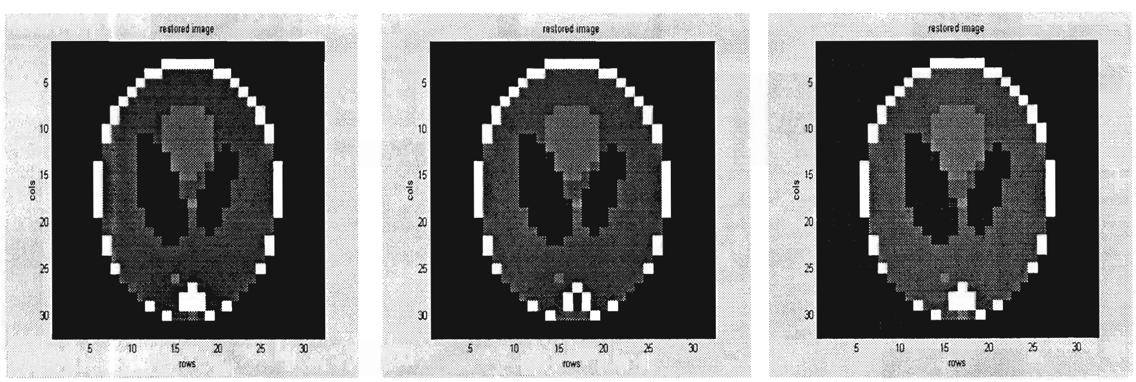


Figure 5.12 Reconstruction of phantom image with a high density object using parallel projections. a) Original image (size 32x32) with high density object, b) Ideal sonogram, without photon starvation, c) Simulated real sonogram, with shadow created by photon starvation, d) Restored image using the Van Cittert method without a priori information.



(a) (b) (c)

Figure 5.13 a) Reconstructed image with circular mask and metal mask, b) Reconstructed image with circular mask and no metal mask, c) Reconstructed image without circular mask but with metal mask used.

From Figure 5.14, it can be observed that the clam-shell geometry shows results similar to that shown using parallel projections. However, if prior information of the metal location is not provided, the reconstructed image is much more blurred as compared to the images resulting from parallel projections. Thus, prior information of the metal is more important in case of clam-shell geometry.

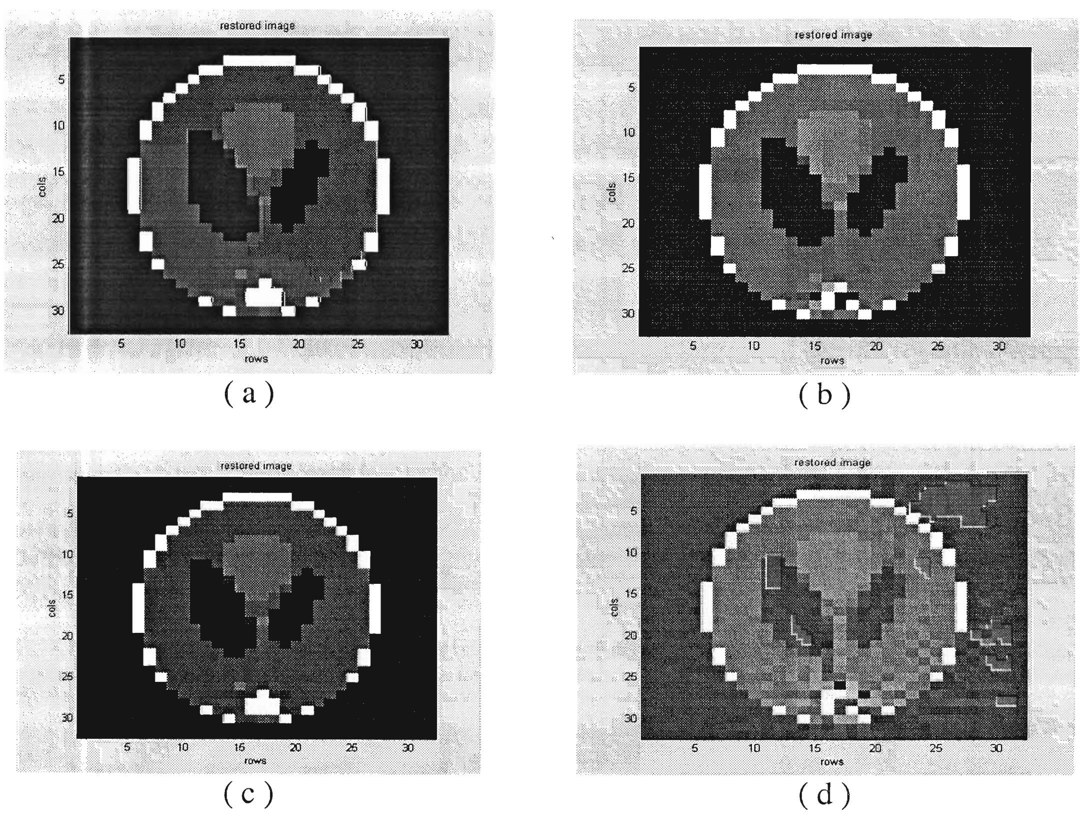
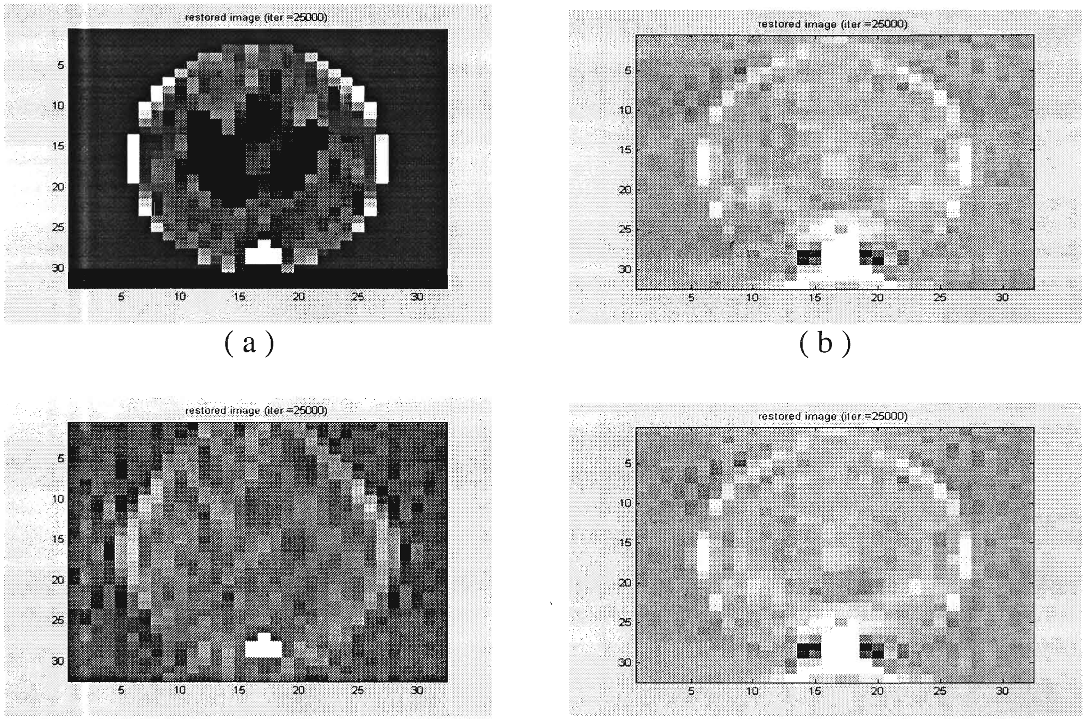


Figure 5.14 Reconstruction of a phantom image with a high density object using the clam-shell geometry. a) with circular mask and metal mask, b) with circular mask and no metal mask, c) without circular mask but with metal mask, d) without a priori information.

For the parallel structure, no improvement was observed by using a priori information. The streaking artifacts visible in Figure 5.15 are even more pronounced when high attenuation objects exist in the body.



(a) (b) (c) (d)

Figure 5.15 Reconstruction of phantom image with a high density object using the parallel plate geometry. a) with circular mask and metal mask, b) with circular mask and no metal mask, c) without circular mask but with metal mask, d) without a priori information.

Location of High Density Object not Known

In practice, none of the masks is available as side information. The mask information has to be extracted from the projection data. The metal mask is estimated as follows:

- Start with a mask equal to zero.
- For each line integral with a value greater than a threshold value (determined using the histogram, as mentioned before), add 1.0 to the mask values of the pixels that are touched by the line integral.

- When all line integrals have been processed, set mask values larger than a threshold (about 25% of the maximum value of the mask) to 1.0 and the rest to 0.0.

The circular mask is estimated in a similar fashion, only the threshold in the second step is set to zero. The mask estimates are remarkably accurate, and the reconstruction results are indistinguishable from the case when the masks were known.

6 Chapter Six - Conclusion

The aim of this thesis research was to develop reconstruction algorithms for unconventional tomography geometries that can be implemented in a miniaturized CT scanner. We investigated three such structures, namely, parallel projections, clam-shell and parallel plate. Our experiments prove that the parallel projections geometry produce excellent results compared to clam-shell and parallel plate geometries. However the clam shell geometry shows dramatic improvement in results with increasing sources and detectors.

Another factor that influences the output of the clam–shell geometry is the positioning of sources and detectors. The sources and detectors can either be placed alternatively or opposite to each other. If placed alternatively then the geometry is similar to that of the parallel projections, resulting in clearer images. The parallel plate geometry is the least effective of all the three. The images generated by the parallel plate geometry are very blurry. This is due mostly to the gap in the pattern of X-rays. Besides, it requires a higher dose of radiation. However by increasing the length of the plates, the quality of the image may be improved.

The Van Cittert technique, also known as the plain conjugate method is one of the most commonly used distance minimization techniques for the reconstruction of CT images. However, it is slower as compared to other minimization techniques. To increase the speed of iterations the Conjugate Gradient (CG) Method was implemented, which proved to be faster but less accurate than the plain gradient method. Several choices for the initial guess have been evaluated, and the generalized backprojection has been found to best help achieve fast convergence.

We also tried to tackle the issue of streaking artifacts in images due to presence of metallic artifacts in the vicinity of the region of interest in the body. It can be seen that if prior information is provided, there is a substantial improvement in the quality of the image. We implemented a priori information in the form of masks that distinguish the metal from the rest of the image and separate the patient body from its surrounding. We have shown that the masks can be accurately estimated from the projection data.

Several topics that we suggest for further research are iterative methods based on the minimization of the I-divergence (instead of the MSE), the effect of the finite size of detectors, the effect of the square pixel approximation, ignored in our research so far. These factors can be integrated into the reconstruction algorithms we developed, as steps towards a real, practical, miniaturized CT scanner.

REFERENCES

[1] Jiang Hsieh, Computed Tomography: Principles, Design, Artifacts and Recent Advances.

[2] A. Docef, MiniCAT- A miniaturized CT scanner. SCEEE Report. 2004

[3] Euclid Seeram, Computed Tomography: Physical Principles Clinical Applications & Quality Control, Philadelphia : W.B. Saunders, 2001.

[4] http://www.gemedicalsystems.com/medical/ct/msctprod.html

[5] http://www.analogic.com/Level4/CT folder/Medical Imaging.html

[6] Butler WE, Piaggio CM, Constantinou C, Niklason L, Gonzalez RG, Cosgrove GR, Zervas NT : A mobile computed tomographic scanner with intraoperative and intensive care unit applications. Neurosurgery 42:1304-10, 1998.

[7] White CS, Meyer CA,Wu J, Mirvis SE: Portable CT: assessing thoracic disease in the intensive care unit. AJR Am J Roentgenol 173:1351-6, 1999.

[8] Mayo-Smith, L. M. Davis, N. C. Clements et al., "CT of the brain: a comparison of transportable and fixed-platform scanners," American Journal of Roentgenology 173:1481-1484, 1999. MiniCat - SCEEE Report 33

[9] http://www.digiray.com/medical/brain-ct.htm.

[10] S. Gondrom, J. Zhou, M. Maisl et al., "X-ray computed laminography: an approach of computed tomography for applications with limited access," Nuclear Engineering Designs 190(1-2):141-147, 1999.

[11] Y. Arai, E. Tammisalo, K. Iwai et al., "Development of a compact computed tomographic apparatus for dental use," Dentomaxillofacial Radiology 28(4):245-248, 1999.

[12] M. Rokitta, E. Rommel, U. Zimmermann et al., "Portable nuclear magnetic resonance imaging system", Review of Scientific Instruments 71(11):4257-4262, 2000.

[13] http://www.improvedoutcomes.com/docs/WebsiteDocs/Clustering/Clustering_ Parameter/ Euclidean and Euclidean Squared Distance Metrics.htm.

[14] http://encyclopedia.laborlawtalk.com/Maximum_norm.

[15] http://encyclopedia.laborlawtalk.com/Taxicab_geometry.

[16] http://mathworld.wolfram.com/ExtremeValueTheorem.html.

[17] Csiszar, Budapest., Information Theoretic Methods in Probability and Statistics.

[18] Ge Wang, Donald I. Snyder, J.A. Sullivan, M.W. Vannier, Iterative Deblurring for CT Metal Artifact Reduction, IEEE Transactions on Medical Imaging, vol. 15, no. 5 Oct 1996.

[19] S.Zhao, D.D. Robertson, G. Wang, B.Whiting, K.T. Bae, X-ray CT Metal Artifact Reduction Using Wavelets: An Application for Imaging Total Hip Prostheses, IEEE transactions on Medical Imaging, vol.19, no.12, Dec 2000.

[20] C.L. Epstein, Introduction to the Mathematics of Medical Imaging.

[21] L.A. Shepp and B.F. Logan, "The Fourier reconstruction of a head section," *IEEE Trans. Nucl. Sci.*, vol. NS-21, pp. 21-43, 1974.

[22] http://www.physics.nist.gov/PhysRefData/XrayMassCoef/cover.html

[23] P. H. VanCittert, "Zum Einfluss der Splatbreite auf die Intensitatswerteilung in Spektrallinien II", Z. fur Physik, 69 (1931), pp 298-308.

[24] http://www.library.cornell.edu/nr/bookcpdf/c2-7.pdf.

[25] Numerical recipes in C: The Art of Scientific Computing. Ch 10, pg 420

[26] http://www.ictp.trieste.it/~mpking/cg.html.

[27] Polak E. Computational Methods in Optimization (New York Academic Press), 1971 Chapter 2.3.

[28] Jacobs D.H. (ed) The State of the Art in Numerical Analysis (London Academic Press), 1977, Chapter III, 1.7 (by K.W.Brodle).

[29] http://www.mathworks.com

[30] J.F. Williamson, B.R. Whiting, J. Benac, R.J.Murphy, G.J. Blaine, J.A. O'Sullivan, D.G. Politte & D.L. Snyder, Prospects for Quantative CT Imaging in the presence of foreign Metal Bodies Using Statistical Image Reconstruction, Biomedical Imaging, 2002. Proceedings 2002 IEEE International Symposium on 7-10 July 2002, Page(s):649 – 652.

[31] http://exchange.manifold.net/manifold/manuals/5_userman/mfd50Masks.htm.

APPENDIX

To generate initial guess using backprojection technique

```
function x = backproj(H,y)
% The (unfiltered) backprojection solution
x = zeros(size(H,2),1);
for r=1:size(H,1),
       x1 = zeros(size(H,2),1);
       idx = find(H(r,:)~=0);
       x1(idx) = y(r) .* H(r,idx) / sum(H(r,idx)) ;
       x = x + x1;
end
x = x/size(H,1);
```

Reconstruction algorithms for parallel projections

```
%%%%%%%%%%%%%   to generate the matrix    %%%%%%%%%%%%%%
% image: always 1.0x1.0 area centered at origin
% pixels are located in the middle of their rectangular areas
rows = 32;
cols = 32;
x_coords = ((-cols+1):2:(cols-1) ) / (2*cols);
y_coords = ((-rows+1):2:(rows-1) ) / (2*rows);
global mtx
% source-detector geometry
theta_vals = [0:1:32]*180/32;
thetas = length(theta_vals);
ds = 2*ceil(norm([rows cols]-floor(([rows cols]-1)/2)-1))+3;
d_vals = sqrt(2) * ((-ds+1):2:(ds-1) ) / (2*ds);

% compute projection matrix
mtx = zeros(thetas*ds,rows*cols);
D = 1.5;      % distance between sources and detectors
L = 1.5;      % width of line of sources and detectors
NS = ds;      % number of sources
ND = ds;      % number of detectors

for itheta=1:thetas,
   itheta;
   theta = theta_vals(itheta);
   theta_rad = theta*pi/180;
   r = 0.5*sqrt(L*L+D*D);
   beta = atan(L/D);
   src_angle1 = theta_rad + pi - beta;
   src_angle2 = theta_rad + pi + beta;
   src_z1 = r*exp(j*src_angle1);
   src_z2 = r*exp(j*src_angle2);
   src_z = src_z1 + (src_z2-src_z1)*[1:2:(2*NS-1)]/(2*NS);
   src_x = real(src_z);
   src_y = imag(src_z);
```

```
   det_angle1 = theta_rad + beta;
   det_angle2 = theta_rad - beta;
   det_z1 = r*exp(j*det_angle1);
   det_z2 = r*exp(j*det_angle2);
   det_z = det_z1 + (det_z2-det_z1)*[1:2:(2*ND-1)]/(2*ND);
   det_x = real(det_z);
   det_y = imag(det_z);

   for id=1:ds,
      x1 = src_x(id);
      y1 = src_y(id);
      x2 = det_x(id);
      y2 = det_y(id);
      h=line_disp_32(x1,x2,y1,y2,x_coords,y_coords);
      mtx((itheta-1)*ds+id,:) = h(:)';
   end
end
save h.txt mtx -ascii;

%%%%%%%%%%%%  reconstruction algorithm    %%%%%%%%%%%%%%%%
N=32; % number of sources/detectors
rows = 32;
cols = 32;
lambda=0.5;
load h.txt;
S=sparse(h);

% testing
%img32=zeros(32,32); img32(8:24,8:24)=1;
img32=phantom('Modified Shepp-Logan',32);% img32=img32/max(img32(:));
figure(1); subplot(1,3,1); imagesc(min(img32,1)); title('original
image');xlabel('rows');ylabel('cols');colormap(gray(256));

global sino;
sino=S*img32(:); sino =reshape(sino,49,33);
save sino.txt sino -ascii;
subplot(1,3,2);imagesc(sino);
colormap(gray(256));title('sinogram');xlabel('detectors');ylabel('sourc
es');

m=backproj(S,sino);
for l=1:60000,
   a=sino-reshape(S*m(:),49,129);
   fprintf(1,'%i %f\n', l, norm(abs(a(:))));
   z=reshape(S'*a(:),N,N);
   m=reshape(m,32,32)+lambda*z;
   %yk+1(n1,n2)=yk(n1,n2)+lambda*z(n1,n2)*h
   m = max(m,0);
   m = min(m,1);
   if l==1000,
      subplot(2,2,3); imagesc(min(m,1));
      title('restored image (iter=1000)');
   end
   if mod(l,10)==1,
```

```
      figure(2);imagesc(min(m,1));
      title('restored image (iter =25000)');
      colormap(gray(256));drawnow;
   end
```

Reconstruction algorithm for clam-shell geometry

```
%%%%%%%%%% to generate matrix and reconstruct the image %%%%%%%%%
rows = 32;
cols = 32;
x_coords = ((-cols+1):2:(cols-1) ) / (2*cols);
y_coords = ((-rows+1):2:(rows-1) ) / (2*rows);
clam_rad=1; lambda=0.5;
Nsource= 64; theta_min=95; theta_max=445;
Ndetector= 64; %dtheta_min=-85; dtheta_max=+85;
mtx = zeros(Nsource*Ndetector,rows*cols);
step=(theta_max-theta_min)/(Nsource-1);
theta_vals=[theta_min:step:theta_max];
theta=theta_min;
for iN=1:64,
   theta_rad = theta*pi/180;
   x_source=clam_rad*cos(theta_rad);
   y_source=clam_rad*sin(theta_rad);
   x_detector=clam_rad*cos(theta_rad+(step/2));
   y_detector=clam_rad*sin(theta_rad+(step/2));
   source_mat(iN,:)=[x_source y_source];
   detector_mat(iN,:)=[x_detector y_detector];
   theta=theta+step;
end
for i=1:Nsource,
   for j=1:Ndetector,
      h=line_disp_32(source_mat(i,1),detector_mat(j,1),
         source_mat(i,2),detector_mat(j,2),x_coords,y_coords);
   end
end
save source_clam.txt detector_mat -ascii;
save detector_clam.txt detector_mat -ascii;
save clam.txt mtx -ascii;
S=sparse(clam);

% testing
f = fopen('Head1.img','rb','ieee-be');
img = fread(f,[512 512],'short');
fclose(f);
img = img(1:16:512,1:16:512);
img=img/max(img(:));
max(img(:));
min(img(:));
figure(1); subplot(2,2,1); imagesc(img); title('original image');
colormap(gray(256));
sino = S*img(:); sino=reshape(sino,Nsource,Ndetector);
save sino.txt sino -ascii;
```

```
figure(2);subplot(2,2,2); imagesc(sino); colormap(gray(256));
title('sinogram'); m=backproj(clam,sino);
m=reshape(m,32,32);
for l=1:10000,
   a=sino-reshape(S*m(:),Nsource,Ndetector);
   fprintf(1,'%i %f\n', l, max(abs(a(:))));
   z=S'*a(:);
   m=m+lambda*z;  %yk+1(n1,n2)=yk(n1,n2)+lambda*z(n1,n2)*h
   m = max(m,0);
   m = min(m,1);
   if l==1000,
      subplot(2,2,3); imagesc(m);title('restored image (iter=1000)');
   end
   if mod(l,10)==1, subplot(2,2,4);
      imagesc(reshape(m,32,32));title('restored image'); drawnow;
   end
end
```

Reconstruction algorithm for parallel plate geometry

```
%%%%%%%%%  to generate matrix and reconstruct image %%%%%%%%%%%
rows = 32;
cols = 32;
x_coords = ((-cols+1):2:(cols-1) ) / (2*cols);
y_coords = ((-rows+1):2:(rows-1) ) / (2*rows);
xmax=0.2; ymax=0.7;
xmin=-0.2; ymin=0.3;
n_rows=32;n_cols=32;
slength_min=1; slength_max=32;
dlength_min=1; dlength_max=32;
d=1; Nsource=32; Ndetector=32; lambda=0.05;

s_step=(slength_max-slength_min)/Nsource-1;
d_step=(dlength_max-dlength_min)/Ndetector-1;
mtx = zeros(Nsource*Ndetector,n_rows*n_cols);
slength=slength_min;
for iN=1:Nsource,
   x_source=-d/2;
   y_source=slength;
   source_mat(iN,:)= [x_source y_source];
   slength=slength+s_step;
end
save source_plate.txt source_mat -ascii;

dlength=dlength_min;
for iN=1:Ndetector,
   x_detector=d/2;
   y_detector=dlength;
   detector_mat(iN,:)= [x_detector y_detector];
   dlength=dlength+d_step;
end
save detector_plate.txt detector_mat -ascii;
```

```
for i=1:Nsource,
for j=1:Ndetector,
   h=pplate_line_32(source_mat(i,1),detector_mat(j,1),source_mat(i,2),
      detector_mat(j,2),xmin,xmax,ymin,ymax,n_cols,n_rows);
   mtx((i-1)*Ndetector+j,:) = h(:)';
end
end
save pplate.txt mtx -ascii;
S=sparse(pplate);

% testing
img32 = zeros(32,32); img32(8:24,8:24)=1;
img32=img32/max(img32(:));
max(img32(:));
min(img32(:));
figure(1); subplot(2,2,1); imagesc(img32); title('original image');
colormap(gray(256));
sino = S*img32(:); sino=reshape(sino,Nsource,Ndetector);
save sino.txt sino -ascii;
subplot(2,2,2);imagesc(sino); colormap(gray(256));title('sinogram');

pinv(pplate)*sino(:);
for l=1:60000,
   a=sino-reshape(S*m,Nsource,Ndetector);
   fprintf(1,'%i %f\n', l, max(abs(a(:))));
   z=S'*a(:);
   m=m+lambda*z;  %yk+1(n1,n2)=yk(n1,n2)+lambda*z(n1,n2)*h
   m = max(m,0);
   m = min(m,1);
   if l==1000,
      subplot(2,2,3); imagesc(reshape(m,32,32));
      title('restored image (iter=1000)');
   end
   if mod(l,10)==1,
      subplot(2,2,4); imagesc(reshape(m,32,32));
      title('restored image'); drawnow;
   end
end
```

To compute line integrals for parallel projections and clam-shell geometry.

```
function h=line_disp_32(x1,x2,y1,y2,x_coords,y_coords)
% to compute the coordinates of the intersection points
a=1/length(x_coords)/2;
b=1/length(y_coords)/2;
for ix=1:length(x_coords),
   xc=x_coords(ix);
   for iy=1:length(y_coords),
      yc=y_coords(iy);
      % translate so the pixel is centered at the origin
      xp=x1-xc;
      yp=y1-yc;
      xq=x2-xc;
```

```
      yq=y2-yc;
      % if the line is horizontal
      if abs(yp-yq) < 0.0001
         if abs(yp)<b
            iLength=2*b;
         elseif abs(yp)==b
            iLength=b;
         else
            iLength=0;
         end
      % if the line is vertical
      elseif abs(xp-xq) < 0.0001
         if abs(xp)<=a
            iLength=2*a;
         elseif abs(xp)==a
            iLength=a;
         else
            iLength=0;
         end
      else
         iSlope=(yq-yp)/(xq-xp);
         xp1=xp+((-b-yp)/iSlope);
         xq1=xp+((b-yp)/iSlope);
         yp1=yp+((-a-xp)*iSlope);
         yq1=yp+((a-xp)*iSlope);
         x_int=sort([-a,a,xp1,xq1]);
         y_int=sort([-b,b,yp1,yq1]);

         % line does not intersect the pixel
         if ( min(abs(xp1),abs(xq1))>a & min(abs(yp1),abs(yq1))>b )
            iLength=0;
         else
            iLength=sqrt((x_int(2)-x_int(3))^2+(y_int(2)-y_int(3))^2);
         end
      end
      h(ix,iy)=iLength;
   end
end
```

To compute line integrals for parallel projections and clam-shell geometry.

```
function h=pplate_line_16(x1,x2,y1,y2,xmin,xmax,ymin,ymax,
                          n_cols,n_rows)
% to compute the coordinates of the intersection points
yc0=0.5;
xc0=0.5;
a=((xmax-xmin)/n_cols);
b=((ymax-ymin)/n_rows);
for cols=1:n_cols,
   xc=xmin+(cols-1)*((xmax-xmin)/n_cols);
   xc=xc+xc0*a;
   for rows=1:n_rows,
      yc=ymin+(rows-1)*((ymax-ymin)/n_rows);
```

```
      yc=yc+yc0*b;
      % translate and rotate so the pixel is centered at the origin
      xp=x1-xc;
      yp=y1-yc;
      xq=x2-xc;
      yq=y2-yc;
      % if the line is horizontal
      if abs(yp-yq) < 0.0001
         if abs(yp)<b
            iLength=2*b;
         elseif abs(yp)==b
            iLength=b;
         else
            iLength=0;
         end
      % if the line is vertical
      elseif abs(xp-xq) < 0.0001
         if abs(xp)<=a
            iLength=2*a;
         elseif abs(xp)==a
            iLength=a;
         else
            iLength=0;
         end
      else
         iSlope=(yq-yp)/(xq-xp);
         xp1=xp+((-b-yp)/iSlope);
         xq1=xp+((b-yp)/iSlope);
         yp1=yp+((-a-xp)*iSlope);
         yq1=yp+((a-xp)*iSlope);
         x_int=sort([-a,a,xp1,xq1]);
         y_int=sort([-b,b,yp1,yq1]);
         % line does not intersect the pixel
         if ( min(abs(xp1),abs(xq1))>a & min(abs(yp1),abs(yq1))>b )
            iLength=0;
         else
            iLength=sqrt((x_int(2)-x_int(3))^2+(y_int(2)-y_int(3))^2);
         end
      end
      h(cols,rows)=iLength;
   end
end
```

Conjugate Gradient Method

```
% Martin King, ICTP, 24 Nov 2004
%1. Conjugate Gradient Method with Flecther-Reeves (or Polak-Ribiere)
% to find a vector x that gives a MINIMUM of a function (a scalar.)
%put initial guess here (x is an n-dimensional vector).
x=ones(32*32,1);
global mtx;
global sino;
[F] = func(x);
```

```
[F_prime] = dfunc(x);

r=-1.*F_prime;
d=r;

%g is for use in Polak-Ribiere
g=r;
delta_new=r'*r;
delta_0=delta_new;
fp = F;
%ftol is a convergence tolerance
ftol=1.e-7;
ITERMAX =20000;
%Don't worry too much about this
EPS=1.e-10;
N=32;

for iter = 1 : ITERMAX
   %Doing the line search here. First bracket a minimum, then
   % use Golden section to find it. Not using Newton-Raphson as in
   % Shewchuk. So you don't need the second derivative.
   [ax,bx,cx,fa,fb,fc] = func_mnbrak(0,1,x,d);
   [xt,golden] = func_golden(ax,bx,cx,x,d);
   %To recover vector x, which is along d at xt away from initial x.
   x = x + xt.*d;
   %The function value at x is golden as returned by func_golden.
   F = golden;
   [F_prime] = dfunc(x);
   r = -1.*F_prime;
   delta_old = delta_new;
   %This is Fletcher-Reeves
   delta_new = r'*r;
   %This is Polak-Ribiere
   % delta_new = (F_prime+g)'*F_prime;
   beta = delta_new/delta_old;
   d = r + beta * d ;
   g = r;
   if r'*d <= 0
      d=r;
   end
   %this convergence criterion is taken from NR.
   %     if 2.*abs(F-fp) < ftol.*(abs(F)+abs(fp)+EPS)
   fp = F;
   fprintf(1,'%i %f\n', iter, F);
end


%%%%%%%%%%%%%% to minimize a function %%%%%%%%%%%%%%%%
function [f_out] = func(m)
% %give your own function here. it is a scalar output

%MSE
global mtx;
global sino;
```

```
% I-divergence
if ((min(sino)<0) | (min(mtx*m)<0))
     f_out=inf;
else
     f_out=sum(sino(:).*log((sino(:)+eps)./(mtx*m+eps))-sino(:)+
           mtx*m));
end

%%%% %%% to find the first derivative of the matlab function
%%%%%%%%%%%
function [f_prime_out] = dfunc(m)

%for parallel projections
% global mtx;
% global sino;
% a = sino(:)-mtx*m;
% f_prime_out = -2*mtx'*a(:);

% %for clam-shell geometry
global mtx;
global sino;
a = sino(:)-mtx*m;
f_prime_out = -2*mtx'*a(:);


%%%%% to find the minimum of a function with minimum tolerance  %%%%%
function [xmin,golden] = func_golden(ax,bx,cx,x_in,d_in)

%1. Given bracket points ax, bx, cx, this function finds the minimum
%   return independent variable as xmin and function value as golden.
%   See NR F77, p.394

tol=1.e-7;
R=0.61803399;
C=1.-R;

x0=ax;
x3=cx;
if abs(cx-bx) > abs(bx-ax)
   x1=bx;
   x2=bx+C.*(cx-bx);
else
   x2=bx;
   x1=bx-C.*(bx-ax);
end
x1t=x_in+x1.*d_in;
[f1]=func(x1t);
x2t=x_in+x2.*d_in;
[f2]=func(x2t);

while abs(x3-x0) > tol.*(abs(x1)+abs(x2))
      if f2 < f1
          x0=x1;
```

```
            x1=x2;
            x2=R.*x1+C.*x3;
            f1=f2;
            x2t=x_in+x2.*d_in;
            [f2]=func(x2t);
        else
            x3=x2;
           x2=x1;
            x1=R.*x2+C.*x0;
            f2=f1;
            x1t=x_in+x1.*d_in;
            [f1]=func(x1t);
        end
end

if f1 < f2
   golden=f1;
   xmin=x1;
else
   golden=f2;
   xmin=x2;
end


%%%%%%% to bracket  minimum a  function from it s initial guess
%%%%%%%%%
function [ax,bx,cx,fa,fb,fc] = func_mnbrak(ax,bx,x_in,d_in)

GOLD=1.618034;
GLIMIT=100;
TINY=1.e-20;

axt=x_in+ax.*d_in;
[fa]=func(axt);
bxt=x_in+bx.*d_in;
[fb]=func(bxt);

if fb > fa
   dum=ax;
   ax=bx;
   bx=dum;
   dum=fb;
   fb=fa;
   fa=dum;
end

%first guess for c
cx=bx+GOLD.*(bx-ax);
cxt=x_in+cx.*d_in;
[fc]=func(cxt);

while fb >= fc
r=(bx-ax).*(fb-fc);
      q=(bx-cx).*(fb-fa);
```

```
      u=bx-((bx-cx).*q-(bx-ax).*r)./
        (2.*abs(max(abs(q-r),TINY)).*sign(q-r));
      ulim=bx+GLIMIT.*(cx-bx);
    if (bx-u).*(u-cx) > 0.
          ut=x_in+u.*d_in;
          [fu]=func(ut);
          if fu < fc
              ax=bx;
              fa=fb;
              bx=u;
              fb=fu;
              break;
          elseif fu > fb
              cx=u;
              fc=fu;
              break;
          end
          u=cx+GOLD.*(cx-bx);
          ut=x_in+u.*d_in;
          [fu]=func(ut);
      elseif (cx-u).*(u-ulim) > 0.
          ut=x_in+u.*d_in ;
          [fu]=func(ut);
          if fu < fc
             bx=cx;
             cx=u;
             u=cx+GOLD.*(cx-bx);
             fb=fc;
             fc=fu;
             ut=x_in+u.*d_in;
             [fu]=func(ut);
         end
     elseif (u-ulim).*(ulim-cx) >= 0
            u=ulim;
            ut=x_in+u.*d_in;
            [fu]=func(ut);
     else
           u=cx+GOLD.*(cx-bx);
           ut=x_in+u.*d_in;
           [fu]=func(ut);
     end
     ax=bx;
     bx=cx;
     cx=u;
     fa=fb;
     fb=fc;
     fc=fu;
end
```

Algorithm to reconstruct images in the presence of high density object when location of object is given

```
load sino_thres.txt;
```

```
load h.txt;
sino1=sino;
mtx1=h;;
N=32;
lambda=0.5;
S=sparse(h);

% testing
img32=phantom('Modified Shepp-Logan',32);
mask=(img32>0); % setting the background values equal to 0
figure(1); subplot(2,2,1); imagesc(min(img32,1));
title('original image');xlabel('rows');ylabel('cols');
colormap(gray(256));

%make sinogram value greater than threshold make it equal to some large
value
sino=S*img32(:); sino =reshape(sino,49,129);
for i=1:49,
    for j=1:129,
        if sino(i,j)>=0.25,
            sino(i,j)=1.25;
        end
     end
end
save sino_thres.txt sino -ascii;
%figure(1);subplot(2,2,2);imagesc(sino);
colormap(gray(256));title('sinogram');
xlabel('detectors');ylabel('sources');


% make all entries in sinogram and matrix equal to zero if value
greater than 'large'
for i=1:49,
    for j=1:129,
        if sino_thres(i,j)==1.25,
            sino1(i,j)=0;
        end
     end
end
for i=1:length(sino_thres(:)),
     if sino_thres(i)==1.25,
          mtx1(i,:)=0;
     end
end
save sino1.txt sino1 -ascii;
%figure(1);subplot(2,2,3);imagesc(sino1);
colormap(gray(256));title('sinogram with
zeros');xlabel('detectors');ylabel('sources');
S=sparse(mtx1);

% Reconstruction procedure
m=backproj(S,sino1);
for l=1:60000,
     a=sino1-reshape(S*m(:),49,129);
```

```
    fprintf(1,'%i %f\n', l, norm(abs(a(:))));
    m=reshape(m,32,32)+lambda*z;
    %yk+1(n1,n2)=yk(n1,n2)+lambda*z(n1,n2)*h

    % location of object is specified
    if l>=100,
        m=m.*mask;
        for i=1:32,
            for j=1:32,
                if img32(i,j)>3;
                    m(i,j)=6;
                end
            end
        end
    end

    if mod(l,10)==1,
        figure(1); imagesc(min(m,1)); colormap(gray(256));
        title('restored image'); xlabel('rows');ylabel('cols');
        drawnow;
    end
end
```

Algorithm to reconstruct images in the presence of high density object when prior information not provided

```
N=32;
rows = 32;
cols = 32;
lambda=0.5;
load h.txt;
mtx1=h;
S=sparse(h);

% testing
img32=phantom('Modified Shepp-Logan',32);% img32=img32/max(img32(:));
% mask1=(img32>0);
% figure(1); subplot(1,2,1); imagesc(min(img32,1));
% title('original image');xlabel('rows');ylabel('cols');
% colormap(gray(256));

sino=S*img32(:); sino =reshape(sino,49,129);
for i=1:49,
   for j=1:129,
      if sino(i,j)>=0.25,
         sino(i,j)=1.25;
      end
   end
end

save sino_thres.txt sino -ascii;
figure(1);subplot(2,2,2);imagesc(sino);
colormap(gray(256));title('sinogram');
```

```
xlabel('detectors');ylabel('sources');

% make all entries in sinogram and matrix equal to zero if value
greater than 'large'
sino1=sino;
mtx1=mtx;
for i=1:49,
    for j=1:129,
          if sino(i,j)==1.25,
              sino1(i,j)=0;
          end
     end
end
for i=1:length(sino(:)),
     if sino(i)==1.25,
          mtx1(i,:)=0;
     end
end
save sino1.txt sino1 -ascii;

%figure(1);subplot(2,2,3);imagesc(sino1); colormap(gray(256));
title('sinogram with zeros');xlabel('detectors');ylabel('sources');

% metal mask
mask=zeros(size(img32));
for i=1:length(sino(:)),
     if sino(i)>=1.25,
          idx=find(mtx1(i,:)>0);
          mask(idx)=mask(idx)+1;
     end
end
for i=1:32,
    for j=1:32,
          if mask(i,j)<138,
              mask(i,j)=0;
          else
              mask(i,j)=1;
          end
     end
end

% large mask
mask1=zeros(size(img32));
for i=1:length(sino(:)),
     if sino(i)>0,
          idx=find(mtx1(i,:)>0);
          mask1(idx)=mask1(idx)+1;
     end
end
for i=1:32,
    for j=1:32,
          if mask1(i,j)<147,
              mask1(i,j)=0;
          else
```

```
                mask1(i,j)=1;
            end
    end
end

figure(2); imagesc(mask1); colormap(gray(256));

S=sparse(mtx1);
m=backproj(S,sino1);
for l=1:600,
     a=sino1-reshape(S*m(:),49,129);
     fprintf(1,'%i %f\n', l, norm(abs(a(:))));
     z=reshape(S'*a(:),N,N);
     m=reshape(m,32,32)+lambda*z;
     %yk+1(n1,n2)=yk(n1,n2)+lambda*z(n1,n2)*h
     for i=1:32,
          for j=1:32,
               if mask(i,j)==1;
                    m(i,j)=6;
               end
          end
     end
     m=m.*mask1;

     if mod(l,10)==1,
        figure(1);imagesc(min(m,1));colormap(gray(256));
        title('restored image'); xlabel('rows');ylabel('cols');
        colormap(gray(256));drawnow;
     end
end
```

VITA

Abey Mukkananchery was born in April 1979 in city of Bombay, also known as Mumbai, a metropolitan city in India. He graduated from the Mumbai University in 2001 with Bachelor of Engineering in Biomedical with honors. He came to United States of America and joined the graduate program of Biomedical Engineering at Virginia Commonwealth University in 2002.