GRAPHICAL USER INTERFACE (GUI) TO STUDY DIFFERENT RECONSTRUCTION ALGORITHMS IN COMPUTED TOMOGRAPHY

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering

By

SHITAL ABHANGE
B.E., D. J. Sanghvi College of Engineering, 2005

2009
Wright State University
I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Shital Abhange ENTITLED Graphical User Interface (GUI) to Study Different Reconstruction Algorithms in Computed Tomography BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Engineering

______________________________
Thomas N. Hangartner
Thesis Director

______________________________
S. Narayanan, Ph.D.
Department Chair

Committee on Final Examination

______________________________
Thomas N. Hangartner, Ph.D.

______________________________
Julie A. Skipper, Ph.D.

______________________________
David F. Short, M.S.

______________________________
Joseph F. Thomas, Jr., Ph.D.
Dean, School of Graduate Studies
Abstract

Abhange Shital, M.S.Egr., Department of Biomedical, Industrial and Human Factors Engineering, Wright State University, Dayton, OH, 2009. Graphical User Interface (GUI) to Study Different Reconstruction Algorithms in Computed Tomography.

Computed tomography (CT) imaging relies on computational algorithms to reconstruct images from projections gathered from the CT scan. Depending on the scanner geometry, different types of reconstruction algorithms can be used. To study these different types of reconstruction algorithms in a user-friendly way, a software tool was built.

The aim of the thesis was to provide a software platform to access a number of previously implemented reconstruction algorithms with ease and minimal knowledge of the reconstruction code. The goal was accomplished by building a Graphical User Interface (GUI) using MATLAB 7.7.0 (R2008b). In addition to creating mathematical objects and invoking the various reconstruction algorithms, the tool provides newly developed features to analyze the reconstructed images.

This thesis first presents an overview of CT and associated reconstruction algorithms. It then describes the features to simulate two-dimensional as well as three-dimensional objects. The reconstructions available are categorized on the basis of different scanner geometries. The tool has the flexibility to specify a range of parameter
values for the reconstruction. Finally the tool allows qualitative and quantitative analysis of reconstructed images by using the analysis tool.

A couple of test phantoms were simulated to demonstrate the capabilities of the GUI tool. The tests performed included the mask analysis to study the relationship between the standard deviation of reconstructed values and the relevant reconstruction parameters, image subtraction to demonstrate differences in reconstructed values, line profile analysis to show variation of reconstructed image values in more detail, and lastly qualitative image display to visualize reconstruction artifacts using the available reconstruction algorithms.

The implemented GUI tool, thus, allows the user to study different reconstruction algorithms with ease using a single panel. It also systematically arranges the available reconstruction algorithms under each scanner geometry. Overall, the tool allows the user to study various objects and reconstruction algorithms by varying different input parameters.
# Table of Contents

1. **Introduction** .......................................................................................................................................................... 1

2. **Background** ........................................................................................................................................................ 3

   2.1 Principles of Image Reconstruction in CT ........................................................................................................ 3

      2.1.1 Scanner Generations and Reconstruction Algorithms ........................................................................... 5

         2.1.1.1 Parallel Beam Geometry .................................................................................................................. 6

         2.1.1.2 Fan Beam Geometry ....................................................................................................................... 6

         2.1.1.3 Cone Beam Geometry .................................................................................................................... 16

         2.1.1.4 Spiral Beam Geometry ................................................................................................................... 17

   2.1.2 Factors Affecting the Reconstructed Images Qualitatively and Quantitatively ........................................... 19

3. **Generation of Object and Projection** ................................................................................................................ 21

   3.1 GUI (Graphical User Interface) ......................................................................................................................... 21

      3.1.1 Object Generation ........................................................................................................................................ 23

      3.1.2 Scanner Geometry ....................................................................................................................................... 26

      3.1.3 Calculation of Projections .......................................................................................................................... 27

      3.1.4 Type of Reconstruction Algorithm ............................................................................................................ 31

4. **Analysis of Reconstructed Data** ....................................................................................................................... 35

   4.1 Description of Analysis Tool ............................................................................................................................... 35

   4.2 Sample Analysis .................................................................................................................................................... 39

      4.2.1 Mathematical Test Phantoms ...................................................................................................................... 39

      4.2.2 Analysis Results ........................................................................................................................................... 40

5. **Discussion and Conclusion** ............................................................................................................................... 53
Appendix A ........................................................................................................................................59
References .......................................................................................................................................66
List of Figures

Figure 2.1: The projection $P_\theta(t)$ is given by the line integral along line $(t, \theta)$ ............4

Figure 2.2: The Fourier slice theorem relates the one-dimensional Fourier transform of a projection to the two-dimensional Fourier transform of an object cross-section along a radial line$^3$ ...............................................................5

Figure 2.3: The figure on the left is back-projected image of circular object without a filter showing the star artifact. The figure on the right shows the object image after use of a ramp filter to modify the projections before back-projecting .................................................................5

Figure 2.4: Parallel-beam geometry in the case of a first generation scanner ...............7

Figure 2.5: Flow chart showing steps for filtered back projection for parallel-beam case7

Figure 2.6: Fan-beam geometry in the case of a second generation scanner .............8

Figure 2.7: Fan-beam geometry in case of third generation scanner$^7$ .........................9

Figure 2.8: Two types of fan-beam geometry (a) equiangular fan-beam with detector arranged along an arc; (b) equidistant fan-beam with detectors arranged along a straight line$^3$ ..............................................................................10

Figure 2.9: Two fan-beams with source angles $\beta_1, \beta_2$ and fan angles $\gamma_1, \gamma_2$ respectively ...........................................................................................................11

Figure 2.10: Location of projections collected over $180^\circ$ is shown on the left in the Radon space (t, \theta coordinate system). The right side shows the projection map over
180º in the (β, γ)-coordinate system. The regions labeled ‘A’ represent redundant data, and the regions labeled ‘B’ represent missing data (adapted from Kak & Slaney\textsuperscript{3})........................................................................................................................................11

Figure 2.11: Location of projections collected over 180º+2γ\textsubscript{m} are shown on the left in the Radon space. The right side shows the projection map over 180º+2γ\textsubscript{m} in the (β, γ) coordinate system. The shaded regions represent redundant data (adapted from Kak & Slaney\textsuperscript{3}) ...........................................................................12

Figure 2.12: Scanning segments λ\textsubscript{1} [P\textsubscript{s},P\textsubscript{a}], λ\textsubscript{2} [P\textsubscript{a},P\textsubscript{b}], and λ\textsubscript{3} [P\textsubscript{b},P\textsubscript{e}] for a point object x (adapted from Chen et al.\textsuperscript{8}) ...........................................................................13

Figure 2.13: Plots of local, global, and smooth short-scan weighting scheme of the central ray for all the view angles .........................................................................................14

Figure 2.14: Flow chart showing steps followed in FBP for fan-beam case. In the equiangular detector geometry, the cosine corrections results in a constant 15

Figure 2.15: Flow chart showing steps followed in DFBP for fan-beam case..............16

Figure 2.16: Cone-beam geometry using x-ray point source ......................................17

Figure 2.17: Translation motion of patient table for helical scanning mode\textsuperscript{14} .............18

Figure 2.18: (a) Single-slice CT scanner setup, (b) multi-slice CT scanner setup\textsuperscript{13} ........19

Figure 3.1: The main GUI window layout showing all components to access different reconstruction algorithms........................................................................................................23

Figure 3.2: The Circle / Ellipse window layout showing five object-specific parameter edit boxes for the 2D object case ..................................................................................................25

Figure 3.3: Layout of Sphere / Ellipsoid / Cylinder window showing the object selection options for the 3D-case ..................................................................................................................26
Figure 3.4: GUI figure shows five object-specific parameter edit boxes for 3D spherical object case

Figure 3.5: The dashed line $s(r, \beta, \gamma)$ represents the integration path for the projection point $P_{\beta, \gamma}$ for the fan bean case

Figure 3.6: Intersection between the projection line and the boundary of the ellipse; $d$ is the distance between the two intersection points $(x_1, y_1)$ and $(x_2, y_2)$, respectively

Figure 3.7: Geometry of cone-beam parameters. Vector $\vec{r}$ denotes the source position and is normal to the detector plane D.P. The source rotates in the x-y plane, by incrementing angle $\beta$. The location of the integration path $s$ is defined by angles $\gamma$ and $\delta$

Figure 3.8: Task bar showing the progress of reconstruction algorithm

Figure 4.1: Window showing object data and analysis panel for a selected two-dimensional object (a) and a three-dimensional object (b). The lower window shows the reconstructed object in three planes, x-y, x-z, and y-z, respectively. Both (a) and (b) show the types of analysis options available to the user

Figure 4.2: Schematic representation of the simulated two-dimensional phantom. The density values represent the material composition of the cortical and trabecular regions of the bone

Figure 4.3: (a) Schematic representation of the simulated three-dimensional cylindrical phantom. The table specifies the geometry and densities of the phantom
components. (b) Two-dimensional slices of the three-dimensional phantom shown in part (a) in the x-y, x-z, and y-z planes.................................40

Figure 4.4: Variation of standard deviation in reconstructed values with number of source positions and detectors for cortical bone region and trabecular bone region. The data are simulated for a two-dimensional phantom (a) and (b) and for a three-dimensional phantom (c) and (d). The legends along the graphs denote the number of detectors used.................................42

Figure 4.5: The green line passing through the centre of the 2D (a) and 3D (b) selected image was used for the line-profile analysis. Due to computer storage limitations, the 3D image was reconstructed only for a limited number of slices along the z-axis. The slice at z = 64 is shown in (b). The line profiles obtained are also displayed .................................................................43

Figure 4.6: Line profiles through the centre of the two-dimensional phantom showing variations in reconstructed image values. The plots were taken for 64-1024 detectors and a fixed number of 250 source positions..............................44

Figure 4.7: These plots represent line profiles through the central slice of the three-dimensional phantom, showing variations in reconstruction noise. The plots were taken for a range of detectors varying from 64-512 and fixed number of 250 source positions .................................................................45

Figure 4.8: The difference images for original object image subtracted from a reconstructed image are shown for both a 2D (a) as well as 3D (b) phantom. In both cases, the left image is the reconstructed image and right image is the original object image.................................................................47
Figure 4.9: Parallel-beam reconstruction algorithm. 128 projections were collected over $\pi$ radians with 64 samples per projection. The Shepp and Logan filter$^4$ was used, and the data were back-projected on a 256x256 matrix. A smoothing effect at the inner edges is visible due to under sampling of projections. The image values above the upper display limit are color coded equivalent to the upper display limit, and the values below the lower display limit are color coded equivalent to the lower display limit ..................................................48

Figure 4.10: Equidistant fan-beam filtered back-projection algorithm. 256 projections were collected over $2\pi$ radians with 64 numbers of samples per projection. The ramp filter$^5$ was used, and the data were back-projected on a 256x256 matrix. The Gibbs phenomenon due to under sampling of projections is visible as dark red rings at both the inner and outer boundary of cylinder...48

Figure 4.11: Equidistant fan-beam Differentiated Filtered back-projection algorithm. 256 projections were collected over $\pi$ radians with 200 numbers of samples per projection. Local and global weighting schemes were used (left and right), respectively, and the data were back-projected on a 256x256 matrix. The local only through $\pi$ radians and zero for the rest. In contrast the global weighting scheme uses a smooth filter weighting filter, and data beyond $\pi$ radians can be seen with blurring..........................................................49

Figure 4.12: Equiangular fan-beam differentiated filtered back-projection. The reconstructed image (a) is obtained using local weighting scheme, (b) is obtained using global weighting scheme, both for short-scan angular interval $[0, \pi+\text{fan angle}]$ (fan angle = 0.6769 radians). 256 projections with
200 samples per projection were used. The local weighted image shows fish-scale artifacts, whereas the global weighted image shows blurring due to a smooth weighting. To see the artifact, a display window size of [0.3 0.31] was used. Display characteristics can be adjusted using the contrast/brightness adjustment button.

Figure 4.13: 3D cone-beam filtered back-projection. 200 projections were collected over $2\pi$ with 200x128 2D detector array (128 is along z-axis). The image was back-projected on 128x128x128 matrix. Due to computer limitations, the image was back-projected for a slice range $z = 50$ to $z = 80$. (a) The reconstructed image at $z = 64$ shows a Moiré pattern due to under sampling. The image window used was [0.28 0.3] to display the artifact. (b) shows a slice at $x = 55$, showing high intensity streaks.

Figure 4.14: Cone-beam reconstructed image using the 3D local weighting scheme. 200 projections collected over $\pi$ radians with 200 samples per projection. Due to the local weighting scheme data are reconstructed over $\pi$ and missing in other scanning regions. The image on left is at $z = 64$ and on right is at $y = 64$. The image is back-projected on a 128x128x128 matrix.

Figure 4.15: Cone-beam reconstructed image shown at $y = 64$ on left and at $z = 64$ on right using the 3D global weighting scheme for a scan angular interval $[0, \pi]$. The image shows smooth weighted data due to the smooth weighting function in both x-y and x-z planes. The parameters used for reconstruction are the same as in Figure 4.14.
Figure 4.16: 360 LI spiral-beam reconstruction showing slices at z = 64, y = 64 and x = 64. (a) Single-slice spiral-beam 360 LI algorithm smoothing due to slice interpolation can be seen at y = 64 and x = 64, respectively. (b) Multi-slice spiral-beam 360 LI algorithm also shows smoothing due to slice interpolation at the same slice locations as in the single-slice case but with additional edge effects at the left and right boundaries of the cylinder. In both cases 200 projections were collected over $2\pi$ radians with 200x128 2D detector array (128 is along z-axis). The images were back projected on a 128x128x128 matrix. The number of spirals used to transverse the entire object was 128. In both cases a pitch of 1 was used.

Figure 5.1: Flow chart showing steps to calculate projections for change in orientation

Figure 5.2: Flow chart showing steps to add noise to projection data

Figure 5.3: Flow chart showing steps to calculate projection data for energy dependent beam spectrum

Figure 5.4: Flow chart showing steps to calculate projection data with finite beam width

Figure 5.5: Steps to calculate dose
List of Tables

Table 3.1: Projection parameters for implemented scanner geometries……………………27
Table 5.1: Overview of the created GUI tool...............................................................53
Acknowledgement

I would like to take this opportunity to thank all those who have contributed to this thesis, directly or indirectly.

First and foremost I would like to express my gratitude towards my advisor Dr. Thomas N. Hangartner, who gave me the opportunity to work on this thesis. His constant guidance, stimulating suggestions and encouragement have helped me throughout the span of my research.

I would like to thank my committee members Dr. Julie Skipper and David Short for their constant support and kind cooperation.

I am thankful to Sangeetha Alladi for advising me in the earlier stages of this project and for sharing the reconstruction code with me. Without her support I couldn’t have achieved this milestone in my life. I would like to express my gratitude to all staff and students who are part of the BioMedical Imaging Laboratory for their continued motivation and friendly guidance to complete the research work. I would also like to acknowledge Mihirkumar Naik and Aarti Raheja for their help in solving problems related to Graphical User Interface. I would like to thank my friends Darshan, Sakina, Kutbi, Jasmin, Vinit, Himanshu, Pratik, Ruchit, Sachi, and Rikki for lending me a helping hand whenever needed. Special thanks to the students who evaluated the tool and provided suggestions for improvement.
This thesis would have been incomplete without the support of my parents, Krishnaji Abhange and Surekha Abhange, and my sisters Minal and Komal. I would like to thank them for all their love and support, which turned any fears of failure into desires to succeed.
Dedicated

To my grandfather, for all his love and sacrifice.
1. Introduction

Computed Tomography (CT) is one of the most frequently used modern diagnostic imaging modalities. CT is based on the principle of reconstruction from projections, which creates a tomographic image representing the distribution of x-ray attenuation coefficients. The computed tomography images are obtained by using different reconstruction algorithms. These reconstruction algorithms vary for different scanner generations. Each scanner generation further uses different types of reconstruction algorithms. Hence it becomes a challenge to study the different types of reconstruction algorithms available.

The main aim of this thesis was to provide a tool to study the different types of reconstruction algorithms with ease and minimal knowledge of the computer code. To achieve this goal, a GUI (Graphical User Interface) tool was implemented using MATLAB 7.7.0 (R2008b). The tool allows the user to create phantom simulations for two-dimensional as well as for three-dimensional objects and to test the available reconstruction algorithms. The reconstructed images can also be analyzed using the implemented analysis tool option. The outline of this thesis is as follows:

Chapter 2 starts with the CT background, covering the basic principles of CT, followed by an explanation of the different scanner generations. The scanner generations considered are parallel, fan, cone, and spiral beam. The chapter further explains the different types of reconstruction algorithms used in each case.
Chapter 3 explains the created GUI program in detail. This chapter begins with a brief introduction about the GUI as a software tool, followed by the user options available to create objects, to calculate projections for the created objects, and to reconstruct them to obtain a cross-sectional image. The objects created by the user are limited to circular and elliptical shapes in the two-dimensional case and to ellipsoidal, spherical, and cylindrical shapes in the three-dimensional case.

Chapter 4 deals with a brief explanation of the analysis tool. It consists of the mask analysis, image subtraction, and line-profile analysis. This chapter further introduces some test objects and describes the sample analyses conducted to test the tool. The last part of this chapter draws conclusions and gives suggestions for future work.
2. Background

The evolution of computed tomography (CT) has made non-invasive medical imaging of internal body organs more accurate and easy to perform. In 1917 Johann Radon suggested the mathematical solution for image reconstruction,\(^1\) which further gave rise to the development of new reconstruction algorithms. In 1972 Godfrey Hounsfield patented the first CT scanner, and he was awarded a Nobel Prize together with Allan Cormack for this invention in 1979.\(^1\) Since then, new developments have led to faster scanning methods with lower dose and better image quality. The image reconstruction algorithms played a vital role in the success of CT.

2.1 Principles of Image Reconstruction in CT

CT is a method for acquiring and reconstructing an image of a cross-section of an object. Consider an object to be imaged using a single x-ray source and a single detector. The x-ray photons emitted by the source undergo attenuation depending on shape, size, and material of the object to be imaged, and they are measured by the detector placed opposite the source. This forms the basic CT principle.\(^2\) The measurement of the linear attenuation coefficients or densities of the object along the x-ray beam passing through it at a particular angle \(\theta\) is termed a projection point (Figure 2.1). The collection of projection points under a given angle \(\theta\) is considered a projection. Projections collected for angular intervals from 0\(^\circ\) to 180\(^\circ\) provide the information necessary to reconstruct the image cross-section.
The mathematical formula used to calculate a projection is given as:

\[ P_\theta (t) = \int_{s(t,\theta)} f(x, y) ds, \]  

(1)

where \( f(x,y) \) = function used to describe density of object,

\( t \) = location of the projection point, and

\( \theta \) = angle of rotation.

The key to CT imaging is the Fourier-slice theorem, which relates the measured projection data to the two-dimensional Fourier transform of the object cross-section. The Fourier-slice theorem states that the Fourier transform of a parallel projection of an image \( f(x,y) \) taken at angle \( \theta \) represents a central slice of the two-dimensional transform \( F(u,v) \) subtending an angle \( \theta \) with the \( u \)-axis (Figure 2.2).

The measured projections are back-projected to obtain an image. To remove star artifacts in the reconstructed image due to over estimation of density in certain pixels of the matrix, projections are convolved with a filter before back-projection (Figure 2.3). The commonly used convolution filters are those by Shepp & Logan⁴ and G. H.
Figure 2.2: The Fourier slice theorem relates the one-dimensional Fourier transform of a projection to the two-dimensional Fourier transform of an object cross-section along a radial line.\textsuperscript{5}

Ramachandran & A.V. Lakshminarayanan\textsuperscript{5} (ramp filter). Back projection with a convolution filter, known as filtered back projection (FBP), is widely applied in CT reconstructions.

![Figure 2.3](image)

Figure 2.3: The figure on the left is the back-projected image of a circular object without a filter showing the star artifact. The figure on the right shows the object image after use of a ramp filter to modify the projections before back-projecting.

2.1.1 Scanner Generations and Reconstruction Algorithms

Different types of CT geometries have been developed to acquire the x-ray transmission data for image reconstruction. These geometries, defined as the projection sampling geometries, are useful in differentiating scanner designs. The following sections
explain the development of different types of beam geometries used in the various scanner generations, along with the reconstruction algorithms associated with them.

2.1.1.1 Parallel-Beam Geometry

The EMI scanner was the first commercial CT scanner invented by Hounsfield in 1973. This scanner acquired data with a narrow collimated x-ray beam directed to a single detector across the patient or object to obtain the parallel projections for a given projection angle \( \theta \) (Figure 2.4). This sampling geometry is known as parallel-beam geometry. The parallel-beam projection data \( P_\theta (t) \) for a given projection angle \( \theta \) are collected while the assembly is translated along a straight line \((t, \theta)\). The next projection is collected after incrementing the angle \( \theta \) by a small amount. Hence, this data collection scheme is also known as translate-rotate geometry. A single scan, collecting the projections necessary for reconstruction, took about four to five minutes.\(^6\) The long scanning time restricted the use of this scanner to regions of the patient that could be immobilized (for example, the head region) to avoid serious image quality issues associated with patient motion.

The reconstruction of the object from measured projection data requires the data to be collected over an interval of \( \theta \) varying from 0° to 180°. In the case of parallel-beam geometry, projections measured from 180° to 360° are mirroring those measured from 0° to 180°. The reconstruction algorithm used to obtain images in parallel-beam geometry is mainly the filtered back projection (FBP). The steps followed in filtered back projection are represented in the flow chart below (Figure 2.5).

2.1.1.2 Fan-Beam Geometry

Demand for reduction in scanning time led to the invention of the second-
Early second generation scanners consisted of an x-ray source, which emits radiation spread over an angle or fan, on one side of the object and a bank of detectors on the other side\(^2\) (Figure 2.6). The detector bank in place of a single detector allows measurement of several projections simultaneously and, thus, reduces the scanning time. The beam geometry is known as fan-beam. The scanner system used is the translate-rotate type. Scanning time in a second generation scanner system with 30 detectors is less than 20 seconds.\(^6\)
With improvement in detector and data acquisition technology, it was possible to design a detector array with sufficiently small detector cells and a large enough detector array to cover the entire patient cross-section. With such a large detector array, it is no longer necessary for the detector-tube assembly to translate past the object. Instead, the detector-tube assembly simply rotates around the object. The scanner consists of an x-ray tube collimated to a wide, fan-shaped x-ray beam and a bank of detectors on the opposite end (Figure 2.7). The beam geometry is called fan-beam and the data collection geometry is rotate only. The scan time is reduced to 0.5 seconds or less.\textsuperscript{6}

The fan-beam geometry is further divided into two sub-categories, namely equiangular and equidistant fan-beam geometry, depending on the arrangement of the detectors. In the equiangular fan-beam, the detectors are arranged on the arc of a circle and are spaced equally along the arc (Figure 2.8a). In the case of an equidistant fan-beam, the detectors are arranged along a straight line, again with equal spacing between them (Figure 2.8b).
As mentioned in the parallel-beam section, parallel-beam projections are collected over 180° to avoid redundant data. In the case of a third-generation fan-beam scanner, if projections are collected over 180°, then some regions of the object will not be measured.

To explain this condition, consider two ray integrals represented by fan-beam angles \((\beta_1, \gamma_1)\) and \((\beta_2, \gamma_2)\) respectively (Figure 2.9). The left plot in Figure 2.10 shows the location of projections collected over 180° for the source position between the curved lines, which represent the starting and ending projections of fan angle \(\pm \gamma_m\). The right side shows the stacked data in the \((\beta, \gamma)\) coordinate system. These two rays are identical only if

\[
\beta_1 - \gamma_1 = \beta_2 - \gamma_2 - 180° \tag{2}
\]

and

\[
\gamma_1 = -\gamma_2
\]

To map these rays in the \((\beta, \gamma)\) coordinate system, the following transformation is used:

\[
t = D \sin \gamma; \quad \theta = \beta + \gamma \tag{3}
\]

where \(D =\) isocentre-to-source distance.
Figure 2.8: Two types of fan-beam geometry: (a) equiangular fan-beam with detectors arranged along an arc; (b) equidistant fan-beam with detectors arranged along a straight line.

The regions labeled ‘A’ represent duplicated data for the same line integral, whereas the regions labeled ‘B’ represent the areas in Radon space, for which no data are collected. To collect the missing projection data, it is necessary to measure projections with source positions over $180^\circ + \text{fan angle}$ (i.e., $180^\circ + 2\gamma_m$).
Figure 2.9: Two fan-beams with source angles $\beta_1$, $\beta_2$ and fan angles $\gamma_1$, $\gamma_2$ respectively.

The data collected over 180°+fan angle are shown in Figure 2.11. Here, the shaded regions represent redundant data, and no data are missing.

Figure 2.10: Location of projections collected over 180° is shown on the left in the Radon space ($t$, $\theta$ coordinate system). The right side shows the projection map over 180° in the ($\beta$, $\gamma$)-coordinate system. The regions labeled ‘A’ represent redundant data, and the regions labeled ‘B’ represent missing data (adapted from Kak & Slaney$^3$).
Figure 2.11: Location of projections collected over $180^\circ+2\gamma_m$ are shown on the left in the Radon space. The right side shows the projection map over $180^\circ+2\gamma_m$ in the $(\beta, \gamma)$ coordinate system. The shaded regions represent redundant data (adapted from Kak & Slaney\textsuperscript{3}).

For improved reconstruction, redundant data is weighted using different weighting schemes.

1. Local weighting scheme: This scheme was introduced by Guang-Hong Chen.\textsuperscript{8} It is also called the equal weighting scheme. To explain this scheme, let us consider a simple diagram representing three segments, namely $\lambda_1$, $\lambda_2$, and $\lambda_3$. These three segments have end points $P_s$, $P_a$, $P_b$, and $P_e$ (Figure 2.12). A circular trajectory of the x-ray source was considered with radius $R$ around the point object $x$. The point object $x$ is connected to the end-points of the trajectory through ray $(P_s, P_b)$ and ray $(P_e, P_a)$. Thus, point object $x$ has three scanning paths $\lambda_1[P_s, P_a]$, $\lambda_2[P_a, P_b]$, and $\lambda_3[P_b, P_e]$. Any ray starting from segment $\lambda_1$ and passing through point object

12
x will have a redundant ray or twice the registration of the same data in region \( \lambda_3 \), whereas a ray starting from the \( \lambda_2 \) region and passing through point object x has no redundant ray. The weighting function used for such redundant data is given as follows:

\[
w(x, \lambda) = \begin{cases} 
0.5 & \lambda \in [P_s, P_a] \\
1 & \lambda \in [P_a, P_b] \\
0.5 & \lambda \in [P_b, P_e] 
\end{cases}
\] (4)

The sharp transition at the scanning segment boundaries may cause problems if there are slight inconsistencies of the data scanned 180° apart due to potential patient movement. The other two weighting schemes discussed below attempt to alleviate this problem.

2. Global weighting scheme: Frederic Noo implemented a smooth weighting function for redundant data.\(^9\) \( c(\lambda) \) is a smooth weighting function given as follows:

\[
c(\lambda) = \begin{cases} 
\cos^2 \frac{\pi(\lambda - P_s - d)}{2d} & \text{if } P_s < \lambda < P_s + d \\
1 & \text{if } P_s + d < \lambda < P_e - d \\
\cos^2 \frac{\pi(\lambda - P_e - d)}{2d} & \text{if } P_e - d < \lambda < P_s 
\end{cases}
\] (5)
where \( P_s \) = Starting angle of scan,
\( P_e \) = Ending angle of scan, and
\( d \) = Angular interval for rotation.

3. Smooth short-scan weighting scheme: Dennis L. Parker stated that by using a proper weighting scheme, image quality equivalent to the quality of a reconstruction from a 360° data set can be obtained. The weighting function is given below.

\[
w(\alpha, \beta) = \begin{cases} 
\sin^2\left(\frac{\pi\beta}{4(\delta-\alpha)}\right) & 0 < \beta < 2\delta - 2\alpha \\
1 & 2\delta - 2\alpha < \beta < \pi - 2\alpha \\
\sin^2\left(\frac{\pi(\pi+2\delta-\beta)}{4\alpha}\right) & \pi - 2\alpha < \beta < \pi + 2\delta 
\end{cases}
\]  

(6)

where \( \beta \) = Angle of projection,
\( 2\delta \) = Fan angle, and
\( \alpha \) = Angle within projection.

The weighting schemes are illustrated in Figure 2.13.

Figure 2.13: Plots of local, global, and smooth short-scan weighting scheme of the central ray for all the view angles.
Fan-beam projections are reconstructed using two types of reconstruction algorithms: filtered back projection (FBP) and differentiated filtered back projection (DFBP). The steps followed in the algorithms are represented in the flow chart below (Figure 2.14).

![Flow chart showing steps followed in FBP for fan-beam case. In the equiangular detector geometry, the cosine correction results in a constant.](image)

In FBP, the projections are measured and then weighted by a cosine function, followed by filtering with a ramp filter. The cosine function is used to correct for the divergence of the fan-beam projection data. The projections are then weighted for redundancy using any of the three weighting schemes mentioned earlier. In the present implementation, the algorithm for fan-beam filtered back-projection only allows data collected over 360° and is, thus, not weighted for redundancy. Finally, the projections are weighted by the inverse-squared distance between the point to be back-projected and the x-ray source and back projected to obtain the reconstructed image.

In DFBP, the projections are measured and then differentiated with respect to the source position. The differentiated projections are then filtered using the Hilbert filter.
This differentiation followed by Hilbert filtering is equivalent to ramp filtering in the FBP method (refer to Appendix A). The filtered data is then weighted for redundancy followed by inverse-distance weighting and back-projection as described earlier.

### 2.1.1.3 Cone-Beam Geometry

Further development in CT scanning demanded representation of volumes instead of single cross-sections. Cross-sectional images were stacked together to form a three-dimensional image. This stacking led to inaccuracies due to misalignment. Concurrent measurement of projections for multiple slices led to the development of cone-beam CT.

![Flow chart showing steps followed in DFBP for the fan-beam case.](Image)

Instead of illuminating a single slice of the object with a fan of x-rays, the entire object is illuminated with a point source, and the x-ray flux is measured in a 2D plane (Figure 2.16). The main advantage of the cone-beam geometry is the reduction in data collection time. With a single source, ray integrals are measured through every point in the object in the time it takes to measure one slice in a conventional 2D scanner. The three-dimensional reconstruction is based on filtering and back-projecting individual
planes within the cone. These planes are tilted within the cone, and the contributions from all tilted planes result in the final 3D image.

The reconstruction methods used for the cone-beam geometry are similar to those of the fan-beam geometry, because the cone-beam is essentially a three-dimensional representation of the fan-beam geometry.\(^\text{12}\)

2.1.1.4 Spiral-Beam Geometry

Spiral CT scanning is a mode of acquiring projection data continuously, while the patient is translated at a constant speed\(^\text{13}\) (Figure 2.17). The patient translation distance per gantry rotation is defined by the table speed. The pitch of a spiral scan refers to the ratio of the table-translation distance per gantry rotation to the thickness of the x-ray-beam-defined slice in the Z-direction.\(^\text{13}\) Spiral CT scanners are further classified into two types: single slice and multi-slice CT scanners. In the case of single slice CT, the illumination with x-rays is the same as that in the case of a fan-beam scanner (Figure 2.18a). A multi-slice CT scanner is equipped with multiple rows of detectors (Figure

Figure 2.16: Cone-beam geometry using x-ray point source.
2.18b). Spiral CT scanning attempts to realize volume coverage at high speed with minimum induced artifacts.

Figure 2.17: Translation motion of patient table for helical scanning mode.\textsuperscript{14}

There are two main algorithms used for helical reconstruction: 180LI (Linear Interpolation) and 360LI. They have different properties in terms of slice-profile broadening and artifacts due to patient-table translation.\textsuperscript{13}
The 360° LI algorithm considers 360° periodicity in the projection data, as projection data 360° apart would be identical in the absence of patient motion, noise variation and other errors. It uses two sets of helical CT projections 360° apart to estimate one set of projections at a prescribed location within the spiral. The 180° LI algorithm, also called the half-scan algorithm, utilizes the 180° periodicity in the projection data, as two measurements along the same path but in the opposite directions would be the same in the absence of patient motion, noise variation and other errors. It uses two sets of helical CT projections, 180° apart, to estimate one set of projections at a prescribed location within the spiral.\textsuperscript{13}

2.1.2 Factors Affecting the Reconstructed Images Qualitatively and Quantitatively

The main objective of reconstruction is to achieve an artifact-free, back-projected image, but there are some errors that may be present. The main types of reconstruction errors present are those caused by insufficiency of data or by random noise. Insufficiency
of data arises mainly due to under-sampling of the projection data or because not enough projections are recorded. The artifacts caused due to insufficiency of data are known as aliasing distortions. The noise in the projection data can be categorized into two types: electrical noise and shot noise. Electrical noise is the noise generated from the electronics of the scanner system. Shot noise occurs when the number of detected photons is small enough to give rise to detectable statistical fluctuations in the measurement.

These errors affect reconstructed images qualitatively as well as quantitatively. We will assess qualitative aspects of the displayed image by various methods, including zoom and pan as well as brightness and contrast controls. Two images can be subtracted from each other and the result displayed for assessment. Line profiles can be viewed for lines drawn by the user through the reconstructed object and exported to an Excel sheet. Quantitative analysis will be performed by applying user-selected masks within the object, and calculating the standard deviation, mean and coefficient of variation for the region covered by the mask. Detailed explanations follow in Chapter 4.
3. Generation of Object and Projection

To study the different types of reconstruction algorithms, objects were created, and corresponding projections were calculated using MATLAB. To perform a comparative study of all these objects with different scanner geometries and reconstruction methods, a GUI (Graphical User Interference) was built to facilitate easy analyses.

3.1 GUI (Graphical User Interference)

A GUI is a useful tool when navigation complexity is an issue. In our case, complexities are represented in the different types of CT reconstruction algorithms. The GUI provides information in graphic form, which helps the user to invoke a reconstruction algorithm and associated parameters with minimal knowledge about the computer code. The GUI is created using a tool called GUIDE in MATLAB. This tool allows the programmer to build the model by placing components into a GUI layout space. The three main elements required to build a GUI are explained briefly below.

Components: The GUI components are placed on the blank figure to build the interface layout. They consist of graphical controls: push buttons, edit boxes, lists, sliders; static elements: text strings and frames; menus; and axes.

Figures: Blank figures are created using the figure function. The components mentioned above can also be placed within a blank figure to build a GUI.

Callbacks: Links need to be created to respond to events of components like the pressing
of a push button or the writing of text in an edit box. The MATLAB program needs to respond to these events to perform appropriate functions. The code executed in response to an event is known as a callback. There must be a callback to implement the function of each graphical component of the GUI. The component properties can be modified as required using the property editor.

The current project’s main GUI window is divided into 4 parts (Figure 3.1): The object generation is accessed by panels labeled 2D Objects and 3D Objects; the scanner geometry is represented by panels labeled Parallel Beam Reconstruction, Fan Beam Reconstruction, 3D Cone Beam Reconstruction, and Spiral Beam Reconstruction; the calculation of projections is accessed by the push buttons labeled Calculate parallel beam projections, Calculate equidistant fan beam projections, Calculate equiangular fan beam projections, Calculate 3D equidistant cone beam projections, Calculate single slice projections, and Calculate multi slice spiral projections; and the last part is the type of reconstruction algorithm used to obtain the reconstructed images, which is executed by pressing push buttons labeled Filtered back-projection in the Parallel Beam Reconstruction panel; Filtered back-projection, Local weighting scheme, and Global weighting scheme in the Equidistant Fan Beam panel; Filtered back-projection, Smooth weighting short-scan method, Local weighting scheme, and Global weighting scheme in the Equiangular Fan Beam panel; 3D Local weighting scheme, 3D Filtered back-projection, and 3D Global weighting scheme in the 3D Equidistant Cone Beam panel; 360 LI single slice in the Single Slice Spiral CT panel, and 360 LI multi slice in the Multi Slice Spiral CT panel. All of these functions are explained in detail below. The reconstructed algorithms are then analyzed using analysis tool panels built for 2D and 3D
objects separately. The detailed explanation for the analysis panels follows in Chapter 4.

3.1.1 Object Generation

The panels **2D Objects** and **3D Objects** of the main GUI window (Figure 3.1) give the user an option to simulate either 2D or 3D objects. All objects created to study the reconstruction algorithm are considered to be elliptical in shape; 2D objects with straight-line edges are beyond the scope of this GUI tool. A push button labeled **Circle / Ellipse** allows the user to create circular- or elliptical-shaped two-dimensional objects. Similarly, for three-dimensional objects, a push button labeled **Sphere / Ellipsoid / Cylinder** allows the user to create spherical, ellipsoidal or cylindrical objects. These push buttons open a new window, which allows the user to enter the parameters to create the respective objects. General parameters that a user needs to enter (in both the 2D- and 3D-cases) are **Reconstruction diameter**, which defines the diameter of the object-bounding circle,
within which the 2D object is contained, *Reconstruction matrix size (x/y)* in the x-y plane, and the *Number of objects* to be created. The three-dimensional case additionally requires the user to enter *Reconstruction diameter (z)* and *Reconstruction matrix size (z)* along the z-axis. The z-axis dimension of the matrix is also taken as the dimension of the detector array in the z-direction. These general parameters are stored by pressing the button labeled *Enter the reconstruction diameter (x/y), reconstruction matrix size (x/y), and the number of objects* (Figure 3.2).

In the 2D-case, a user can create either an ellipse or a circle as discussed above. The circle is considered as a special case of an ellipse if the x-half axis and y-half axis parameters are equal. The 2D-object selection allows selection of up to 10 objects at one time. Each object creation requires five object-specific parameters to be entered using edit boxes (Figure 3.2). The five parameters are x-half axis, y-half axis, x-center, y-center, and the density of the object. The density value entered by the user for overlapping objects is implemented additively. Hence, the user needs to enter the values appropriately, possibly as a negative number to reduce the density in the interior of a given object. To explain this, let us consider an example of two concentric circular objects centered at (0, 0) with radius 5 and 3 centimeters, respectively. Now, in order to assign the density value of 1 to the larger circle and 0.3 to the smaller circle, the user needs to enter the density values as 1 and -0.7, respectively. The density for each object created is considered to be uniform. Depending on the number of objects selected, the appropriate numbers of object-specific parameter edit boxes are enabled or disabled (grayed out). Once the object-creation parameters are entered, the push button labeled *Store object* allows the user to save the object parameters in a user-specified file. The
Figure 3.2: The *Circle / Ellipse* window layout showing five object-specific parameter edit boxes for the 2D object case.

The current created object is displayed by pushing the button labeled *Display the current created object*, and the button labeled *Display any created object* displays a previously saved object. Finally, the push button labeled *Done* is pressed to exit the current window.

In the 3D-case, a user can create spherical, ellipsoidal, and cylindrical objects (Figure 3.3). The final 3D object may contain a combination of several ellipsoids, spheres, and cylinders. At any given instance, up to 10 objects of each type may be defined, i.e., the object may consist of up to 30 user-defined shapes.

The 3D-case has three options of objects as mentioned earlier, and each object has its own object-specific parameters. The object-specific parameters for a sphere are x-center, y-center, z-center, radius, and density (Figure 3.4). The object-specific parameters for an ellipsoid are x-center, y-center, z-center, x-half axis, y-half axis, z-half axis, and density.
Figure 3.3: Layout of Sphere / Ellipsoid / Cylinder window showing the object selection options for the 3D-case.

The object-specific parameters for a cylinder are x-center, y-center, z-center, radius of cylinder, length of cylinder, density, and orientation of cylinder. As in the 2D-case, density for overlapping objects is also implemented additively.

After entering all object-specific parameters, they are saved with a user-specified file name and displayed as in the 2D-case. The only difference in the display screen is that we have sliders, which allow the user to select a specific slice to be displayed. The display screen consists of three figures, which display the x-y, x-z, and y-z planes, respectively. The Done push button exits the GUI window as in the 2D-case.

3.1.2 Scanner Geometry

The GUI window has four different panels for the type of scanner geometry. The four scanner geometries are parallel beam, fan beam, cone beam and spiral beam geometry (refer to Chapter 2). Each of these panels specifies the geometry of the projections and the reconstruction algorithm used to create the reconstructed image for
the particular scanner geometry.

3.1.3 Calculation of Projections

The projections are calculated for a range of projection angles varying from 0 to \(2\pi\) for both 2D and 3D objects. The GUI window for calculating projections allows the user to enter the projection parameters for each type of scanner geometry (Table 3.1).

Table 3.1: Projection parameters for implemented scanner geometries.

<table>
<thead>
<tr>
<th>Scanner Geometry</th>
<th>Projection Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Two-Dimensional</strong></td>
<td></td>
</tr>
<tr>
<td>Parallel-beam</td>
<td>number of source positions and number of detectors</td>
</tr>
<tr>
<td>Fan-beam</td>
<td>number of source positions, number of detectors, and source-to-isocentre distance</td>
</tr>
<tr>
<td><strong>Three-Dimensional</strong></td>
<td></td>
</tr>
<tr>
<td>Cone-beam</td>
<td>number of source positions, number of detectors, and source-to-isocentre distance</td>
</tr>
<tr>
<td>Single-slice spiral-beam</td>
<td>number of source positions, number of detectors, source-to-isocentre distance, number of spirals, and helical pitch distance</td>
</tr>
<tr>
<td>Multi-slice spiral-beam</td>
<td>number of source positions, number of detectors, source-to-isocentre distance, number of spirals, helical pitch, and detector row distance</td>
</tr>
</tbody>
</table>
To explain the projections calculated in detail for 2D and 3D objects created by
the user, let us consider an example of each case.

In the 2D-case consider the example of an ellipse. To calculate the projections for
an elliptical object, the equation of an ellipse is given by

$$
\left(\frac{x-x_0}{a}\right)^2 + \left(\frac{y-y_0}{b}\right)^2 = 1
$$

(7)

where $x_0 = x$-centre ,

$y_0 = y$-centre ,

$a = x$-half axis, and

$b = y$-half axis.

The values for the $x$-centre, $y$-centre, $x$-half axis, $y$-half axis, and density of the ellipse
are entered by the user (refer to Section 3.1.1). The density of the ellipse represents the
linear attenuation coefficient of the object and is considered to be uniform throughout the
given ellipse. The user also enters the projection parameters as discussed above,
depending on the scanner geometry. The ellipse is created using equation (7) by
accepting the values entered by the user as input.

The sum of all density values $f(x,y)$ along a projection line $s(t,\theta)$ represents the
value for a projection point $P(t,\theta)$ (Equation 1) for the parallel-beam scanning geometry.
Similarly for the fan-beam scanner geometry, projection point $P_{\beta,y}$ is defined as follows:

$$
P_{\beta,y} = \int_{s(r,\beta,y)} f(x,y) ds
$$

(8)

where $f(x,y) =$ the density distribution of a two-dimensional object,

$s = s(r,\beta,y) =$ integration line (Figure 3.5),

$\beta =$ source angle,
\( \gamma = \) fan angle,

\( S = \) source position, and

\( r = \) source-to-centre distance.

Figure 3.5: The dashed line \( s(r, \beta, \gamma) \) represents the integration path for the projection point \( P_\beta \gamma \) for the fan beam case.

In the simplified case of a homogenous object, we calculate the path length along \( s \) within the object and multiply it with the density to obtain \( P \). To obtain the path length \( d \), we calculate the intersection points of the projection line with the boundary of the ellipse and compute the Euclidean distance between the two intersection points (Figure 3.6).

Figure 3.6: Intersection between the projection line and the boundary of the ellipse; \( d \) is the distance between the two intersection points \((x_1, y_1)\) and \((x_2, y_2)\), respectively.
The projection profiles for a given object are stored in a matrix. In the case of more than one object, the projections for each object are calculated separately, and the resulting matrices are summed together to form the final projection matrix.

Now let us consider an example of a sphere for the 3D-case. The sphere centered at \((x_0, y_0, z_0)\) is given by

\[
(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2,
\]

where \(R = \text{radius}\).

As discussed in the elliptical object case above, summation of all density values along a projection line represents a projections point. The projection line equation for the 3D-case is

\[
P_{\beta, \gamma, \delta} = \int_{s(r, \beta, \gamma, \delta)} f(x, y, z) ds,
\]

where \(f(x, y, z) = \text{density distribution of a three-dimensional object}\),

\(s = s(r, \beta, \gamma, \delta) = \text{integration line (Figure 3.7)},\)

\(\beta = \text{source angle},\)

\(\gamma = \text{fan angle},\)

\(\delta = \text{cone angle},\)

\(S = \text{source position},\) and

\(r = \text{source-to-centre distance in x-y plane}.\)

The intersections of the projection line with the surface of the sphere define the distance required for further calculations. The product of this distance and the density of the object is calculated as in the 2D-case and is stored in a matrix. The sum of all matrices in the case of multiple objects represents the final set of projections. These are reconstructed using different reconstruction algorithms as explained in the following part.
Figure 3.7: Geometry of cone-beam parameters. Vector $\vec{r}$ denotes the source position and is normal to the detector plane D.P. The source rotates in the $x$-$y$ plane by incrementing angle $\beta$. The location of the integration path $s$ is defined by angles $\gamma$ and $\delta$.

### 3.1.4 Type of Reconstruction Algorithm

The calculated projections for the created objects are saved under a user-defined filename. They are input into one of the available reconstruction algorithms to obtain the final image and saved with the same main filename as the projections but with a different extension. All reconstruction algorithms are associated with a task bar, which displays the progress of the program (Figure 3.8). The GUI window in all reconstruction algorithms allows the user to select the user-saved projection file by using a push button and displays the back-projected image in the same window.

Figure 3.8: Task bar showing the progress of reconstruction algorithm.
This section gives the steps followed for each available reconstruction algorithm. The basic algorithms for the parallel-beam, equidistant fan-beam, equiangular fan-beam, and cone-beam geometry follow the FBP reconstruction steps outlined in Sections 2.1.1.1 to 2.1.1.3. Additional algorithms allowing specific weighting schemes are implemented as follows.

**Equidistant fan-beam geometry:** For the local weighting scheme, the projections are equally weighted and back-projected using the DFBP algorithm. The GUI window allows the user to enter the start and end scan angle between 0 and 2\(\pi\). The set of projections in the specified angular range are then back-projected, and an image ROI (Region of Interest) of the object is obtained. The GUI window has two push buttons assigned to display the differentiated and convolved projections.

The global weighting scheme is implemented using the same steps and GUI components as those used for the local weighting scheme discussed above. The only difference is the weighting scheme itself.

**Equiangular fan-beam geometry:** The GUI window for this case provides four reconstruction algorithms. Three of these algorithms are similar to the equidistant case, except that the beam geometry is now the equiangular fan beam. The fourth reconstruction algorithm uses the smooth short-scan weighting method. This algorithm is differentiation free and uses FBP. The projection data are first corrected for divergence by cosine weighting. The corrected projection data are then weighted with Parker’s weighting. The weighted data are filtered using a ramp filter\(^5\) and back-projected. The GUI allows the user to select the projection file by using a push button, and it displays the back-projected image on the same window.
**Cone-beam geometry:** The GUI tool considers only the equidistant case, because most practical cone-beam scanners use flat-panel detectors. Three reconstruction algorithms are available to reconstruct cone beam data.

In all algorithms, the projection file is pre-weighted for cone- and fan-angle divergence by two cosine weighting functions. The GUI window allows the user to enter the range of slices (i.e., the minimum and maximum slice positions) and the saved projection file to be reconstructed. It also allows the user to display back-projected slices in the x-y, x-z and y-z planes. The user has the option to enter the slice number in all three planes. The slice number to be displayed should be within the previously entered reconstructed slice range.

In addition to the 3D filtered back-projection algorithm by Feldkamp\(^{12}\), the 3D local weighting scheme is available. The cone-beam corrected projection data are treated by steps similar to those used in the local weighting scheme in the 2D-case using the DFBP method.

The third algorithm uses the 3D global weighting scheme. This algorithm differs from 3D equal weighting only in the weighting scheme applied. The weighting function used in this case is Noo’s weighting scheme.

**Spiral-beam geometry:** There are two main algorithms used for spiral-beam reconstruction: 180 LI (Linear Interpolation) and 360 LI. Only the 360 LI algorithm is currently implemented (Section 2.1.1.4).

The GUI window has two separate panels for single-slice and multi-slice spiral CT scanners. The GUI window in both spiral-beam cases allows the user to select the
projection file using a push button. It also allows the user to display back-projected slices in the x-y, x-z, and y-z planes. A push button labeled *Done* is used to close the current GUI window.

In the spiral-beam case, the projection data are not aligned within a slice due to the helical scanning path. Hence, slice interpolation must be implemented to obtain a complete data set for reconstruction. The reconstruction algorithm implemented for both single- and multi-slice spiral-beam requires slice interpolation to obtain a complete projection data set. The projection data are corrected for divergence, then filtered using a ramp filter\(^5\), and back-projected to obtain a reconstructed image. The reconstruction algorithm used for multi-row detectors considers multiple parallel fan-beams. The implemented multi-slice spiral CT 360 LI algorithm\(^{11}\) is restricted to the 4-slice spiral CT geometry (4 rows of detectors) and has a fixed helical pitch of 1.
4. Analysis of Reconstructed Data

The GUI-driven software can be used to study the different types of reconstruction algorithms. For this purpose an analysis tool was created to perform qualitative and quantitative analyses of reconstructed images.

4.1 Description of Analysis Tool

The analysis tool was created to handle both 2D and 3D images. The following description is divided into three sections: selection of test image, type of analysis to be performed, and display of results of the performed analysis.

Selection of test image: The user selects a reconstructed object file by using the browsing option (Figure 4.1). In the 2D-case, the analysis tool displays the reconstructed image in a single figure (Figure 4.1 a), whereas in the 3D-case it is displayed in x-y, x-z, and y-z planes (Figure 4.1 b). The 3D analysis is a slice-based analysis, and the user is given the option to enter the slice number of the reconstructed 3D image to be displayed and analyzed. The analysis-tool window in both cases provides the reconstructed image information, such as file name and path of the image, number of detectors, number of source positions, reconstructed matrix size, and start and end scanning angle.

In the 3D-case, the GUI also displays the range of slices back-projected and the reconstruction matrix size along the z-axis.
Figure 4.1: Window showing object data and mask analysis panel for a selected two-dimensional object (a) and a three-dimensional object (b). The lower window shows the reconstructed object in three planes, x-y, x-z, and y-z, respectively. Both (a) and (b) show the types of analysis options available to the user.

**Type of analysis to be performed:** The user can perform either qualitative or quantitative analyses. As a part of the qualitative analysis, the user is provided with...
brightness and contrast adjustment for each figure and a MATLAB tool bar at the top of each GUI window (e.g. Figure 4.1). The image-subtraction tool allows the user to study the difference between images. The two options for quantitative analysis are: *Mask analysis* and *line-profile analysis*. In all analysis options available, the user will follow the push buttons that are highlighted in blue color. Now let us discuss each of the analysis options in detail.

1. **Image subtraction**: This type of analysis allows the user to select two images and obtain a difference image. The selected images can either be both reconstructed images or one reconstructed and one original object image.

2. **Mask analysis**: The mask analysis option is further divided into two types: user-defined and object-defined mask analysis.

   - **User-defined object mask analysis**: This type of analysis allows the user to select either one or two masks to be drawn. In the 3D-case, as mentioned earlier, the user is first required to select one of the three display planes for analysis. For both 2D and 3D objects, elliptical masks are available. For the 3D-case, a rectangular mask is also available to allow appropriate analysis of a cylinder. If the user selects to draw two masks, then a combined mask can be created using a logical operation. The logical operations available are XOR, OR, and AND. Only one logical operation can be performed at a time. The x-centre, y-centre, x-half length, and y-half length values of the user-selected mask are displayed in the corresponding edit boxes. If the user draws a wrong mask, the mask can be adjusted by using the mouse. The mask values are updated by clicking the pushbutton labeled *Update position to be saved*. Each mask is also provided with
the option to be eroded or dilated by a certain number of pixels.

- Object-defined mask analysis: This type of analysis is similar to the user-defined mask analysis, except that the parameters used to create the mask are those of the original object. These parameters are stored with each original object. Similarly, in the 3D-case, the slice to be analyzed is associated with a pre-defined mask as in the 2D-case. In both cases the masks are defined for the first object in the one-mask analysis case, and for the first and second object in the two-mask analysis case. The user needs, thus, to consider the analysis plan when creating the mathematical objects. The masks can logically be combined and eroded or dilated using the appropriate option as in the user-defined mask analysis.

3. **Line-profile analysis:** This type of analysis allows the user to draw a line across the reconstructed image, which provides the image-value profile of the pixels through which the line passes. The start- and end-point coordinates of the drawn line are displayed in the corresponding display boxes of the window. Line profile analysis can be performed for one or two selected images. If two images are selected for analysis, the location of the line in both images is identical. In this case, a difference line-profile plot can also be obtained.

**Display of results of the performed analysis:** The display of the results obtained in each of the analysis types is initiated through the same GUI window as the analysis type. The results for user-defined and object-defined analyses are displayed in a similar way. In both cases, standard deviation, mean and coefficient of variation are calculated for the region selected by the user.
The images of the binary mask and the masked region of the image are displayed in a new window. The masks created in all analysis types can be saved and can be reloaded later to perform the same mask analysis on other images. These actions are performed by activating the push buttons labeled *Save mask* and *Load saved mask*, respectively.

The difference image in the image-subtraction tool is displayed adjacent to the two selected images. Similarly, the line-profiles are displayed adjacent to the images used for selecting the location of the lines. In the case of the two-image line-profile analysis, the user is given the option to display the line through the first image, the line through the second image, the difference between the two lines or any combination of the three. The line-profile data can be saved in an Excel spreadsheet under a user-specified file name by using the push button labeled *Save line profile data*.

**4.2 Sample Analysis**

**4.2.1 Mathematical Test Phantoms**

To study the tool, mathematical test phantoms were created for both the 2D and 3D-case, reflecting the cylindrical shape and densities of a long bone. These phantoms are explained in detail below.

The 2D test phantom consisted of two concentric circles with outer and inner densities equivalent to cortical and trabecular bone (Figure 4.2). The test phantom for the 3D-case was composed of two concentric cylinders aligned along the z axis, again simulating the outer cortical and inner trabecular bone regions (Figure 4.3). The geometry and density parameters of the two phantoms are tabulated in Figures 4.2 and 4.3, respectively. The central sections of the 3D simulated phantom are displayed in the
x-y, x-z and y-z planes in Figure 4.3b. The analysis performed on the reconstructed mathematical test phantoms is explained in the following section.

![Figure 4.2: Schematic representation of the simulated two-dimensional phantom. The density values represent the material composition of the cortical and trabecular regions of the bone.](image1)

<table>
<thead>
<tr>
<th></th>
<th>Radius (cm)</th>
<th>Density (cm$^{-1}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortical Bone Region</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Trabecular Bone Region</td>
<td>4</td>
<td>0.3</td>
</tr>
</tbody>
</table>

![Figure 4.3: (a) Schematic representation of the simulated three-dimensional cylindrical phantom. The table specifies the geometry and densities of the phantom components. (b) Two-dimensional slices of the three-dimensional phantom shown in part (a) in the x-y, x-z, and y-z planes.](image2)

<table>
<thead>
<tr>
<th></th>
<th>Radius (cm)</th>
<th>Length (cm)</th>
<th>Density(cm$^{-1}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortical Bone Region</td>
<td>5 ($r_1$)</td>
<td>10 ($l_1$)</td>
<td>1</td>
</tr>
<tr>
<td>Trabecular Bone Region</td>
<td>4 ($r_2$)</td>
<td>8 ($l_2$)</td>
<td>0.3</td>
</tr>
</tbody>
</table>

4.2.2 Analysis Results

To test the GUI tool, a few sample tests were performed, using different types of
reconstruction algorithms and applying the above discussed analysis tools for both 2D and 3D objects.

Test 1: The object-defined analysis option of the tool was tested using the 2D and 3D mathematical phantoms. The 2D phantom was reconstructed on a 256 x 256 matrix using the equidistant fan-beam filtered back-projection algorithm. The number of source positions were varied from 50-1050 in steps of 40 and the number of detectors from 64-1024 in multiples of 2. Two masks were drawn, covering the cortical and trabecular regions, using the object-defined mask analysis option. To avoid the partial-volume effect areas, the masks were kept away from object edges by eroding or dilating the masks. A mask1 was formed by eroding the larger circular mask by 2 pixels, dilating the smaller circular mask by 2 pixels, and combining the two masks with an XOR operation. This mask was used to analyze the cortical region. Another mask, mask2, was created by eroding the smaller circular region by 2 pixels. This second mask was applied to the trabecular region. The masks used for this sample analysis were saved for one reconstructed image. The saved mask was loaded later to perform the same mask analysis for different 2D reconstructed images under the different combinations of numbers of source positions and detectors.

Similarly, the 3D phantom was reconstructed using the 3D filtered back-projection algorithm. The matrix size was 128x128x128 corresponding to the x-, y-, and z- direction. The range of source positions and detectors used were 50-370 in steps of 40 and 64-512 in multiples of 2, respectively. The 2D slice at z = 64 in the x-y plane was selected for analysis. The masks selected in this case were similar to those of the 2D-case discussed above.
The standard deviations for the above selected mask regions are shown in Figure 4.4 for both 2D and 3D reconstructed objects. These plots can be used to examine the relationship between the standard deviations of the reconstructed values and the relevant reconstruction parameters. For the cortical bone region, the standard deviation decreases with increasing number of source positions and with increasing number of detectors. However, there is no benefit in increasing one parameter independently of the other. The standard deviation for the trabecular region is small compared to that of the cortical region of the bone and appears to be mainly influenced by the number of detectors and not by the number of source positions.

Figure 4.4: Variation of standard deviation in reconstructed values with number of source positions and detectors for cortical bone region and trabecular bone region. The data are simulated for a two-dimensional phantom (a) and (b) and for a three-dimensional phantom (c) and (d). The legends along the graphs denote the number of detectors used.
Test 2: To study the variation in the reconstructed image values in more detail, the line profile analysis tool was used.

Figure 4.5: The green line passing through the centre of the 2D (a) and 3D (b) selected image was used for the line-profile analysis. The line profile analysis for two images is shown in (a). Due to computer storage limitations, the 3D image was reconstructed only for a limited number of slices along the z-axis. The slice at z = 64 is shown in (b). The line profiles obtained are also displayed.
The analysis was performed across both the cortical and trabecular regions of the 2D and 3D reconstructed images discussed in Test 1. To perform the line-profile analysis, a line was drawn passing through the centre of the selected image (Figure 4.5).

Figure 4.6: Line profiles through the centre of the two-dimensional phantom showing variations in reconstructed image values. The plots were taken for 64-1024 detectors and a fixed number of 250 source positions.
The line-profile plots shown in Figure 4.6 were obtained from the 2D reconstructed image for 250 source positions and 64, 128, 256, 512, and 1024 detectors.

In the 3D-case the line profiles were obtained for 64-512 detectors and also for 250 source positions (Figure 4.7). The line profiles can be compared to the mask analysis performed earlier. The cortical regions show a decrease in the variation of image values with an increase in the number of detectors, which also was reflected in the mask-analysis graph (Figure 4.4). The trabecular region in the line-profile analysis showed a smaller variation with an increase in the number of detectors.

Figure 4.7: These plots represent line profiles through the central slice of the three-dimensional phantom, showing variations in reconstruction noise. The plots were taken for a range of detectors varying from 64 to 512 and a fixed number of 250 source positions.
The line-profile analysis can also be used to study the noise properties in the reconstructed images. Aliasing results from over-sampling of the projections. This phenomenon can clearly be seen in Figures 4.6 and 4.7, where overshoots and undershoots result in areas of sharp density variation. These overshoots and undershoots appear as rings in the reconstructed image and is commonly known as Gibbs phenomenon. The variations in the outer regions of the phantom are also due to sampling errors, where the number of source positions and the number of detectors used for reconstruction do not match. For a fixed number of source positions, an increase in the number of detectors increases the noise outside the phantom.

Test 3: The image-subtraction tool was used to study the qualitative deterioration of the reconstructed image. In this analysis, difference images were obtained by subtracting the original object image from the reconstructed image (Figure 4.8).

The 2D phantom was reconstructed by the equidistant fan-beam filtered back-projection algorithm on a 256x256 matrix using 256 projections with 200 samples per projection. Blurring of edges for both trabecular and cortical regions of the phantom are clearly visible in the difference image.

Similarly, the 3D phantom was reconstructed on a 128x128x128 matrix by the 3D filtered back-projection algorithm using 200 projections with a 200x128 detector array. The slice at z = 64 is displayed for analysis. The difference image shows blurring similar to that in the 2D case.

Test 4: The tool was also used to visualize images reconstructed by the available reconstruction algorithms. The 2D and 3D mathematical phantoms simulated above were
used. Figures 4.9 to 4.16 show the variations in the reconstructed images using different reconstruction algorithms.

Figure 4.8: The difference images for the original object image subtracted from a reconstructed image are shown for both a 2D (a) as well as 3D (b) phantom. In both cases, the left image is the reconstructed image and the right image is the original object image.
Figure 4.9: Parallel-beam reconstruction algorithm. 128 projections were collected over $\pi$ radians with 64 samples per projection. The Shepp and Logan filter was used, and the data were back-projected on a 256x256 matrix. A smoothing effect at the inner edge is visible due to under sampling of the projections. The image values above the upper display limit are color coded equivalent to the upper display limit, and the values below the lower display limit are color coded equivalent to the lower display limit.

Figure 4.10: Equidistant fan-beam filtered back-projection algorithm. 256 projections were collected over $2\pi$ radians with 64 samples per projection. The ramp filter was used, and the data were back-projected on a 256x256 matrix. The Gibbs phenomenon due to under sampling of the projections is visible as dark red rings at both the inner and outer boundary of the cylinder.
Figure 4.11: Equidistant fan-beam differentiated filtered back-projection algorithm. 256 projections were collected over $\pi$ radians with 200 samples per projection. Local and global weighting schemes were used left and right, and the data were back-projected on a 256x256 matrix. The local weighting scheme applies the pixel-weighted data only through $\pi$ radians and zero for the rest. In contrast, the global weighting scheme uses a smooth weighting filter, and data beyond $\pi$ radians can be seen with blurring.
Figure 4.12: Equiangular fan-beam differentiated filtered back-projection. The reconstructed image (a) is obtained using the local weighting scheme, (b) is obtained using the global weighting scheme, both for the short-scan angular interval $[0, \pi + \text{fan angle}]$ (fan angle = 0.6769 radians). 256 projections with 200 samples per projection were used. The local weighted image shows fish-scale artifacts, whereas the global weighted image shows blurring due to a smoother weighting. To see the artifact, a display window size of [0.29 0.33] was used. The display characteristics can be adjusted using the contrast/brightness adjustment button.
Figure 4.13: 3D cone-beam filtered back-projection. 200 projections were collected over $2\pi$ with a 200x128 2D detector array (128 is along z-axis). The back-projection image size was on 128x128x128 matrix. Due to computer limitations, the image was back-projected for a slice range $z = 50$ to $z = 80$. (a) The reconstructed image at $z = 64$ shows a Moiré pattern due to under-sampling. The image window used was $[0.28 , .3]$ to display the artifact. (b) shows a slice at $x = 55$, showing high intensity streaks.

Figure 4.14: Cone-beam reconstructed image using the 3D local weighting scheme. 200 projections collected over $\pi$ radians with 200 samples per projection. Due to the local weighting scheme, data are reconstructed over $\pi$ and missing in other scanning regions. The image on the left is at $z = 64$ and on the right at $y = 64$. The image is back-projected on a 128x128x128 matrix.
Figure 4.15: Cone-beam reconstructed image shown at $y = 64$ on left and at $z = 64$ on right using the 3D global weighting scheme for a scan angular interval $[0, \pi]$. The image shows smooth weighted data due to the smooth weighting function in both x-y and x-z planes. The parameters used for reconstruction are the same as in Figure 4.14.

Figure 4.16: 360 LI spiral-beam reconstruction showing slices at $z = 64$, $y = 64$ and $x = 64$. (a) Single-slice spiral-beam 360 LI algorithm. Smoothing due to slice interpolation can be seen at $y = 64$ and $x = 64$, respectively. (b) Multi-slice spiral-beam 360 LI algorithm. Smoothing due to slice interpolation at the same slice locations as in the single-slice case is seen but with additional edge effects at the left and right boundaries of the cylinder. In both cases 200 projections were collected over $2\pi$ radians with a 200x128 2D detector array (128 is along z-axis). The images were back projected on a 128x128x128 matrix. The number of spirals used to transverse the entire object was 128. In both cases a pitch of 1 was used.
5. Discussion and Conclusion

Other than the tests described in Chapter 4, the tool was also evaluated by five users. The ease of using the tool was checked by this exercise, and modifications suggested by the five users were implemented in the tool. The aim of building the tool was to provide a unified platform to invoke the basic types of reconstruction algorithms and study their performance using defined objects. The resultant tool was flexible in some parts and was limited in a few areas. This section will summarize the developed GUI tool and point out improvements that could be implemented in the future.

Table 5.1: Overview of the created GUI tool

<table>
<thead>
<tr>
<th>Implemented Parts</th>
<th>2D</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object Shapes</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ellipse / Circle</td>
<td></td>
<td>Sphere, Ellipsoid, and Cylinder</td>
</tr>
<tr>
<td><strong>Scanner Geometry</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel beam</td>
<td>Equidistant fan beam</td>
<td>Equiangular fan beam</td>
</tr>
<tr>
<td>Equiangular fan beam</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Reconstruction Algorithm</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filtered back-projection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local weighting scheme</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Analysis Tool</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User-defined mask analysis</td>
<td>User-defined mask analysis</td>
<td></td>
</tr>
<tr>
<td>Object-defined mask analysis</td>
<td>Object-defined mask analysis</td>
<td></td>
</tr>
<tr>
<td>Image subtraction</td>
<td>Image subtraction</td>
<td></td>
</tr>
<tr>
<td>Line-profile analysis</td>
<td>Line-profile analysis</td>
<td></td>
</tr>
</tbody>
</table>

The available code allowed use of the reconstruction algorithms listed in Table 5.1 for each scanner geometry. The tool enables the user to select the relevant parameters
for reconstruction and study their influence on the image. The reconstructed image can then be saved under a user-defined file name.

The current tool is limited to elliptical and symmetrical object shapes, which prohibits the study of objects with other shapes and symmetries. Although the orientation of the object is currently limited parallel to the x-, y-, and z-directions, change in orientation could be implemented for both 2D and 3D objects as part of future modifications. The following steps are needed to calculate projections for 2D elliptical objects at arbitrary orientations (Figure 5.1):

1. Specify the rotational angle $\xi$
2. Multiply the elliptical object definition with a 2D rotational matrix
3. Find distance of projection line intersecting the elliptical object
4. Calculate projections as the product of distance and density

Figure 5.1: Flow chart showing steps to calculate projections for change in orientation.

The rotational angle $\xi$ is the angle between the horizontal half-axis of the ellipse and the x-axis of the coordinate system. The 2D rotational matrix $\begin{bmatrix} \cos \xi & \sin \xi \\ -\sin \xi & \cos \xi \end{bmatrix}$ is used for 2D objects, whereas in the case of 3D, the Euler matrix

$\begin{bmatrix} 
\cos \xi_1 \cos \xi_3 - \sin \xi_1 \cos \xi_2 \sin \xi_3 & -\cos \xi_1 \sin \xi_3 - \sin \xi_1 \cos \xi_2 \cos \xi_3 & \sin \xi_1 \sin \xi_2 \\
\sin \xi_1 \cos \xi_3 + \cos \xi_1 \cos \xi_2 \sin \xi_3 & -\sin \xi_1 \sin \xi_3 + \cos \xi_1 \cos \xi_2 \cos \xi_3 & -\sin \xi_2 \cos \xi_1 \\
\sin \xi_2 \sin \xi_3 & \sin \xi_2 \cos \xi_3 & \cos \xi_2 
\end{bmatrix}$

is used. Three angles $\xi_1, \xi_2, \text{and} \xi_3$ are required to be entered as input for the 3D object case, with $\xi_1, \xi_2, \text{and} \xi_3$ representing the Euler rotational angles about $z$-, $x'$-, and $z'$-
axes, respectively. The $x'$- and $z'$- axes represent the x-axis after rotation by $\xi_1$ and the z-axis after rotation $\xi_1$ followed by $\xi_2$.\(^{16}\)

The currently implemented reconstruction algorithms do not simulate photon noise in the reconstructed images. The addition of noise to the calculated projections based on a given photon flux would allow the study of the effect of such noise in conjunction with the reconstruction algorithm. A number of steps are necessary to add noise to the projection data (Figure 5.2), requiring an assumption of the number of photons $I_o$ in the ray impinging on the object. To add noise, the MATLAB function `poissrnd` can be used. The inputs to the `poissrnd` function are the number of photons and the matrix size consisting of the number of projections times the number of data points per projection.

The simulation programs currently implemented assume a mono-energetic x-ray beam. The calculation of projection data for an energy-dependent beam spectrum could nicely follow the introduction of $I_o$, which was necessary for the introduction of photon noise.

---

![Flow chart](image)

**Figure 5.2:** Flow chart showing steps to add noise to projection data.

- Consider fixed number of initial photons $I_o$
- Add noise to $I_o$
- Calculate the measured photons $I$ using currently implemented projection calculations
- Add noise to $I$
- Calculate projections using Beer–Lambert law $\ln \left( \frac{I_o}{I} \right)$

The simulation programs currently implemented assume a mono-energetic x-ray beam. The calculation of projection data for an energy-dependent beam spectrum could nicely follow the introduction of $I_o$, which was necessary for the introduction of photon noise.
Rather than assuming a single value for $I_o$, a number of $I_o(E)$, representing the energy spectrum of the x-ray beam, would be introduced. The remaining steps are outlined in Figure 5.3.

![Flow chart showing steps to calculate projection data for energy dependent beam spectrum.](image)

Figure 5.3: Flow chart showing steps to calculate projection data for energy dependent beam spectrum.

A number of blurring-related effects cannot be studied with the currently assumed infinitesimally narrow beam width. To consider a finite beam width for the projection data, the steps to be followed are given in Figure 5.4. The Gaussian or trapezoidal functions can be used as beam profile weighting functions.

![Flow chart showing steps to calculate projection data with finite beam width.](image)

Figure 5.4: Flow chart showing steps to calculate projection data with finite beam width.

The masks used for 2D analysis are also limited to circular or elliptical shapes, but those for 3D analysis allow rectangular shapes to accommodate cylinders. The masks
drawn manually can be modified using the mouse. To allow for more accurate mask parameters, future modifications could accommodate entry of mask parameters through the keyboard.

The object-defined mask analysis currently does not allow the user to select the objects for which masks should be created. The program automatically selects the first object for the one-mask analysis and the first two objects for the two-mask analysis. Thus, in future work an option could be provided to select the objects for which masks are to be defined.

Simple dose calculations could be implemented following the steps in Figure 5.5. The dose along each ray path is calculated and mapped onto an image matrix. The sum of these image matrixes for all projection angles gives the dose distribution of the cross-section. To calculate the dose along a projection ray, the $I$ profile through each object along the ray is calculated using $I = I_o e^{-\mu d}$, where the distance $d$ is considered in increments of the pixel width $w$ and the $\mu$ value is taken from the user-entered analytical object definition. At each increment along the ray, the original object matrix (Section 3.1.1) will be sampled to obtain the appropriate $\mu$ of the combined objects. Sequential differencing of the $I$ profiles then allows calculation of the dose deposited in each voxel.

New reconstruction algorithms other than the ones implemented can also be added later to update the tool with current reconstruction approaches.

In spite of the short-comings listed above, the created tool is useful for the basic understanding of reconstruction algorithms. The provided analysis tool can help the user to study the noise variations due to reconstruction parameters and the type of algorithm in different simulated objects.
To introduce this reconstruction simulation tool to the user, a user’s manual with a brief explanation of each tool component has been created. The parameter of each tool component is explained, and a suggested range of values to be used is also specified where meaningful.

Figure 5.5: Steps to calculate dose.
Appendix A

Transformation of reconstruction by filtered back-projection using ramp filter (FBP) to reconstruction using differentiation and Hilbert filter (DFBP).

1. Parallel-Beam Geometry

Let an object function (Figure 2.1) be defined by

\[ f(x, y) = \int_0^\pi \int_{-t_m}^{t_m} P_\theta(t) \, h(t - t') \, dt \, d\theta \]  \hspace{1cm} (A1)

where \( t' \) is the distance to the centre from the projection ray through \((x, y)\), \( t \) the distance to the centre from the other projection rays parallel to that of \( t' \), and \( \theta \) the angle of these projection rays to the x-axis.

The integral over \( t \) is actually a convolution (*):

\[ Q_\theta(t) = P_\theta(t) \ast h(t). \]  \hspace{1cm} (A2)

Applying the Fourier slice theorem, Eq. A1 can be rewritten as

\[ f(x, y) = \int_0^\pi \int_{-t_m}^{t_m} \hat{P}_\theta(\omega) |\omega| e^{2\pi j \omega t} \, d\omega \, d\theta, \]  \hspace{1cm} (A3)

where \(|\omega|\) represents a ramp filter in the frequency domain with Fourier transform \( h(t) \) and \( \hat{P}_\theta(\omega) \) the Fourier transform of the projection data \( P_\theta(t) \).

Based on Eq. A3, \( Q_\theta(t) \) is the ramp filtered projection data given by

\[ Q_\theta(t) = \int_{-\infty}^{\infty} \hat{P}_\theta(\omega) |\omega| e^{2\pi j \omega t} \, d\omega. \]  \hspace{1cm} (A4)

We can use the signum function to relate \(|\omega|\) and \( \omega \):

\[ |\omega| = sgn(\omega) \cdot \omega, \]  \hspace{1cm} (A5)

where \( sgn(\omega) \) is the signum function given by
\[
\text{sgn}(\omega) = \begin{cases} 
-1 & \text{if } \omega < 0 \\
0 & \text{if } \omega = 0 \\
1 & \text{if } \omega > 0 
\end{cases} \quad (A6)
\]

Substituting \(|\omega|\) by Eq. A5, we can rewrite Eq. A4 as

\[
Q_\theta(t) = \int_{-\infty}^{\infty} \hat{P}_\theta(\omega) \cdot \omega \cdot \text{sgn}(\omega) e^{2\pi j \omega t} d\omega. \quad (A7)
\]

Multiplying and dividing by \(2\pi^2 j^2\), Eq. A7 can be rewritten as

\[
Q_\theta(t) = \frac{1}{2\pi^2 j^2} \int_{-\infty}^{\infty} 2\pi j \omega \hat{P}_\theta(\omega) \cdot \pi j \text{sgn}(\omega) e^{2\pi j \omega t} d\omega.
\]

Now,

\[
j \cdot j = j^2 = -1,
\]

\[
\therefore Q_\theta(t) = \frac{1}{2\pi^2 j^2} \int_{-\infty}^{\infty} 2\pi j \omega \hat{P}_\theta(\omega) \cdot -\pi j \text{sgn}(\omega) e^{2\pi j \omega t} d\omega. \quad (A8)
\]

Based on the Fourier convolution theorem, \(Q_\theta(t)\) (Eq. A10) can be written as a convolution in the time domain:

\[
Q_\theta(t) = \frac{1}{2\pi^2} \text{Inverse Fourier Transform of} \left(2\pi j \omega \hat{P}_\theta(\omega)\right)^* \\
\text{Inverse Fourier Transform of} \left(- \pi j \text{sgn}(\omega)\right)
\]

Following is some theory to allow the subsequent transformations.

The Hilbert transform of \(\frac{\partial (P_\theta(t))}{\partial t}\) is defined as

\[
H \left(\frac{\partial (P_\theta(t))}{\partial t}\right) = \frac{1}{\pi \cdot t} \ast \left[\frac{\partial (P_\theta(t))}{\partial t}\right]. \quad (A12)
\]

The Fourier derivative theorem states that, if \(f(x)\) has the Fourier transform \(F(s)\) then the derivative \(f'(s)\) has the Fourier transform \(2\pi j s F(s)\). Thus, in our case, the

\[
\text{Fourier transform of} \quad \frac{\partial (P_\theta(t))}{\partial t} = 2\pi j \omega \hat{P}_\theta(\omega), \quad (A13)
\]

which represents the first term in Eq. A12. For the second term, we know that

\[
\text{Fourier Transform of} \quad \frac{1}{t} = -\pi j \text{sgn}(\omega). \quad (A14)
\]
Substituting Equations A13 and A14 in Equation A12, we get

\[ Q_\theta(t) = \frac{1}{2\pi} \left[ \frac{\partial(P_\theta(t))}{\partial t} \right] * \left[ \frac{1}{t} \right] \tag{A15} \]

\[ Q_\theta(t) = \frac{1}{2\pi t} \frac{\partial(P_\theta(t))}{\partial t} \tag{A16} \]

\[ Q_\theta(t) = \frac{1}{2\pi} \text{Hilbert transform of} \left[ \frac{\partial(P_\theta(t))}{\partial t} \right] \tag{A17} \]

\( Q_\theta(t) \), which is the result of convolving the projection \( P_\theta(t) \) with the ramp filter in the time domain, represents, in essence, the Hilbert transform of the differentiated projection \( P_\theta(t) \).

The reconstruction equation for the parallel-beam case now is

\[ f(x, y) = \frac{1}{2\pi} \int_0^\pi \int_{-t_m}^{t_m} \frac{\partial(P_\theta(t))}{\partial t} \frac{1}{t-t'} dt d\theta. \tag{A18} \]

or

\[ f(x, y) = \frac{1}{2\pi^2} \int_0^\pi \left[ \frac{\partial(P_\theta(t))}{\partial t} \right] * \left[ \frac{1}{t} \right] d\theta. \tag{A19} \]

2. Fan-Beam geometry

Considering the parallel-beam object function (Eq.A1) for data collection over \( 2\pi \), we obtain

\[ f(x, y) = \frac{1}{2} \int_0^{2\pi} \int_{-t_m}^{t_m} P_\theta(t)h(t' - t) dt d\theta \tag{A20} \]

The integral over \( 2\pi \) provides duplication of the data set, thus we apply a correction factor \( \frac{1}{2} \).

Equation A18 then is modified accordingly to

\[ f(x, y) = \frac{1}{4\pi^2} \int_0^{2\pi} \int_{-t_m}^{t_m} \frac{\partial(P_\theta(t))}{\partial t} \frac{1}{t-t'} dt d\theta \tag{A21} \]
We will now transform the coordinates \((t, \theta)\) of the parallel-beam system into the more suitable coordinates \((\beta, \gamma)\) of the fan-beam system (Figure A.1) with the following definitions.

- **S** = source; location defined by \(\beta\) relative to y-axis,
- **D** = source-to-isocentre distance,
- **L** = distance from source to point \((x, y)\),
- **\(\beta\)** = source angle,
- **\(\gamma\)** = angle of ray in fan,
- **\(\gamma'\)** = angle of ray passing through point \((x, y)\),
- **\(\gamma_m\)** = half fan angle,
- **\((r, \varphi)\)** = location of \((x, y)\) in polar coordinates,
- **\(t\)** = distance of the projection ray from isocenter, and
- **\(\theta\)** = angle of \(t\) relative to x-axis.
The coordinate transformation is given by
\[ t = D \sin \gamma \quad \text{and} \quad \theta = \beta + \gamma \quad \text{(A22)} \]

By reordering the parallel-beam projection rays into the fan-beam geometry, we define
\[ P(\theta, t) = R(\beta, \gamma) \quad \text{(A23)} \]

with application of Eq. A22.

The following two transformations are obvious
\[ \pm t_m \rightarrow \pm \gamma_m \quad \text{(A24)} \]
\[ dt \, d\theta \rightarrow D \cos \gamma \, d\gamma \, d\beta, \quad \text{(A25)} \]

with \( D \cos \gamma \) representing the Jacobian.

The third transformation
\[ \left[ \frac{\partial (p_\theta(t))}{\partial t} \right] \rightarrow \frac{1}{D \cos \gamma} \left( \left[ \frac{\partial R(\beta, \gamma)}{\partial \gamma} \right] - \left[ \frac{\partial R(\beta, \gamma)}{\partial \beta} \right] \right) \quad \text{(A26)} \]

needs more explanation.

From Eq. A21 we need to transform \( \frac{\partial (p_\theta(t))}{\partial t} \) into the \((\beta, \gamma)\) coordinate system.

We start by differentiation using the chain rule:
\[ \left[ \frac{\partial (p_\theta(t))}{\partial t} \right] = \left[ \frac{\partial R(\beta, \gamma)}{\partial \gamma} \right] \left[ \frac{\partial \gamma}{\partial t} \right] + \left[ \frac{\partial R(\beta, \gamma)}{\partial \beta} \right] \left[ \frac{\partial \beta}{\partial t} \right]. \quad \text{(A27)} \]

From Eq. A22 we can write
\[ \sin \gamma = \frac{t}{D} \quad \text{(A28)} \]
\[ \gamma = \sin^{-1} \left( \frac{t}{D} \right). \quad \text{(A29)} \]

Differentiating Eq. A29 by \( t \), we get
\[ \frac{\partial \gamma}{\partial t} = \frac{1}{D \sqrt{\frac{d^2 - t^2}{D^2}}} \quad \text{(A30)} \]

From Eq. A22 we can write
\[ \beta = \theta - \gamma. \]  

(A31)

By substituting the value of \( \gamma \) from Eq. A29, we get

\[ \beta = \theta - \sin^{-1}\left(\frac{L}{D}\right). \]  

(A32)

Differentiating Eq. A32 by \( t \) results in

\[
\frac{\partial \beta}{\partial t} = \frac{-1}{D\sqrt{\frac{D^2 - t^2}{D^2}}}.
\]  

(A33)

Now substituting Eq. A30 and A33 in Eq. A27 we get

\[
\left[ \frac{\partial (P_\beta(t))}{\partial t} \right] = \frac{1}{D\sqrt{\frac{D^2 - t^2}{D^2}}} \left( \left[ \frac{\partial R(\beta, y)}{\partial y} \right] - \left[ \frac{\partial R(\beta, y)}{\partial \beta} \right] \right) \]  

(A33)

Further simplifying \( \frac{1}{D\sqrt{\frac{D^2 - t^2}{D^2}}} \) by substituting \( t = D \sin \gamma \) we get

\[
\frac{1}{D\sqrt{\frac{D^2 - t^2}{D^2}}} = \frac{1}{D\sqrt{\frac{D^2 - D^2 \sin^2 \gamma}{D^2}}} \]

\[
= \frac{1}{D\sqrt{1 - \sin^2 \gamma}} \]

(A34)

We know that \( \cos^2 \gamma + \sin^2 \gamma = 1 \)

\[ \therefore \] \[ \frac{1}{D\sqrt{\frac{D^2 - t^2}{D^2}}} = \frac{1}{D\sqrt{\cos^2 \gamma}} = \frac{1}{D\cos \gamma} \]

(A35)

Substituting Eq.A35 in A33, we get

\[
\left[ \frac{\partial (P_\beta(t))}{\partial t} \right] = \frac{1}{D\cos \gamma} \left( \left[ \frac{\partial R(\beta, y)}{\partial y} \right] - \left[ \frac{\partial R(\beta, y)}{\partial \beta} \right] \right) \]

(A36)

Now we will enter the transformations A24, A25 and A26 into Eq. A21:

\[
f(x, y) = \frac{1}{4\pi^2} \int_0^{2\pi} \int_{-\gamma_m}^{\gamma_m} \frac{1}{D\cos \gamma} \left( \left[ \frac{\partial R(\beta, y)}{\partial y} \right] - \left[ \frac{\partial R(\beta, y)}{\partial \beta} \right] \right) \frac{1}{D \sin \gamma - D \sin \beta} D\cos \gamma \ dy \ dB. \]  

(A37)

The \( D\cos \gamma \) terms cancel out.
In order to obtain a proper convolution kernel showing a difference in \( \gamma \) and not in \( \sin \gamma \), some further modifications are necessary.

\( D \sin \gamma \) identifies \( t' \) for the projection ray passing through point \((x,y)\) to be reconstructed. In polar coordinates, \((x,y)\) is defined by \((r,\phi)\). From Figure A.1, we can conclude that

\[
t' = r \cos (\theta - \phi) .
\]  
\[
(A38)
\]

Thus,

\[
D \sin \gamma' - D \sin \gamma = r \cos (\theta - \phi) - D \sin \gamma .
\]  
\[
(A39)
\]

Applying Eq. A22, we obtain

\[
D \sin \gamma' - D \sin \gamma = r \cos (\beta + \gamma - \phi) - D \sin \gamma
\]  
\[
= r \cos (\beta - \phi + \gamma) - D \sin \gamma .
\]  
\[
(A40)
\]

Using the cosine angle addition theorem \( \cos(\alpha + \gamma) = \cos \alpha \cos \gamma - \sin \alpha \sin \gamma \), we obtain

\[
D \sin \gamma' - D \sin \gamma = r \cos \beta \cos \gamma - r \sin (\beta - \phi) \sin \gamma - D \sin \gamma
\]  
\[
= r \cos (\beta - \phi) \cos \gamma - (r \sin (\beta - \phi) + D) \sin \gamma
\]  
\[
(A41)
\]

Referring to Figure A.1, we get

\[
r \cos (\beta - \phi) = L \sin \gamma' \text{ and } D + r \sin (\beta - \phi) = L \cos \gamma'
\]  
\[
(A43)
\]

Substituting Eq. A43 in Eq. A42 we get

\[
D \sin \gamma' - D \sin \gamma = L \sin \gamma' \cos \gamma - L \cos \gamma' \sin \gamma
\]  
\[
= L \sin (\gamma' - \gamma)
\]  
\[
(A44)
\]

Now applying Eq. A44 to Eq. A37, we obtain the final expression for the differentiated filtered back-projection

\[
f(x, y) = \frac{1}{4\pi^2 L} \int_0^{2\pi} \int_{-\gamma_m}^{\gamma_m} \left( \frac{\partial R(\beta, \gamma)}{\partial \gamma} - \frac{\partial R(\beta, \gamma)}{\partial \beta} \right) \frac{1}{\sin(\gamma' - \gamma)} \, d\gamma \, d\beta ,
\]  
\[
(A45)
\]

or writing the inner integral as a convolution

\[
f(x, y) = \frac{1}{4\pi^2 L} \int_0^{2\pi} \left( \frac{\partial R(\beta, \gamma)}{\partial \gamma} - \frac{\partial R(\beta, \gamma)}{\partial \beta} \right) * \frac{1}{\sin \gamma} \, d\beta.
\]  
\[
(A46)
\]
References


11. Reconstruction and the phantom generation codes implemented by Sangeetha Alladi, Ph.D. candidate, Wright State University, Dayton, Ohio, USA.


