5-2009

Iterative transmission image reconstruction for the DPET positron emission tomograph

Mark Wayne Lenox
University of Tennessee

To the Graduate Council:

I am submitting herewith a dissertation written by Mark Wayne Lenox entitled "Iterative transmission image reconstruction for the DPET positron emission tomograph." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Science.

Jens Gregor, Major Professor

We have read this dissertation and recommend its acceptance:

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Mark Wayne Lenox entitled "Iterative Transmission Image Reconstruction for the DPET Positron Emission Tomograph" I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Science.

Jens Gregor, Major Professor

We have read this dissertation
and recommend its acceptance:

Michael E. Casey

Michael G. Thomason

David W. Townsend

Lynne E. Parker

Jonathan S. Wall

Accepted for the Council:

Carolyn R. Hodges, Vice Provost and
Dean of the Graduate School

(Original signatures are on file with official student records.)

Iterative Transmission Image Reconstruction for the DPET Positron Emission Tomograph

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Mark Wayne Lenox
May 2009

Acknowledgements

This dissertation is dedicated to my wife, Katey, and my two children, Sarah and Scott. Neither of whom were even born the last time I was in graduate school.

This work was made possible, in a large part, through the generosity of Ronald Nutt and the Nutt Graduate Fellowship for Image Formation.

Abstract

Positron emission tomography (PET) systems use transmission imaging to compensate for attenuation. One commercial example of this approach is the Siemens Inveon Dedicated PET (DPET), a 120mm bore system dedicated to the study of small animals. DPET transmission images are currently reconstructed using single slice rebinning followed by filtered backprojection. Single slice rebinning attributes the attenuation associated with an oblique line integral to the direct midplane intersected thereby. This leads to position-dependent axial blurring, especially for large diameter animals, and objects with abrupt axial changes in diameter. The mathematics underlying filtered backprojection are based on assumptions that are not met by the scanner, including but not limited to data being sampled in a uniform fashion. These limitations can be alleviated by an iterative algorithm if the associated system model is made to match the physical set-up. The downside is typically viewed as a potentially prohibitive increase in the computational cost. In this dissertation, we report on the implementation and use of Simultaneous Iterative Reconstruction Technique (SIRT) (a weighted least-squares solver) for transmission imaging on the DPET. We provide experimental evidence regarding the improvement in transmission image quality. We also show that these new, higher quality images can be computed in less than two minutes on the existing DPET host computer thus making the approach practical. Computational speed is gained both algorithmically through relaxation and use of ordered subsets and implementation-wise through vector based arithmetic and multi-core program execution.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and General Information

1.0 Introduction

Positron Emission Tomography (PET) is used to measure concentrations of chemical compounds that have been tagged with a positron emitting isotope within live tissue. It is based on the properties of positrons and their interaction with electrons [Phelps, 1975]. There are many factors that can affect a PET scan, not the least of which is the density of the subject to be scanned. Techniques have been developed to compensate for these effects.

One correction that is required for valid quantification is attenuation correction. This correction requires either estimation or measurement of the attenuation coefficients of the object being scanned, then it applies those coefficients to the emission PET information to compensate for losses [Bailey, 2005]. It is true that it takes relatively large errors in an attenuation map to generate significant error in the resulting PET image due to the line integral nature of the correction. However, a more accurate measure is always preferable if it is possible to do so in an efficient way. Most modern clinical PET scanners are paired with Computed Tomography (CT) systems [Townsend, 2001] [Beyer, 2000] that can provide attenuation maps with both high resolution and contrast [Kinahan, 1998] in addition to the anatomical information they normally provide. Systems that do not have this advanced capability, instead rely on transmission scanning with an encapsulated source as a substitute [Huesman, 1988]. Using this method, a radioactive source is moved around the outside edge of the field of view, casting a field of radiation through the subject and against the PET detectors allowing the measurement of the attenuation of the field. This configuration adds a mechanism to hold the source, but does not add any more radiation detection hardware than is already present for PET. This simplicity makes it cost efficient, but no substitute for CT in terms of anatomical information. Examples of this configuration can be seen on several commercial PET systems: The Philips MOSAIC [Hichwa, 2004], Siemens Inveon DPET [Kemp, 2009], and Siemens/CTI HRRT [Wienhard, 2002]. One particular example of this configuration is the Siemens Inveon Dedicated PET (DPET). This design uses a single cone-beam collimated source mounted inside one edge of the axial field of view, combined with bed motion to measure the subject [Beach, 2004]. Due to the limited motion of the source, not all lines of response through the subject are directly measured, so they must be estimated in some way. The current system software uses the single slice rebinning (SSRB) technique to estimate 2D projections combined with filtered backprojection to perform this estimate. SSRB techniques approximate the values of direct planes with the values of oblique planes that pass through their midpoint [Daube-Witherspoon, 1987]. This produces axial blurring and inaccurate representation when the diameter of the object to be scanned is large or changes abruptly in the axial direction [Sossi, 1994]. More advanced techniques have been previously proposed for multi-row computed

tomography systems [Bruder, 2000] as an efficient solution to CT image reconstruction for narrow fan angles. The physical design of the transmission system suggests the use of more sophisticated iterative mathematical methods such as Simultaneous Iterative Reconstruction Technique (SIRT) [Gilbert, 1972][Kak, 2001] that do not suffer from these drawbacks, but those methods have not been utilized up to this point, perhaps due to their perceived long computational times. This dissertation describes the development of the system models used by SIRT, the implementation and optimization of SIRT itself for this application, and presents experimental results regarding the expected image quality and processing requirements.

1.1 PET Fundamentals

Positron Emission Tomography is an often used functional imaging modality for the study of disease and development of new drugs and therapies. It can be used to measure the concentration of various labeled compounds within live tissue, in real time [Cherry, 2003].

PET is an emission modality, meaning that the measured signal comes from within the subject by a tracer. The tracer is in the form of a positron emitting radioisotope that is bound to a molecule whose distribution or kinetics are to be measured. Common radioisotopes for PET include isotopes of Oxygen-15, Fluorine-18, Nitrogen-13, Carbon-11, and Iodine-124 that are subject to β^+ decay. As the radioisotope undergoes β^+ decay, a positron is generated. That positron moves some distance, the mean free path, that is dependent on its energy. Eventually, it annihilates with an electron. The result of that annihilation is the creation of two gamma photons of 511 keV energy each, that travel away from each other at essentially 180 degrees. The resulting gamma photons are always the product of the annihilation, and always have the energy 511 keV. Depending on the energy of the positron, the co-linearity of the resulting gammas may not be exactly 180 degrees, but for the purposes of this discussion, we assume that it is. Since they were created at exactly the same time, the detection of these photons within detectors mounted in a ring defines a line of response (LOR) if they were detected within a certain period of time to each other (Figure 1).

Problems occur during the measurement of the line of response if for some reason the gamma photons are scattered or lost completely. For example, if one of the gammas were scattered but still detected, and the direction changed, the line of response itself would be placed in the wrong location (see Figure 2). In addition, it is possible to completely lose an event that should otherwise be detected. Depending on the object in the field of view, at this energy, a loss can occur for one of two reasons: photoelectric absorption, and scatter [Cherry, 2004]. Photoelectric events in water at 511 keV are rare, and account for less than 0.1% of all losses, so we neglect that case. Only scatter has a significant contribution for this application. In this case, if one of the gammas were scattered outside the field of view, or its energy reduced to below the detection window, then the line of response would not be recorded at all as shown in Figure 3. The combination of these effects can cause parts of the imaging volume to record lower than

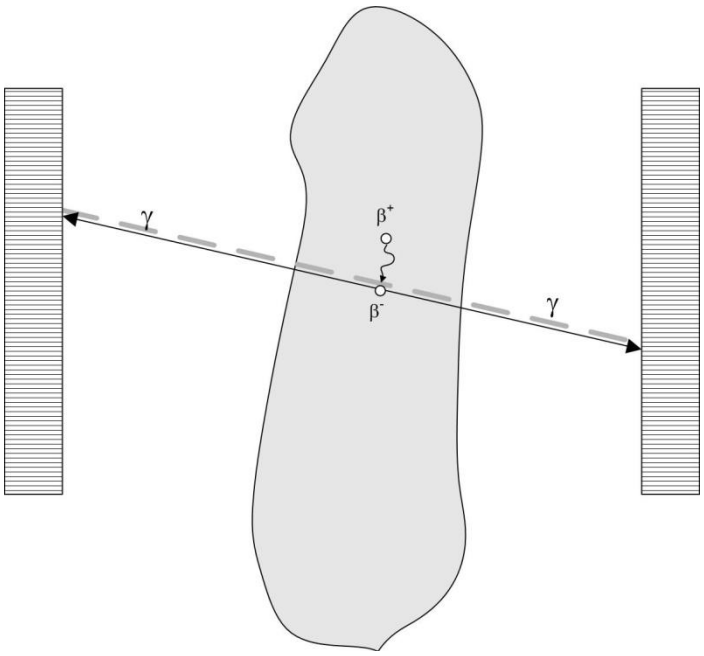

Figure 1. PET Line of Response.

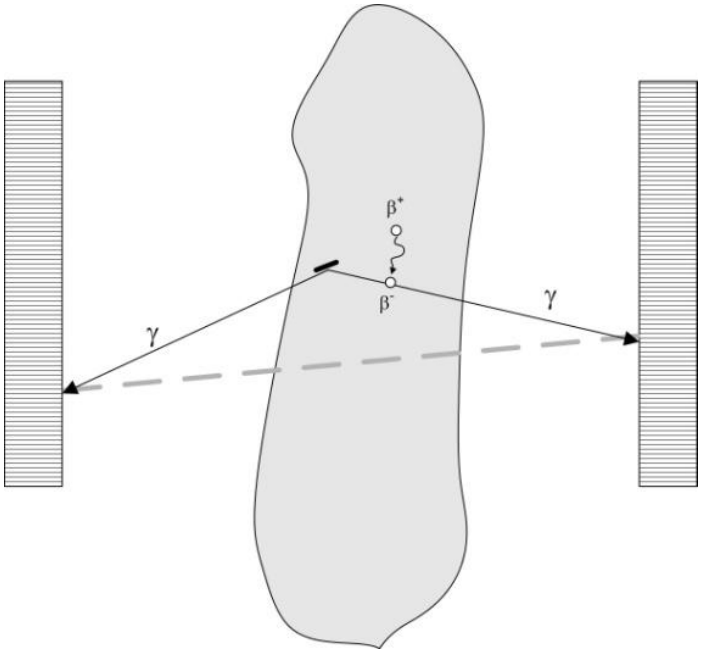

Figure 2. PET scattered event with erroneous line of response.

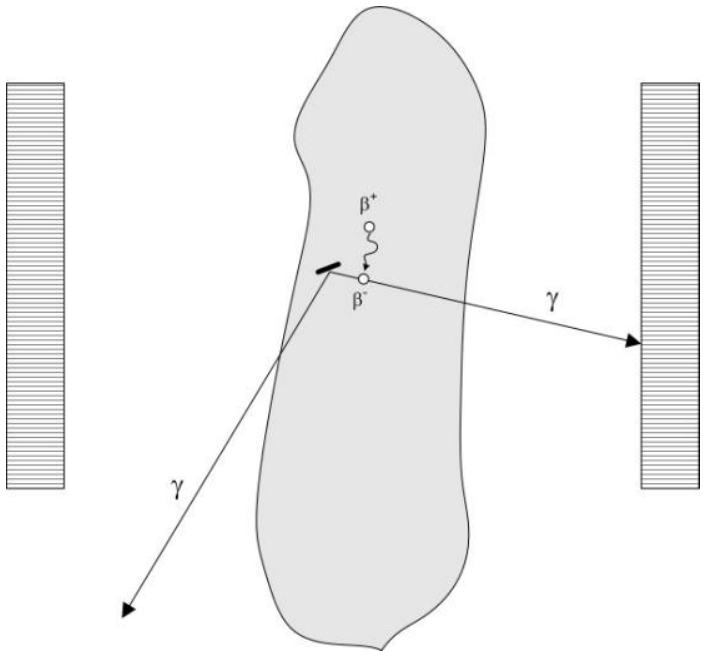

Figure 3. PET lost event with no line of response registered.

actual statistics based on the ability of the subject to scatter and otherwise prevent 511 keV gamma photons from being detected. We refer to the cumulative effect of all losses as attenuation.

1.2 Measuring Attenuation

For a narrow beam, the transmission (survival) of a photon through a thick absorber is exponentially related to the thickness and attenuation coefficient of the absorber [Cherry, 2003]. This is expressed through (1.1), where I_0 is the initial (blank) beam flux, I_1 is the resulting (measured) beam flux, u(x,y,z) is the distribution of attenuation coefficients in 3-dimensional space, and the line integral dictates the path that the beam travels.

$$I_1 = I_0 e^{\int_L u(x,y,z)dl} \tag{1.1}$$

It should be noted that *u(x,y,z)* is dependent on energy, however, in the methods presented here, energy is a constant value (a mono-energetic source is used). In this case, corrections to accommodate variable energy, like beam hardening, are not needed. In addition, the energy of the transmission scanning source does not always match the 511 keV energy gamma photons generated by positron annihilation, but this transformation is a known process [Knoess, 2003][Kinahan, 1998] and for simplicity will not be dealt with here.

Equation (1.1) is rewritten to express the attenuation coefficients in terms of the log normalized projection data.

$$\int_L u(x,y,z)dl = -log\left(\frac{I_1}{I_0}\right) \tag{1.2}$$

The right hand side of (1.2) is referred to as the log normalized projection information. Further simplification of (1.2) leads to a linear system of equations:

$$Ax = b \tag{1.3}$$

where vector b represents the log normalized ratio of measured flux to blank flux, vector x represents the unknown distribution of attenuation coefficients, and matrix A embodies a discrete representation of line integral L between the source and any given detector. We will henceforth refer to A as the system matrix. This system of equations is not solvable directly by inverting the system matrix A, but it can be solved with numerical methods appropriate for most linear systems, for example, weighted least squares [Stoer, 1980]. In such a calculation, the solution *x** can be expressed as:

$$x^* = argmin\|Ax - b\|_R^2 \tag{1.4}$$

where R denotes the weight matrix determined by the weighted least squares method chosen. Many algorithms exist that solve (1.4). We use the Simultaneous Iterative Reconstruction Technique (SIRT)[Kak, 2001]. In SIRT, the weight matrix R is defined as a diagonal matrix containing the inverse row sums of A.

Given that we want to solve for x*, we must first measure the log normalized projection information. Measuring the flux of gamma radiation presented by a source across the field of view (FOV), both with nothing in the FOV (I_0), and with the subject in the FOV (I_1), yields the ratio in equation (1.2).

By moving the source to different locations within the system, it is possible to directly measure the attenuation coefficients for every possible voxel within the system. It should be noted that depending on the size of the system and the composition of the subject, it can take considerable time to do this. Just the process of moving the source and spending enough time at every location to get a solid statistical measurement can take a long time depending on how many detectors are installed in the scanner.

There are several approaches that have been used to measure this information on PET systems. The use of a rod source that extends across the field of view [Huesman, 1988] was commonly used in early PET scanners like the ECAT EXACT family (including the HR, HR+, and Accel) [Adam, 1997] and the Advance [Kohlmyer, 2003]. This method is a coincidence based approach, requiring a positron emitter as the source, and uses the detector nearest the rod to electronically collimate the response. This allows the entire axial field of view to be sampled concurrently, however, it also suffers from high deadtime in the near detectors, and has not been designed into a new system in many years. Non-coincidence (singles) type transmission methods were created to deal with the detector deadtime issue at the expense of a more complex mechanism. One example of this approach is the partial ring PET system ECAT ART [Bailey, 1997]. In this system, the entire detector array rotated. A point source was moved axially through the field of view with a chain driven mechanism as the entire assembly rotated around the subject. In a system where the detectors are fixed in space, a point source driven in a helical motion around the subject can provide full coverage of all lines of response. A hydraulically driven helical point source was used in the EXACT3D system [Spinks, 2002]. This system used a thin stainless steel tube bent in a helical fashion and filled with hydraulic fluid and a source that was pumped through the field of view. A similar geometry, but different implementation, was used in some preclinical PET systems. The microPET [Chatziioannou, 1999] utilized a singles method that consisted of a helical source mechanism that moved a source across the entire field of view using a screw-drive mechanism, sampling all lines of response. Although precise, the mechanism was problematic due to the exposed mechanism. The MOSAIC took a simplified approach by removing the axial motion from the source, limiting the motion to rotation only [Hichwa, 2004]. This simplifies the mechanism, but requires bed motion to sample the entire field of view and for attenuation coefficients to be estimated for those lines not actually measured. Furthermore, the MOSAIC did not have the source within the field of view, and was thus not able to directly sample perpendicular lines of response. The DPET [Kemp, 2009] revisited this issue with a mechanism that does not axially move the source through the entire field of view, similar to the MOSAIC. However, unlike the MOSAIC,

the source has enough axial motion that it can be moved in and out of the field of view as needed, allowing direct measurement of perpendicular lines of response.

1.3 Inveon DPET

The Siemens Inveon Dedicated PET (DPET) is a tomograph designed for positron emission tomography in a preclinical research environment [Kemp, 2009]. The system has a bore size of 120mm and an axial field of view of 120mm, and is thus equipped for imaging small animals such as mice and rats. This system is pictured in Figure 4.

The DPET includes a fully enclosed and shielded transmission source with a cone-beam collimator. Axial coverage of the subject is performed by moving the bed. In addition, accurate source and bed location information is embedded in the raw data stream, thus, a spiral cone-beam transmission scan can be accomplished by continuously moving the bed slowly while scanning, or step-and-shoot methods can be employed.

1.3.1 Source Selection

The DPET collimator uses a standard capsule design that allows for different source types, and thus energies. Two capsules are currently available, a 122 keV Cobalt-57 source, and a 60 keV Americium-241 source. These two sources have a different response as measured on the detector. The reduced energy deposited in the detector by the Am-241 source results in reduced energy resolution and positioning accuracy. Given the size of the source window (2.0mm)[Beach, 2004], and the positioning accuracy of the detector at the specified energy, the solid angle determined by a beam between the source and detector could be either divergent, convergent, or parallel. This effect must be properly accounted for in the generation of the system model in order to reasonably match the system model to the solution vector. Experimental results related to the size and divergence of the beam and the related effects on the system model are located in Appendix A. For this specific configuration, those results determined that a parallel model could be used for a system matrix with voxel sizes of $1.0mm^3$ and larger.

1.3.2 Determining Attenuation Coefficients

If all lines of response in the PET field of view are not directly measured by moving the source to the correct position, then they must be estimated in some way. This can be done by computing the attenuation coefficients within the image volume, and forward projecting to determine the attenuation factor for a given line of response. The DPET can measure nearly all lines of response by moving the bed in steps the same size as the detector crystals, but it is not normally used in that fashion due to the amount of time required to do so. Instead, 8 bed steps are taken, providing direct measurement for approximately 20% of the available lines of response. The DPET computes the attenuation coefficient matrix using single-slice-rebinning (SSRB) to create a set of 2D projections [Newport, 2001][Daube-Witherspoon, 1987], followed by filtered

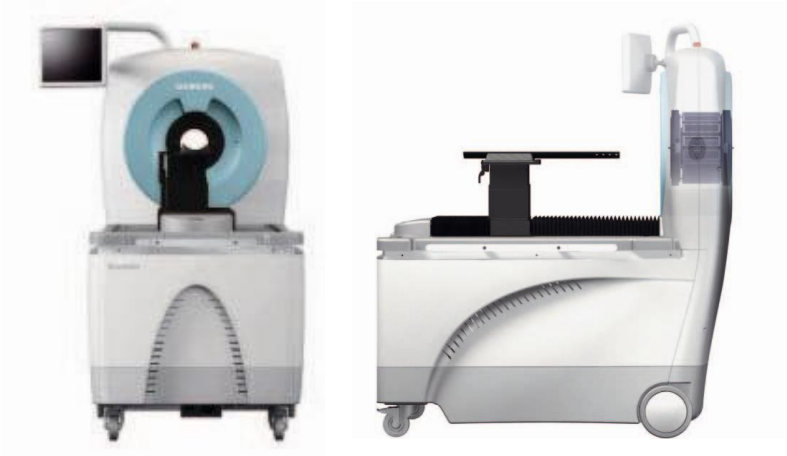

Figure 4. Inveon DPET front and side views.

backprojection [Kak, 2001]. This method uses the oblique planes to estimate attenuation coefficients for the direct plane that passes through the midpoint thereof, resulting in axial blurring. If the object is elliptical, and relatively uniform, SSRB performs an acceptable first order approximation of the attenuation in the subject. Most rats and mice roughly fit this description, but problems ensue if those conditions are violated. These concepts are shown graphically for an axially stationary source in Figure 5. The actual measured direct plane is shown in green, an estimated direct plane is shown in yellow, and the oblique plane used to estimate the direct plane is shown in red.

Once the attenuation coefficients are computed for the image volume, forward projection is used to compute the attenuation factor for a given line of response [Newport, 2001]. The accuracy of the forward projection is thus only as good as the attenuation coefficient matrix that it is based on. Since the 2D projections created by the SSRB processing are estimates of the actual object, the attenuation coefficient matrix reflects the same potential errors, and these are propagated through the forward projection process into the PET image. Thus, a more accurate approach to computing the attenuation coefficients would represent a desirable improvement.

A variety of 3D reconstruction methods have the potential to compute the attenuation coefficients more accurately than the SSRB/FBP approach currently used. The DPET transmission mechanism is configured to radiate a cone-beam against the opposing detectors, yet this is at odds with the single slice rebinning (SSRB) techniques used to process attenuation data. Only a few planes are measured nearly perpendicularly, the majority are measured obliquely. SSRB techniques approximate vertical planes with oblique information, so for a given plane, the more oblique the measurement, the more the estimation deviates from the correct answer [Sossi, 1994]. Given the cone-beam design of the collimator, a more accurate approach derived from cone-beam reconstruction would be more suitable. An analytic cone-beam methods for a similar calculation are used in microCT systems [Feldkamp, 1984]. Unfortunately, the geometry of the DPET does not entirely conform to the cone-beam geometries assumed by analytic methods like Feldkamp.

1.3.3 Source/Detector Geometries with Uniform Sampling

Dedicated microCT systems are often equipped with a moving flat panel detector as shown in Figure 6. In this configuration, the imaging space is sampled in an equidistant uniform fashion. This allows analytic cone-beam reconstruction algorithms to be used effectively. Clinical systems generally use a curved detector geometry with a relatively small axial fan angle as shown in Figure 7.

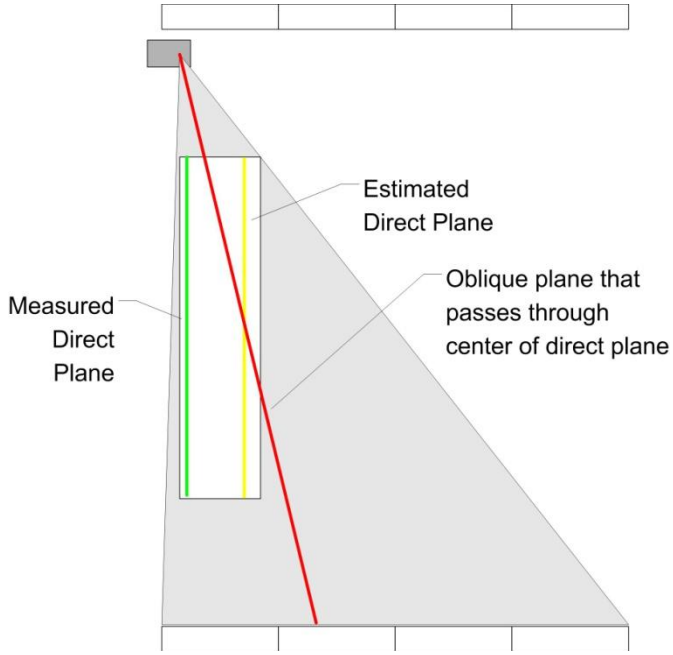

Figure 5. Single Slice Rebinning direct plane estimation from oblique information.

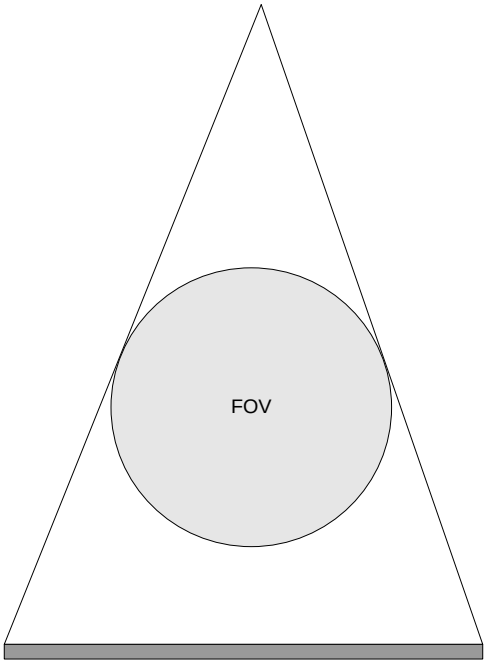

Figure 6. MicroCT flat panel geometry has equidistant sampling against the detector.

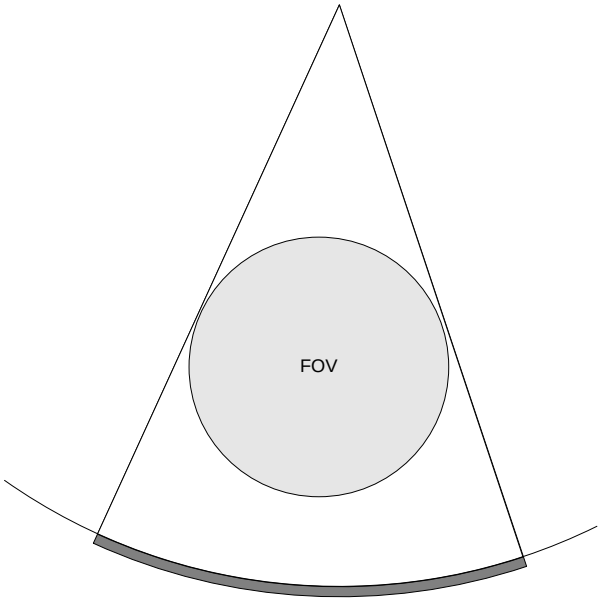

Figure 7. Clinical CT curved detector geometry has equiangular sampling against the detector.

1.3.4 DPET Source/Detector Geometry

The DPET has a segmented curved detector but the source is not placed at the radius of curvature of the detector (that would be at the center of the FOV), but instead at a distance of approximately twice that (2R). This is illustrated in Figure 8.

Traditional analytic cone-beam reconstruction algorithms for a curved detector geometry are based on the source being located at a distance equal to the radius of curvature of the detector. In this way, sampling uniformity is preserved in an equiangular fashion.

The DPET places the source at a distance greater than the radius of curvature of the detector (Figure 9), and uses a much wider beam angle, resulting in irregular angular sampling. In an equiangular configuration, from the perspective of the source looking across the field of view at the detectors, the angle seen from one detector to the next would be the same. If there are irregular gaps between detectors, then the angle between two detectors could be different depending on the presence or absence of a gap. In addition, if the detector array is not at least approximately curved, but straight or otherwise oblique to the line of sight from the source, then the angle from one detector to the next will be different. The DPET detector array exhibits all of these conditions, and the resulting pattern is shown in Figure 10.

The sampling varies between 0.5 degrees and 0.7 degrees per angular sample. Thus, the assumption of uniform sampling made in analytical reconstruction is clearly violated, and if left uncompensated, significant errors may result.

1.3.5 Effects of Non-Uniform Sampling

A transmission scan taken on a DPET and reconstructed with an analytic approach yields disjoint areas of the image where the detector gaps are located (high differential angles). Substantially degraded resolution is seen as the radial distance from the center is increased (due to cumulative mispositioning of the lines of response the further from zero). Figure 11 shows such a scan done of an aluminium flat (3.2 mm by 38.2 mm) and reconstructed from data that has not been resampled to make the sampling uniform, it is simply assumed to be an approximation of a curved detector. The reconstructions were done with both an analytic approach that is not compensated for the non-uniform sampling (Feldkamp) and a model-based iterative approach (SIRT) that does compensate. Both are presented here to illustrate the potential problems.

Defects are clearly visible in the resulting analytic image in Figure 11 (a). Holes are shown in the flat where none actually exist due to the gaps in the detector geometry, and the resolution is degraded the further the position from the center of the FOV due to the cumulative mis-positioning induced by the irregular sampling. The model-based method (b) shows far fewer problems, from the same data, and is dimensionally correct.

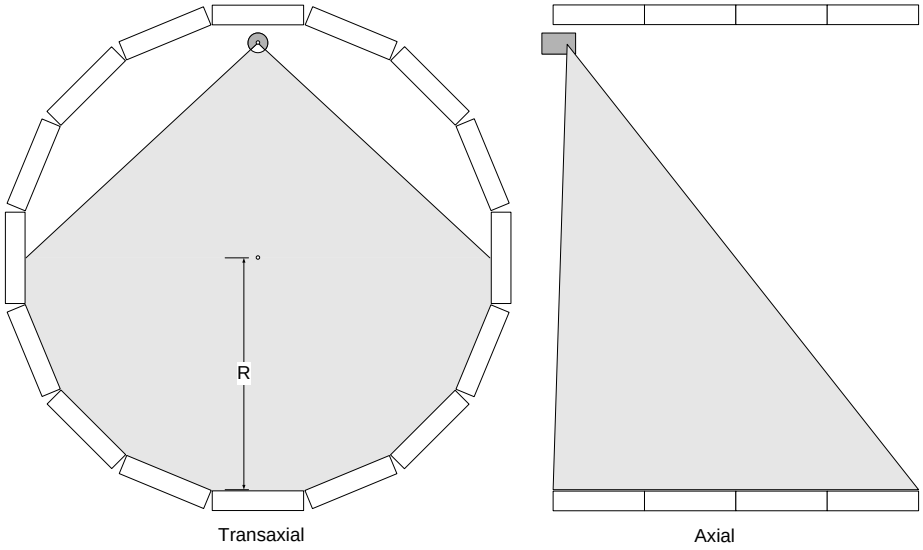

Figure 8. The Inveon DPET has a 16-sided polygon with fixed axial source geometry.

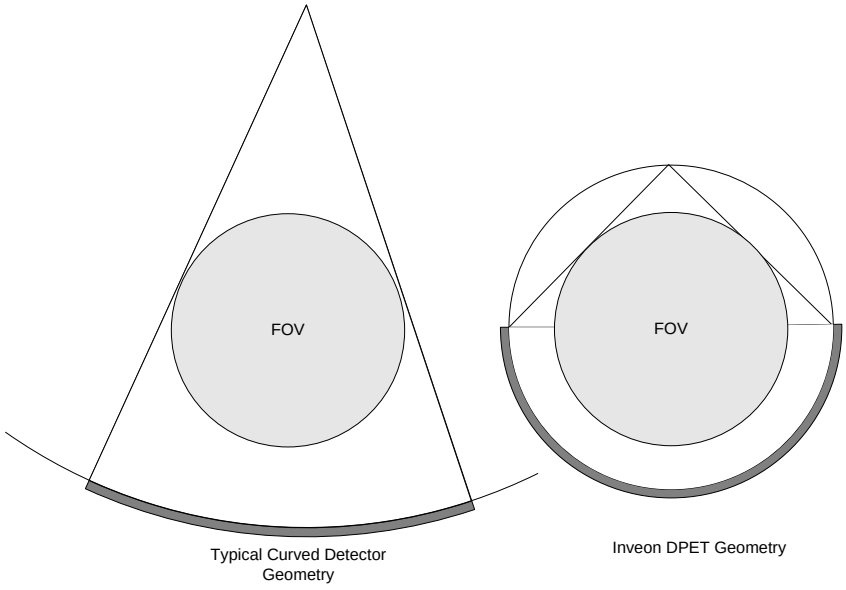

Figure 9. Typical fan beam vs. Inveon DPET fan beam geometry.

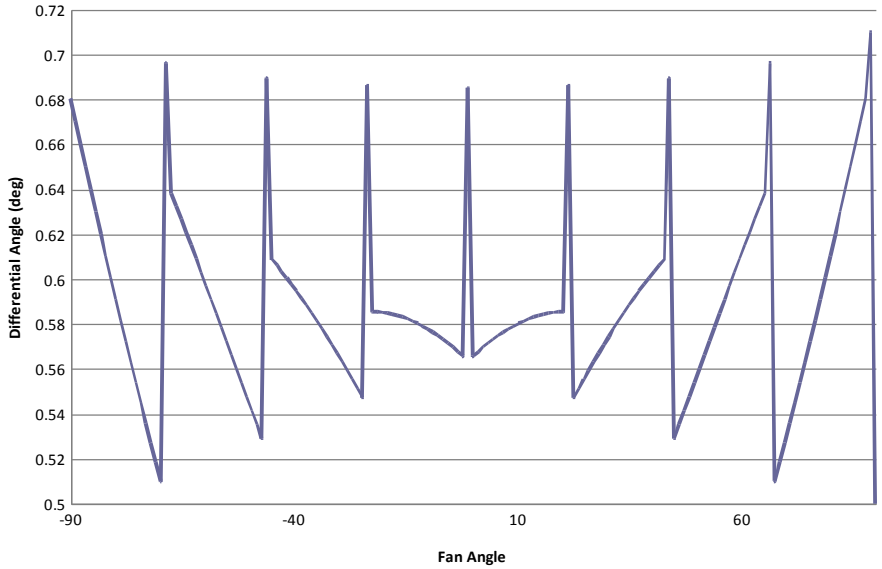

Figure 10. Angular sampling of the Inveon DPET is not uniform. Large jumps to high angles indicate gaps, and sloping sections indicate flat detectors placed obliquely to the line of sight.

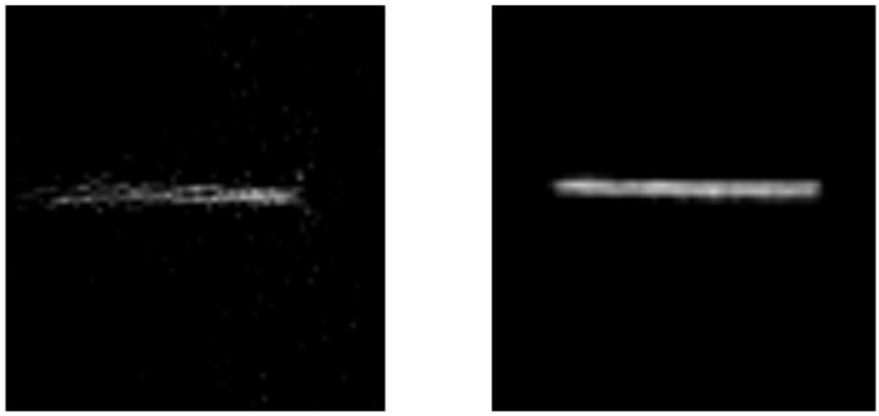

Figure 11. Reconstruction of an aluminum flat, 3.2 mm by 38.2mm, transaxial view, analytic cone-beam (Feldkamp) and model based (SIRT) on non-uniformly sampled data.

1.3.6 Methods for Handling Non-Uniform System Geometries

Although computationally intensive, iterative reconstruction algorithms form a viable alternative to analytic methods due to their ability to incorporate detailed models of the sampling geometry. They are not dependent on sampling uniformity in order to achieve correct results, although their performance may vary across the field of view if the sampling varies widely [Kak, 2001]. Unlike analytic methods that have rigid assumptions about the relation of individual voxels to the projection information, iterative algorithms like SIRT individually specify those contributions. The individual handling of the contributions increases the computational complexity of the overall computation.

Significant improvements in the availability of fast computational hardware combined with careful optimization of the process allows most criticisms of the processing speed of iterative approaches to be dismissed, and this will be explored in the following chapters.

1.4 Contribution

The solution to improved transmission imaging on the Inveon DPET is the subject of this work. The base algorithm itself, SIRT, is well known [Kak, 2001], but successful implementation on a given architecture requires work in several areas.

- Determination of the system matrix.
- Physical modeling from 3D manufacturing models and detector performance parameters.
- Numerical acceleration of the system matrix calculation with a new method, the volume intersection approximation.
- Physical acceleration of the volume intersection approximation with graphics and vector hardware.
- Implementation of the SIRT algorithm.
- Numerical acceleration of SIRT with standard numerical methods techniques.
- An orthogonal approach to update order selection based on priority queues.
- Physical acceleration of SIRT with development of a mutex-free multi-core implementation.
- Validation of the final result using synthetically derived information.
- Statistical evaluation of the synthetic validation.
- Phantom and biological subject scans to show system compatibility.

The result of the combination of these contributions is an improved transmission measurement on the Inveon Dedicated PET with sufficient computational speed to not interfere operationally with the normal use of the system.

1.5 Outline

This dissertation is divided into 5 principal components. Chapter 1 has provided a basic description of some of the fundamentals of PET and attenuation correction, as well as a description of the DPET and the reasons why measurement of attenuation factors on that system are complicated. Chapter 2 deals with the development of the system matrix around which the iterative reconstruction is centered. This is done through the volume intersection approximation and the implementation of three different hardware approaches used to generate that information in a time efficient way. Chapter 3 is concerned with the reconstruction algorithm SIRT, and the ordered subsets variant OSSIRT, as well as the implementation for use on DPET transmission data. In addition, an analysis is presented regarding the use of convergence enhancing numerical techniques, as well as hardware acceleration with a multi-core implementation to achieve time efficient operation. The validation of the system is presented in Chapter 4 with results presented for both synthetic data, as well as phantoms and biological subjects selected to show key aspects of system performance. Statistical testing methods were employed to evaluate the algorithms over a wide range of expected use and these results are presented as well. Chapter 5 discusses final results and expectations for future work in this area. Finally, Appendix A discusses the measurements needed as input to characterize the system matrix.

Chapter 2

System Modeling

2.0 System Model Background

The Inveon DPET is made up of a a 16-sided array of detectors located around the imaging volume, and a radioactive source in a tungsten collimator that can rotate around the subject and illuminate the opposing detectors. For every possible location of the transmission source, a series of lines can be drawn to all opposing detectors. Each of these lines is termed a ray (from ray tracing), and can also be thought of as a line integral or line of response (from PET terminology). Every ray projects a particular pattern through the imaging volume, interacting with individual voxels as it proceeds forward from the source to the detector. The combination of all of these interactions represents the system model.

This concept is often modeled in terms of segment length through line intersections in the voxel [Siddon, 1985][Wu, 1991]. Others have examined this calculation due to complications in image quality due to aliasing in the direct Siddon-based line intersection approach [Vandenberghe, 2002]. Aliasing, seen as jagged edges in the ray as it is projected through the imaging volume, can induce high frequency noise into the calculation. Due to sampling geometry symmetries, this noise is coherent, and results in ring and moiré pattern artifacts [Brinks, 2006]. The applications described by Brinks and Vandenberghe required better anti-aliasing at higher computational cost. As a result of this research, and further based on volumetric concepts previously proposed [Joseph], a more efficient approach was developed that involves a tube of response [Schretter, 2006]. Although it was not stated as such, a tube of response is a volume, not just a line with special properties. A volume intersection provides inherent anti-aliasing because it does not allow sharp changes in voxel interaction based on small changes in ray position. It is important to note that volume is proportional to average length for a beam of a given diameter, so the necessary calculation result units of measure (inverse length) are preserved. In addition, the Siddon line intersection approach was shown to be 60% slower than the tube of response approach. Thus, we chose to model volume intersection due to the simplicity (and thus potential speed) of the calculation with included anti-aliasing.

2.1 Ray Projection Methods

In the optics field, a procedure exists by which a ray would interact with a given volume as it passed through [Spencer, 1962]. This approach is focused on what the volume does to the ray, usually bending or otherwise deforming it. The information required for the system model is the relative magnitude of the interaction of the ray on the volume. This is the inverse of the problem described by Spencer.

This problem has certain similarities to standard ray tracing algorithms [Hearne, 1994], but is in fact a much more computationally intensive problem. In standard ray tracing algorithms, a ray is used to determine the contents of a 2D planar image as shown in Figure 12. As such, this class of algorithm can be categorized as complexity class $\Theta(N^2)$ for an image size NxN, given constant and finite objects within the scene.

The determination of the system model weights can be more closely related to the drawing of a line in 3 dimensions. Thus, in terms of computational complexity, the projection of a single ray represents a series of voxel intersections through a volume, increasing the computational complexity to $\Theta(N^3)$. Since there are multiple projections (P) to be formed, and P can potentially be a large number, the overall complexity of the problem is thus $\Theta(PN^3)$.

The easiest method to compute this relation is a binary projection (Figure 13a) through the volume. If the ray touches a particular voxel, then the voxel weight is set to 1, and it will be updated with that weight. This method requires only a floating point position calculation to determine if the line is anywhere within the voxel. Potentially severe aliasing can occur under these conditions, and this approach is not used for this reason.

Extending the binary projection model to a line intersection model [Siddon, 1985] requires computing the length of the line intersection within the volume (Figure 13b). This approach doubles the number of floating point operations to compute as compared to the binary projection, as the endpoint of each line must be determined. With this approach, aliasing normally should be better than the binary projection method, but has the potential to be inconsistent if the ray is close to a series of voxel vertices. Thus, this approach is sensitive to small changes in ray location near the voxel vertices.

Further extension of the projection method into an area/volume intersection removes the inconsistencies of the line intersection approach by making the calculation less sensitive to proximity to voxel vertices (Figure 13c). If done by the definition, it also requires double the floating point operations to compute, as it is essentially two line intersections. In practice, the very large number of rays in a 3D system require long calculation times, potentially days long, depending on the method used to compute it. Fortunately, the calculation need be done only once for the image volume parameters desired.

2.2 Volumetric Intersection Approximation

By approximating a volume/area intersection, the computational complexity can be reduced. We will henceforth refer to this method as the Volume Intersection Approximation (VIA) algorithm. A volume intersection can be approximated by calculating the tube of response in space with a simple binary projection algorithm, then downsampling the voxel volume by summing the number of subvoxels hit. This can be expressed similarly to anti-aliasing methods used in graphics ray tracing [Hearne, 1994][Glassner, 1989], extended to 3 dimensions (Figure 14).

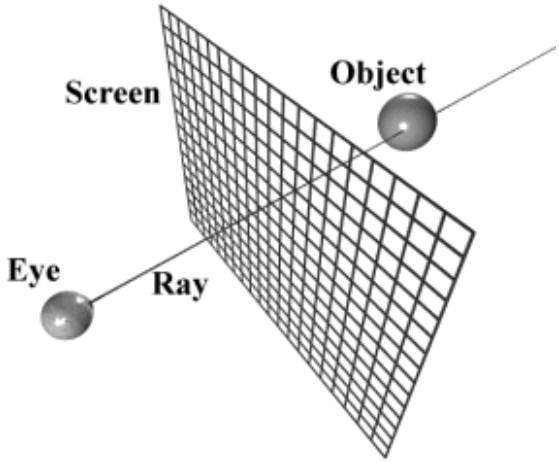

Figure 12. Ray tracing 2D solution [Buck, 1999].

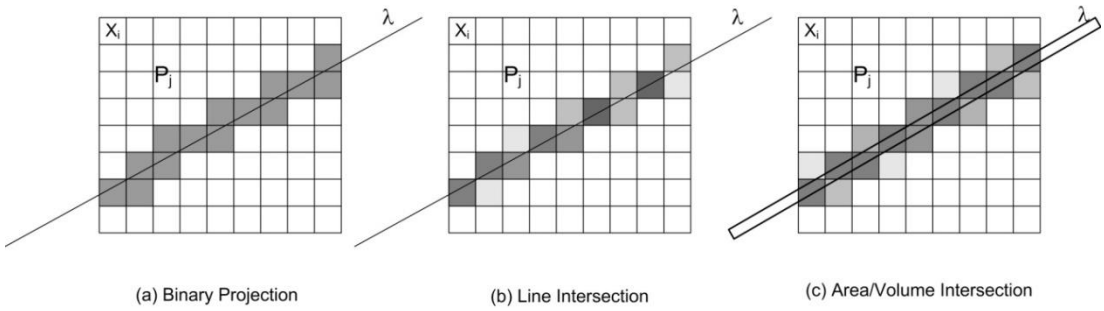

Figure 13. Projection Methods and their effect on aliasing. Shown in 2D for simplicity.

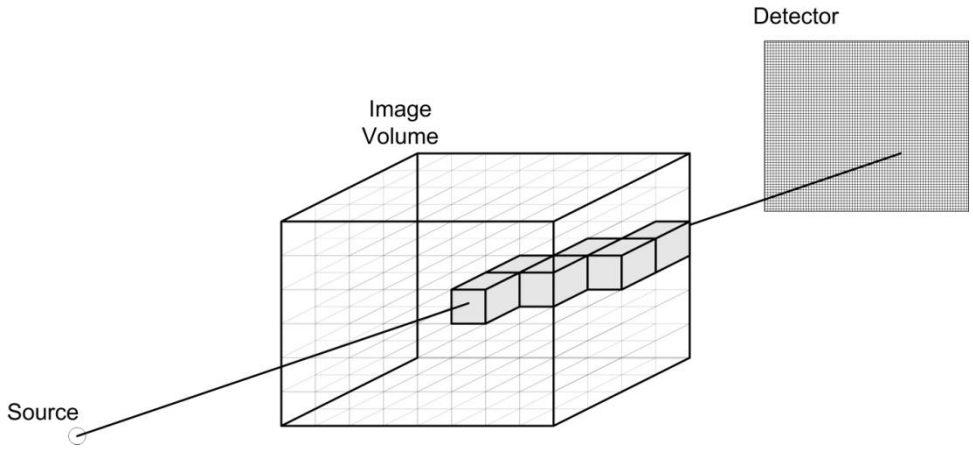

Figure 14. Volumetric Intersection in 3D. A primary voxel is subdivided into subvoxels whose volume is added together to determine the contribution to the primary voxel.

The method used to compute VIA, is described as follows. For a volume V containing N voxels, subdivide each voxel V_i into an arbitrary number (n) of smaller voxels v_j. Thus, by definition, n >> N. Using a binary projection method, for projection j (from source to detector), through the subvoxels v_k, setting the value of each subvoxel to 1 if intersected. Next, for each voxel V_i, count the number of subvoxels v_k contained within voxel V_i whose value has been set to 1. Multiply this count by the volume of each voxel v. This value is the weight of the system matrix A_{ij}.

Thus, the maximum error *e* can be determined by examining the volume of the voxels in the two physical volumes. Since it is given that k >> i, the volume of a voxel in volume v is much smaller than the volume of a voxel in volume V. Thus, since a binary projection is used in volume v, the value computed in the elements of system matrix A can be different from the analytically derived volume by a maximum of the volume of a voxel in v divided by 2. In our specific application, the ratio of the volumes of the voxels is 1/64, resulting in a maximum error of 1.56%. For a given application, if this level of error is not accurate enough, the relative voxel sizes can be adjusted to lower the error at the expense of computational demand. We refer to the cube root of the inverse of this ratio as the downsample factor D. Since computational complexity increases with $\Theta(D^3)$, and computational demands for the calculation of the system matrix are largely dependent on the downsampling, a downsampling factor of 4 gives a good tradeoff of accuracy versus size.

2.2.1 Direct Calculation of the Volume Intersection Approximation

The volume intersection approximation (VIA) technique can be approached in multiple ways. The simplest solution is to literally compute the system matrix according to the VIA algorithm previously described. This technique is relatively easy to validate, because intermediate results are visible, and it is physically possible to calculate by hand for a given ray. Only simple arithmetic calculations are required. The drawback is that a volume of memory of the subsampled image volume size is required for each thread making the calculation, thus limiting the number of process threads that can work on the whole job. In addition, the entire volume must be checked in steps 8-10 and this takes considerable time.

2.2.2 Calculation of VIA with a volume walk

By noticing a convenient fact about the system geometry and borrowing from graphics ray tracing theory[Hearne, 1994], the algorithm can be rewritten to not require any interaction with memory. We will refer to this method as the optimized volume intersection approximation (OVIA) technique. Since the purpose of the model is to determine the voxel interactions with the particular ray, the concept of scatter or reflection (in the ray tracing sense, not the nuclear physics sense) does not occur. In addition, voxels are a convex hull. In practice, borrowing from algorithms regarding the computation of hulls, this means that all rays project forward only, and once a ray leaves a voxel, it will never return [Cormen, 2006]. An algorithm can take advantage of this

optimization by maintaining the current location only, and storing the value in a list when the ray departs the voxel. A flow chart of the algorithm is defined in Figure 15. This type of volume walk makes the calculation entirely computational in nature, and thus ideal for mass parallelization on computational hardware with very fast computational capability but minimal memory resources such as a General Purpose Graphics Processor Units (GPGPU), or streaming vector hardware such as Streaming Simultaneous Execution (SSE) systems.

2.3 GPGPU Implementation

Taking advantage of the available General Purpose Graphics Processing Unit (GPGPU) available on many systems allows highly parallelized and high performance optimized implementation of the OVIA algorithm. These GPGPUs are capable of far more than graphics functions. Their processing engines are optimized for complex, high speed mathematical computations. Examples such as the NVIDIA series of GPU contain many-core implementations with very high performance numerical capability. Demand for realism from high end graphics applications, primarily games, has driven both computational and memory bandwidth performance according to Moore's Law [CUDA, 2009], as shown in Figure 16 and Figure 17. It is possible to configure most modern computers with a GPGPU at low cost.

GPGPUs are designed to implement graphics algorithms. Generally, this class of algorithm performs short, but specific operations to many elements of a multidimensional array. Architecturally, their implementation utilizes more transistors for processing, and fewer transistors for control and cache applications than a generic CPU would. In practice, this means that they can perform very specific (in this case mathematical) functions in fewer clock cycles, as long as the algorithm can be formed such that all processors run identical code. All processing elements must run the same code. Their instruction set is laid out in a very regular and specific manner to minimize decode requirements, and their many-core implementations leverage this optimized architecture by replicating it in a form that many would interpret as a vector supercomputer. Individual processing elements can perform complex mathematical constructs on specific members of an array with high arithmetic intensity in parallel.

One example of a common GPGPU is the NVIDIA Quadro NVS series [NVIDIA, 2009]. This is a low power device used on a large number of laptop displays. The model 140M is an available option on some IBM business laptops. This device has two processors, 512 Mbytes of memory. Individual vector operations of up to 128 elements can be executed in parallel, but it is limited to vectors of length 512. Floating point calculations are IEEE compatible, and hardware support is available for single precision operations. Given the size of the typical image reconstruction problem, a vector length of 512 is sufficient for most algorithms, as is the single precision floating point. More advanced devices, such as the Tesla series, contain much longer vector hardware, and are capable of handling double precision floating point in IEEE format at a price point of a few hundred dollars.

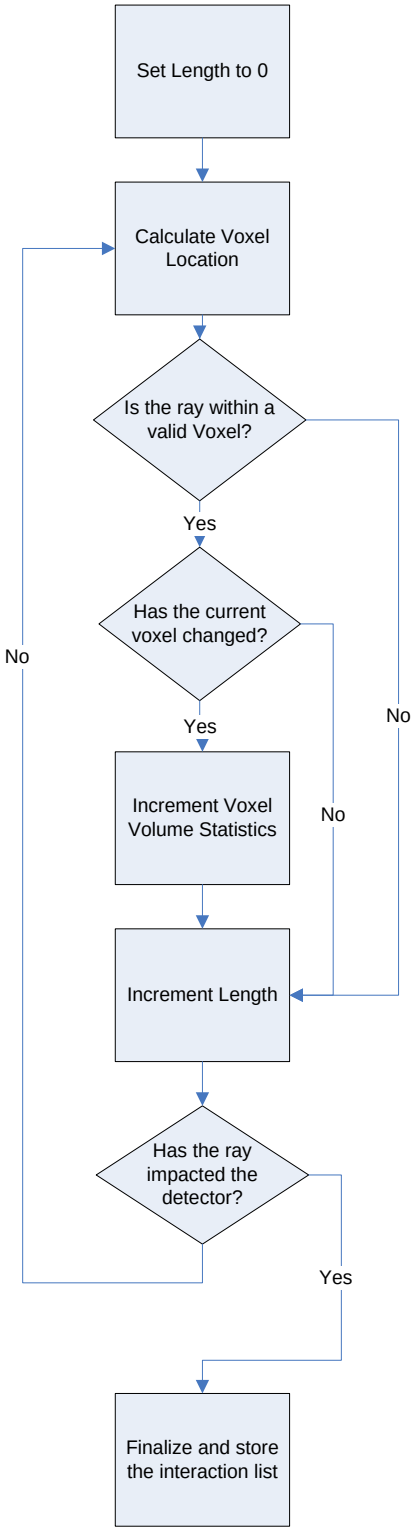

Figure 15. Flow chart of Optimized VIA method.

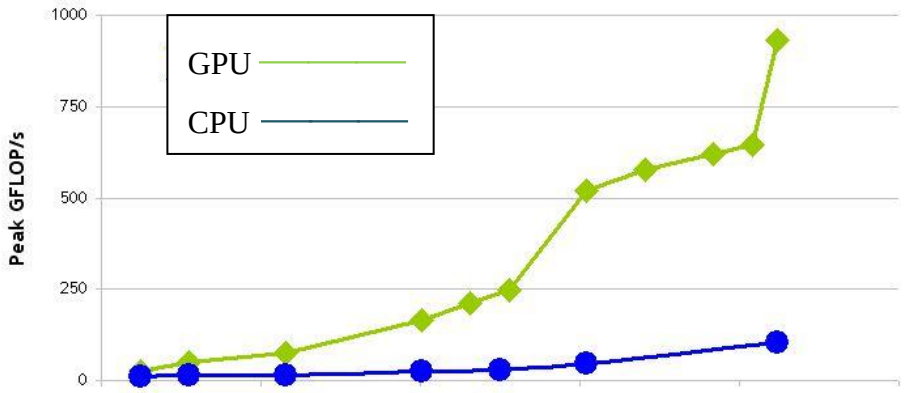

Figure 16. NVIDIA GPU Numerical Processing Performance [NVIDIA, 2009].

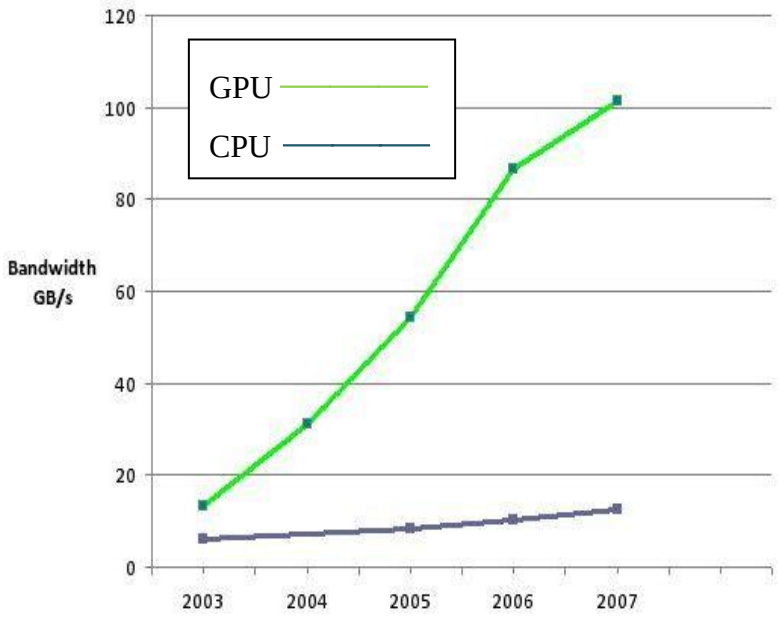

Figure 17. NVIDIA Graphics Processor Memory Bandwidth [NVIDIA, 2009].

Programming on GPGPU based systems is simplified with a programming environment called Compute Unified Device Architecture [CUDA, 2009] that abstracts the hardware in such a way that the algorithms can be expressed in the "C" programming language [Kernigan, 1976], independent of the hardware actually used. The environment is compatible with the full line of NVIDIA products from the relatively simple, but useful Quadro NVS to the high performance NVIDIA Tesla. Code can be independently developed on any hardware, and run on any hardware.

Within CUDA, expression of a vector operation is handled with macro expansion as in the following example for vector addition [CUDA, 2009].

```
__global__ void vecAdd(float *A, float *B, float *C)
{
        int i = threadIdx.x;
        C[i] = A[i] + B[i];
}

int main()
{
        vecAdd<<<1,N>>>(A,B,C);
}
```

The definition of the code to be executed by the GPU operates on a single element of the array, and is automatically loaded into the GPU. The entire array is processed through the <<<1,N>>> macro, and the results are automatically transferred as necessary.

By applying abstraction to the OVIA algorithm, there are multiple approaches. The parallelization can be done to make the calculation of a single ray as fast as possible, or multiple lines can be computed simultaneously. Eventually, as hardware gets faster, it would be advantageous to be able to compute a single line on the fly within the reconstruction so that the system matrix would not need to be calculated and then stored. At this time, the system requires 5 seconds on standard Inveon production hardware to read the 1.4 GB system matrix from disk. However, reading it in is still faster than calculation. Even so, by laying out the problem in this fashion, as faster hardware becomes available, the computation speed will exceed the speed to read from disk and it will become more efficient to calculate than to store the matrix. Namely, the ray can be split down its length into segments, and the contributions of individual segments of the ray can be calculated independently as shown in Figure 18. The length of each segment is equal to the minimum dimension of each voxel so that the algorithm is guaranteed to visit each voxel at least once. It is possible that a voxel can be sampled twice, so this needs to be accounted for in the summation process.

By splitting the problem into multiple lengthwise segments, each vector unit is sent the source position, detector position, and segment index. Using that information, the vector unit can independently calculate the position along the segment, and thus the individual contribution to the overall ray. This approach provides fast calculation of the

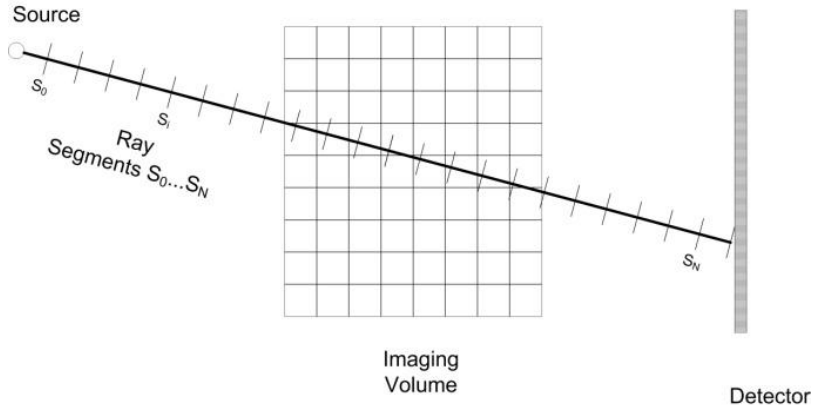

Figure 18. Division of ray into independent segments for the OVIA Method.

geometry of the ray, which is a bulk of the problem. Once the individual components are created, simply summing them in to the resulting response is done very efficiently by the top level CPU as shown in Figure 19.

If there are sufficient vector units (cores) available to complete the calculation of the entire vector length in one pass, and all calculations are independent, the overall calculation can be fast, but there is a problem in the storage. Once calculated, the voxel contributions must be stored. Access to real memory on the GPU is very expensive compared to compute cycles, and is not cached. Delays of up to 400 clock cycles [CUDA, 2009] can exist for a memory write to the internal memory. After the entire calculation is performed, the result must be transferred back to main memory on the CPU. This takes additional time as well, reducing the effective performance in this case.

The ray segmentation code for GPU is shown below for reference:

```
__global__ void segment_ray_GPU(float src_x, float src_y, float src_z, float diff_x,
float diff_y, float diff_z, float ray_length, float voxel_size, int seg_size, int
*subvoxel, int *voxel)
{
	// model constants
	int seg_index = threadIdx.x * 16 + blockIdx.x;
	float g = seg_index * voxel_size;

	x_mm = src_x + diff_x * g; // location in mm
	y_mm = src_y + diff_y * g;
	z_mm = src_z + diff_z * g;

	// limit the FOV to 50mm radius
	if ((x_mm * x_mm) + (y_mm * y_mm) > 2500.0) // outside 50mm radius
	{
		voxel[seg_index] = -1;
		subvoxel[seg_index] = -1;
		return;
	}

	int x = (int)(x_mm / voxel_size + x_dim / 2);
	int y = (int)(y_mm / voxel_size + y_dim / 2);
	int z = (int)(z_mm / voxel_size);

	if (x >= x_dim || x < 0 || y >= y_dim || y < 0 || z >= z_dim || z < 0)
	{
		voxel[seg_index] = -1;
		subvoxel[seg_index] = -1;
		return;
	}

	int p_addr = z * x_dim * y_dim + y * x_dim + x;

	subvoxel[seg_index] = p_addr;

	if (p_addr < 0)
		voxel[seg_index] = -1;
	else
	{
		int dx = (int)(p_addr % x_dim) / xresample_factor;
		int dy = (int)(p_addr % (x_dim * y_dim) / y_dim) / yresample_factor;
		int dz = (int)(p_addr / (x_dim * y_dim)) / zresample_factor;

		int downsampled_address = dx + dy * rx_dim + dz * rx_dim * ry_dim;

		if (downsampled_address < 0 || downsampled_address >= rvolumesize)
			downsampled_address = -1;
```

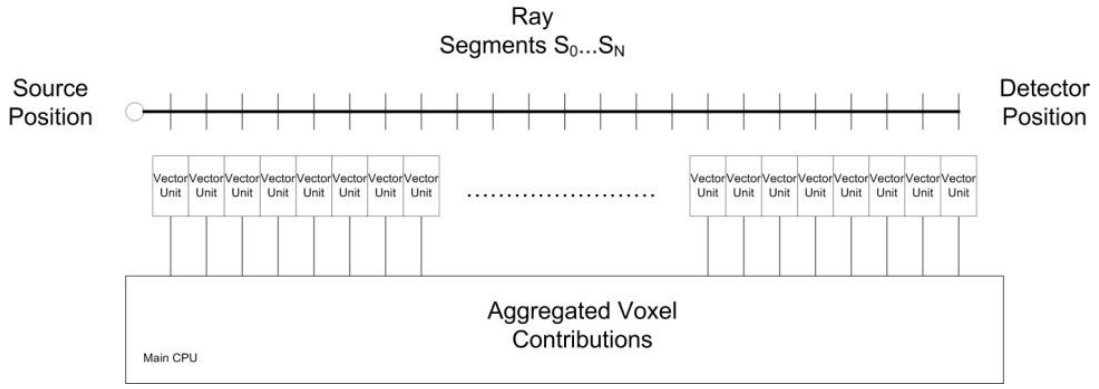

Figure 19. Vectorized OVIA Hardware Organization for many-core GPGPU.

```
                voxel[seg_index] = downsampled_address;
        }

        return;
}
```

2.4 SSE2 Implementation

Streaming Simultaneous Execution (SSE) instructions available on many modern computer architectures allow some level of vectorization depending on data type [Intel, 2009] [AMD, 2009]. This approach is relatively universal, inexpensive, and offers significant computational improvement, but is not easy to implement. SSE comes in 5 different versions, SSE1 thru SSE5 with increasing capability in the higher numbered versions. Nearly all new processors support SSE5, including the latest Intel and AMD devices.

SSE level 2 supports floating point formats of 32-bits within 128 bit registers. In practice, this means that one SSE2 instruction can perform four 32-bit floating operations simultaneously. Theoretically, this results in a 4X improvement in floating point throughput. Unfortunately, 32-bit integer support isn't quite as good in SSE2 as SSE4 and 5, and the SSE4 32-bit integer multiply _mm_mullo_epi32() intrinsic was emulated in our implementation. This does not represent a significant degradation in performance as the normal CPU 32-bit multiply is fast if pipelined properly.

Using the same vectorization methodology derived in the GPU implementation of the OVIA algorithm, some compromises had to be made. Given the same total vector length as shown in Figure 18 19, the SSE2 implementation is limited to vectors of length 4, so some compartmentalization had to occur to complete the implementation. Each vector of 4 segments is computed sequentially as shown in Figure 20.

There are advantages to the SSE methodology over GPU. Namely, memory is in place at all times. The results are computed and stored only once. Furthermore, the main memory subsystem of the CPU has both level 1 (L1) and level 2 (L2) caching, and this has up to an order of magnitude effect on average system memory access time [Hennessey, 2006].

The principal disadvantage of SSE is the vector length. 32-bit floating point vectors are of length 4, requiring multiple iterations to calculate longer vectors. Also, a high degree of programmer knowledge of hardware interaction is required to achieve performance improvements. For example, the portion of the segment_ray() kernel that calculates the (x,y,z) position of the segment is shown below:

```
float t[4];
__m128 dfx_v = _mm_set1_ps(diff_x);
__m128 dfy_v = _mm_set1_ps(diff_y);
__m128 dfz_v = _mm_set1_ps(diff_z);
t[0] = seg_index + 0;
t[1] = seg_index + 1;
t[2] = seg_index + 2;
t[3] = seg_index + 3;
```

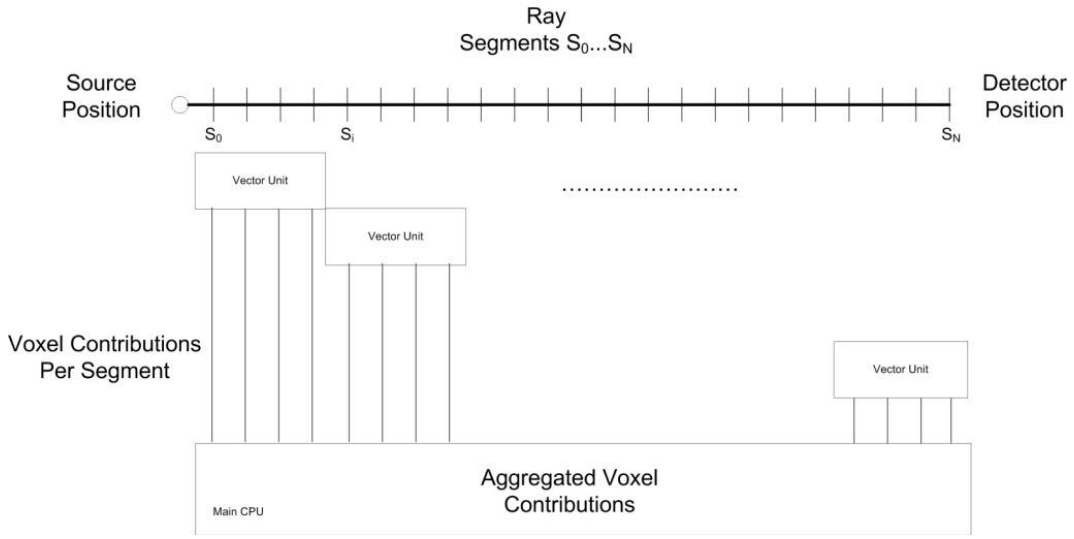

Figure 20. Vectorized OVIA Hardware Organization SSE.

```
__m128 si_v = _mm_loadu_ps(t);
__m128 t1_v = _mm_mul_ps(si_v,vs_v);
__m128 gdiff_x = _mm_mul_ps(t1_v,dfx_v);
__m128 gdiff_y = _mm_mul_ps(t1_v,dfy_v);
__m128 gdiff_z = _mm_mul_ps(t1_v,dfz_v);


__m128 x_mm_v = _mm_add_ps(gdiff_x,sx_v);
__m128 y_mm_v = _mm_add_ps(gdiff_y,sy_v);
__m128 z_mm_v = _mm_add_ps(gdiff_z,sz_v);
```

Inspection of the complexity of the ray segmentation code written for SSE shows the hazards of the programming environment. Even though the intrinsic functions are not straight assembler, in many respects performing these calculations in assembler would be easier and clearer, but the later Microsoft compiler products do not allow this.

From a performance perspective, SSE performance acceleration will fail if pipelining fails [Hennessy, 2006]. A straight numerical calculation can be executed very rapidly because the instructions are queued up and efficiently executed. Decision making severely degrades the performance. Branches break the pipeline, and in this case, since the boundary conditions can occur at any element within the vector, traditional per element testing would result in performance issues. In order to resolve this issue, boundary testing is done after all calculations are performed, and out of range conditions are masked through simple binary logic. This type of calculation can be done in SSE vector form as well, calculating boundaries for all four voxels simultaneously, and will not break the pipeline. This is one advantage of the SSE implementation over the GPU implementation.

2.5 DPET System Model

The system was modeled by first extracting the locations of all detectors from the mechanical CAD models used to build the DPET system. Next, all possible rays were calculated with a volume intersection approximation, and their response recorded. Tight manufacturing tolerances are specified for the Inveon DPET (+/- 0.001 inches for detector components)[Beach, 2004], so in practice, the mechanical models match the actual production systems with a high degree of accuracy.

Since the detector material, LSO [Melcher, 1992], is very dense with respect to the low energy photons emitted by Co-57 and Am-241, the depth of interaction of the photon is very small, and is set to zero. In this work, the interaction point is set to be at the center of the crystal face.

From information in Appendix A, the expected ability of the system to resolve position due to the size of the transmission source window is 2.0 mm. Thus, we choose to model the imaging volume in voxels approximately half that size, or 1.0 mm. For each ray (identified by source angle and destination detector address), a variable length set of projection weights, P_j (32-bit float), are stored, with their corresponding voxel address (32-bit integer). This results in a system matrix for a 128x128x128 image volume of 1.0 mm voxels of 18 gigabytes. Reducing the system matrix to only those rays that can

contribute to the field of view in the area support region reduces this value dramatically to approximately 1.4 gigabytes.

2.6 Algorithm Performance

Performance testing for the OVIA algorithm was performed on an DELL D7400 System with an Intel X5450 CPU clocked at 3.0 GHz, 16 GB of main memory, a 1TB disk array, and an NVIDIA Quadro FX 1700 graphics card. The performance gain of the optimized algorithm is summarized in Table 1.

A large performance advantage was gained by eliminating the memory bottleneck through the use of the OVIA volume walk method. Elimination of the memory access greatly improves system performance. The instruction sequence maintains a tight loop, allowing the CPU to perform well and resulting in a performance improvement of around 5000:1.

Overall, GPGPU performance was not high enough to be considered for general use in this case, given current technology. This is likely due to slow local memory access and high overhead. GPU applications are best suited to tasks that require calculations to deposit in framebuffer memory for display. The next generation of GPU from nVidia to replace the Quadro FX 1700 multiplies the memory throughput performance by a factor of 3 from 12 GB/sec to 38 GB/sec. As this technology continues to evolve, it is expected to better satisfy the needs of this algorithm.

Streaming instruction performance turned out to be as expected with a 2X improvement over the plain CPU instructions. Even though SSE could theoretically show a 4X improvement as the vector length is 4, in practice a 1.5X to 2.5X improvement is considered normal due to overhead [Kaldeway, 2007][Hennessey, 2006], with speedups in the 1.5X range for less dense operations like matrix multiply and greater speedups in denser signal processing type computations. Since many commercial CPUs support SSE in some form, this makes it a very cost effective solution.

In the higher performance techniques, the disk write is the largest single factor, requiring 8 seconds. On our development systems, caching the system matrix to disk is entirely optional as it can be computed in a similar time as it would take to read from disk. However, on the high throughput production systems with their very fast disk subsystems, this is less than 5 seconds and is considered inconsequential.

Table 1. Volume Intersection Approximation Calculation Performance.

Algorithm	Time
Baseline memory based VIA Calculation	61,300 sec. (est from subset)
OVIA on CPU, inline voxel address calculations	13 sec.
OVIA on GPU	14 sec.
OVIA on CPU with SSE2	9 sec.
Time to write to disk	8 sec.

Chapter 3

Reconstruction Computation

3.0 Simultaneous Iterative Reconstruction Technique

Simultaneous Iterative Reconstruction Technique (SIRT) as described by Kak and Slaney [Kak, 2001] is used to to perform iterative reconstruction on the log normalized projection data. The basic idea behind this method was first proposed by Gilbert in 1972 [Gilbert, 1972] although other variants have been proposed [Oppenheim, 1975][Gregor, 2008].

As mentioned in Chapter 1, a linear relation exists between the unknown image of attenuation coefficients and the log-normalized projection data. Representing the former by vector x and the latter by vector b, while using matrix A to connect the two as outlined in Chapter 2, we obtain the linear system of equations:

$$Ax = b \tag{3.1}$$

SIRT formulates the solution x* as a matter of solving the weighted least squares problem:

$$x^* = argmin \parallel Ax - b \parallel_R^2 \tag{3.2}$$

where R is a diagonal weight matrix described below. Expanding the residual error norm, taking the derivative with respect to x, and setting equal to zero yields:

$$\frac{\partial}{\partial x}(Ax - b)^T R(Ax - b) = 0 \tag{3.3}$$

$$A^T RAx - A^T Rb = 0 \tag{3.4}$$

Preconditioning the normal equations by a matrix C, defined later leads to:

$$CA^T RAx = CA^T Rb \tag{3.5}$$

By using matrix splitting, adding and subtracting the identity matrix I associated with CA^TRA and rearranging some terms, the form can be changed as follows:

$$CA^T RA = I - (I - CA^T RA) \tag{3.6}$$

Substituting the value of CA^TRA from (3.6) into (3.5) and rearranging some terms yields:

$$x^{k+1} = (I - CA^TRA)x^k + CA^TRb \tag{3.7}$$

With some additional rearranging we arrive at the stated vector matrix representation for SIRT:

$$x^{k+1} = x^k + CA^TR(b - Ax^k) \tag{3.8}$$

where R and C are defined as diagonal matrices containing the reciprocals of the row and column sums of A respectively.

Nominally, (3.8) is expressed with individual components and temporary indexing variables i, j, h, and k as:

$$x_j^{k+1} = x_j^k + \frac{\sum_i [a_{ij}\,(b_i - \sum_h a_{ih}x_h^k)/\sum_h a_{ih}]}{\sum_i a_{ij}} \tag{3.9}$$

where a_{ij} represents the individual components of the system matrix A for ray *i* and voxel *j*, and b_i represents the log normalized projection data for the ray *i*.

3.1 Row and Column Sum Matrices

The matrix R consists of a diagonal matrix containing the reciprocals of the row sums of the system matrix A. The use of R within the SIRT algorithm has the purpose of normalizing the values of the individual voxel contributions along a ray. After the application of R, only the relative magnitudes of the voxel contributions are part of the contribution. Thus, there is one value for each ray projected through the volume.

The matrix C is a diagonal matrix containing the reciprocals of the column sums of the system matrix A. This matrix represents the sum of the contributions of all projections that pass through each voxel in the image volume. Thus, there is one value for each voxel. Histogramming the values of the column sums (Figure 21) is indicative of the complexity of the geometry. Ideally, since there are 320 angles in this case, every voxel would be visited uniformally 320 times, and all column sums would equal 320. Since some rays only slightly touch some voxels, thus the ideal case is reduced more for some voxels than others. Non-uniform sampling further distorts the picture as shown in Figure 22. Since this difference in coverage is compensated for with the column sum matrix C, all voxels within the volume will reflect their correct value after convergence. However, some voxels may take longer to converge depending on the number of rays that pass through that voxel and the subject being imaged.

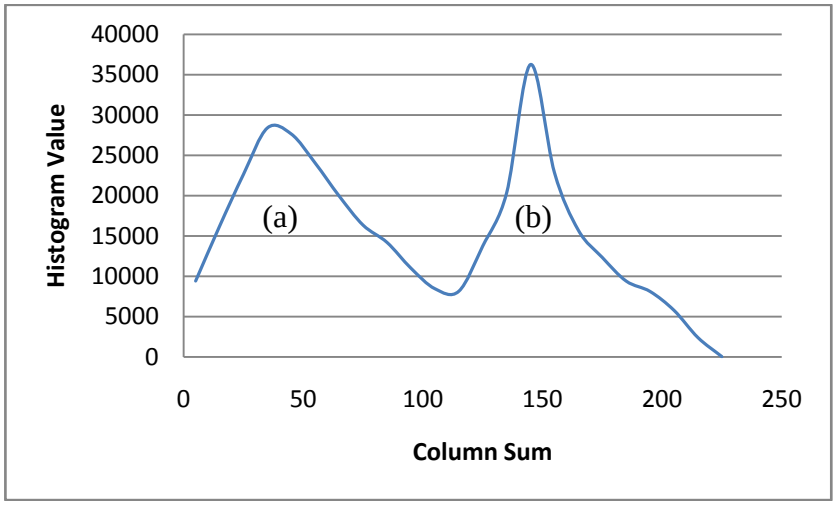

Figure 21. Histogram of column sums in the entire field of view.

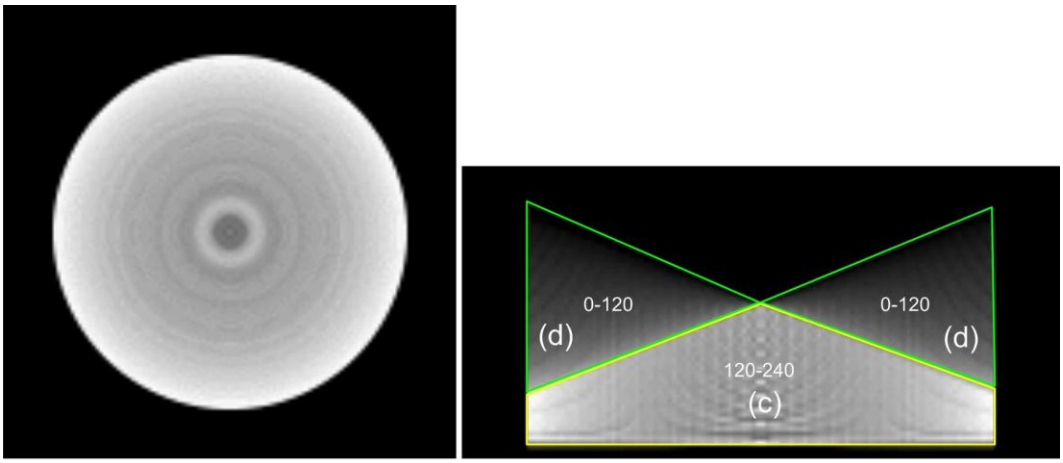

Figure 22. Colum sum images show variation in coverage.

The transaxial view in Figure 22(a) shows the limits of the image volume radially, and the coronal view in Figure 22(b) shows the dual fan nature of the geometry for a single bed position.

In areas that are covered by all projection angles (Figure 22(c)), the column sums correspond to Figure 21 area (b). The leftmost distribution of Figure 21, area (a), pertains to the locations within the matrix that are only covered by a portion of the projection angles (Figure 22 (d)). In addition, the 16-sided nature of the detector array can be seen as 16-lobed interference patterns in Figure 22(a). This is not a failure, it is simply an expression of the normalization required to compensate for the specific geometry. This effect causes the sharp peaks seen in the histogram (a) and (b) in Figure 21. The reason that these are sharper in the rightmost distribution (b) of the histogram is because geometrically they occur at the center, and are thus double-covered. Narrowing the system model down to just the volume with full 360 degree support, the column sums vary less (Figure 23).

Multiple bed positions are required to sample the entire field of view. The DPET is programmed to do this in 8 bed steps of 16 mm each. As the fan overlaps at each step, the entire field of view is covered. A single bed step is shown in Figure 24.

As the bed is moved, more of the field of view is sampled. This process is shown in Figure 25.

Reconstructing small sections independently does not make best use of all information. In order to take advantage of as many of the measured projections as possible, the entire image volume is reconstructed as a single unit, with all projection measurements contributing to as much of the image volume as they cover. This process is shown in Figure 26.

The multiple fan coverage generated through the whole volume computation complicates the column sum considerably. The coverage of the field of view becomes very non-uniform as shown in Figure 27. The single independent reconstruction field of view isn't uniform either, but the combination of overlapping scan fields creates much greater variation in the column sums in the whole volume method. This non-uniformity is expressed in the matrix C in the SIRT equations, and is corrected. However, those voxels with less coverage may converge slower than the surrounding voxels depending on the object in the field of view, so it is important to iterate until full convergence is achieved, or apparent artifacts may result, in this case banding behind the direct planes in the fan.

In conclusion, it was deemed necessary to compensate for column sum on a voxel by voxel basis. Further validation of this effect is discussed in Chapter 4 in the section on statistical testing. The column sum is computed from the system matrix during the first iteration, held in memory, and applied during future iterations as needed. This adds approximately 2 seconds to the total computation.

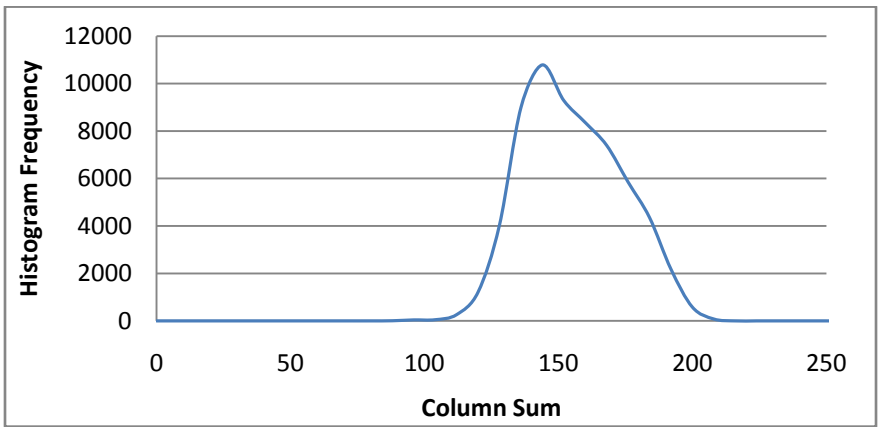

Figure 23. Histogram of column sums in the full support region only.

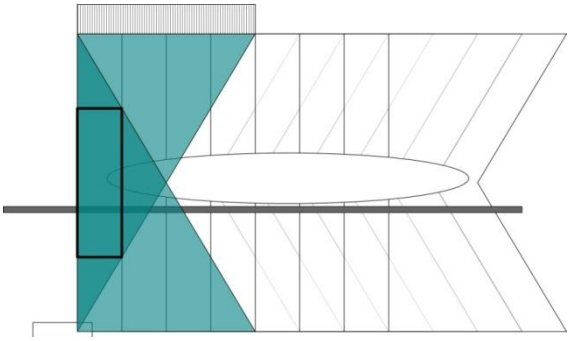

Figure 24. Fan coverage for a single step within the field of view.

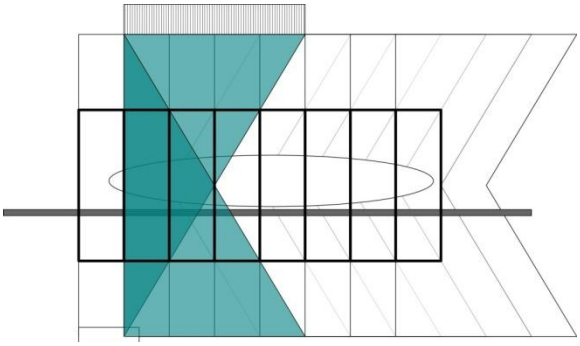

Figure 25. Multiple independent bed steps stitched together.

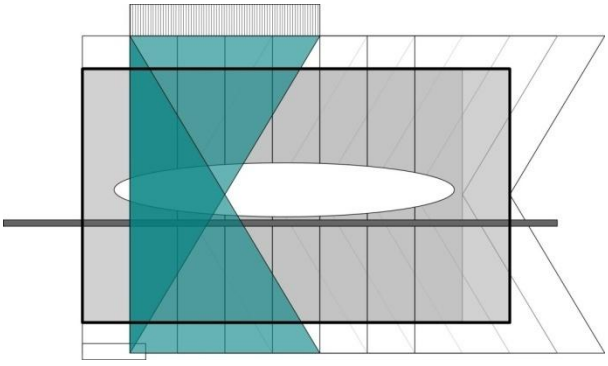

Figure 26. Fan contributions to the entire volume in a 3D approach.

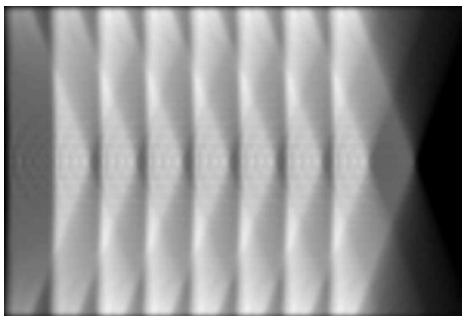

Figure 27. Column sums combined for multiple bed steps shown axially in the FOV.

3.2 Optimization of total performance

There are two principal aspects to the optimization of any iterative computational problem. First, the algorithm itself must be mathematically stable and tuned to converge in the least possible number of iterations. This reduces the work necessary to achieve the desired end result. Second, the iterative portion of the computation must be tuned to function with the available hardware to the largest degree possible. Knowledge of system architecture is key to achieving this result. In the end, it is the product of all optimizations, both algorithmic and systemic, that achieve the full result.

When discussing convergence acceleration, a comparison is made from the base approach to the approach under test. For this comparison, a plot is made that compares the residual error norm of one approach on one axis, and the second approach on the other. The slope of the line thus formed is the speedup factor. There are some potential errors in determining speedup with this approach, namely, the differences can be very small, so computing the first derivative (slope) can be unstable. For this reason, it is important to choose an object to reconstruct that challenges the algorithm. Typically reconstructed objects like cylindrical or uniform phantoms, or even actual subjects converge too quickly for use in a convergence comparison, thus making it difficult to determine a comparison slope accurately. In all convergence comparison cases, we use the stress testing synthetic phantom that will be introduced later in section 4.1.2. It contains wide variation in object density and orthogonal geometric shapes that cause difficulty in convergence.

3.2.1 Relaxation

Adding relaxation to SIRT changes (3.8) with the addition of the relaxation constant α.

$$x^{k+1} = x^k + \alpha C A^T R (b - A x^k) \tag{3.10}$$

This is consistent with the Richardson Iteration [Saad, 2000], a classical method of solving linear systems of equations that takes the form of:

$$x^{k+1} = x^k + \alpha (v - U x^k) \tag{3.11}$$

The optimal value of relaxation constant α, i.e., the value that leads to the fastest rate of convergence, is one over the average sum of the largest and smallest eigenvalues of matrix U [Saad, 1996]. Gregor and Benson [Gregor, 2008] observed that CA^TRA is a non-negative, double-stochastic matrix which implies that its largest eigenvalue is 1.0. They further argued that the smallest eigenvalue is likely to be substantially smaller than the largest eigenvalue for many image reconstruction problems. Setting α to a value close to 2.0 thus has great potential for leading to a faster rate of convergence than the default value of 1.0. The performance improvement is shown below in Figure 28.

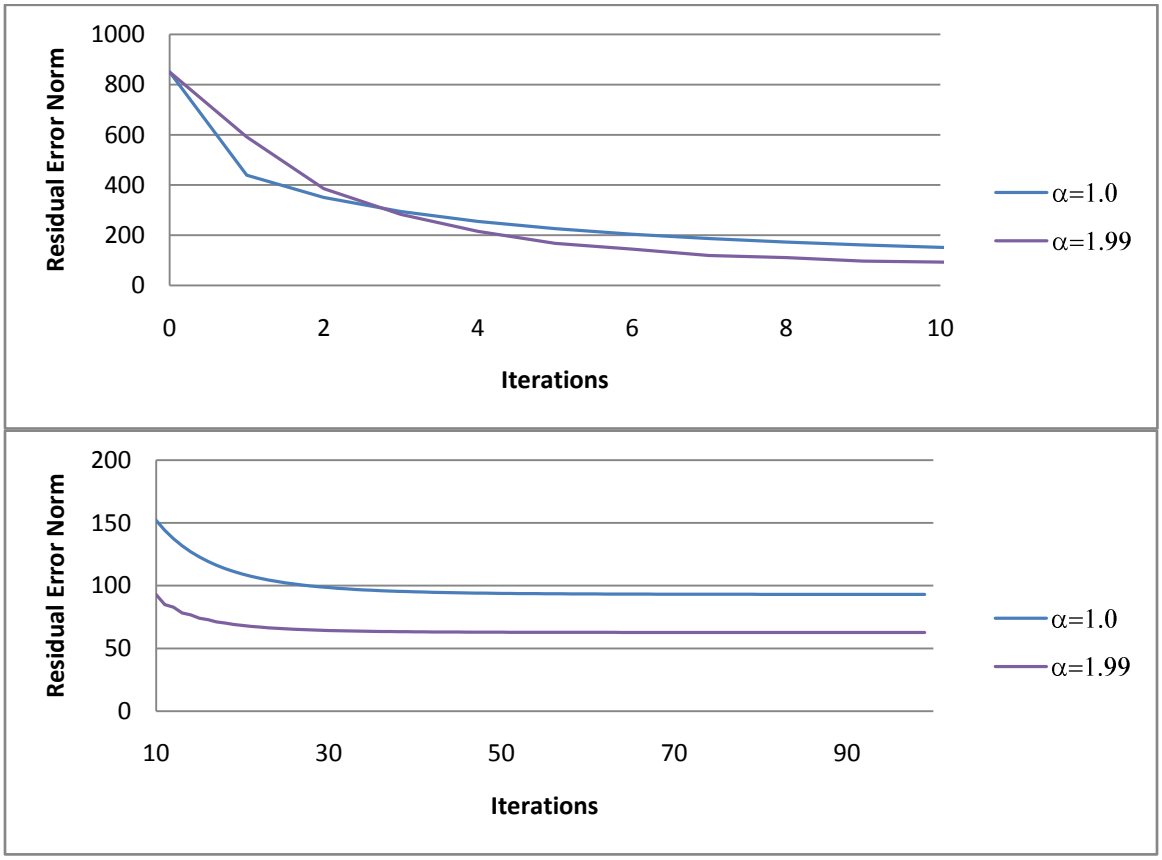

Figure 28. Convergence with respect to relaxation constants 1.0 and 1.99.

Initially, the rates are not significantly different, but further out in the computation after iteration 20, the higher relaxation factor improves the rates by a factor of 2X.

Comparing the residual error norms of different runs with relaxation constants to the standard case without relaxation yields the graph shown in Figure 28Figure 29. Initially all relaxation levels yield fast convergence, but as the algorithm needs to apply finer and finer changes, the higher relaxation constants make for a larger difference and get the solution closer to the final solution in fewer iterations.

The slope of the line in Figure 29 represents the speedup factor as referenced to the case without relaxation (a relaxation factor of 1.0). A tabulation of these values is shown in Table 2. Iteration was performed until the residual error norm reached a stable state, less than 2% change per iteration). The relaxation value of 1.99 gives better performance without the risk of failed convergence. Selecting a relaxation value of less than 1.0, or 2.0 or greater would alter the spectral radius such that convergence could no longer be analytically guaranteed. Overall, a speedup factor of 2X can be achieved using this approach.

3.2.2 Ordered Subsets

Hudson and Larkin showed the mathematical basis for ordered subsets [Hudson, 1994]. More specifically it has also been shown that ordered subsets can be applied to iterative transmission techniques [Kamphuis, 1998]. The overall concept involves choosing a subset of the projection data to update the image volume estimate. This divides the projection dataset into multiple subsets where a full iteration is completed only after all subsets are processed. Figure 30 (a) shows the linear pie ordering approach, ordering subsets into simple pie slices. This approach does not work particularly well as all the angles in each slice contain similar views. Linear spoke ordering, shown in Figure 30 (b), selects one angle from each pie slice and combines those into a subset, then chooses the next angle in each slice, etc. This method provides a more orthogonal mix and thus a better update estimate.

In practice, projections that are close to one another do not contribute substantially different update information. By selecting a subset of projections that are orthogonal in terms of content and treating that subset as an independent sub-iteration, a complete estimate of the final image volume can be computed in a much smaller length of time. This improvement in estimated image volume accuracy results in further gains in the downstream processing, accelerating the entire process.

By definition, SIRT uses a single subset. All projections are forward projected from the same image estimate before corrections are then sequentially back projected into the image volume. By modifying SIRT to use ordered subsets, thus OSSIRT, the update rate is improved. For example, if 16 subsets are chosen, the image volume is updated 16 times per iteration. When the number of subsets is equal to the number of projection angles (thus one update per angle), OSSIRT becomes analogous to the Simultaneous Algebraic Reconstruction Technique (SART)[Kak, 2001] in terms of update

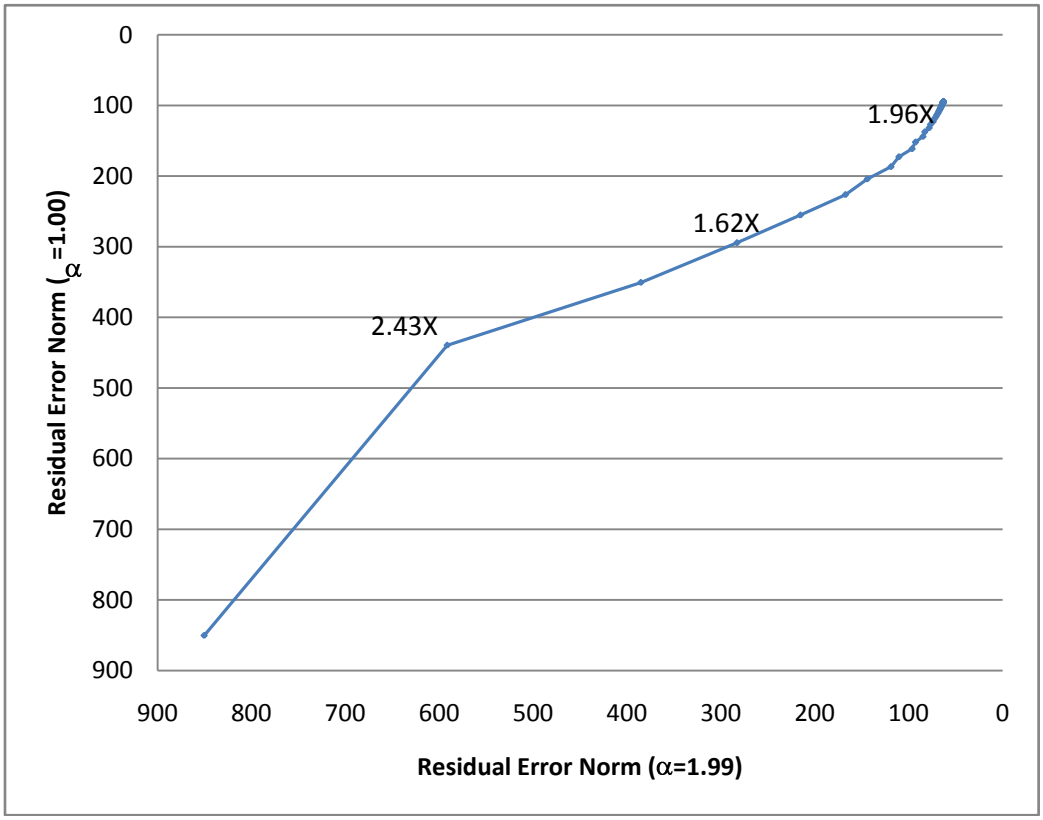

Figure 29. Convergence of maximum relaxation (α=1.99) vs. no relaxation (α=1.0).

Table 2. Relaxation Factor Speedup Results.

Relax Factor	**Iterations to Stable State**	**Residual Norm**	**Speedup**
1.0	75	47.70	1.0X
1.99	35	45.82	1.96X

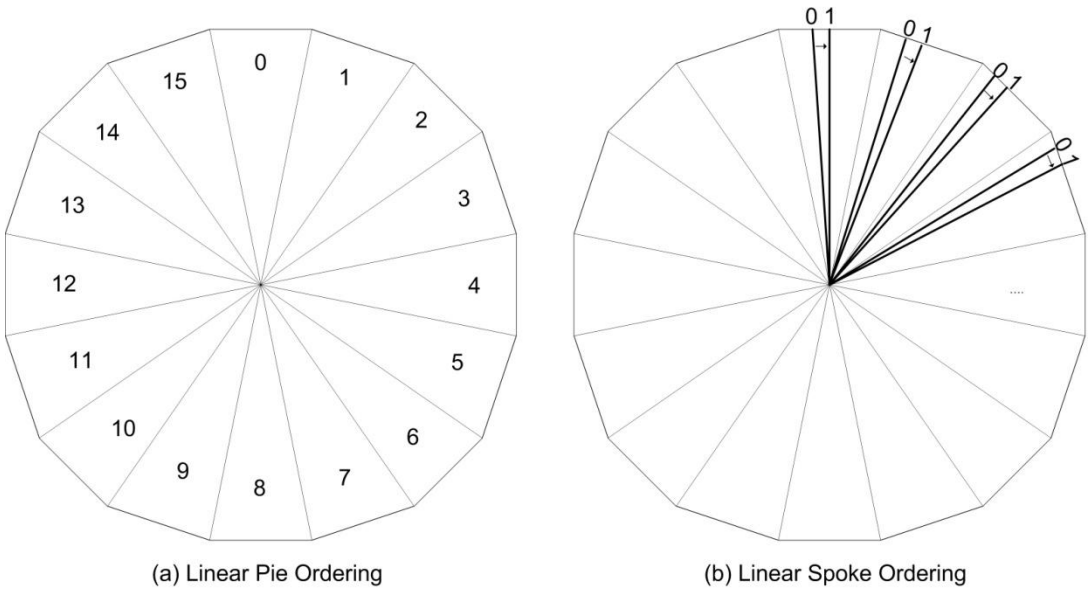

Figure 30. Ordered Subsets, linear pie ordering and linear spoke ordering.

methodology. By definition, SART updates the estimated image after processing each projection angle.

Projection data can be divided into an arbitrary number of subsets. As the number of subsets is increased, convergence rates increase, to a point, as shown in Figure 31. If the number of subsets is too large, then the number of projections within each subset will be insufficient to independently produce an accurate image estimate, and convergence rate will decrease. In our case, memory constraints limit us to 16 subsets, and improved convergence is shown up to that level.

By comparing the convergence rates of the 1 subset case to the other subset cases, relative comparisons can be made concerning convergence speedup. Plotting the 1 subset error norm on the Y axis, and the others on the X axis, yields the graphs shown in Figure 32, 33, and 34. The slope of the convergence comparison determines the speedup factor for the particular number of subsets. In this case, the ordered subsets method is more important further into the computation, when the necessary changes are small, and this can be seen in the progression on the comparison graphs as the slope turns more steadily upward (faster), until it asymptotically approaches the individual limits. In this way, speedup factors are determined for 4, 8, and 16 subsets of 4X, 8X, and 16X respectively.

The results of the convergence testing are summarized in Table 3. We choose an arbitrary image quality standard of 80 iterations, and note that the standard is achieved (or exceeded in this case) in the 4, 8, and 16 subset cases, with 4X, 8X, and 16X fewer iterations. In all cases, this yields a stable final state where the residual error norm is changing less than 2% per iteration.

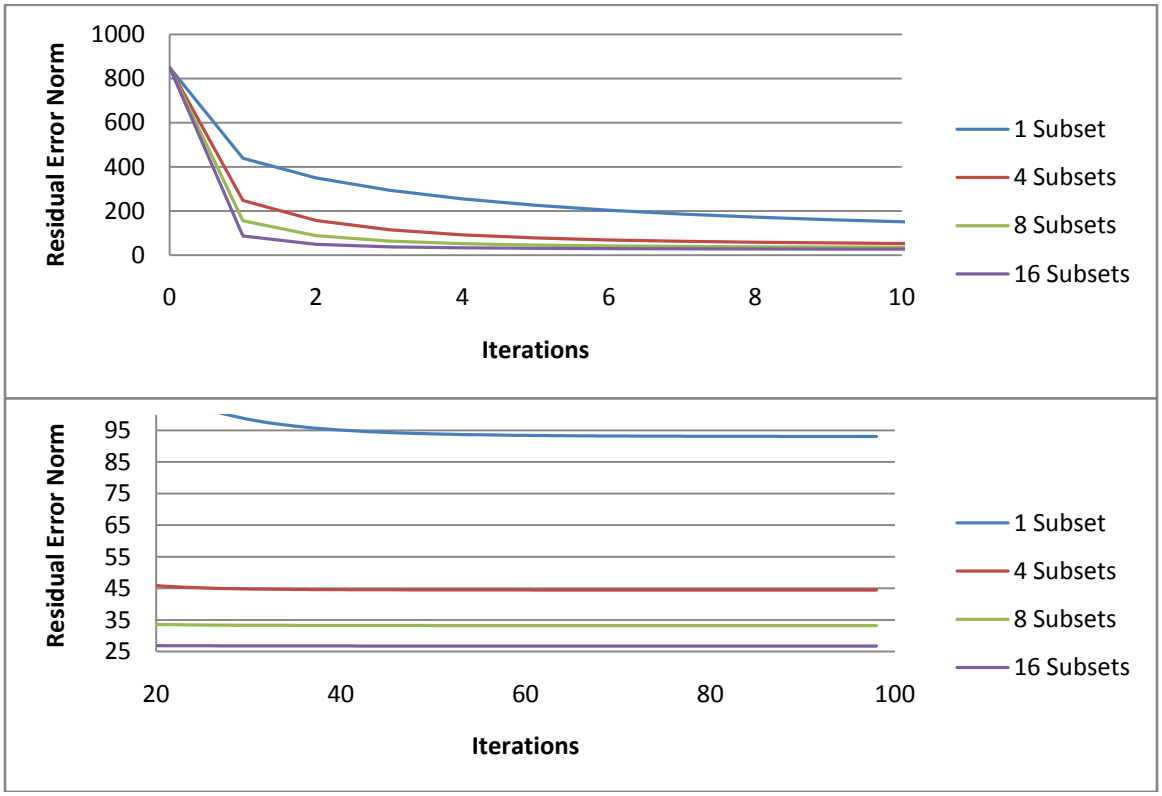

Figure 31. Convergence behavior for 1, 4, 8, and 16 subsets.

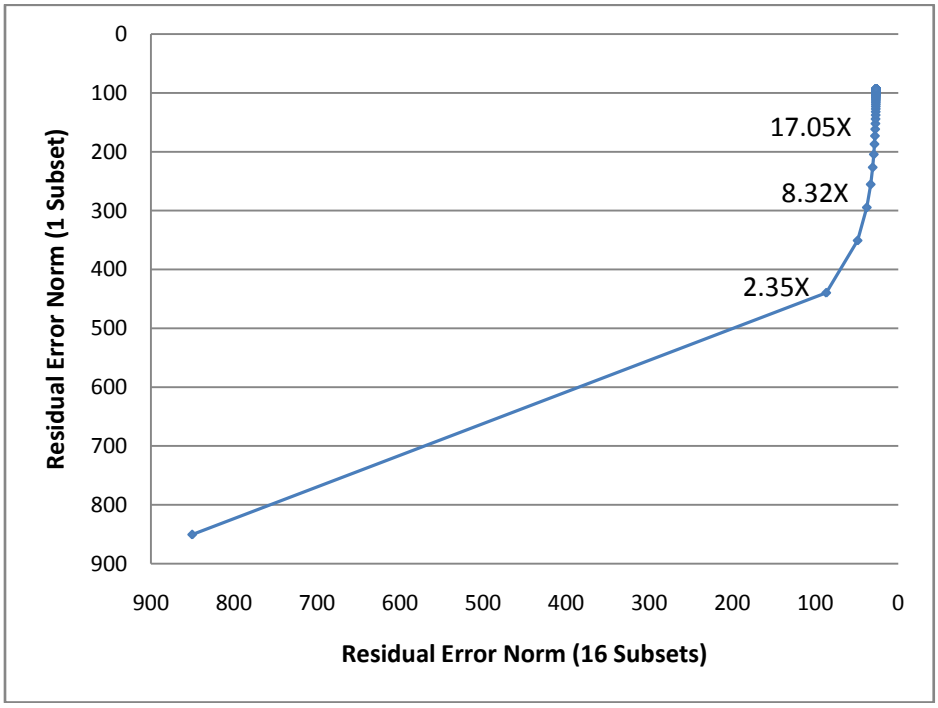

Figure 32. Ordered Subsets Convergence Comparison, 16 vs. 1.

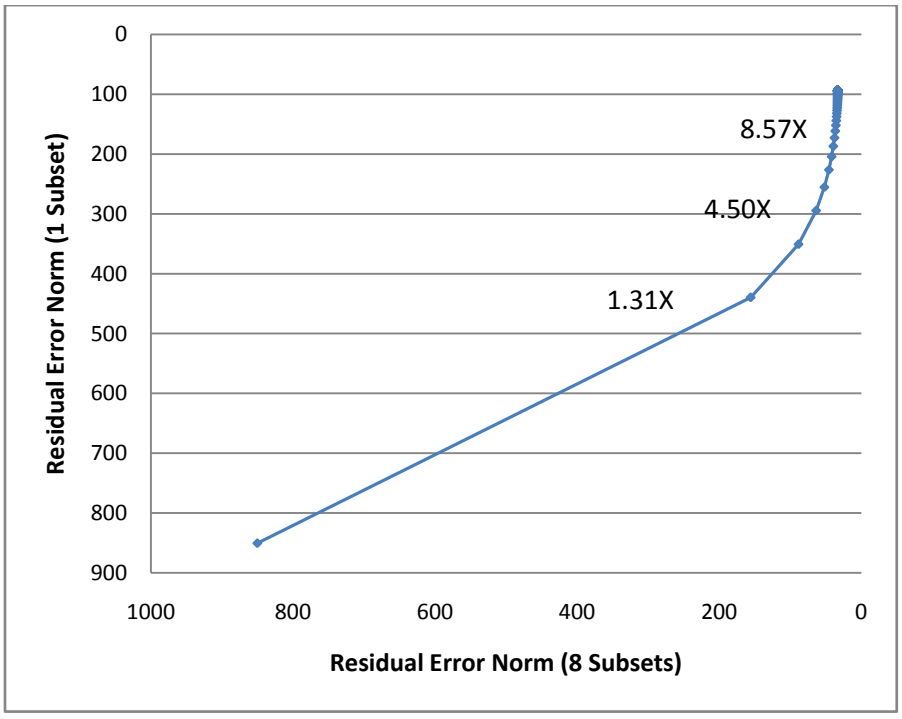

Figure 33. Ordered Subsets Convergence Comparison, 8 vs. 1.

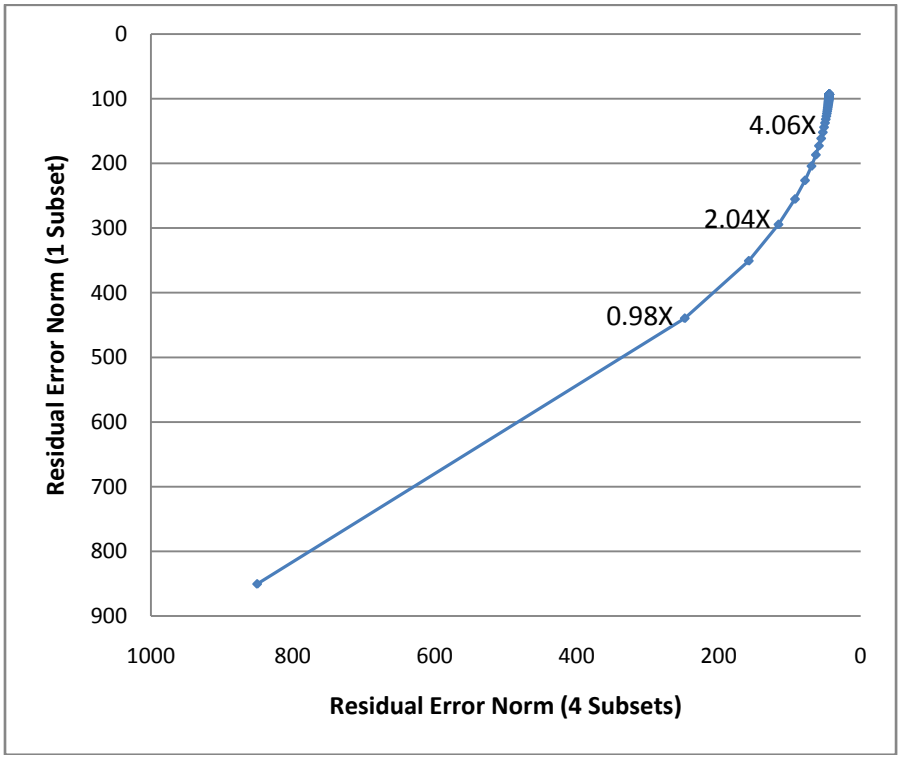

Figure 34. Ordered Subsets Convergence Comparison, 4 vs. 1.

Table 3. Subset Acceleration.

No. Subsets	Iterations to Stable State	Residual Norm
1 (SIRT)	80	93.1
4	20	44.4
8	10	33.2
16	5	26.7

The end result of the entire ordered subsets technique is the reduction of the number of iterations necessary to achieve a particular image quality goal. It should be noted that the residual error norms are not exactly the same. The starting error norm in this case is approximately 1000, and while the residual norms differ, the corresponding images do not show visible differences.

3.2.3 Subset Order

In his early work in CT reconstruction, Hounsfield noticed that convergence was improved by arranging the processing order to use the most orthogonal projections [Hounsfield, 1972][Kak, 2001]. He attributed the speedup to the high correlation of information in adjacent projections. This was later proven mathematically [Ramakrishnan, 1979]. The selection of subset order can be determined in many ways, even randomly, but ideally, by keeping the subset updates as orthogonal as possible, convergence is improved. Selection of subset calculation order does not change equation (3.10) in itself, it merely describes the approach to dividing the data and the rate of update.

A recently suggested quasi-orthogonal approach is based on a Fibonacci series [Kohler, 2004]. The technique is based on a natural observation in plants whereby the sequential leaves of the plants form at angles of approximately 222.5 degrees to maximize their sun exposure. This is referred to as the Golden Ratio. In this particular case, with 320 angles, the selection delta is 200. The next angle selected is determined by adding 200 to the current angle, modulo 320. This results in a sequence of 0, 200, 80, 280, 160, etc. If the number of angles available in the dataset to be reconstructed is divisible by 222.5 degrees, then all of the angles are correctly sequenced. This is the case in this particular geometry, as there are 320 detectors in the ring, representing 320 angles. For a non-modulo number of angles, such as 360 for example, rounding and resorting must be used to correctly determine the update order, and it is no longer precise.

A more truly orthogonal approach selects the next angle to compute based on the angle section it is currently working on and what is left to compute. First angle zero is computed, followed by angle 180 degrees. Next the two hemispheres are divided with computations at 90 and 270 degrees, ad infinitum. There is a possibility due to the section splitting depending on the number of projections that can cause conflicts in the selection of the next angle. That conflict can be resolved by selecting from the lower angle side, or the higher angle side consistently. Such orthogonalization can be implemented using a priority queue to sort update order (implemented as a heap) [Cormen, 2006]. As each angle is processed, the next orthogonal components to be computed are added to the queue and then withdrawn in the order processed. Finally, we note that, unlike the Golden Ratio method, the priority queue approach continually

divides the search space in a symmetric fashion (the largest open angle is never more than double the smallest). Also, it will function correctly for any arbitrary number of angles without the need to post-correct the result.

Both the orthogonal and Kohler's golden ratio update orders were implemented. Performance evaluations shown in Figure 35 and Figure 36 show no significant performance differences between the two approaches in this case. Thus, by using the orthogonal approach, we are guaranteed a precise update sequence without concern for the number of projection angles.

3.3 Parallelization

Dividing the work involved to compute equation (3.10) on parallel hardware ordinarily requires the use of a mutex to control access to portions of memory during the update to prevent conflict between different rays [Dongarra, 2003]. The use of a mutex in this way is very expensive computationally. Instead, this work focuses on the upfront separation of rays into multiple groups that do not conflict, then executing them in an alternating fashion. This allows the use of only a single synchronization event in the middle of the calculation to ensure no conflicts will occur.

The forward projection step is completely independent of parallel dependencies, as all lines of response are computed from a stable image space. Parallelization of this step is straight forward, spinning multiple threads to compute independent ray forward projections without regard to synchronization.

The back projection step has memory contention issues in that sections of image space are being updated continuously, and the potential exists for multiple threads to overwrite each other. Due to the divergent nature of the system matrix, it can be shown that adjacent lines of response do interfere, but a separation of some arbitrary amount in the detector relieves this restriction.

A conflict occurs when two rays share voxel updates. An illustrative example of this type of limitation is shown in Figure 37. If separate processors are performing an operation where this conflict occurs, the result is not deterministic, as one processor may interfere with the other. In order to resolve this conflict, a mutex is normally used to lock the area of memory in question while one processor performs the update and the second one waits. If two lines of response have no common voxels, they can be computed simultaneously without regard to locking the memory volume in question. By searching the system matrix for commonality in voxel addresses, the minimum separation necessary to relieve the conflict can be determined. In the case of this particular geometry, a separation of one pixel in the detector prevents any conflict. From a practical perspective, this means that the entire set of rays can be divided into multiple disjoint sets. All CPU cores are given two sets, A_i and B_i. All sets A_i are disjoint from each other, and all sets B_i are disjoint from each other, but all sets of A_i are not necessarily disjoint from all sets of B_i. When the process starts, all CPU cores start work on their

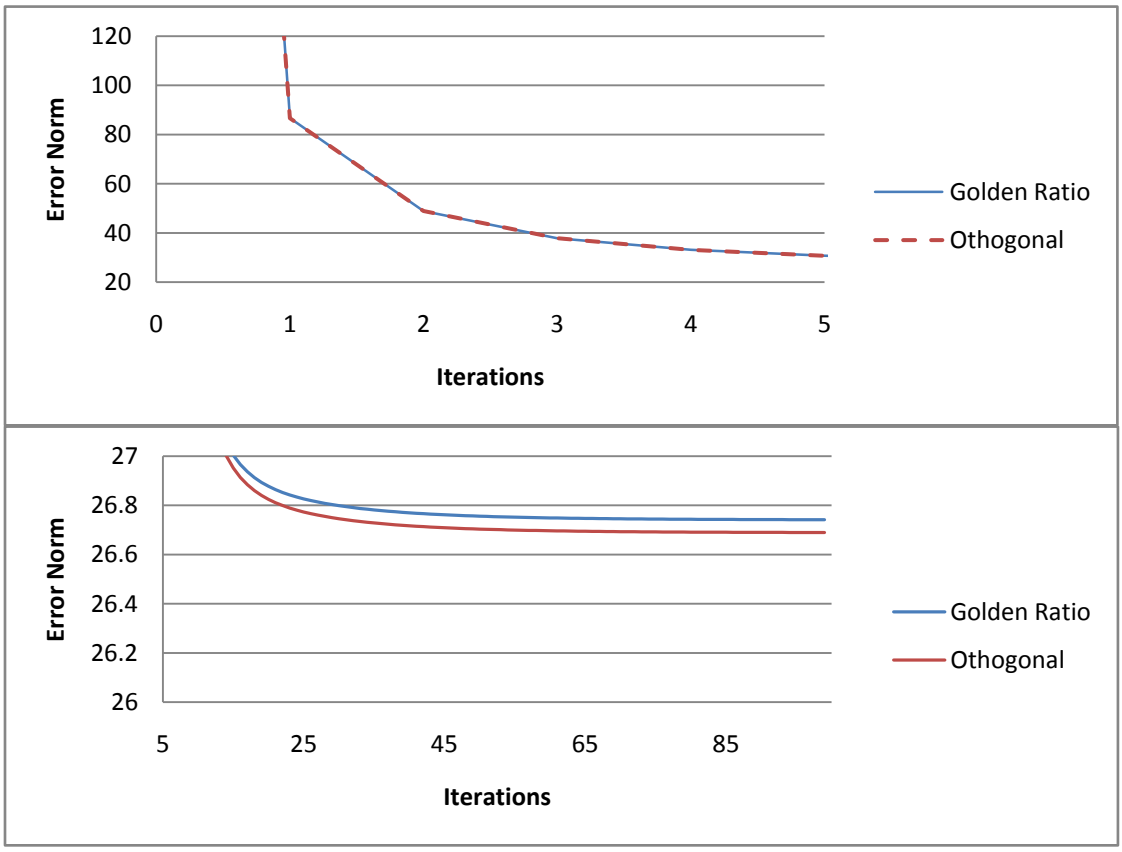

Figure 35. Convergence for Golden Ratio and Orthogonal Update Schemes.

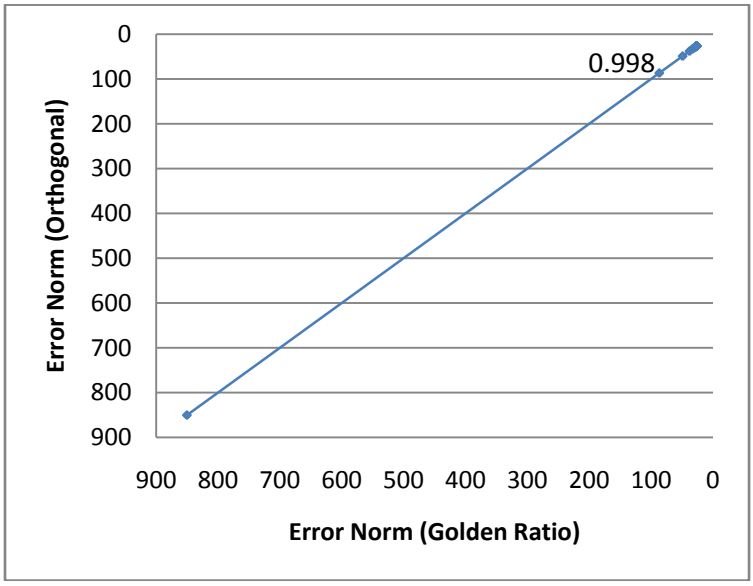

Figure 36. Update techniques convergence comparison golden vs. orthogonal.

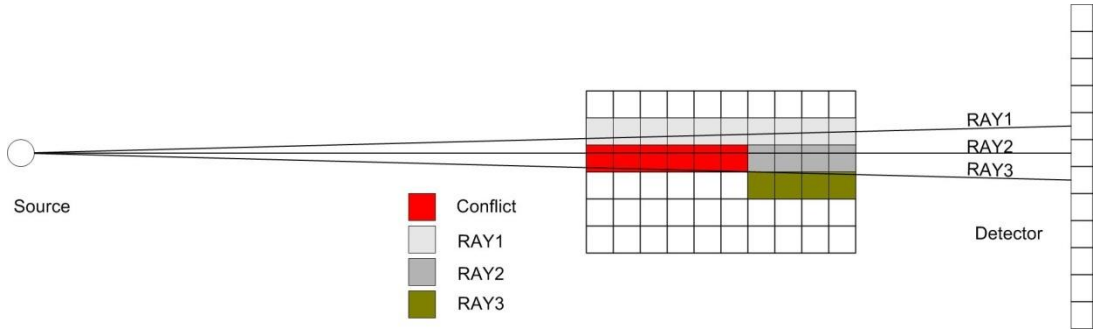

Figure 37. Conflicting update requirements in the system matrix require mutex locks.

respective A sets, and when finished, they jointly signal each other to start the set B. This method allows a single synchronizing point in the middle of each iteration, not millions of potential stopping points and semaphore checks. Under these conditions, multi-core operations are efficient if loaded equally, and this can be assured by forcing the set sizes to be equal.

3.3.1 Multi-core Implementation

Commonly available multi-core CPU systems can provide significant improvements in calculation time. Modern trends in computer system design are tending toward multi-core designs placing multiple CPU cores on a single system [Intel, 2009][AMD, 2009]. Currently available systems typically employ a symmetric multiprocessing architecture which allows all processors equal memory access. This can be advantageous when working with iterative algorithms with large data sizes in that only a single copy of the larger portions of data need be resident.

Dividing the whole volume computation described earlier into sections by bed position gives a convenient organization for up to 8 cores. If fewer cores are used, each core can work on multiple beds. If more cores are available, each bed can be split into sections by angle with care to protect against conflict (Figure 37). No additional memory is required to store the update image for each core, and thus there is no additional processing required to sum all of the update images into the final image at each iteration.

If the column sum normalization is applied to image space during the back projection step on a voxel by voxel basis, it can cause roundoff errors that will impose a column sum like variation on the final image. This occurs because if the column sum is applied during the individual ray projection step, then a relatively small weight is repeatedly being divided by a significantly (two orders of magnitude) larger column sum. By summing all back projection updates into a single update volume, and dividing all voxels by their appropriate column sums once, only a single division is performed. This reduces the computational complexity as division is expensive, and improves the accuracy by reducing the potential for roundoff. Unfortunately, this is a non-parallelizable step, and will not contribute to speedup in multi-core implementations. This overhead, however, is quite small and grows only linearly with the addition of subsets. For this reason, we chose to use this method of update. A diagram of the data flow used in this approach is shown in Figure 38.

3.3.2 Multi-core Performance

For a single example executed on the standard DPET system processing console (3.0 GHz, Quad Core Intel X5450, 16 GB RAM), with 1, 8, and 16 subsets, the performance improvements for multiple processors and subset configurations are shown in Table 4.

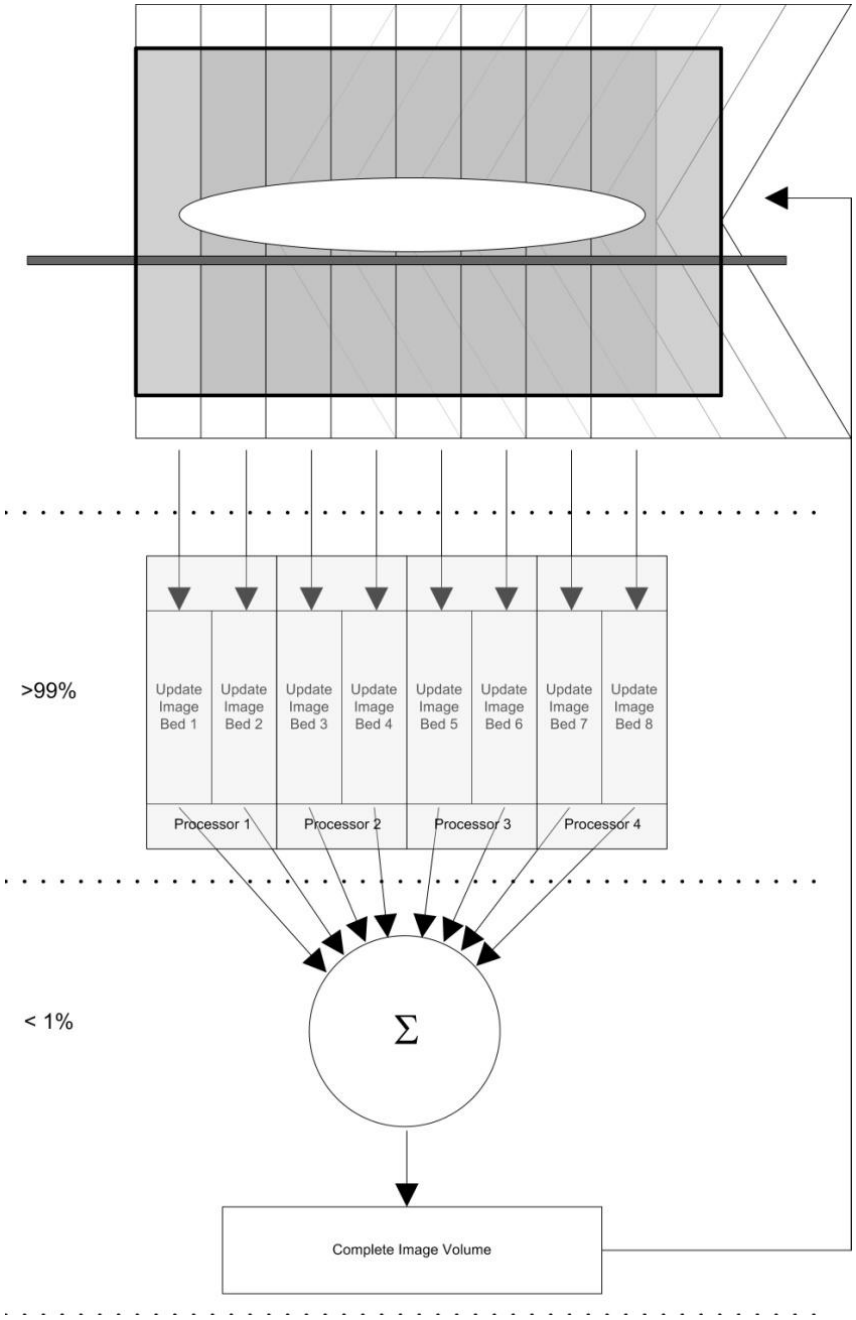

Figure 38. Data flow for multi-core reconstruction.

Table 4. Multi-core Performance.

Proc	Subsets	Calc Time per Iteration Sec.	Calculation Speedup Factor	Sync Time Sec.	Total Time per Iteration Sec.	Total Speedup Factor
1	1	11.7	1.0X	0.017	11.7	1.0
2	1	6.4	1.8X	0.018	6.4	1.8
4	1	4.5	2.6X	0.021	4.5	2.6
1	8	11.7	1.0X	0.15	11.9	1.0
2	8	6.4	1.8X	0.16	6.6	1.8
4	8	4.4	2.7X	0.17	4.5	2.6
1	16	11.7	1.0X	0.33	12.0	1.0
2	16	6.5	1.8X	0.34	6.8	1.8
4	16	4.3	2.9X	0.34	4.6	2.6

The speedup achieved from 1 to two processors (1.8X) is a reasonable result, but the speedup from 2 to four processors (2.6X total) does not meet expectations. The overall level of synchronization overhead should not degrade the total result to this degree, as it is not particularly large, and does not change depending on the number of processors, only on the number of subsets.

The reason why performance is limited can be determined by looking at overall memory throughput. The system matrix contains 175 million voxel interactions that are invoked in every 360 degree rotation around the field of view. In a forward projection operation, three memory operations are required: read the voxel weight (4 byte float), read the voxel index (4 byte int), and read the voxel value (4 byte float). This totals 12 bytes, or 2.1 GBytes of memory transferred after one rotation. In a backprojection operation, four memory operations are required: read the voxel weight (4 byte float), read the voxel index (4 byte int), read the voxel value (4 byte float), and write the new voxel value (4 byte float). This operation totals 16 bytes, for a total of 2.8 GBytes of memory transferred after one rotation. Thus the total memory transferred per 360 degree rotation is:

2.1 GBytes + 2.8 GBytes = 4.9 GBytes/rotation

There are 8 bed positions to be considered during the process to cover the entire field of view for one iteration, thus:

4.9 GBytes/rotation * 8 = 39.2 GBytes/iteration

By dividing the total amount of memory moved though the system by the time the system actually used to perform the operation, a conservative estimate of memory bandwidth can be determined. Cache hits will improve these values, however, the system matrices at 1.4 GB are much larger than the L2 cache at 12 MB, so it is expected that the caches are overwhelmed. The results of this calculation are shown in Figure 39.

From the datasheet on the Intel X5450 CPU used in this test, the front side bus operates at 333 MHz, and is 4X clocked to raise the transfer frequency to 1333 MHz for a maximum memory bandwidth of 10.66 GBytes per second [Intel, 2009]. As the number of processors increases, the demand for memory increases as well, and the memory bus becomes saturated.

Ordinarily, one would argue that this is indicative of poor overall algorithm design. However, the design criteria in this case is to optimize performance for the standard system, and that system is a quad core Intel X5450 processor. One could argue that instead of performing a memory lookup, the values could be calculated using the OVIA method independent of main memory and achieve greater speedup. From Chapter 2, calculating the system matrix requires 9 seconds for a single processor, and the total time required to calculate a single iteration for a single processor is 12 seconds. Thus a total of 9 + 12 = 21 seconds of processing work (single processor) is required per iteration under this scheme. If that work product achieved a consistent speedup factor that was the same as from 1 to 2 cores (1.8X), in order to achieve the same level of

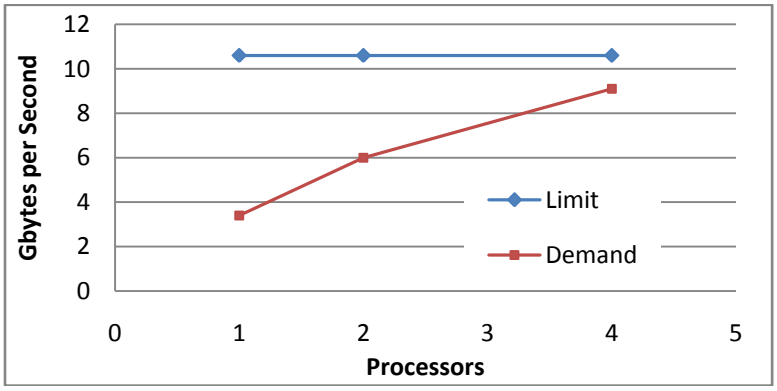

Figure 39. Memory Bandwidth Requirements.

performance achieved on the quad-core system (4.6 seconds), 8 cores would be required. Thus, an 8 core system would need to be procured to achieve a similar performance goal. For reference at the time of publication, a Dell T7400 quad-core currently costs approximately $2,500, and a Tyan 8-core system costs around $6,000. So, for approximately 2X the cost, similar performance can be envisioned. The point of this exercise is to demonstrate that computers are made up of a variety of resources: CPU cores, memory, disk, and bandwidth. The ideal optimization is one that makes maximum use of all available resources.

3.4 Summary of Reconstruction Performance Improvements

By combining both the numerical convergence advantages with the architectural compute performance improvements, a significant speedup was achieved, making the total computation acceptable from an operational point of view, as it requires less time than the currently used approach. All approaches have a constant level of startup overhead of 7 seconds, relaxation is turned on, and the whole volume is computed simultaneously. This includes time to read the raw data from the disk, and compute the column sum normalization matrices. The overall level of performance improvements for datasets of similar quality is summarized in Table 5.

This approach yields a computationally efficient solution in 30 seconds, (23 seconds calc time + 7 second startup overhead) on the standard production computer hardware sold with every DPET. This compares well with the current production SSRB code which runs in approximately 45 seconds including reconstruction.

Table 5. Summary of Reconstruction Performance Improvements.

Improvement	**Number of Iterations**	**Time Per Iteration**	**Total Time to Final Solution (seconds)**	**Total Speedup**
Standard SIRT 1 Processor	80	11.7 sec	11.7 * 80 = 936	1X
16 Subsets 1 Processor	5	11.7 sec	11.7 * 5 = 58	16X
16 Subsets 2 Processor	5	6.8 sec	6.8 * 5 = 34	27X
16 Subsets 4 Processors	5	4.6 sec	4.6 * 5 = 23	41X

Chapter 4

Validation

4.0 Overview

Validation was performed with pre-defined synthetically generated test cases, synthetic data derived from a statistical usage model, and actual scans. Pre-defined test cases can test specific conditions, and the statistical usage model can provide a measure of system reliability confidence by measuring operation over a large range of usage [Walton, 1995]. Actual scans provide integration testing with the scanner itself. For consistency, all images shown are reconstructed to the equivalent of 80 single subset iterations with relaxation. This is an arbitrary value derived from convergence testing done in Chapter 3 that gives consistent quality results. Unless otherwise specified, all images were reconstructed with 16-subsets, 5 iterations, relaxation enabled (α=1.99). This is the high speed (23 second solution) configuration (Chapter 3), and represents the conditions that would be expected during normal operation.

4.1 Results with Synthetic Data

Testing with synthetic data allows complete control of the introduction of noise into the system. Three different types of synthetic data were tested using the algorithm: a large cylinder to check uniformity, a pathological set of ellipsoids to measure response in extreme conditions outside of normal use, and a statistical usage modal derived to test the algorithm under a wide variety of conditions and establish an envelope of known performance as a basis for testing future modifications.

4.1.1 Synthetic Cylinder Test Case

A uniform cylinder of large diameter covering the entire axial field of view and expected attenuation coefficient value of 0.115 cm^{-1} (an arbitrary value between water and bone) was synthetically derived and input into the algorithm to ensure that the system model has a uniform response. A transaxial view is shown in Figure 40. Drawing a profile through the center of the cylinder yields the graph shown in Figure 41. Overall results for the synthetic cylinder indicate uniform coverage of the field of view with minimal distortion.

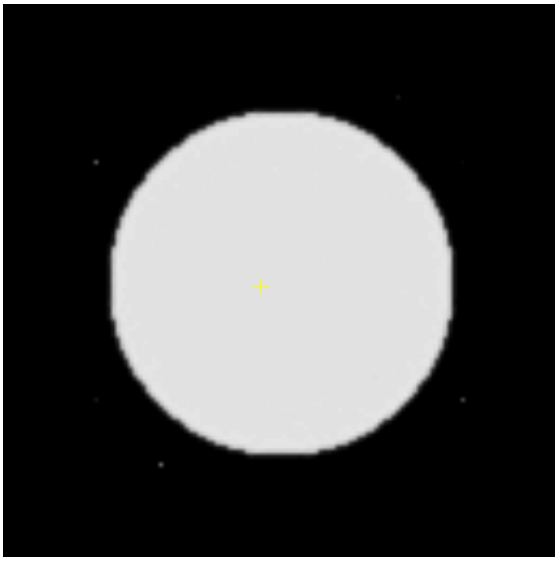

Figure 40. Uniform Synthetic Cylinder, transaxial view.

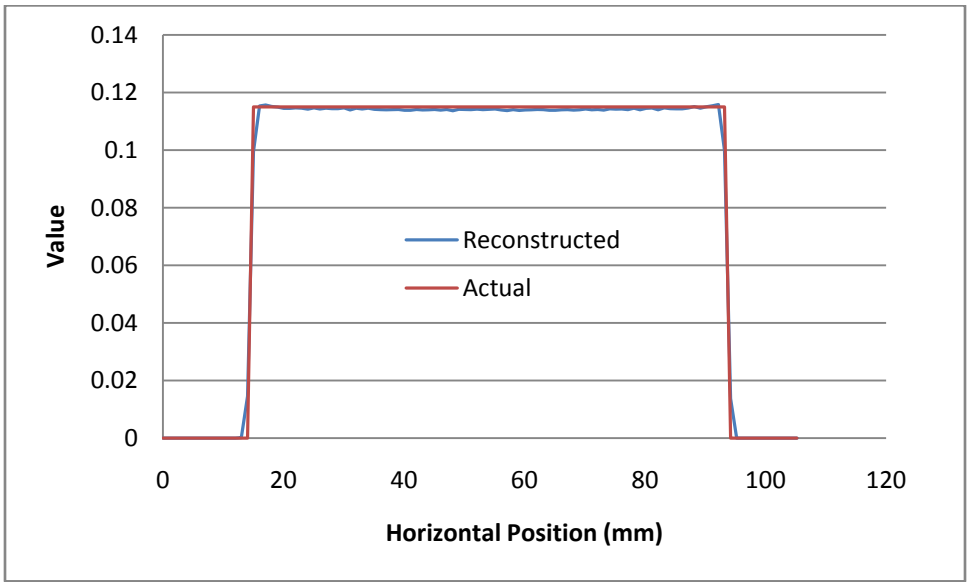

Figure 41. Profile through the center of the synthetic cylinder.

4.1.2 Stress Testing Object Test Case

This particular example, a pathological case with densities well outside ordinary usage, contained four areas, one main, two internal lesions, and one surface lesion. The parameters are shown in **Error! Not a valid bookmark self-reference.**. The results are shown in Figure 42. This structure was pulled from the set of statistical test cases as one of the cases that caused difficulty in convergence.

The values of both the synthetic and reconstructed images were measured along a profile path (shown in green in Figure 42). The results of that profile are shown below in Figure 43.

The algorithm converged with the residual error norm for the reconstruction at 31.84 and the image error norm at 13.45 in this case. Dividing the image error norm by the number of voxels in the supported image volume (160k) yields an expected error per voxel of 8.4×10^{-4} image units. This is not the worst case error, but a more general error measurement over the entire supported FOV. The worst case error can be visualized by looking at an image of the difference between the synthetic and reconstructed images as shown below in Figure 44.

The maximum errors are clearly located around the edges of the primary structures. The steep cutoffs induced by the highly attenuating structures causes the algorithm to over or under estimate the actual value. The principal cause of this error is the system model, which is itself a reflection of the DPET construction. Given the physical size of the detectors and source, this limits the ability of the system to position a given line of response, thus blurring the end result. The reconstructed image is only as clear as the system model can make it, and resolving the sharp edges of the noise free synthetic phantom points out that limitation. By plotting a profile through the same segment as the prior profiles, the error magnitudes and locations are seen in Figure 45.

In summary, if highly attenuating objects are placed in the field of view, there is a potential for localized errors at the edge of the object on the order of 20% of the attenuation value due to the steep gradients in the synthetic data. It is important to note that within the context of this image, this represents objects made of such dense materials as lead or tungsten. Imaging this type of material in an actual scanner would likely render the image completely useless due to exceeding the dynamic range of the system. We test it here because synthetically we can test the algorithm beyond what could ordinarily be expected in real world usage to get an idea about how it will fail. It would take a very long time to acquire enough statistics to image such large high density objects with low energy gamma radiation. For normal soft tissue with reasonable transitions, the measured errors are under 1%. In this particular case, the expected error per voxel over the entire volume is 0.84%.

Table 6. Stress Testing Object Phantom Parameters.

Area Name	Density
Main	0.1
Internal #1	1.5
Internal #2	0.7
Surface	0.6

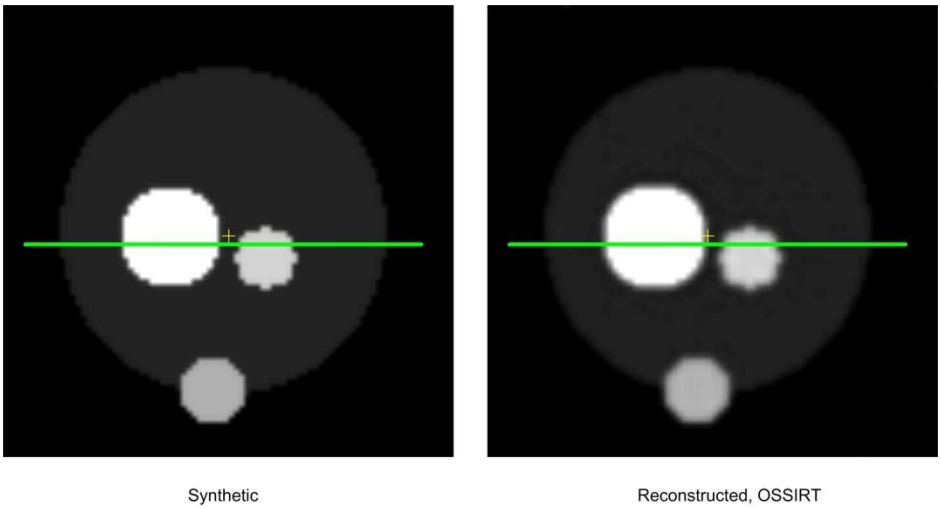

Figure 42. Synthetic vs. Reconstructed Images, stress testing test case.

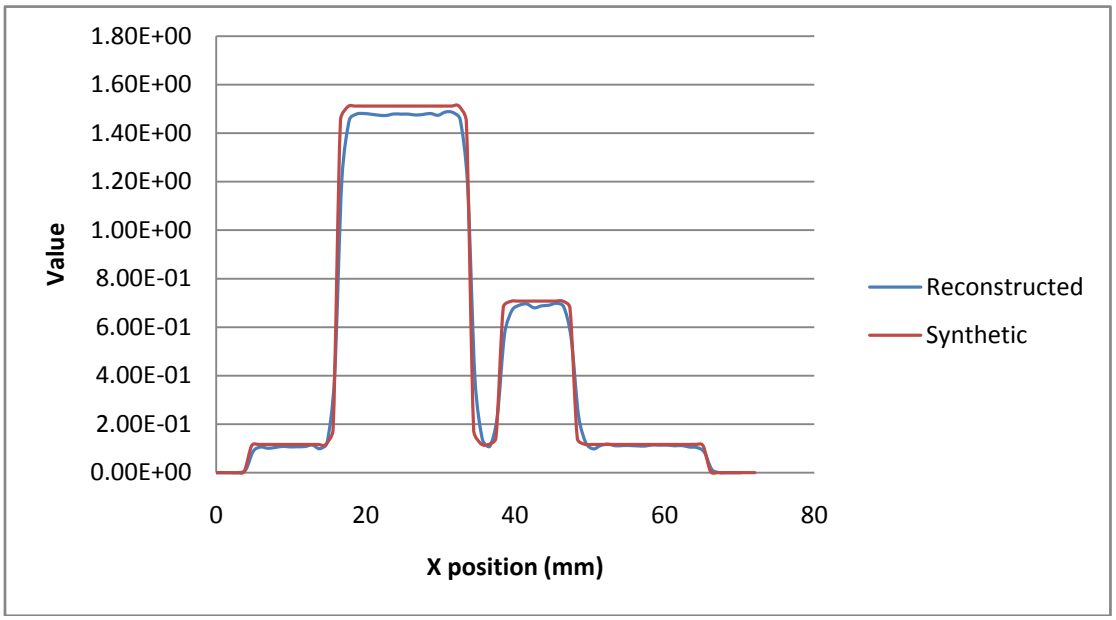

Figure 43. Synthetic vs. Reconstructed Profile, stress testing test case.

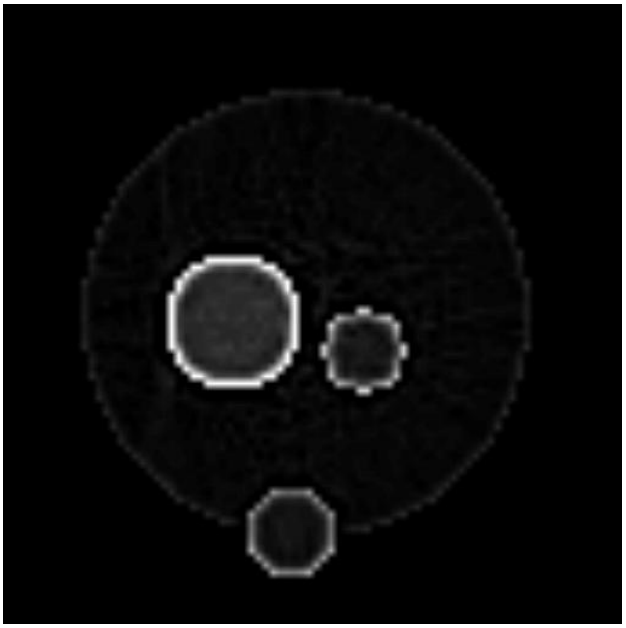

Figure 44. Difference Image, synthetic vs. reconstructed stress testing test case.

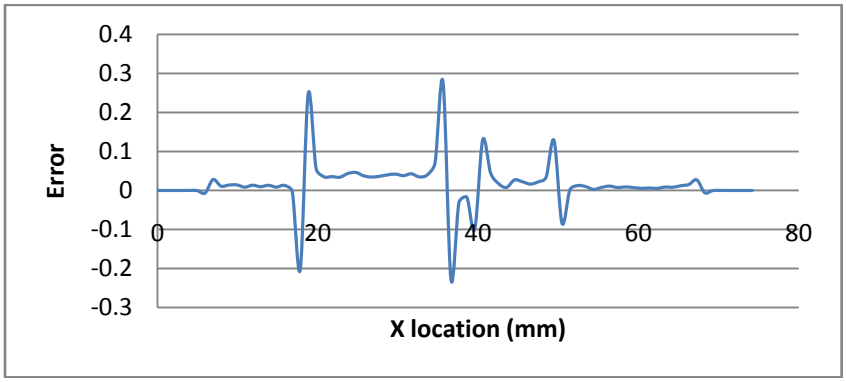

Figure 45. Error profile through difference image, stress testing test case.

4.1.3 Statistical Testing

Statistical testing is based on a statistical model of the usage of the system [Walton, 1995]. In this particular case, the system is designed to image mice and other small rodents, and we choose to model this usage as a set of ellipsoidal volumes within the image volume of the system. There is a single volume that represents the body, and a variable number of smaller volumes that may be embedded, surface, or completely disjoint from the body that vary in density. The contents of the entire test volume are then blurred together such that it represents the physical resolution of the scanner itself at 1.6 mm full width at half maximum (FWHM). The contents of the usage model are described in Table 7.

Iterative reconstruction methods lend themselves to automated statistical testing because there are two inputs, both image and projection data. Normally, the initial image is specified as zero, and projection information is fed into the process, but it is possible to simply use the forward projector to start from a synthetic image with zero projection information by negating the result. The end result of this process should be a final image that should match the initial image that was fed into the system. Any differences in the two images are either errors, or the physical limitations imposed by the system model. This concept is shown in Figure 46.

Once the algorithm has converged, two results are presented. The first result is the residual error norm estimated by the algorithm normalized with respect to the input projection data. This is the weighted difference between the actual scanned projection information and the forward projected data from the estimated image, and is represented by the formula (4.1)

$$\text{Normalized Residual Error Norm} = \frac{\|Ax-b\|_R^2}{\|b\|_R^2} \tag{4.1}$$

The algorithm attempts to minimize this value by design. After the minimization is complete, a second error norm is computed based on image results. This image error norm represents the difference between the synthetic start image and the final reconstructed image normalized with respect to the synthetic start image, and is represented by formula (4.2)

$$\text{Normalized Image Error Norm} = \frac{\|x^*-x\|^2}{\|x^*\|^2} \tag{4.2}$$

where x* is the synthetically generated correct value, and x is the reconstructed image.

Statistical testing allows many different cases to be measured such that an envelope of correct operational performance is determined for the parameters to be measured. In this instance, we desire to determine the worst case expected error. The usage model previously described allows for a wide variety of both normal scans and

Table 7. Usage Model Parameters.

Object	**Center**	**Length**	**Density**
Body	0 ± 5 mm	20-60 mm	0.1-1.5
Ancillary (1-3)	0 ± 20 mm	0-5 mm	0.5-1.5

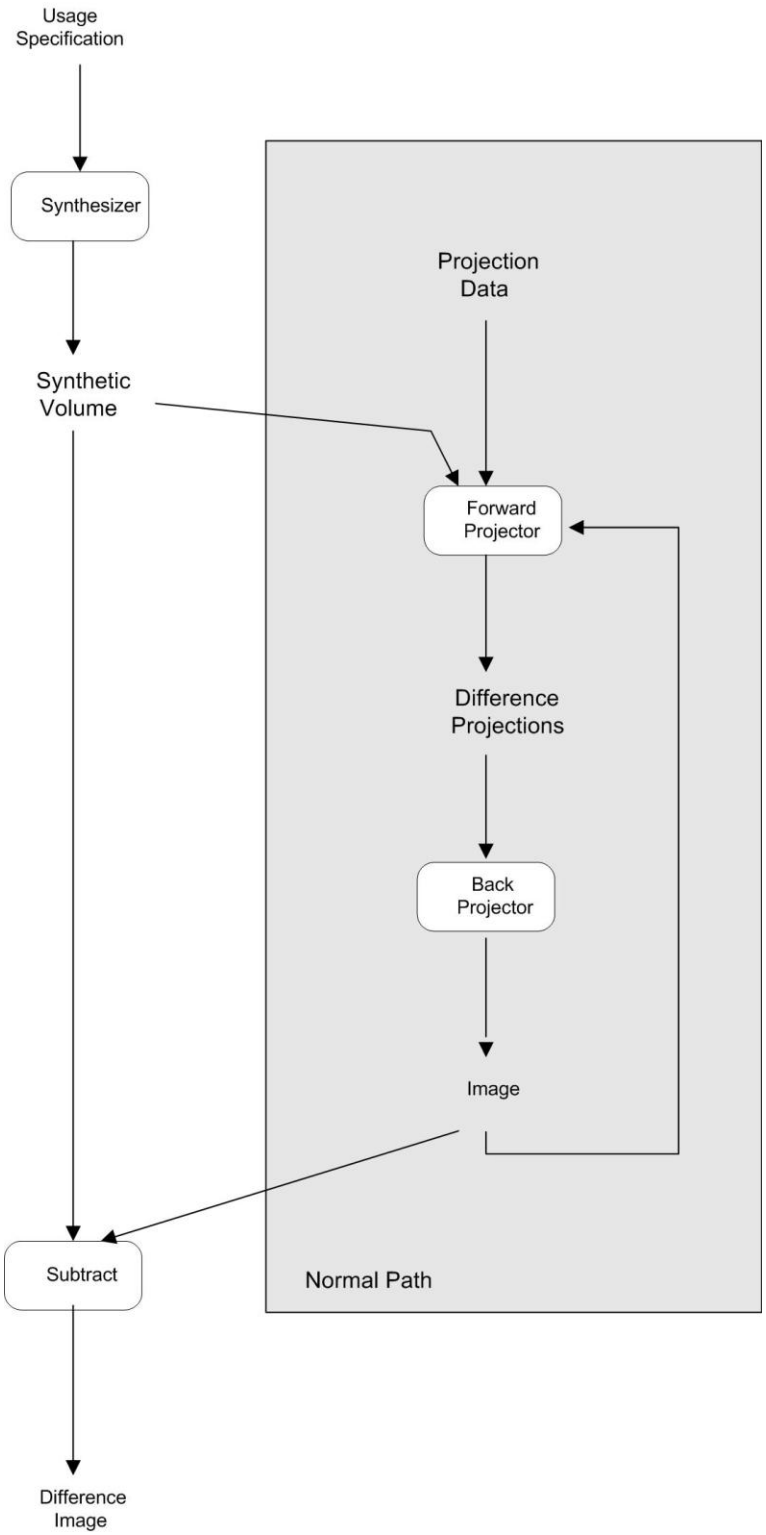

Figure 46. Statistical Testing Data Flow.

potentially problematic ones. For example, high density objects that span the support region boundaries can cause problems with the convergence of any reconstruction algorithm, and these are included. Two hundred test cases were run, and the results reported here.

Plotting the measured image error norm versus the residual error norm for the two hundred test cases yielded the relationship shown in Figure 47. In the post-analysis of this information, it was noted that three zones of operation exist. The primary zone contains 60% of the results. All test cases in this zone contain objects that are reasonably close to what is expected during normal operation, images with structure completely contained in the field of view and attenuation constants between 0.08 and 0.15. In the next zone, containing approximately 30% of the results, one or more of the objects in the synthesized field of view contain highly attenuating materials, the equivalent of lead or tungsten. The wide dynamic range of this scenario causes the algorithm to converge with higher residual norms. The algorithm did converge, however, and did so with final image error norm values consistent with the values presented in the normal case. The final zone contains approximately 10% of the test cases, and represents cases where the object in the field of view extends beyond the edges of the field of view in such a way that the algorithm cannot account for it. If the system matrix does not allow updates of voxels where the object actually sits (outside the support region), then this area causes baseline error that cannot be corrected. Even with this built-in error in projection space, the image error inside the support region still had final errors within 20% of the high of the normal zone.

For all test cases, the worst error measured was a cumulative error of 14% in a typical subject. This particular test case involved reconstructing an object that was partially outside the field of view. For cases that did not have problematic scanning conditions, boundary penetrations, etc., the worst case error dropped to 1.5%.

The power of statistical testing is that it allows for code changes to be evaluated quantitatively for improved accuracy. For example, Gregor and Benson [Gregor, 2008] published the PSIRT algorithm as a memory optimization for the SIRT algorithm. In order to accomplish this, the diagonal matrix C is approximated by a single scalar value equal to its smallest non-zero value (which corresponds to one over the largest column sum). Convergence is guaranteed, as this is the worst case, but it is slowed for some voxels due to excessive variation in the column sum matrix. It is possible to easily switch the code to use the necessary scalar constant, and re-execute the same statistical testing sequence with the change. The results of this modification and comparison to the non-estimated method are shown in Figure 48.

In this case, overall accuracy is not improved for the same test conditions. The worst case normalized image error norm increased from 13% to 15%. Test cases that did not converge well to the correct answer in the OSSIRT case similarly struggled in the PSIRT case with the maximum normalized residual norm increasing from 0.16% to 0.35%. A few of the worst cases were directly compared and are shown in Figure 49. The vector between the original baseline method, and the new method for the same test conditions, is termed the accuracy vector. The direction and magnitude of the accuracy vector combine the changes in the final residual norm and the final image error norm.

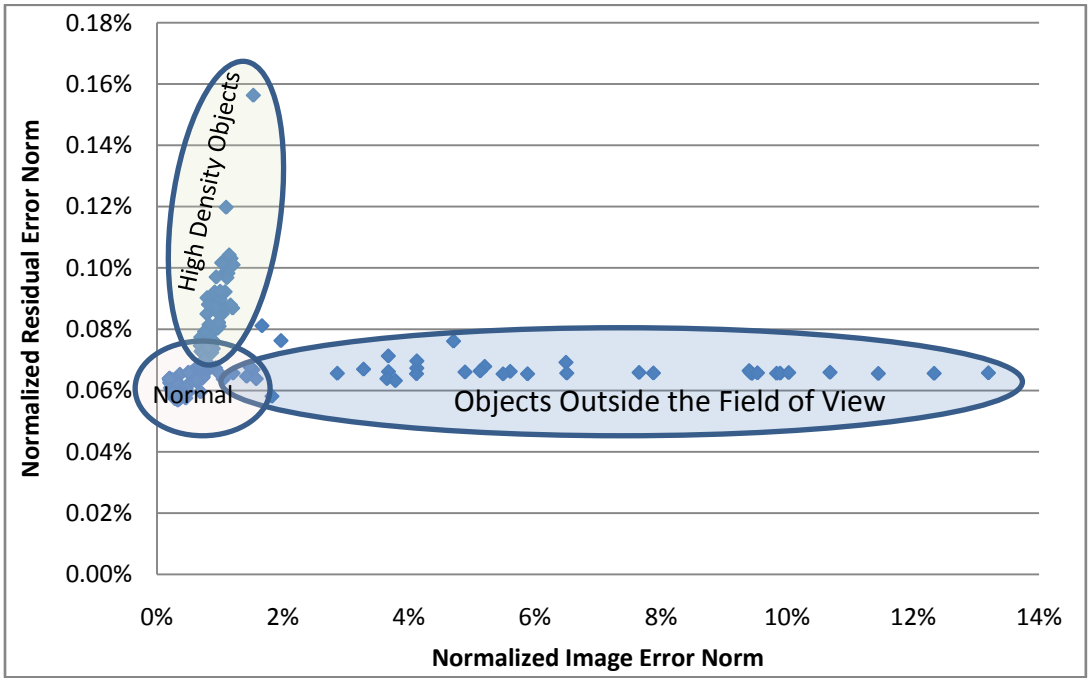

Figure 47. Statistical Testing Results, 200 test cases.

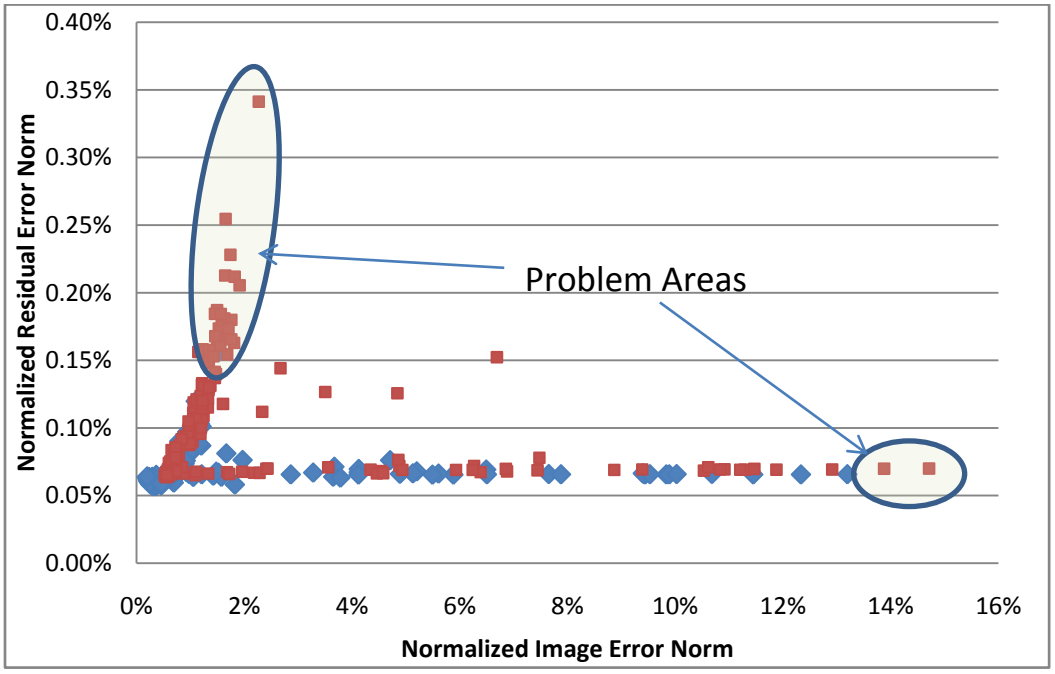

Figure 48. Statistical testing results compare PSIRT(Red) to OSSIRT(Blue).

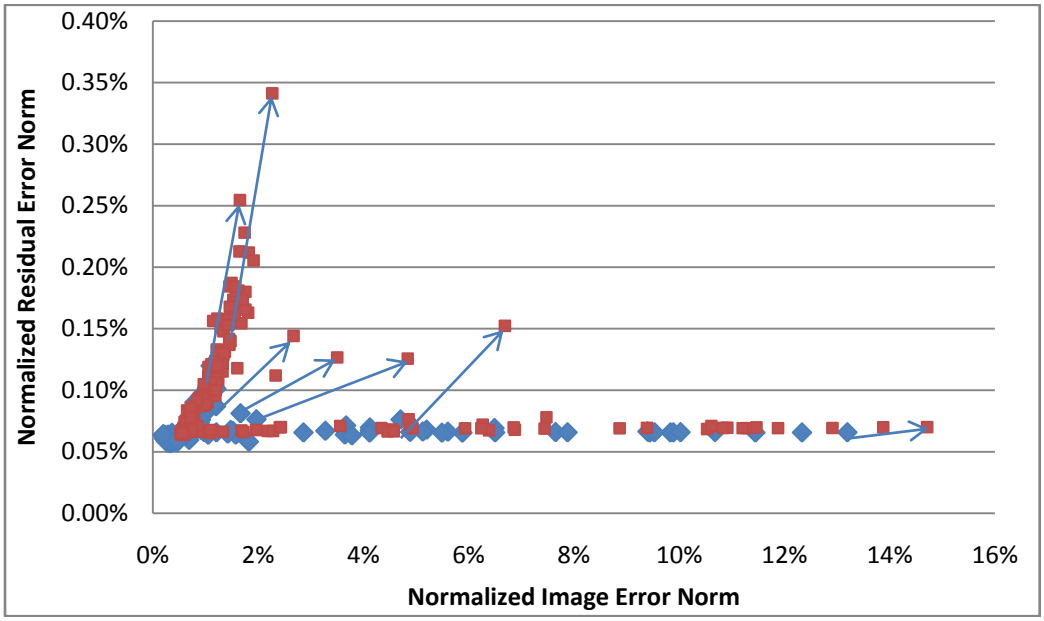

Figure 49. Systematic convergence errors. PSIRT (Red). OSSIRT(Blue).

Given that memory consumption isn't a controlling parameter, we choose to not introduce the PSIRT optimizations. A similar approach can be taken with any potential code change, even those meant to fix a particular artifact or problem. Using this method, an objective measure can be obtained that allows clear determination of whether or not the change actually made the resulting images more closely match reality in a wide variety of conditions. Examination of the accuracy vectors from the baseline method to the method under test for all test cases can give a sense of whether or not the method change consistently reduces or increases the image error norm. Singularities in the results should be examined to detail potential problems. It is entirely possible that a change could improve some test cases and worsen others. This method provides the developer tools to visually understand this behavior for a large number of test cases.

4.2 Phantom Scans

Several different phantoms were imaged to highlight differences between the methods. A cylindrical water phantom was used to calibrate the system for 511 keV attenuation. The attenuation factor of water is a well known constant, and a cylindrical water phantom provides an uninterrupted area to measure this constant. A steel rod with a 90 degree bend was used to evaluate the algorithms for axial and transaxial resolution as well as geometric validity. The assumptions used in single slice rebinning break down as the distance from the center of the system increases, so a measurement of axial resolution on a wire placed in the transaxial plane will highlight this fault. An aluminum flat was used both to validate system geometry when the transaxial diameter of the object in the field of view tapers rapidly, and to show uniformity across bed steps. A micro Derenzo phantom was used to evaluate resolution over an area, and finally, a microCT tissue equivalent phantom was used to evaluate contrast.

4.2.1 Thin Rod Phantom

From Bailey et. al, "Spatial resolution refers to the minimum limit of the system's spatial representation of an object due to the measurement process." [Bailey, 2005]. Spatial resolution is modeled as a Gaussian blur applied to the true image. Thus, by measuring a small item of known size and looking at the spread in the resulting image, the spatial resolution of the system can be determined. This is a useful number, but doesn't necessarily characterize the entire system. This method only measures the resolution at a particular place in the field of view, but by selecting a couple of locations that are representative of the field of view, an approximation of the performance of the system can be evaluated. We chose the center and 1 cm radially off center to evaluate, as this represents the portion of the field most likely to be used in actual production use, as mice in the 30 g size range tend to be approximately 2 cm in diameter.

A 1.59 mm diameter steel wire, bent in an L shape was used to evaluate resolution in the axial and transaxial directions. The phantom itself is shown in Figure 50. Profiles were drawn through the phantom transaxially, axially at the center of the field of view, and axially at 1cm offset from the center. A Gaussian fit was performed on all

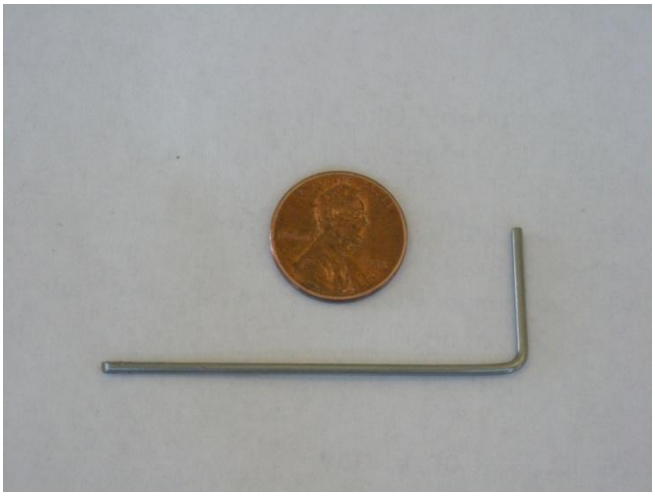

Figure 50. Thin Rod Phantom, 1.59 mm diameter, steel, with 90 deg. Bend.

profiles to determine the full width at half max (FWHM) of the measured width of the wire. The thickness of the wire was then deconvolved from the measured width by subtracting the square of the rod diameter from the square of the measured spread and taking the square root of the difference to determine the system resolution. The results of these measurements are shown in Table 8.

The side views of the line phantom clearly show the limitations of the SSRB/FBP processing. The addition of oblique angles to a vertical view ruins the ability of the system to accurately position an event off-center (Figure 51). The introduction of the model based OSSIRT reconstruction dramatically improved the axial resolution off-center.

The resolution of the system is degraded slightly transaxially with the introduction of the Am-241 low energy source, but the resolution recovered by the use of a system model in the OSSIRT algorithm overcomes this loss.

4.2.2 Aluminum Flat Phantom

A flat, 3.2 mm thick by 38.2 mm wide, made of 6064-T4 aluminum was placed in the system and scanned with Am-241 for 30 minutes acquiring 700M counts. The resulting coronal images are shown in Figure 52.

The problems with the SSRB/FBP techniques are seen in Figure 52(a) at the end of the flat where the corners are truncated. In addition, there is also a secondary effect in the horizontal striping. As the angle of incidence of the ray with the object becomes more oblique, estimation of the direct plane becomes less accurate and a sharp contrast exists at the bed step points. The OSSIRT method does a better job at normalizing over all conditions, and the bed step artifacts do not appear to be present in Figure 52(b), however, they are still there, just much less visible. Scans with very high statistics do show these effects slightly, and will be seen later in the results.

4.2.3 Derenzo Phantom

A Derenzo phantom [Derenzo, 1987] is used to evaluate resolution over an area. The Derenzo phantom contains a series of rods of varying sizes and spacing. In this way, a simple view of the phantom allows a qualitative evaluation of resolution over a significant portion of the FOV. One example of a Derenzo phantom is shown below in Figure 53.

In Figure 54, differences can be observed in the different methods of reconstruction. The SSRB/FBP processing shows the 2.32 mm rods visible. By applying OSSIRT to the same scan data, the 1.54 mm rods start to become resolvable. Note that these images are in the transaxial plane, where the differences between SSRB/FBP and OSSIRT are minimized. In the axial planes, the differences are much greater, and those have been reviewed in previous sections. These values are consistent with the measured resolution from the thin rod experiments in the transaxial direction.

Table 8. Resolution Results.

Source	Type	Outside Dia. (trans) (mm)	Trans-axial Res. (mm)	Outside Dia. (center) (mm)	Axial Res. (center) (mm)	OD (1cm) (mm)	Axial Res. (1cm) (mm)
Co-57	SSRB/FBP	2.75	2.24	2.31	1.68	5.59	5.35
Co-57	OSSIRT	2.08	1.34	1.88	1.00	1.78	0.80
Am-241	SSRB/FBP	2.84	2.35	2.63	2.09	5.83	5.60
Am-241	OSSIRT	2.17	1.48	1.87	0.98	1.77	0.77

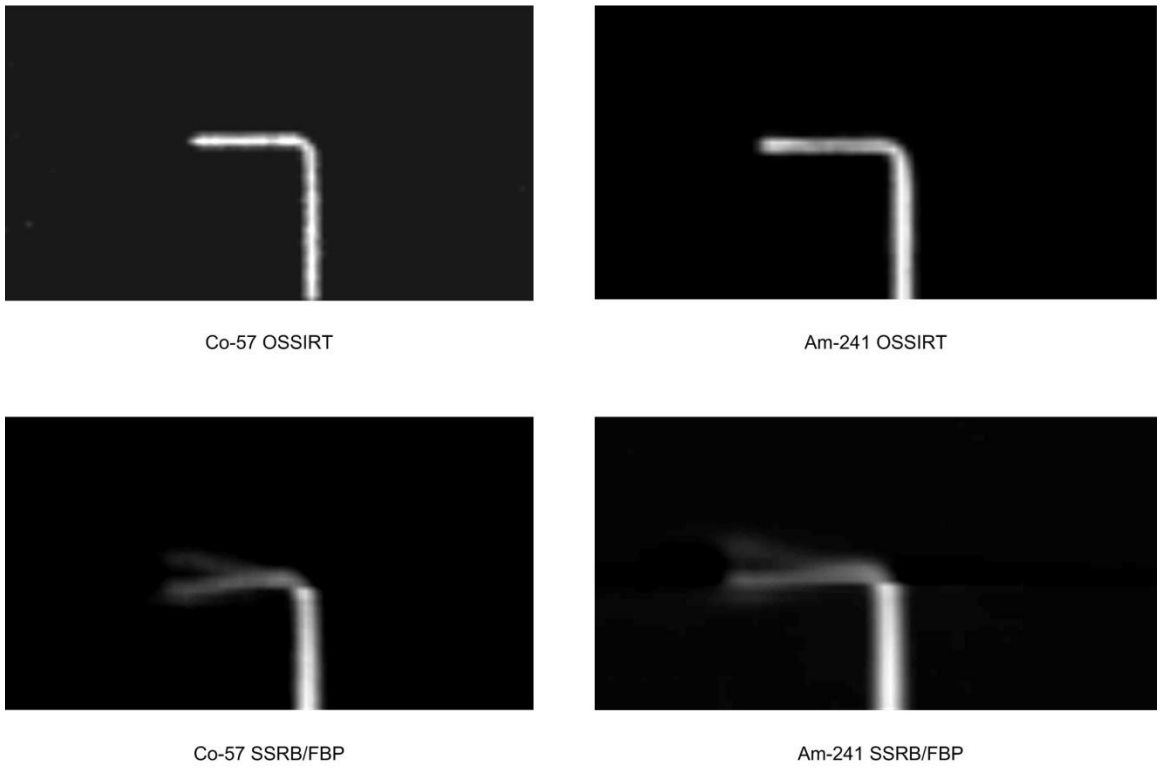

Figure 51. Thin rod scanned with Co-57 and Am-241, 1.0B counts each, processed with OSSIRT and SSRB/FBP.

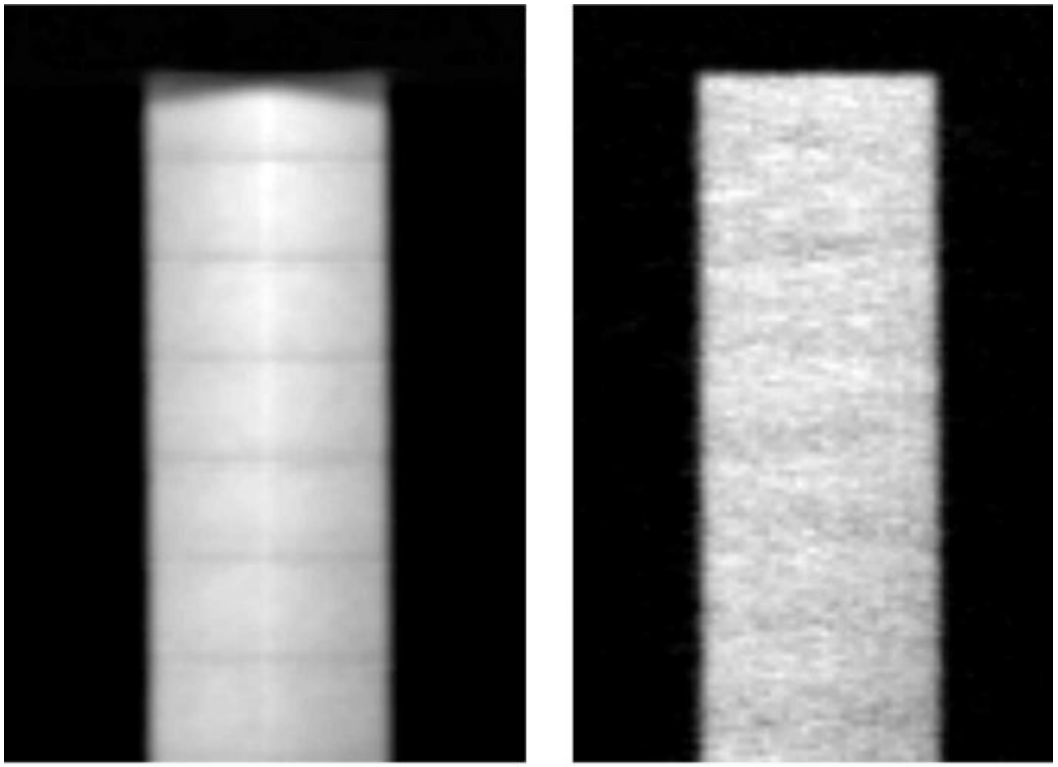

Figure 52. Aluminum Flat, 700M counts, processed with SSRB/FBP and OSSIRT.

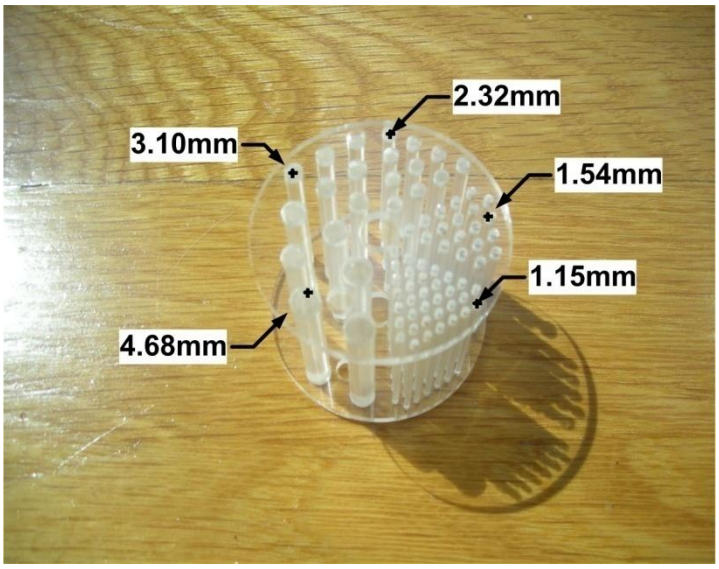

Figure 53. Micro Derenzo phantom picture indicating shaft sizes from 1.15 mm to 4.68 mm.

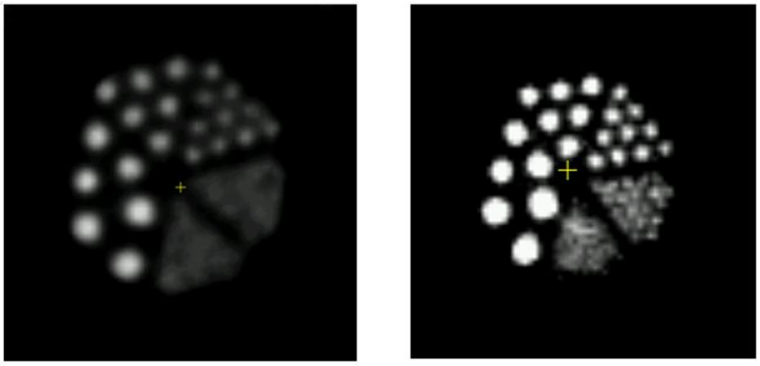

Figure 54. Micro Derenzo transaxial view with 1.2B counts shows improvement in resolution.

4.2.4 Uniform Phantom

A uniform cylinder, 30 mm in diameter, filled with water, is used to calibrate the system to the proper attenuation factors. A three minute scan with Am-241 is shown in Figure 55. Note the foam core carbon fiber bed supporting the phantom.

Drawing a line through the center of the phantom shows the response of the system (Figure 56). The mean response is 0.095 cm^{-1}, this is the 511 keV attenuation coefficient of water.

4.2.5 Tissue Equivalent Phantom

The Tissue Equivalent Phantom was originally designed to calibrate microCT systems. It is made of a cylinder of Lucite with a density of 1040 mg/cc. Holes (1.5mm diameter) are drilled lengthwise and filled with materials with different densities. The layout is highlighted in Figure 57.

The Tissue Equivalent Phantom was modeled by taking the specifications for the materials and, because the shaft sizes are small, compensating for partial volume effects by applying a Gaussian blur kernel with a magnitude equivalent to the spread measured on the detector (in Appendix A) of 1.08mm FWHM. The result of that model is shown in Figure 58. Note that the 1000 mg/cc shaft is barely visible.

Scanning the Tissue Equivalent Phantom, and reconstructing with OSSIRT yields the following image in Figure 59. The open holes and the 1750 mg/cc shafts are clearly visible, but blurred compared to the synthetic version. The 1250 mg/cc, 1000 mg/cc, and 1050 mg/cc shafts are not discernable.

Drawing a profile through the two open holes and the 1750 mg/mm3 shaft and comparing that to the same profile on the synthetic phantom yields the graph in Figure 60. Partial volume effects clearly affect the profile. Since the phantom was originally designed for CT, the shaft sizes are simply too small for this usage, and the result is blurred by comparison. A better measurement would require a larger phantom that more closely matches the resolution of the system it was designed to test. Such a phantom was not available for this work, but should be considered for future work in this area.

4.3 Performance on Biological Subjects

The University of Tennessee does not own an Inveon DPET. Siemens Molecular Imaging graciously offered the use of a DPET for the purposes of testing this algorithm at their factory in Knoxville, TN. However, the scanning of live subjects is prohibited at the Siemens Molecular Imaging factory, so only previously euthanized animals can be scanned. These are procured from a local pet store pre-frozen. In addition, the potential use of a CT contrast agent is not available, but this is accepted as a limitation of facilities.

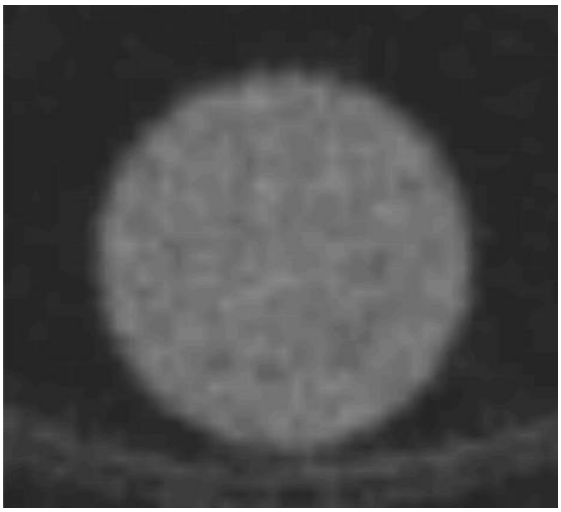

Figure 55. Transaxial view uniform phantom. Scanned with Am-241 and reconstructed with OSSIRT.

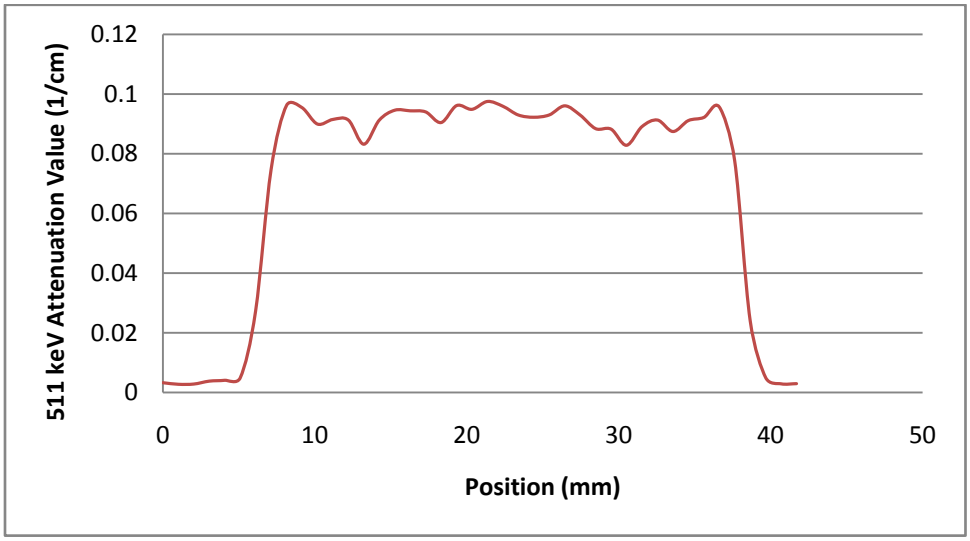

Figure 56. Line profile through 30 mm diameter water phantom.

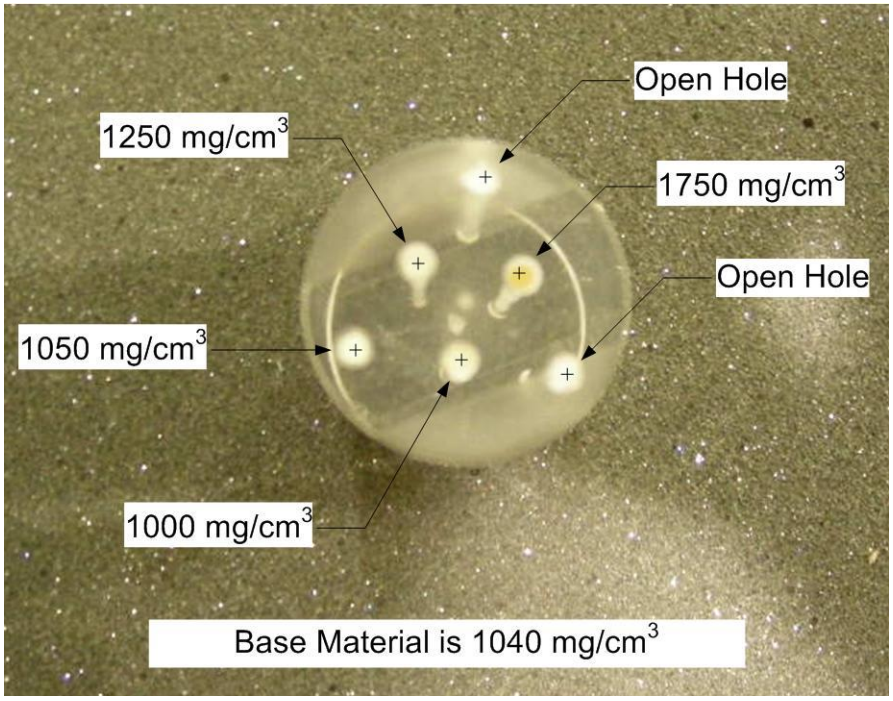

Figure 57. Tissue equivalent phantom with densities labeled.

Figure 58. Synthetically derived tissue equivalent phantom created from CAD models of the object.

Figure 59. Tissue equivalent phantom. Reconstructed with OSSIRT.

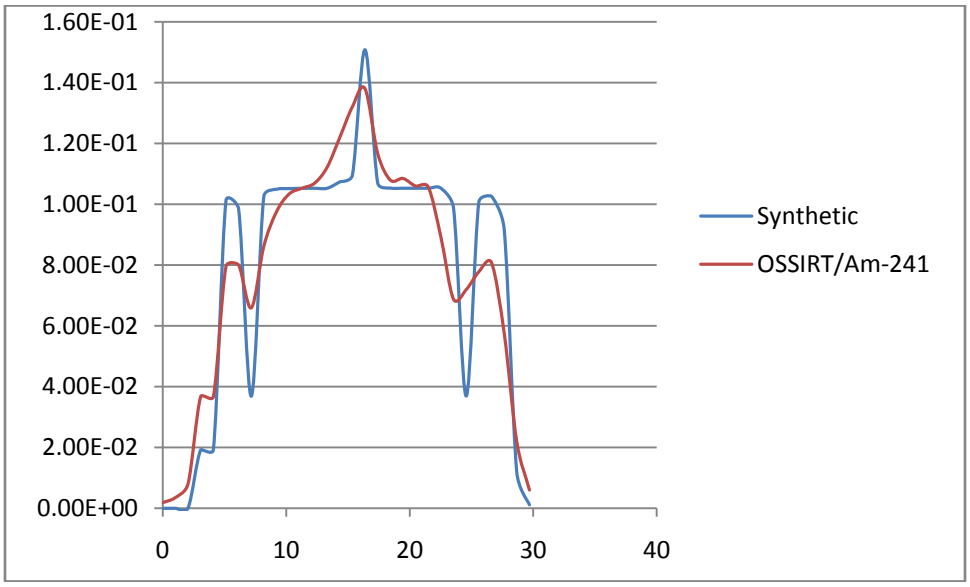

Figure 60. Profile comparison between synthetic and actual tissue equivalent phantoms.

In this case, we imaged a 50 g white rat with both Co-57 and Am-241 in both a 30 minute protocol, and a long 4 hour protocol. The 30 minute protocol represents what would be expected for a study to achieve very good performance for attenuation correction. The 4 hour protocol represents the best that could be done with current technology, removing statistics from the equation, and can relate additional information about structure of the subject.

4.3.1 30 Minute Rat Scan

A 30 minute scan of the 50 g rat shows results with 1.2B counts. Images were reconstructed with SSRB/FBP and OSSIRT against the same Am-241 listmode data, showing only differences in processing.

The images in Figure 61 show that SSRB/FBP exhibit some of the artifacts that have been previously identified. Bed step boundaries have discontinuities and features are covered over that should be more visible. The OSSIRT reconstruction shows more anatomical detail with less significant bed step artifacts.

4.3.2 4 Hour Rat Scan

Scanning the 50 g rat for 4 hours with Am-241 provided 5.5B counts. Issues associated with image quality are principally algorithmic in nature at this level of statistics. Images of the rat are shown in Figure 62, reconstructed with the standard SSRB/FBP methods, as well as OSSIRT at 80 iteration equivalent quality (23 second reconstruction time).

Resolution is improved with OSSIRT over the SSRB/FBP case, and in addition, artifacts that are present with SSRB have also been removed or dramatically suppressed (Figure 62). These artifacts show the limitations of the SSRB/FBP approach as applied in actual use. For example, the coronal view (e) shows areas where the diameter of the subject changes rapidly are underestimated (neck regions), but not in the OSSIRT reconstructed image (f). The tip of the jaw is truncated in (a), but not when reconstructed with OSSIRT shown in (b). Pockets of air in the hind-quarters are visible in both types, but more clearly delineated with OSSIRT in (f) than (e). Overall, the large levels of axial blurring and other artifacts distort the overall result for SSRB/FBP compared to OSSIRT.

Comparing a profile drawn through the right hind quarter air bubbles (location shown in green on Figure 63) illustrates the difference in the profile shown in Figure 64. The three bubbles are well delineated on the OSSIRT case, and severely smoothed over in the SSRB/FBP case. In addition, bone structure in the right rear leg shows a spike in OSSIRT, but is not as delineated in SSRB/FBP.

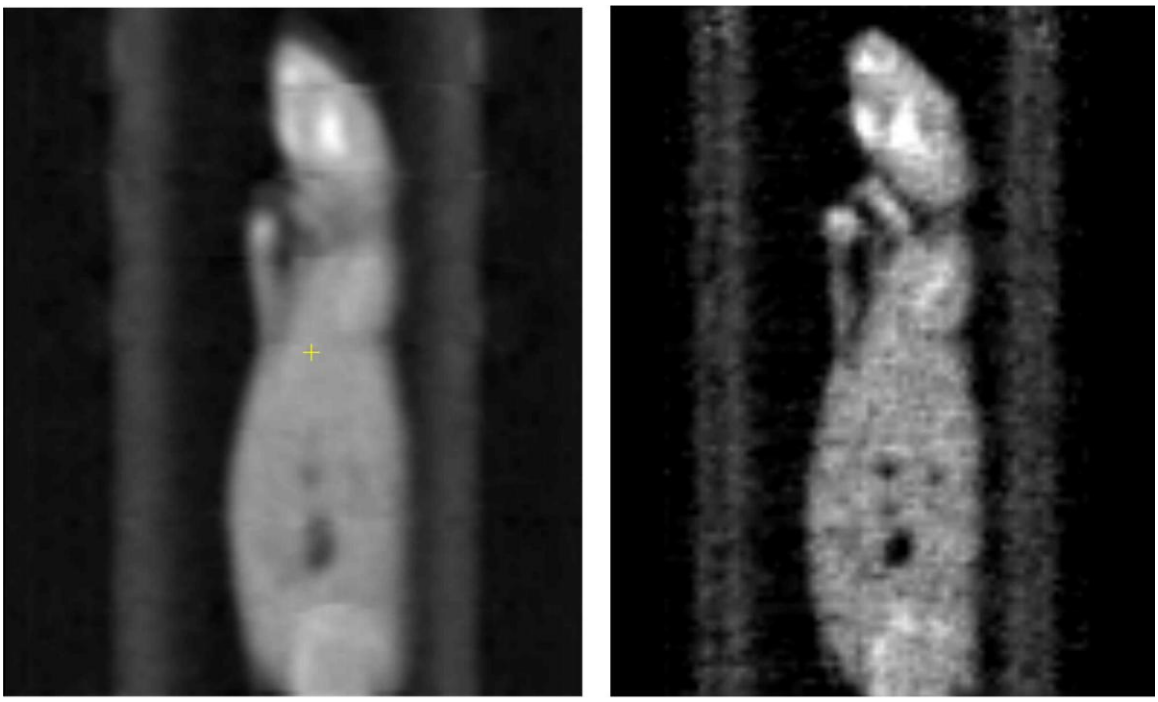

Figure 61. 50 g rat, 1.2B counts, Am-241, comparison of SSRB/FBP and OSSIRT.

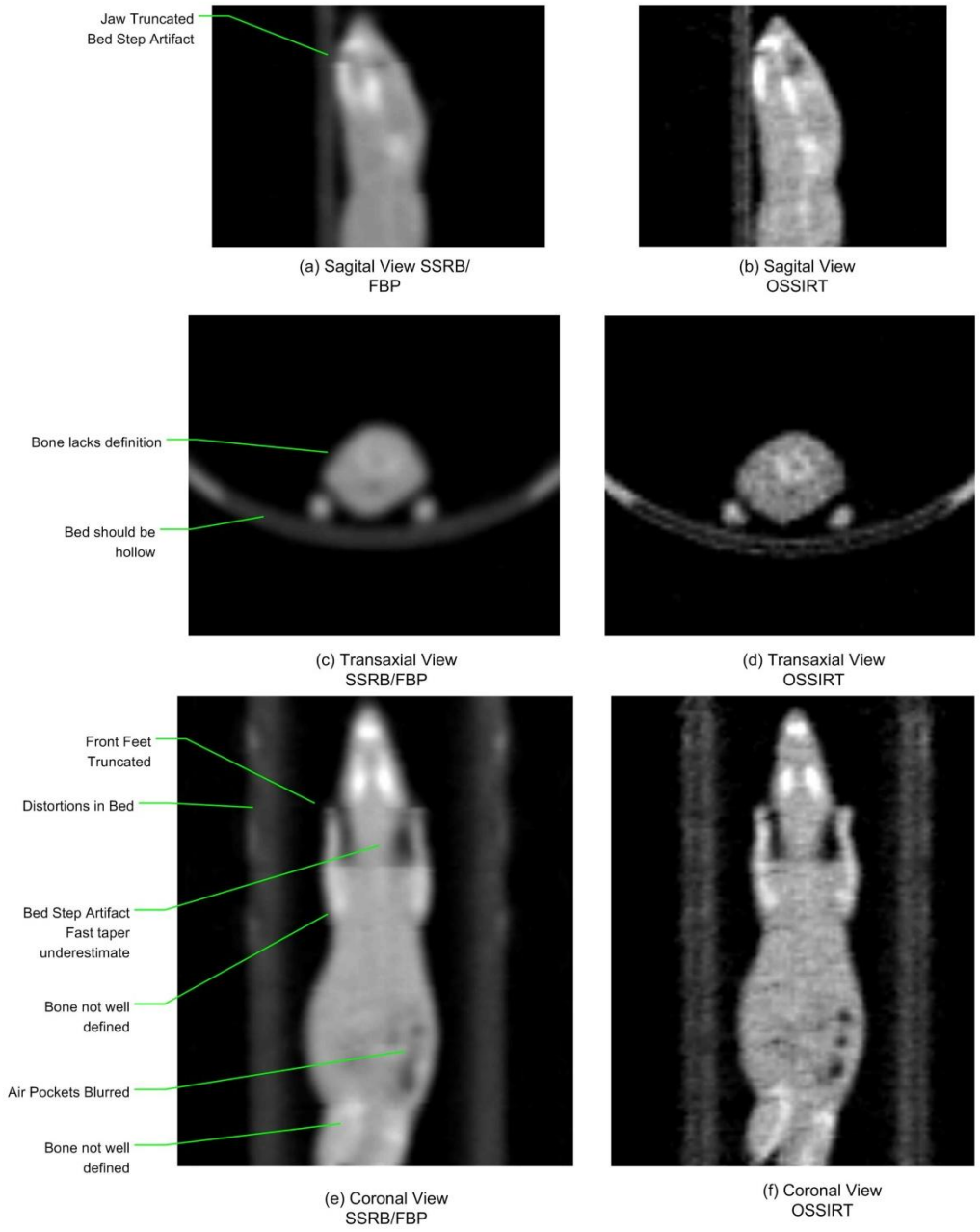

Figure 62. 50 g. rat, 4 hour scan, 5.5B counts, Am-241, processed with SSRB/FBP and OSSIRT to show differences between the methods.

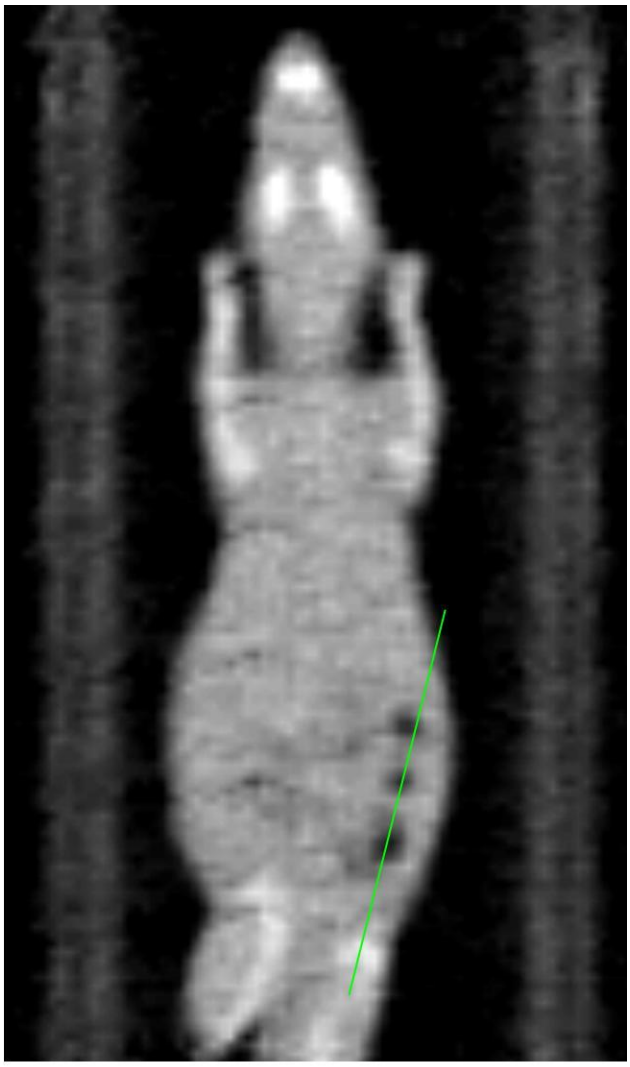

Figure 63. 50 g. rat, 4 hour scan, 5.5B counts, profile location shown through abdomen.

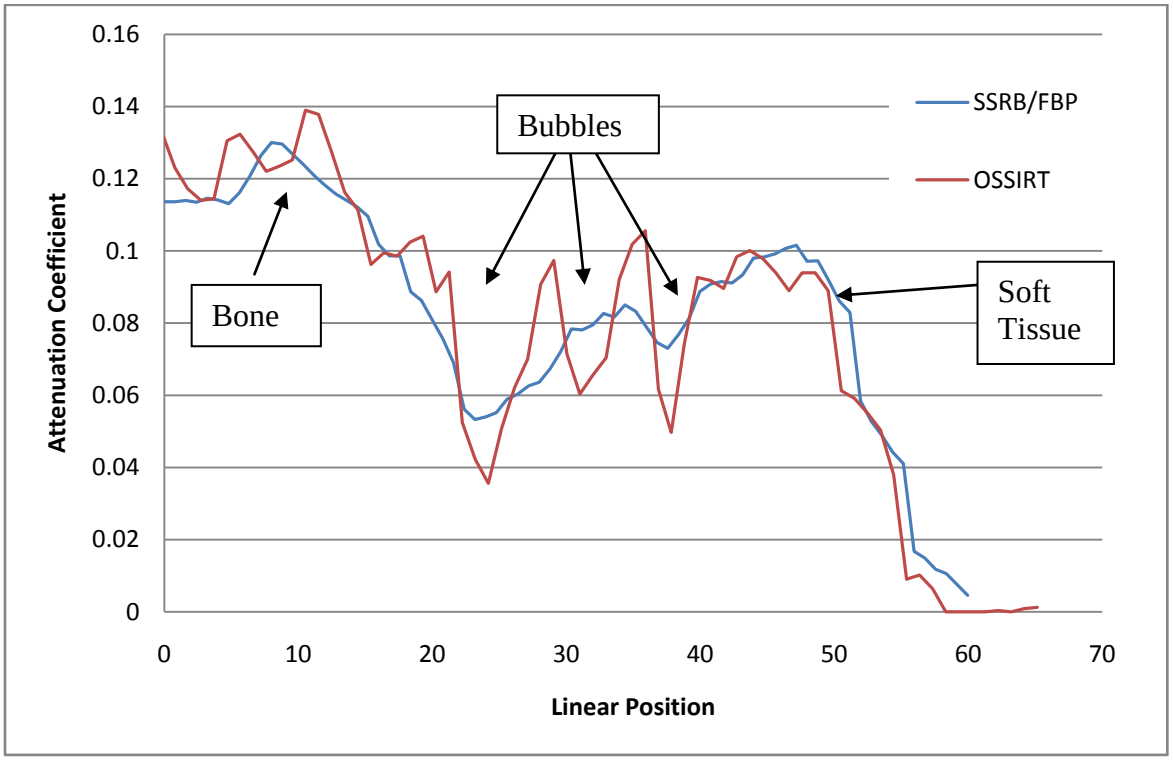

Figure 64. Abdomen profile drawn through the 4 hour rat scan.

Chapter 5

Conclusions and Future Work

The development of a model based iterative method for DPET transmission processing has shown that the benefits of this class of methods can be applied without an undue burden on processing time. Using the same hardware standard on every Inveon DPET, well optimized algorithms can provide improved results in a reasonable period of time. In fact, processing time for OSSIRT on standard Inveon hardware is similar to that required for the SSRB/FBP method, around a minute or less. Overall, there were several new concepts presented in this work to achieve this goal.

The development of the system matrix is the crux of any reconstruction effort. By deriving the physical coordinates of the matrix from the 3D CAD models that are used to build the system, we were able to build a model that is centered around the manufacturing tolerances for the system as a whole. Using measurements from a single system can bias the results, but the manufacturing efforts will always attempt to achieve the dimensions specified in the CAD models. By basing the system model on the same physical parameters, we are likely to achieve consistent results over many different systems. In addition, if the dimensions were to change by design, deriving a new system matrix would become a trivial task. The reconstruction would be ready before the first system was built, all based on the computer models.

Given past performance of various ray projection methods, especially with respect to anti-aliasing, the development and validation of a volume intersection method was a step forward. The concept of using a volume intersection to anti-alias the ray was a logical next step from prior work on this subject. Although initially computationally intensive, the volume intersection approximation algorithm, when implemented as a volume walk, was shown to be efficient enough to remove the need to pre-compute the system matrix if desired. This level of efficiency makes it a practical alternative. Additional physical acceleration with SSE was successful and inexpensive. However, it is likely that higher GPU performance in the future will make that the approach of choice. This concept is likely to see further use in other reconstruction algorithms where the efficiency of the projector and image quality are simultaneously important.

Implementation of OSSIRT provided some new computational concepts. Numerically the conflict resolution method used in the mutex-free multi-core reconstruction implementation was shown to provide a scalable solution with minimal interprocessor synchronization. Techniques to divide the projection data into disjoint sets are ideally done analytically, however, a simple sorting of projections can achieve the same goal if the geometry is not very uniform and it is desirable to operate with a large number of cores. Due to the choice to calculate and store the system matrix, memory access to the system matrix was problematic. Memory bandwidth is clearly a limiting issue with the combination of these methods and a large number of CPU cores, but newer CPU architectures that eliminate the front side bus [Intel, 2009] are being designed to address this bottle-neck, and we look forward to testing on these architectures.

A straight-forward method of selecting orthogonal views that is not constrained or complicated by the number of view angles was proposed and validated against previous techniques as identical in performance. This orthogonal selection approach provides an easy to implement method that handles a wide variety of orthogonal subset problems, more than just image reconstruction applications.

During the validation of the code, an approach for statistically testing this class of algorithm was proposed and used to validate results. By integrating the generation of a statistical usage model into the system, automated testing for a large number of test cases becomes much easier, and is thus done more often. Using the forward projector internal to the SIRT algorithm to insert a noise free test image into the system assures compatibility and system integrity. Any problem in the system matrix or the algorithm itself will result in errors in the final image that can be quantified with this approach. This method was found to be useful in the evaluation of incremental algorithmic improvements and appears deserving of further study, especially in the area of pass/fail criteria.

Validation with a variety of phantoms showed substantially better resolution, especially axially off center. The older SSRB/FBP techniques resulted in significant axial blurring that OSSIRT was able to correctly resolve. The fact that spatial resolution so closely matches what we measured directly on the detector gives us confidence that the system model closely matches the actual physical construction of the system. This result validates the generation of the system model from the 3D CAD physical models.

Geometric problems due to rapid taper in the axial direction are not a limitation with this reconstruction approach as they are with prior methods. Even so, axial banding is still somewhat evident with high statistics scans, and further work should be done changing the bed step parameters. A continuous wholebody acquisition would be a likely next step. In addition, speeding up the acquisition through the use of multiple sources should also be explored.

Results with biological subjects show improvement over the previous methods. Bed step artifacts were dramatically reduced, and discrimination between bone, air, and soft tissue was significantly improved. In this area, the obvious next step is testing with live animals. Potential improvements in image quality through the use of vascular contrast agents should also be explored.

As the system is currently implemented, we have shown that it is possible to extract information from a transmission scan with fewer distortions than the current SSRB/FBP method, and more importantly, that it is possible to achieve this in a computationally time efficient way without new investment in hardware. There are, however, other benefits. More fundamentally, this work represents a framework for solving reconstruction problems for arbitrary geometries. Since no assumptions about geometry were be made in the system matrix or reconstruction code, this approach should be able to compute a solution for nearly any arbitrary geometry. Thus, this work could lend itself to future system developments for very specialized scanners with similar geometric challenges.

List of References

Adam L.E., et.al, "Performance evaluation of the whole-body PET scanner ECAT EXACT HR+ following the IEC standard," *IEEE Trans. Nuc. Sci.*, Vol. 44, pp. 1172–1179, 1997

AMD, Advanced Micro Devices Corporate Website, Product Information, 2009. [Online] http://www.amd.com/us-en/Processors/ProductInformation. [Accessed March, 2009]

Bailey, D.L., et.al., "ECAT ART — a continuously rotating PET camera: Performance characteristics, initial clinical studies, and installation considerations in a nuclear medicine department", *European J. Nuc. Med.*, vol. 24, no. 1, pp. 6-15, 1997

Bailey, D.L., et.al., *Positron Emission Tomography, Basic Sciences*, Second Edition, Springer-Verlag, 2005.

Beach, R.S., Pirshafiey, N., *Inveon Dedicated PET Mechanical CAD Models*, Siemens Molecular Imaging, Preclinical Solutions, 810 Innovation Dr., Knoxville, TN 37932, Internal Document, 2004.

Beyer T., Townsend D.W., Brun T., et al., "A combined PET/CT scanner for clinical oncology", *J. Nucl Med.*, vol. 41, pp. 1369–1379, 2000.

Brinks, R.; Schretter, C.; Meyer, C.**; "**Comparison of maximum-likelihood list-mode reconstruction algorithms in PET**"**, *IEEE Nuc. Sci. Sym. Conf. Rec.*, vol. 5, pp. 2801 – 2803, 2006.

Bruder, H., et.al., "Single-slice rebinning reconstruction in spiral cone-beam computed tomography", *IEEE Trans. Med. Img.*, vol. 19, no. 9, pp. 873 – 887, 2000.

Buck, J., "The Recursive Ray Tracing Algorithm", Course Notes, 1999. [Online] http://www.geocities.com/jamisbuck/raytracing.html. [Accessed March, 2008]

Casey, M.E., Nutt, R, "A multicrystal two dimensional BGO detector system for positron emission tomography", *IEEE Trans. Nuc. Sci.*, vol. 33, no. 1, pp. 460-463, 1986.

Chatziioannou, A.F., et al., "Performance evaluation of microPET: A high resolution LSO PET scanner for animal imaging". *J Nuc. Med.*, vol. 40, pp. 1164-1175,1999.

Cherry, S.R., Sorenson, J.A., Phelps, M.E., *Physics in Nuclear Medicine*, Third Edition, Saunders, 2003.

Cormen, et.al., *Introduction to Algorithms*, Second Edition, McGraw-Hill., 2006.

CUDA, *Compute Unified Device Architecture Reference*, NVIDIA Corporation, 2009. [online]. Available: http://www.nvidia.com/cuda. [Accessed Feb, 2009].

Daube-Witherspoon, M.E., et. al., "Attenuation correction for small-animal PET scanners", SNM 51-st Annual Meeting, *J. Nucl. Med.*, vol. 45, p. 159+, 2004.

Daube-Witherspoon, M.E.; Muehllehner, G., "Treatment of axial data in three-dimensional PET", *J. Nucl. Med.*, vol. 28, pp. 1717-1724, 1987.

Daube-Witherspoon, M.E., et. al., "Rebinning and reconstruction of point source transmission data for positron emission tomography", *IEEE Nuc. Sci. Sym. Conf. Rec.*, vol. 4, pp. 2839–2843, 2003.

Derenzo, S. E., et.al., "Initial results from the Donner 600 crystal positron tomograph", *IEEE Trans. Nuc. Sci.*, vol. 34, no. 1, pp. 321 – 325, 1987.

Dongarra, J., Foster, I., Fox, G., *Sourcebook of Parallel Computing*, Morgan Kaufmann Publishers, Boston, p. 43, 2003.

Feldkamp, L., Davis, L., Kress, J., "Practical cone beam algorithm", *J. Opt. Soc. Am.*, vol. 1, pp. 612-619, 1984.

Foley, J.D., *Computer Graphics : Principles and Practice*, Addison-Wesley, 1990.

Gilbert, P., "Iterative methods for the reconstruction of three dimensional objects from their projections", *J. Theo. Bio.*, vol. 36, pp. 105-117, 1972.

Glassner, A.S., *An Introduction to Ray-Tracing*, Academic Press, 1989.

Gregor, J., Benson, T., "Computational analysis and improvement of SIRT," *IEEE Trans. Med. Img.*, vol. 27, pp. 918-924, 2008.

Hearn, D., *Computer Graphics*, Prentice-Hall, 1994.

Hennessy, J.L., Patterson, D.A., *Computer Architecture: A Quantitative Approach*, Morgan-Kauffman, 2006.

Hichwa, R.D., et. al., "Initial performance measurements and nude mouse imaging with philips small animal PET scanner", SNM 51-st Annual Meeting, *J. Nucl. Med.*, vol. 45, pp. 107+, 2004.

Hounsfield, G.N., "A method of and apparatus for examination of a body by radiation such as x-ray or gamma radiation", Patent Specification 1283915, US PTO, 1972.

Huesman, et al., “Orbiting Transmission Source for Positron Tomography”, IEEE Trans. Nuc. Sci., vol. 35, no. 1, pp. 735-739, 1988.

Hudson, H.M., Larkin R.S., “Accelerated image reconstruction using ordered subsets of projection data”, *IEEE Trans. Med. Img.*, vol. 13, no. 4, pp. 601-609, 1994.

Intel, Product Information, 2009. [Online]. http://www.intel.com/products [Accessed March, 2009].

Joseph, P., “An improved algorithm for reprojecting rays through pixel images”, *IEEE Trans. Med. Img.*, vol. 1, pp. 192–197 , 1983.

Kak, C., Slaney, M., *Principles of Computerized Tomographic Imaging*, Society of Industrial and Applied Mathmatics, 2001.

Kaldeway, T., “SSE Instructions in Commodity CPUs”, Course Notes, CEPE220, UCSC, 2007.

Kamphuis, C., Beekman F., “Accelerated iterative transmission CT reconstruction using an ordered subsets convex algorithm”, *IEEE Trans. Med. Img.*, vol. 17, No. 6, pp. 1101-1105, 1998.

Kemp, B.J., Lenox, M.W.,Mcfarland, A., “NEMA NU 2-2007 performance measurements of the Siemens Inveon preclinical small animal PET system”, *Phy. Med. Bio.*, vol. 54, pp. 2359-2376, 2009.

Kernigan, B., Ritchie, D., *The C Programming Language*, Prentice Hall, 1976.

Kinahan, P.E., Townsend, D.W., Beyer, T., Sashin, D., “Attenuation correction for a combined 3D PET/CT scanner”, *Med. Phys.*, vol. 25, pp. 2046–2053, 1998.

Knoess, C., Rist, J., Michel, C., Burbar, Z., Eriksson, L., Panin, V., Byars, L., Lenox, M., Wienhard, K., Heiss, W.-D., Nutt, R.,” Evaluation of single photon transmission for the HRRT”, *IEEE Nuc. Sci. Sym. Conf. Rec.***,** vol. 3, pp. 1936–1940, 2003.

Knoll, G. F., *Radiation Detection and Measurement*, 2nd edition, John Wiley & Sons., 1989.

Kohler, T, “A projection access scheme for iterative reconstruction based on the Golden Section”, *IEEE Nuc. Sci. Sym. Conf. Rec.*, vol. 6, pp. 3961-3965, 2004.

Kohlmyer, S.G., et al., “NEMA NU2-2001 performance results for the GE Advance PET system, *IEEE Trans. Nuc. Sci.*, vol. 31, 2003.

Melcher, C.L., Schweitzer, J.S., "Cerium-Doped Lutetium Oxyorthosilicate - a fast, efficient new scintillator", *IEEE Trans. Nuc. Sci.*, vol. 39, pp. 502-505, 1992.

Mintzer, R.A., Siegel S.E., "Design and performance of a new pixilated-LSO/PSPMT gamma-ray detector for high resolution PET imaging", *IEEE Nuc. Sci. Sym. Conf. Rec.*, vol. 5, pp. 3418-3422, 2007.

Newport, D.N., *Internal design documentation for transmission processing*, Siemens Molecular Imaging, Preclinical Solution, 810 Innovation Dr., Knoxville, TN, 37932, Internal Document, 2001.

NIST, *Mass Attenuation Constants*, National Institute of Standards and Technology, 2009. [Online] Available: http://www.nist.gov [Accessed July, 2008].

NVIDIA, *Product Information*, Nvidia Corporate Website, 2009. [Online] http://www.nvidia.com/page/products.html [Accessed March, 2009].

Oppenheim, B.E., "Reconstruction tomography from incomplete projections", *Reconstruction Tomography in Diagnostic Radiology and Nuclear Medicine*, Baltimore, MD, University Park Press, 1975.

Phelps, M.E., et al., "Application of annihilation coincidence detection to transaxial reconstruction tomography", *J. Nuc. Med.*, vol. 16, pp. 210-224, 1975.

Ramakrishnan, R.S., Mullick, S.K., Rathore, R.K.S., Subramanian, R., "Orthogonalization, Bernstein polynomials, and image restoration", *App. Opt.*, vol. 18, pp. 464-468, 1979.

Saad, Y., van der Horst, H.A., "Iterative solution of linear systems in the 20^{th} century", *J. Comp. App. Math.*, vol. 123, pp. 1-33, 2000.

Saad, Y., *Iterative Methods for Sparse Linear Systems*, PWS Publishing Co., Boston, 1996.

Schretter, C., "A fast tube of response ray-tracer", *Med. Phy.*, vol. 33, no. 12, pp. 4744-4748, 2006.

Siddon, R.L., "Prism representation: a 3D ray-tracing algorithm for radiotherapy applications", *Phys. Med. Biol.*, vol. 30, pp. 817-824, 1985.

Siegel, S., *Internal design documentation for transmission processing*, Siemens Molecular Imaging, Preclinical Solutions, 810 Innovation Dr., Knoxville, TN 37932, Internal Document, 2001.

Sossi, V., Stazyk, M.W., Kinahan, P. E., Ruth, T. J., "The performance of the single-slice rebinning technique for imaging the human striatum as evaluated by phantom studies", *Phys. Med. Biol.* , vol. 39, no. 3, pp. 369-380, 1994.

Spencer, G.H., Murty, R.K., "General ray tracing Procedure" , *J. Opt. Soc. Am.*, vol. 52 no. 6, pp. 672–678, 1962.

Spinks, T.J., Bloomfield, P.M., "A comparison of count rate performance for ^{15}O-water blood flow studies in the CTI HR+ and Accel tomographs in 3D mode", *IEEE Nuc. Sci. Sym. Conf. Rec.*, vol. 3, pp. 1457-1460, 2002.

Stoer, J., Bulirsch, R., *Introduction to Numerical Analysis*, First Edition, Springer-Verlag, 1980.

Swann, B.K., et.al, "A custom mixed signal CMOS integrated circuit for high performance PET tomography front-end applications", *IEEE Nuc. Sci. Sym. Conf. Rec.*, vol. 1, pp. 24-28, 2002.

Townsend, D.W, "A Combined PET/CT Scanner: The Choices", Letter to the editor, *J. Nuc. Med.*, vol. 42, no. 3, pp. 533-534, 2001.

Vandenberghe, S., et.al., "Correction for external LOR effects in listmode reconstruction for PET"; *Proceedings IEEE Intl. Sym. Biomed. Img.*, pp. 537 – 540, 2002.

Walton, G.H., Poore, J.H., Trammell, C.J., "Statistical testing of software based on a usage model", *Software-Practice and Experience*, vol. 25, no. 1, pp. 97-108, 1995.

Wienhard, K., et.al., "The ECAT HRRT: performance and first clinical application of the new high resolution research tomography", *IEEE Trans. Nuc. Sci.*, vol. 49, no. 1, pp. 104 – 110, 2002.

Wu, X., "An efficient anti-aliasing technique", *ACM Comp. Graph.*, vol. 4, no. 25, pp. 143-152, 1991.

Appendix

General System Information

A.1 Transmission Sources

Two different transmission sources are available for the DPET, Cobalt-57 (122 keV) and Americium-241 (60 keV), in strengths up to 5 mCi [Siegel, 2001]. Both of these sources are packaged in a standard sealed case by Isotope Products, Inc., and are interchangeable physically. The window of the tungsten beam collimator is 2.0 mm across, and the projected fan limits coverage to 90 degrees [Beach, 2004].

A.2 511 keV Detector Performance Evaluation

The DPET is optimized for performance at 511 keV. Scanning with a Ge-68 point source yields typical energy resolution of around 14%, shown in Figure 65. Very good definition between individual pixels is exhibited with additional separation visible every third pixel given the 3:2 multiplexing scheme is shown in Figure 66. Optimum separation for this detector is determined at 511 keV, as this is the energy it was designed to operate at [Mintzer, 2007]. High overall light yield manifests itself with excellent separation with a peak to valley ratio of 50%. Fitting the sample cross section data to a pair of overlapped Gaussian distributions (Figure 67) yields a FWHM of 5.88 bins, or 0.94 mm. for each crystal. This provides an estimate of single crystal performance. Selecting an individual pixel from the reference profile yields a FWHM measurement of well under the width as the crystal boundary itself. In this case, 10 bins in electronic space, with a corresponding energy resolution of 14%. Thus, positioning accuracy is better than a detector element in 95% of all interactions.

A.3 122 keV Detector Performance Evaluation

In transmission mode for Co-57 (122 keV energy), the detector does not perform as well as only 24% of the light yield as the 511 keV case is realized. Typical energy resolution is 24% (Figure 68) with some visible definition between individual pixels and the 3:2 multiplexing scheme (Figure 69). Typical peak to valley performance in the 122 keV case is 20% (Figure 70). Fitting a Gaussian distribution to the 122 keV data yields a FWHM of 8.06 bins, or 1.28 mm.

A.4 60 keV Detector Performance Evaluation

Available light photons for a 60 keV interaction event are approximately 12% of that of a 511 keV event. This further degrades all performance parameters. Typical energy resolution is 35% (Figure 71) with limited definition between individual pixels and only the 3:2 multiplexing scheme visible (Figure 72). Positioning performance at 60 keV is degraded from the 511 keV case. It is not possible to discern individual pixels (Figure 72) so a fit is not obvious, but given the same triple-wide Gaussian form as the previous examples, a FWHM of 10.1 bins and 1.62 mm is estimated (Figure 73).

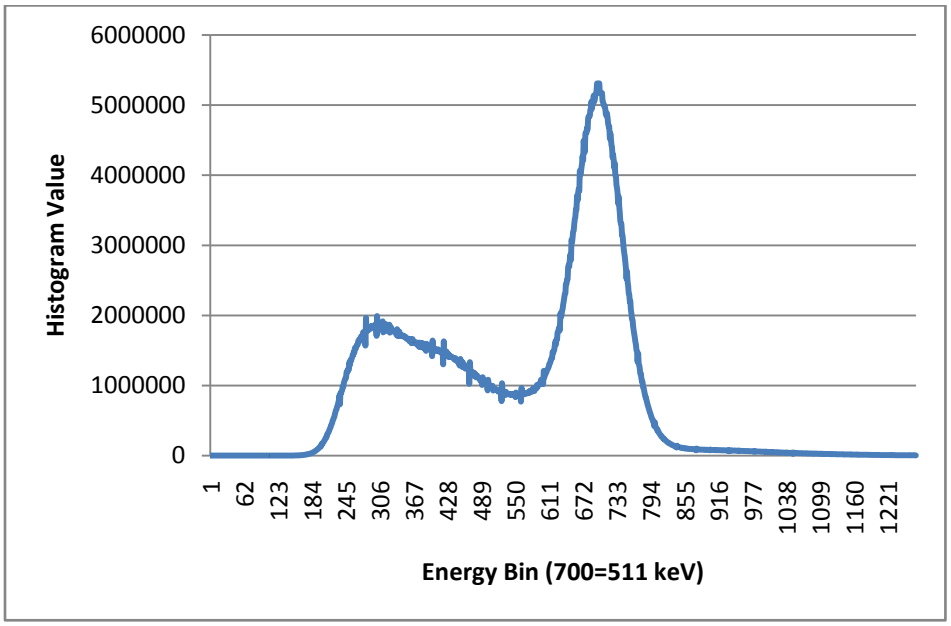

Figure 65. Composite Energy Spectrum 511 keV

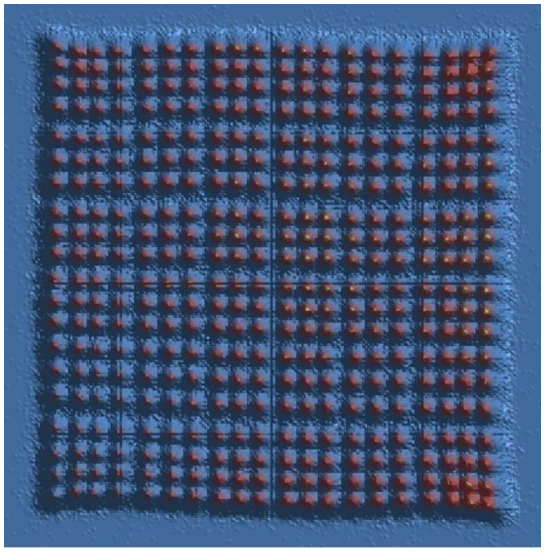

Figure 66. Sample 511 keV position profile

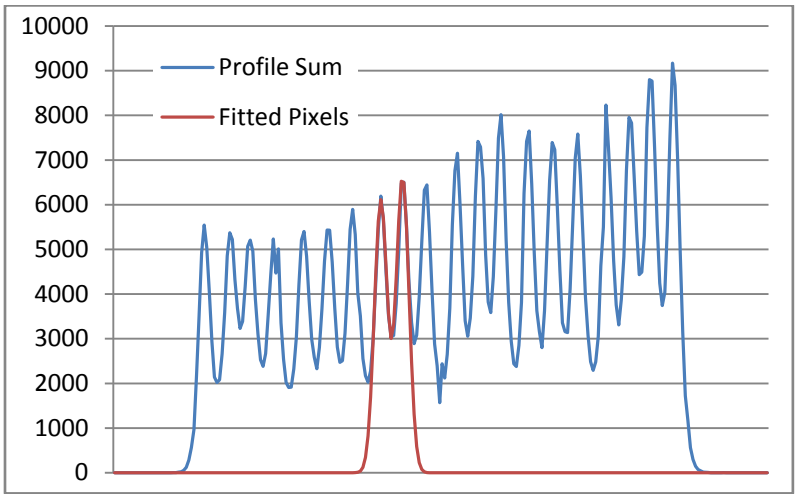

Figure 67. Sample cross section of position profile at 511 keV

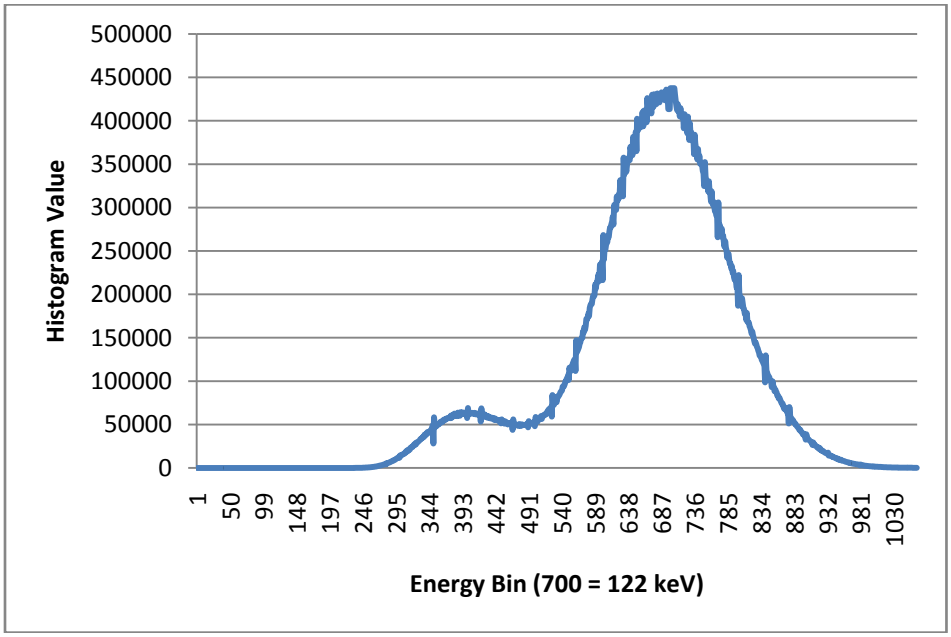

Figure 68. Composite Energy Spectrum 122 keV

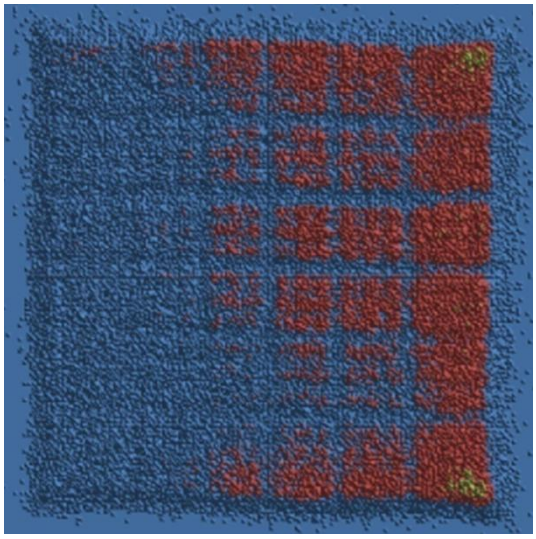

Figure 69. Position profile measured at 122 keV

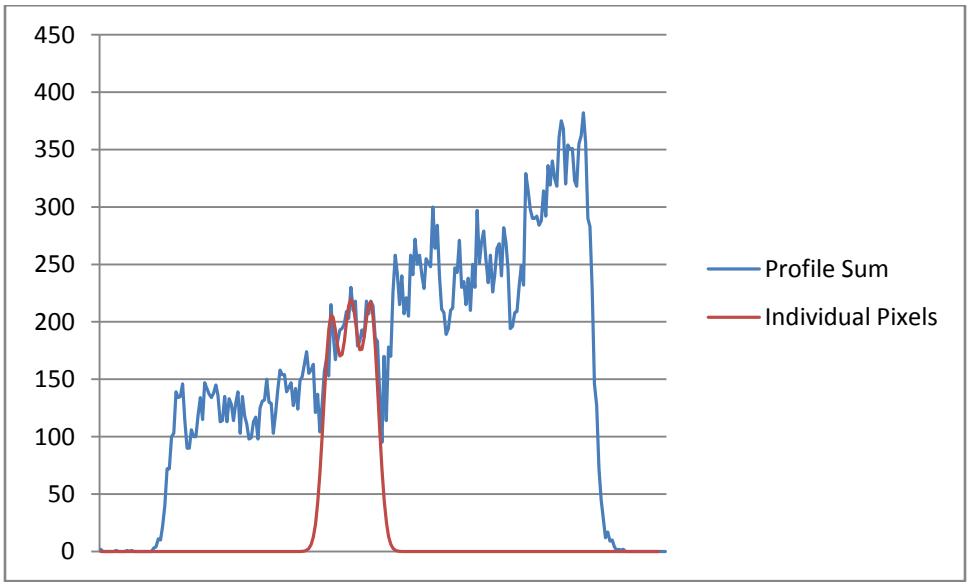

Figure 70. Sample cross section of position profile at 122 keV

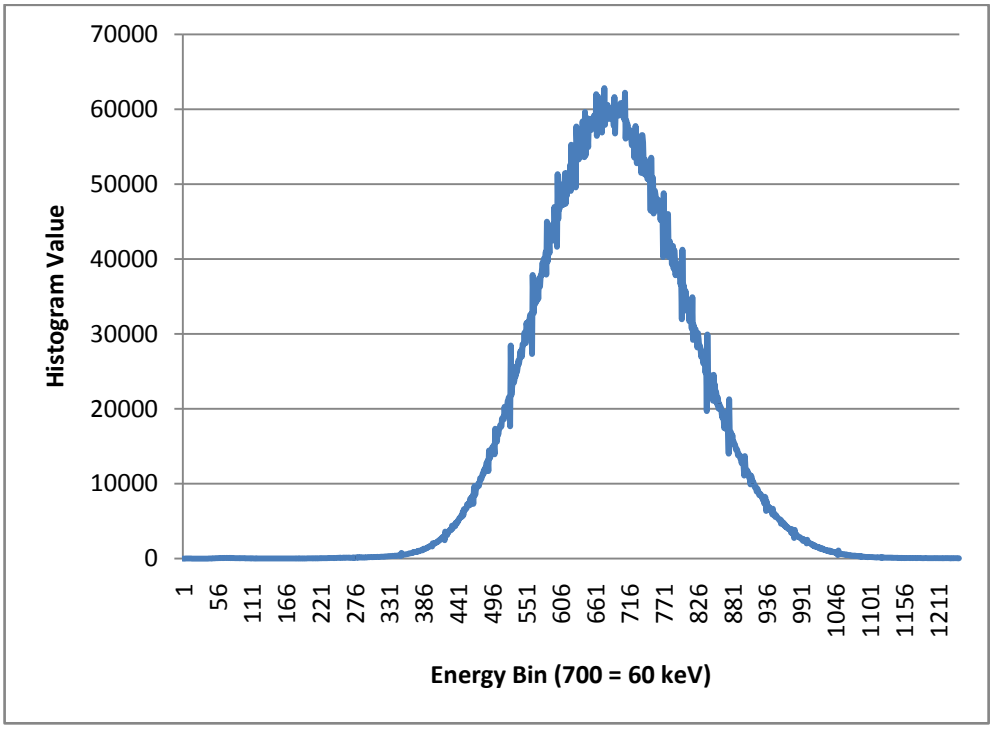

Figure 71. Composite Energy Spectrum 60 keV

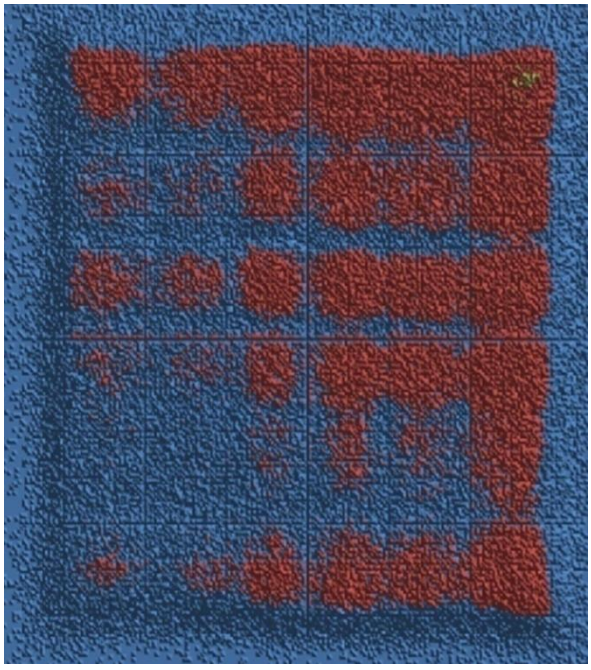

Figure 72. Position profile at 60 keV

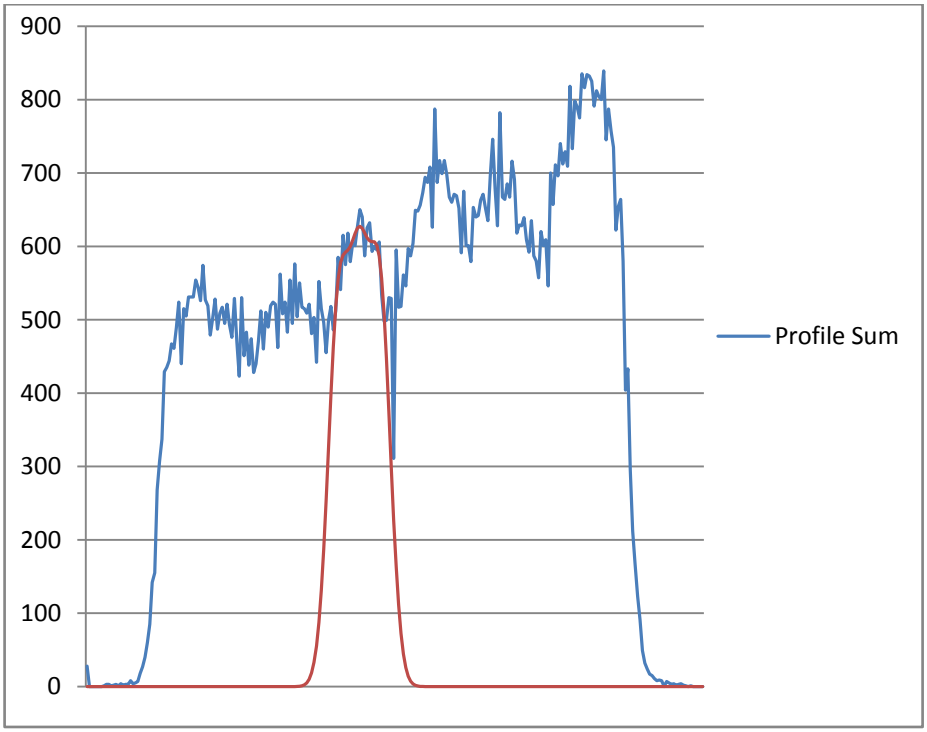

Figure 73. Sample cross section of position profile at 60 keV

A.5 Conclusions on Positioning Accuracy

At energy levels nearly an order of magnitude below the expected operating range of the detector, the system is able to resolve position correctly 75% of the time. The results for all energies are summarized in Table 9. Overall, the detector spatial resolution is similar to the size of the collimator aperture (2.0 mm). The rays are essentially parallel sided, and for this reason they are modeled as parallel sided in the system model. Furthermore, a reasonable voxel size for this system is half the source aperture at 1.0 mm.

Table 9. Detector Spatial Resolution vs. Photon Energy

Photon Energy	**Position Spread FWHM**	**Percent within a single crystal**
511 keV	0.94 mm	95%
122 keV	1.28 mm	85%
60 keV	1.62 mm	75%

Vita

Mark Lenox was born in 1967 in San Diego California. He spent his early childhood in Minnesota, and moved to Arizona in his early teens. He did his undergraduate work at Arizona State University in Systems Engineering, then went on to graduate studies at Texas A&M where he received as MSEE in 1990. Most of his commercial experience is in the field of Positron Emission Tomography, first at CTI PET Systems, then at Siemens Molecular. He went back to school in 2007 to complete a PhD in Computer Science at the University of Tennessee. Upon graduation he intends to relocate back to College Station, Texas, to work at the Texas Institute for Preclinical Studies.