An Area Based Fan Beam Projection Model

Richard E. Steele  
*Georgia Southern University*

Jiehua Zhu  
*Georgia Southern University*

Follow this and additional works at: [https://digitalcommons.georgiasouthern.edu/honors-theses](https://digitalcommons.georgiasouthern.edu/honors-theses)

Part of the [Applied Mathematics Commons](https://digitalcommons.georgiasouthern.edu/applied-mathematics), [Bioimaging and Biomedical Optics Commons](https://digitalcommons.georgiasouthern.edu/bioimaging-and-biomedical-optics), and the [Medicine and Health Sciences Commons](https://digitalcommons.georgiasouthern.edu/medicine-and-health-sciences)

**Recommended Citation**  
[https://digitalcommons.georgiasouthern.edu/honors-theses/321](https://digitalcommons.georgiasouthern.edu/honors-theses/321)

This thesis (open access) is brought to you for free and open access by the Student Research Papers at Digital Commons@Georgia Southern. It has been accepted for inclusion in University Honors Program Theses by an authorized administrator of Digital Commons@Georgia Southern. For more information, please contact digitalcommons@georgiasouthern.edu.
An Area Based Fan Beam Projection Model

An Honors Thesis submitted in partial fulfillment of the requirements for Honors in Mathematics.

Richard Steele

Under the mentorship of Jiehua Zhu

Georgia Southern University

Department of Mathematical Sciences

April 2, 2018

Abstract

Area based projection models for computed tomography mitigate raw data errors by treating X-Rays as beams, whereas traditional line based projection models treat an X-Ray like a line, thus generating significant error. In an existing area based fan beam projection model, a rotation matrix, $Q$, simulates the rotation of the emitter detector pair to reduce computational load, but this introduces approximations by using an approximated rotation matrix. We eliminate approximations by deriving an exact formula for the entries of $Q$. Using a rotation of axes and by considering the neighboring cells’ contributions to the area, the result has formulations for the exact calculation of the matrix $Q$. Thus, approximations are phased out, and error in projection data is minimized for image reconstruction.

Thesis Mentor: __________________

Dr. Jiehua Zhu

Thesis Mentor: __________________

Dr. Steven Engel
Acknowledgements

I wish to acknowledge and give a special thanks to my advisor, Dr. Jiehua Zhu, for her generous support and guidance over the past three years. I would also like to thank the College of Undergraduate Research for funding our project, the University Honors Program for supporting our research, and the Department of Mathematical Sciences for their underlying support throughout the whole project. Finally, I would like to thank my girlfriend Taylor Coleman and my family for their unwavering support throughout my entire college career.
1 Introduction

Computed tomography (CT) refers to image reconstruction from projection data. Projection data is acquired at detectors when a high energy wave passes through a subject and is collected on the other side. This high energy wave is usually an X-ray and will be an X-ray in our case. CT scans are a crucial tool in a doctor’s diagnostic toolbox. In a head trauma situation, doctors do not have time to wait for a brain MRI, and a head X-ray doesn’t show enough detail. This is one area where the CT is invaluable. I once read in an emergency room "time=brain" on a board discussing stroke protocol and other head injuries. There will always be a need for fast and accurate imaging of the brain; CT can give this to us.

To formulate an image from projection data, algebraic reconstruction techniques are commonly used. Algebraic reconstruction requires us to solve the linear system

\[ Ax = b, \]

where \( A \) is the weight matrix, \( x \) is the image, and \( b \) is the projection data. We will discuss various algebraic reconstruction methods in later chapters.

CT has three unique scanning geometries: fan beam, parallel beam, and cone beam. These geometries describe how the emitters and detectors are oriented. In parallel beam, there are the same amount of emitters as detectors and they are parallel to each other like two opposing walls in a square room. In cone and fan beam geometries, there is only one emitter and the detectors orientation determine one or the other. In the fan beam geometry, the detectors are in a fan-shaped orientation in the same plane as the emitter. I like to picture the fan beam like a plastic garden rake, where the detectors have an arc-like shape like the teeth on the rake. The cone beam geometry shares traits with the
fan beam geometry. However, in the cone beam geometry, the detectors are not in the same plane as the emitter. Imagine an ice cream cone that is pointed at the bottom and not flat. This is how the cone beam geometry looks. The point at the bottom is the emitter, and the outside rim at the top of the cone is where the detectors are located. In our case, we will only be concerned with the fan beam geometry.

Lastly, there are two different projection models, line based and area based. Line based projection models are very classical and have been used the longest. Line based models work very well, but they simplify the X-ray into a line which generates significant error but reduces the computational load. For this reason, line based projection models are reasonably accurate. An area based projection model comes from the fact that detectors actually have a finite width, hence X-rays are beams that occupy fractional areas of an object as they pass through it. Thus, we use the area based projection model to get closer to reality by reducing approximations. Figure 1 shows the difference between the area based fan beam model and the line based fan beam model. The line based projection models have been studied extensively so now we focus on the area based. The area based parallel beam projection model has already been completed [5]. Thus we will be focusing on the area based fan beam projection model in this project.

Figure 1: The area based model (left) and line based model (right)
2 Current Area Based Fan Beam Projection Model

An Area Based Fan Beam Projection Model is proposed in [2]. This model works very well, but it doesn’t hold up under scrutiny because of an approximation that it contains. This proposed model starts by partitioning the subject into an $N \times N$ grid, centered on the subject. This partition of the subject is the image matrix, $Img$. After this grid is created, the upper right hand corner of each cell is labeled as coordinates $(i, j)$. The image value at this cell is the $(i, j)$-th entry of the image matrix, $A$. The fractional areas created by the intersection of the beams with the grid are then calculated. The computation is first done with no rotation. Figure 2 shows the orientation of the emitter and detector. Note that the emitter lies on the $45^\circ$ line of the original axes. This is to simplify the area calculations because the intersections above the $45^\circ$ line are mirror cases to the intersections below the $45^\circ$ line.

![Figure 2: Area Based Fan Beam Scanning Geometry](image)

After the areas are calculated for $\theta = 0$, the subject is then rotated to preserve the symmetry about the $45^\circ$ line. This rotation does precisely the same thing as rotating
the emitter and detector about the subject. For this rotation to take place, a rotation matrix, $Q$, is introduced to perform this rotation. Recall that $x$ is the image matrix and hence what we are rotating, thus we can define the rotated image vector

$$x_\theta := Q_\theta x. \quad (2)$$

Let $a^n$ be one of the original rows of $A \in \mathbb{R}^{M \times N^2}$ for when $\theta = 0$. Then there exists an $n$ for each $r \leq M$ such that

$$a^n x_\theta = a^n Q_\theta x = b_r. \quad (3)$$

Suppose a clockwise rotation of angle $\theta$ is applied to each of the cells, and $(i', j')$ are coordinates of the cell $(i, j)$ after the rotation:

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} i - N/2 \\ j - N/2 \end{bmatrix} + \begin{bmatrix} N/2 \\ N/2 \end{bmatrix}. \quad (4)$$

Once the new coordinates $(i', j')$ are obtained, there is a very low probability that they will be integer coordinates. However, we cannot define non-integer coordinates on the grid, so we must introduce the floor and ceiling functions. Let $x \in \mathbb{R}$ then the floor function, denoted by $\text{floor}(x) \equiv \lfloor x \rfloor \equiv \lfloor x \rfloor$, takes only the integer part of $x$. However, the ceiling function, denoted $\text{ceiling}(x) \equiv \lceil x \rceil \equiv \lceil x \rceil$, rounds $x$ up to the next integer. We will use the shorthand notations $\lfloor i' \rfloor = [i']$, $\lfloor i' \rfloor = [i']$, $\lfloor j' \rfloor = [j']$, and $\lceil j' \rceil = [j']$, throughout the paper.

Figure 3 shows in general how a rotated cell would land on the original grid. The fractional areas labeled one through four are needed to calculate $Q$. Since these shapes can’t easily be generalized, an approximation to make each of the fractional areas rectangular is made in the existing area based fan beam projection model as depicted in Figure 4. From this approximation, the entries in $Q$ can easily be calculated. However, this
approximation is the major pitfall of the current area based fan beam projection model. As previously noted, the current model works very well, but room still remains for an exact model to compute $Q$.

![Figure 3: General case of intersection](image1)

![Figure 4: Approximated position](image2)

3 Exact Rotation Matrix for Area Based Fan Beam Projection Model

3.1 Method for Constructing $Q$

Finding an exact model for $Q$ will require the exact area of each intersection region the rotated cell makes with the original grid. Figure 5 shows the areas needed to be calculated. The numbers with square boxes reference the cell numbers on the original grid and the numbers that are circled reference the areas of intersection. For example, area #3 will be the entry in $Q$ corresponding to cell #6 and area #1 will be the entry in $Q$ corresponding to cell #5. There can be intersection regions in up to five or six neighboring cells, so we need a systematic approach to calculate these areas. The derivation of the exact model of $Q$ now requires us to track all four vertices of the rotated cell $(i', j')$ instead of just one vertex. We will also need the equations of the lines passing through adjacent
vertices. Now after a rotation of angle $\theta$ we will label the four vertices $(i, j) \rightarrow (i', j')$, $(i-1, j) \rightarrow (i'_2, j'_2)$, $(i, j-1) \rightarrow (i'_3, j'_3)$, $(i-1, j-1) \rightarrow (i'_4, j'_4)$. We will correspondingly label the lines between adjacent vertices $y_i$, $i = 1, 2, 3, 4$, and seek to find the equations for these lines. Figure 5 depicts the vertices and lines.

![Figure 5: Vertices and lines of rotated cell](image)

**Lemma 1.** As labeled in 5 the line passing through neighboring vertices is denoted as $y_i$ and the slope as $m_i$ where $i = 1, 2, 3, 4$. Then

$$m_1 = m_4 = -\tan \theta,$$

$$m_2 = m_3 = \cot \theta.$$
Proof. By notation and formula for slope,

\[ m_1 = \frac{j'_2 - j'_1}{i'_2 - i'_1}. \]

Thus by equation (4),

\[
m_1 = \frac{-(i - 1 - \frac{N}{2}) \sin \theta + (j - \frac{N}{2}) \cos \theta + \frac{N}{2} - (- (i - \frac{N}{2}) \sin \theta + (j - \frac{N}{2}) \cos \theta + \frac{N}{2})}{(i - 1 - \frac{N}{2}) \cos \theta + (j - \frac{N}{2}) \sin \theta + \frac{N}{2} - [(i - \frac{N}{2}) \cos \theta + (j - \frac{N}{2}) \sin \theta + \frac{N}{2}]].
\]

Simplification indeed yields

\[ m_1 = -\frac{\sin \theta}{\cos \theta} = -\tan \theta. \]

Since \( y_2 \) is perpendicular to \( y_1 \), its slope is \(-\frac{1}{m_1}\). Hence,

\[ m_2 = \cot \theta. \]

We can also see \( y_4 \) is parallel to \( y_1 \) thus

\[ m_4 = m_1. \]

Likewise \( y_3 \) is parallel to \( y_2 \) and therefore

\[ m_3 = m_2. \]

By Lemma 1, the equations of the lines are:

\[ y_1 = -(x - i'_1) \tan \theta + j'_1, \quad (7) \]

\[ y_2 = (x - i'_1) \cot \theta + j'_1, \quad (8) \]
\[ y_3 = (x - i'_4)\cot\theta + j'_4, \] (9)

\[ y_4 = -(x - i'_4)\tan\theta + j'_4. \] (10)

3.2 Algorithm

Our goal to compute \( Q \) exactly still remains and we will dive into doing so. We notice there are eight possible intersections the rotated cell can make with the original grid. Knowing this we next realize these shapes have no apparent symmetry and are all arbitrary polygons. To further deepen our understanding of the shapes created and how to compute the areas of these shapes, we first categorize intersections into four different parent cases. Each of these four cases has many sub-cases but understanding and identifying the parent case is the first step in computing the area. After determining the case, the points of intersection are stored in a matrix whose columns correspond to cells on the original grid. The points are then extracted from the matrix and the areas of intersection by the rotated cell on the original grid are calculated by our area algorithm. These areas make up the entries in \( Q \).

3.2.1 Cases of Intersection

Utilizing the information from the four lines, we can define the eight possible intersection points: \( p_1, p_2, \ldots, p_8 \). Figure (6) shows the location of the eight points.
Figure 6: Intersection points of cell rotated by angle $\theta$

We define the cases by the location of the vertices of the rotated cell.

**Case 1**: $clj'_2 = clj'_3 + 2, clj'_1 = clj'_4 + 2$.

Case 1, as shown in Figure 7, occurs when opposing vertices each have two lines between them. Case 1 is the most general case and least likely.
Case 2: \( cl_j^2 = cl_j^3 + 2, cl_i^1 = cl_i^4 + 1 \).

Case 2, as shown in Figure 8, occurs when the vertically opposing vertices have two horizontal lines between them but the horizontally opposing vertices only have one vertical line between them.

Case 3: \( cl_j^2 = cl_j^3 + 1, cl_i^1 = cl_i^4 + 2 \).

Case 3, as shown in Figure 9, occurs when the vertically opposing vertices have only one horizontal line between them and the horizontally opposing vertices have two vertical
lines between them. Case 3 is the exact opposite of case 2.

Case 4 occurs when there is only one vertical line between horizontally opposing vertices and there is only one horizontal line between vertically opposing vertices. Figure 10 shows the most abundant intersection case.

$$\text{Case 4: } clj_2' = clj_3' + 1, cli_1' = cli_4' + 1.$$
3.2.2 Storage of Intersection Points

Once a case is determined, we then know which intersection points occur and store them in a matrix, \( P \in \mathbb{R}^{8 \times 18} \). There are eight rows in \( P \) due to the fact that there are eight possible intersection points for each cell on the original grid that is intersected by the rotated cell. The 18 columns correspond to the nine neighboring cells surrounding the rotated cell. For each of the nine cells, the intersection points have an \( x \) and \( y \) coordinate value, thus 18 columns. Figure 11 explains the nine neighboring cells around the rotated cell. For example columns 1 and 2 of \( P \) correspond to the \( x \) and \( y \) values, respectively, of the intersection points the rotated cell makes with Cell 1, and the submatrix formed by Columns 1 and 2 is denoted by \( P_1 \).

![Figure 11: Rotated cell on original grid, where the numbers correspond to the nine neighboring cells](image)

3.2.3 Area of Intersection Regions

Once the matrix \( P \) is created for a cell \((i, j)\) and angle \( \theta \) to store intersection points of the rotated cells and the nine relevant cells, \( P \) is divided into 9 submatrices \( P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9 \). The submatrix \( P_k \) \((k = 1, 2, ..., 9)\) consists of the \((2k - 1) - th\) and
2k – th columns of P, containing the coordinates of vertices of the intersection region the rotated cell makes with cell k and thus defining this region.

**Theorem 1.** Let $k = 1, 2, \ldots, 9$ and $l_k = \text{length}(P_k)$, then the area of the region defined in matrix $P_k$ is

$$
\text{Area}_{P_k} = \frac{1}{2} \sum_{t=1}^{l_k-2} \left| \det \begin{pmatrix}
P_k(t+1, 1) - P_k(1, 1) & P_k(t+1, 2) - P_k(1, 2) \\
P_k(t+2, 1) - P_k(1, 1) & P_k(t+2, 2) - P_k(1, 2)
\end{pmatrix} \right|.
$$

**Proof.** Recall that if $a$ and $b$ are two non-colinear vectors in the same plane, and $c = a - b$, the area enclosed by vectors $a$, $b$, and $c$ is $\frac{1}{2}|a \times b|$. Let the points $p_\alpha$, $\alpha = 1, \ldots, 8$, have $x$ and $y$ coordinates $(p_{\alpha x}, p_{\alpha y})$. To calculate the area of an intersection region we break down the region into triangles as in Figure 12.

![Figure 12: Vectors to calculate area of intersection region, where $aT$ is the area of the corresponding triangle](image-url)
We define vector $p_\beta p_\gamma = (p_{\gamma x} - p_{\beta x}, p_{\gamma y} - p_{\beta y})$ as the vector between points $p_\beta$ and $p_\gamma$, where $\gamma \neq \beta$ and $\gamma, \beta = 1, \ldots, 8$. As an example we will compute the area for $P_5$, which consists of $aT_1, aT_2, aT_3, aT_4, aT_5, aT_6$.

By Figure 12 the area $aT_1$, 

$$aT_1 = \frac{1}{2} |p_1 p_2 \times p_1 p_3| = \frac{1}{2} \left| \det \begin{bmatrix} (p_{2x} - p_{1x}) & (p_{2y} - p_{1y}) \\ (p_{3x} - p_{1x}) & (p_{3y} - p_{1y}) \end{bmatrix} \right|,$$

$$= \frac{1}{2} \left| (p_{2x} - p_{1x})(p_{3y} - p_{1y}) - (p_{2y} - p_{1y})(p_{3x} - p_{1x}) \right|.$$ 

The second triangle will have area $aT_2$, 

$$aT_2 = \frac{1}{2} |p_1 p_3 \times p_1 p_4| = \frac{1}{2} \left| \det \begin{bmatrix} (p_{3x} - p_{1x}) & (p_{3y} - p_{1y}) \\ (p_{4x} - p_{1x}) & (p_{4y} - p_{1y}) \end{bmatrix} \right|,$$

$$= \frac{1}{2} \left| (p_{3x} - p_{1x})(p_{4y} - p_{1y}) - (p_{3y} - p_{1y})(p_{4x} - p_{1x}) \right|.$$  

$$:$$

The area of the sixth triangle, $aT_6$, 

$$aT_6 = \frac{1}{2} |p_1 p_7 \times p_1 p_8| = \frac{1}{2} \left| \det \begin{bmatrix} (p_{7x} - p_{1x}) & (p_{7y} - p_{1y}) \\ (p_{8x} - p_{1x}) & (p_{8y} - p_{1y}) \end{bmatrix} \right|,$$

$$= \frac{1}{2} \left| (p_{7x} - p_{1x})(p_{8y} - p_{1y}) - (p_{7y} - p_{1y})(p_{8x} - p_{1x}) \right|.$$ 

From the figure above, we can notice the total area of the intersection of $(i', j')$ with Cell $(cl_1', 1, cl_1')$ is the sum of the individual areas of triangles $aT_1, \ldots, aT_6$. In other words,
from notation previously mentioned,

\[ \text{Area}_{P_5} = \frac{1}{2} \sum_{t=1}^{t=6} \left| \left[ (P_5(t+1,x) - P_5(1,x))(P_5(t+2,y) - P_5(1,y)) - (P_5(t+1,y) - P_5(1,y))(P_5(t+2,x) - P_5(1,x)) \right] \right|, \]

\[ = \frac{1}{2} \left| \text{det} \begin{bmatrix} P_5(t+1,x) - P_5(1,x) & P_5(t+1,y) - P_5(1,y) \\ P_5(t+2,x) - P_5(1,x) & P_5(t+2,y) - P_5(1,y) \end{bmatrix} \right|. \]

Since \( \text{Area}_{P_5} \) is not unique and was only shown for simplicity, this algorithm will work for an intersection in any cell. Thus generally we have,

\[ \text{Area}_{P_k} = \frac{1}{2} \sum_{t=1}^{t_k-2} \left| \text{det} \begin{bmatrix} P_k(t+1,1) - P_k(1,1) & P_k(t+1,2) - P_k(1,2) \\ P_k(t+2,1) - P_k(1,1) & P_k(t+2,2) - P_k(1,2) \end{bmatrix} \right|. \quad (11) \]

\[ \square \]

3.2.4 Matrix of Rotation

With these areas determined we can now update the corresponding entries in \( Q \). Note that we are using MATLAB matrix notation, and that in MATLAB entry \((i,j)\) of a matrix is in the \(i^{th}\) row and \(j^{th}\) column. The image matrix \( \text{Img} \) is reshaped, to a vector \( x \) by setting \( x((j - 1)N + i) = \text{Img}(i,j) \). So for a given pixel \((i,j)\), \( Q_\theta \) is updated based on figure 13.
Based on figure 13, the matrix $P_1$ will be empty. However, if $P_1$ were not empty, the area intersected in cell #1 or cell $(cli'_1 - 2, clj'_1 - 1)$ would correspond to the $((clj'_1 - 1) * N + (cli'_1 - 2), (j - 1) * N + i)^{th}$ entry in $Q$ based on the reshaping of the image matrix, $x((j - 1)N + i) = \text{Img}(i, j)$. Thus we have the following equation:

$$Q((clj'_1 - 2) * N + (cli'_1 - 2), (j - 1) * N + i) = \text{Area}_{P_1}. \quad (12)$$

Cell #2, in this example, yields a different result. The matrix $P_2$ will not be empty. In fact the matrix $P_2$ will contain six entries, contained in three rows. Each of the rows contains an $x$-coordinate in the first column and a $y$-coordinate in the second column. These coordinates make up the three vertices of the triangle intersecting cell $(cli'_1 - 2, clj'_1)$ or cell #2. Therefore we can see the area of this triangle is the $((clj'_1 - 1) * N + (cli'_1 - 2), (j - 1) * N + i)^{th}$ entry in $Q$ based on the reshaping of the image matrix, $x((j - 1)N + i) = \text{Img}(i, j)$. Thus we have the following equation:

$$Q((clj'_1 - 2) * N + (cli'_1 - 2), (j - 1) * N + i) = \text{Area}_{P_2}. \quad (13)$$
2), \((j - 1) \times N + i\)^{th} entry in \(Q\). Thus arriving at the second equation for updating \(Q\):

\[
Q((c_l j'_1 - 1) \times N + (c_l i'_1 - 2), (j - 1) \times N + i) = \text{Area}_{P_2}.
\]  \tag{13}

Cell #3 happens to be the same as cell #1 in this case, but this isn’t always the case. Likewise, if matrix \(P_3\) were not empty the area contained in cell \((c_l i'_1 - 2, c_l j'_1 + 1)\) would correspond to the \(((c_l j'_1) \times N + (c_l i'_1 - 2), (j - 1) \times N + i)^{th} entry in \(Q\). Arriving at the third equation for updating \(Q\):

\[
Q((c_l j'_1) \times N + (c_l i'_1 - 2), (j - 1) \times N + i) = \text{Area}_{P_3}.
\]  \tag{14}

The fourth cell, cell \(((c_l i'_1 - 1, c_l j'_1 - 1)\), has corresponding matrix \(P_4\) similar to \(P_2\), and the entries in \(Q\) corresponding to cell #4 is defined by the following equation:

\[
Q((c_l j'_1 - 2) \times N + (c_l i'_1 - 1), (j - 1) \times N + i) = \text{Area}_{P_4}.
\]  \tag{15}

The way \(Q\) was derived, was derived in such a way that matrix \(P_5\) will never be empty. However, \(P_5\) may not always have all eight rows full, i.e. having eight vertices. The rows and columns in \(P_5\) are arranged in the same manner as described for \(P_2\). The area from the octagon intersecting cell \((c_l i'_1 - 1, c_l j'_1)\) will be the \(((c_l j'_1 - 1) \times N + (c_l i'_1 - 1), (j - 1) \times N + i)^{th} entry in \(Q\). Thus we have the equation:

\[
Q((c_l j'_1 - 1) \times N + (c_l i'_1 - 1), (j - 1) \times N + i) = \text{Area}_{P_5}.
\]  \tag{16}

The areas of intersection for cells #6, #7, #8, and #9 follow the same pattern as the previously mentioned cells. Hence, we have the last of the equations to update \(Q\). Figure 13 can be referenced to obtain the index of the cells.

\[
Q((c_l j'_1) \times N + (c_l i'_1 - 1), (j - 1) \times N + i) = \text{Area}_{P_6}
\]  \tag{17}
We have now given sufficient proof for the following theorem,

**Theorem 2.** For any positive integer \( r \leq M \), there exists a positive integer \( n \) which depends on \( k \), such that

\[
a^n x_0 = a^n Q_\theta x = b_r,
\]

where \( Q_\theta \) can be approximately constructed as

\[
Q((clj'_1 - 2) \ast N + (cli'_1), (j - 1) \ast N + i) = \text{Area}_{P_7} \\
Q((clj'_1 - 1) \ast N + (cli'_1), (j - 1) \ast N + i) = \text{Area}_{P_8} \\
Q((clj'_1) \ast N + (cli'_1), (j - 1) \ast N + i) = \text{Area}_{P_9}
\]

for a given pixel \((i, j)\), where \( \text{Area}_{P_k} \) is given in (11).
4 Numerical Results

To verify the improved area based fan beam projection model with the exact rotation matrix, the algorithm was implemented in Matlab for numerical experiments. The projection data of the well known Shepp-Logan phantom [2] and clinical cardiac image [4] is computed with the proposed model. The images are reconstructed from the projection data using two reconstruction algorithms in order to verify the projection model. Two reconstruction algorithms are used to solve system (1). One is Algebraic Reconstruction Technique (ART) [2] and the other is Block Cyclic Projection Compressed Sensing algorithm (BCPCS) [3]. The experimental parameters for CT scan simulation are listen in Table 1 and the tests were performed on a MacBook Pro (16GB, 2.5GHz CPU).

The original Shepp-Logan and cardiac phantoms can be seen in the first (left most) column of Figures 14 & 15. For both phantoms an image size of $256 \times 256$ was used. This image size along with the experimental parameters generated $A$, from system (1), a size of $20592 \times 65536$. Thus we can see $A$ is very large and extremely underdetermined. For this reason, iterative methods are appropriate for solving the system. In both Figures 14 & 15 the ART reconstruction can be seen in the middle and BCPCS on the right.

<table>
<thead>
<tr>
<th>Image size</th>
<th># of detectors</th>
<th>$\phi$</th>
<th># of views</th>
<th>$d$</th>
<th>$\gamma$</th>
<th>Max Iterations</th>
<th>$\varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$256 \times 256$</td>
<td>287</td>
<td>5</td>
<td>72</td>
<td>50</td>
<td>$\pi/3$</td>
<td>100</td>
<td>$10^{-6}$</td>
</tr>
</tbody>
</table>
As verified by the images, we were able to reconstruct reasonable approximations to the test images using the proposed area based projection model for fan beam geometry. This can be further verified in numerical testing. Tables 2 & 3 show the numerical results as well as time to compute $A$ and the reconstruction. For both test phantoms, the BCPCS yielded better numerical results as well as faster run time.

Table 2: Projection Model Data

<table>
<thead>
<tr>
<th>Image</th>
<th>Size of $A$</th>
<th>Time to compute $A$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shepp-Logan</td>
<td>20592 × 65536</td>
<td>1422.31</td>
</tr>
<tr>
<td>Cardiac</td>
<td>20592 × 65536</td>
<td>1022.04</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Run Time(s)</td>
<td>$k$</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------</td>
<td>-----</td>
</tr>
<tr>
<td>ART Shepp-Logan</td>
<td>1839.27</td>
<td>100</td>
</tr>
<tr>
<td>BCPCS Shepp-Logan</td>
<td>1356.98</td>
<td>100</td>
</tr>
<tr>
<td>ART Cardiac</td>
<td>1643.69</td>
<td>100</td>
</tr>
<tr>
<td>BCPCS Cardiac</td>
<td>1336.76</td>
<td>100</td>
</tr>
</tbody>
</table>

## 5 Conclusion

Area based projection models provide superior error reduction in raw data over their line based counterparts. The current area based fan beam projection model is near completion, but lacking an exact method for computing $Q$. The main contribution of this project to the current area based fan beam projection model is an exact rotation matrix for simulating the rotation of the emitter and detector pair. The improved area based fan beam projection model with exact rotation matrix is verified by reconstruction of Shepp-Logan phantom and a cardiac image by ART and BCPCS algorithms. Future work entails further testing and optimization of the current code. Other work we would like to conduct is using many more reconstruction methods to further verify our model.
6 Bibliography


