NEW ALGORITHMS FOR COMPUTED TOMOGRAPHY IMAGE RECONSTRUCTION TO ELIMINATE ARTIFACTS

By
Eli Lechtman
B.Sc. (Hons.), University of Western Ontario, London, ON, Canada, 2006

A thesis
presented to Ryerson University
in partial fulfillment of the
requirements for the degree of
Master of Science
in the Program of
Biomedical Physics
Toronto, Ontario

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. I authorize Ryerson University to lend this thesis to other institutions of individuals for the purpose of scholarly research.

[redacted]

I further authorized Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

[redacted]

New Algorithms for Computed Tomography Image Reconstruction to Eliminate Artifacts

Eli Lechtman B.Sc. (Hons)

M.Sc., Department of Physics, Ryerson University, 2008

ABSTRACT

Computed tomography (CT) relies on computational algorithms to reconstruct images from CT projections. Current filtered backprojection reconstruction methods have inherent limitations in situations with sharp density gradients and limited beam views. In this thesis two novel reconstruction algorithms were introduced: the Algebraic Image Reconstruction (AIR) algorithm, and the Geometric Image Reconstruction Algorithm (GIRA). A CT simulation was developed to test these novel algorithms and compare their images to filtered backprojection images. AIR and GIRA each demonstrated their proof of principle in these preliminary tests. AIR and its extension, the Parsed AIR algorithm (PAIR), were able to reconstruct optimal images compared to filtered backprojection after empirically determining parameters relevant to the algorithms. While GIRA reconstructed optimal images in preliminary tests, reconstruction was complicated by error propagation for larger imaging domains. The initial success of these novel approaches justifies continued research and development to determine their feasibility for practical CT image reconstruction.

ACKNOWLEDGEMENTS

This thesis was completed with the help and support of many individuals and organizations.

Dr. S. P. Goldman *Thesis Supervisor*	-	Dr. Goldman's guidance, direction, and availability formed the backbone of this research.
Nancy Ford *Committee Member*	-	Nancy offered much advice and knowledge in the practical aspects of CT as well as thesis writing.
Michael Kolios *Committee Member*	-	In committee meetings, Michael's questions and suggestions helped focus the thesis research.
Carl Kumaradas *Director of Graduate Studies*	-	Carl was always available to answer policy related questions and gave useful suggestions.
Ryerson University	-	Ryerson offered both financial support for the past two years, and a helpful environment conducive to learning.
Natural Sciences and Engineering Research Council of Canada (NSERC)	-	Funding from 2007-2008.
Ben and Marilyn Lechtman *My Parents*	-	Without their encouragement and support, this thesis would not have been completed.
Ryerson Staff, Faculty, and Fellow Students	-	The biomedical physics department at Ryerson was an environment filled with people that truly cared about each other. Without their constant support this research would not have been completed.

TABLE OF CONTENTS

Chapters:

LIST OF FIGURES

GLOSSARY OF TERMS

AIR — The Algebraic Image Reconstruction algorithm developed in this thesis.

Beamlet — The path of a beam that contributes to a single detector. Each detector has its own respective **beamlet.** In contrast to a ray, which is defined to have no width, a beamlet has the width of its corresponding detector size.

Data Collection — Denotes the entire computed tomography scanning process.

GIRA — The Geometric Image Reconstruction Algorithm developed in this thesis

Image Artifact — A degradation in image quality that may render the image unfit for its intended purpose.

Linear Attenuation Coefficient — A measure of an x-ray beam's attenuation per unit thickness in a given material (depends on the x-ray energy and material properties). Other terms that are used synonymously in this thesis are **absorption density, density, attenuation.**

PAIR — The Parsed Algebraic Image Reconstruction algorithm developed in this thesis. **PAIR** is an extension of **AIR**

Phantom Grid — A matrix of values defining the Shepp and Logan head phantom used in the CT simulation.

Phantom Pixel A pixel in the phantom matrix.

Projection Describes the integral sum of the attenuation along a **beamlet** path:

$$P = -\ln\left(\frac{I}{I_o}\right) = \int_L \mu(x)dx$$

where I is the detected x-ray intensity, I_o is the input x-ray intensity, and $\mu(x)$ is the attenuation coefficient of the material at a depth x.

Projection Function A set of projection measurements from one beam angle.

Reconstruction Pixel A pixel in the reconstruction grid. Also called image pixel.

Reconstruction Resolution The resolution of the reconstructed image defined in the reconstruction algorithm.

CHAPTER I

INTRODUCTION

1.1 An Introduction to Computed Tomography Imaging

The success of computed tomography (CT) for diagnostics and treatment planning rests on its ability to produce accurate images that represent the true structure and densities of the subject being imaged. CT is aptly named as it depends on computer algorithms (computed) to reconstruct slices (tomography) of images comprising the three-dimensional structure that has been scanned. CT imaging takes advantage of radiation absorption in tissues, relying on the fact that the attenuation of an x-ray beam is a function of the internal structures' electron and proton densities or effectively, attenuation coefficients, μ [1,2].

The process of obtaining CT images can be broken down into two steps: data collection and image reconstruction. In data collection, the object or patient being imaged is placed between the x-ray source, which emits an x-ray beam, and the detector array, which detects radiation that has not been absorbed in the object. For each image slice, the CT source and detector rotate around a gantry to collect information from a full range of angles. The object or patient being imaged is moved through the gantry to allow many slices to be scanned. At each angle the data collected is a discretized function called a projection function, effectively describing the total absorption along the various paths from the source to each detector in the array. Data collection is complete once projections have been obtained for all slices at a range of angles [1,2,3].

In image reconstruction, a mathematical algorithm is used to determine the structures and densities of the imaged region from the collected projections. Unfortunately, exact images are not possible. CT has intrinsic noise considerations related to (but not limited to) the stochastic nature of the interaction of radiation with matter, as well as noise in the detectors. Furthermore, the process of reconstructing an image from the projection functions is not trivial. The image reconstruction algorithm can be a contributing factor towards poor or unrepresentative images, for example yielding unwanted image artifacts (image artifacts will be discussed in more detail in a subsequent section). For this reason there has been much investigation into image reconstruction algorithms. Current accepted reconstruction algorithms are based on a process called filtered backprojection (which will be described in a later section) with subtle variation in implementation between facilities.

The research in this thesis focuses on developing fundamentally distinctive image reconstruction alternatives that could optimize image quality for a given CT scan. An optimal reconstruction algorithm can have important medical implications in areas ranging from radiation treatment planning, where representative attenuation coefficients are needed, to diagnostic imaging, where false positives or undetected disease due to poor images are a concern [1-5].

Optimizing image reconstruction can also have significant implications for radiation safety. By improving image contrast and eliminating artifacts, it may be possible to reduce the dose to patients required to obtain clear images. Furthermore, advancements in reconstruction algorithms are more easily and cheaply adapted in a clinical setting compared to CT system design advancements, which necessitate new and often expensive hardware.

Before going into the details of various reconstruction algorithms it will be useful to examine the basics of CT imaging. This will help lay the groundwork for the investigation, and define terms that will be referred to throughout this paper.

1.2 The Basics of Computed Tomography Imaging

X-ray source

The two main components of any CT machine are the x-ray source and the x-ray detector. The source consists of an x-ray tube, responsible for the production of a high-energy photon beam. This high-energy photon beam is achieved by heating electrons off a cathode, and accelerating them through a potential difference towards an anode consisting of a heavy metal target. As the electrons bombard the target they interact with the metal. While over 99% of the input energy is lost to heat, some electrons are destined for more relevant interactions. There are two important types of interactions that contribute to the intensity spectrum of the output photons (**figure 1.2.1**) [1].

Some electrons will experience Bremsstrahlung interactions producing high-speed photons of peak energy equal to the kinetic energy of the incident electron. X-ray beam energies are typically defined by this peak energy indicated by KVp, while in reality an x-ray beam consists of a spectrum of energies. Bremsstrahlung interactions refer to electrons that are decelerated due to the opposing electric force as they approach the nucleus of an atom. This rapid transfer of energy creates Bremsstrahlung radiation, which results in a spectrum of photon energies, whose highest energy has a forward trajectory relative to the axis of the electron flux [1,2].

The second type of interaction that affects the photon spectrum is so-called characteristic radiation. When an incident electron bumps an inner shell electron from its orbit, and an outer shell electron replaces it, a characteristic photon is emitted with energy equal to the difference between the binding energies of the two shells. This phenomenon heavily impacts the photon intensity spectrum and is responsible for the sharp peaks that can be seen in the figure (**1.2.1**). Ideally what is desired is a monochromatic beam energy rather than a spectrum of energies. For this reason various techniques are employed to narrow the spectrum such as the use of x-ray filters, which are built into the CT machine. Even with beam hardening filters, a monochromatic flux of photons cannot be achieved [1,2].

Typical CT x-ray energies range from 80 KVp-140 KVp, though MVp x-rays are used in certain circumstances [7]. While KeV photons offer better soft tissue contrast, MeV x-rays can be used in combination with a medical linear accelerator, for treatment planning and patient setup. This thesis focuses on CT systems with KeV range photons.

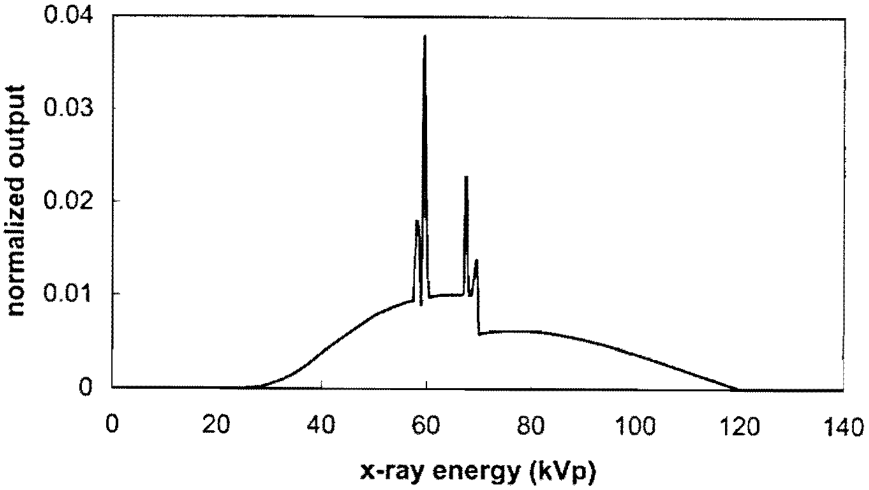


Figure 1.2.1: The normalized output intensity spectrum of photons due to the bombardment of 120kV electrons on a heavy metal target as a function of photon energy. The main curve is due to the contribution from Bremsstrahlung interactions while the sharp spikes are caused by so called characteristic interactions [1].

Interactions of x-rays with matter

At energies in the KeV range, photons interact with matter through a number of interactions. The main interactions that concern CT are the photoelectric effect and Compton scattering. The photoelectric effect refers to a photon that transfers all its energy into liberating an orbiting electron. This is only possible when the photon has greater energy than the binding energy of a given shell. The remaining energy is imparted to the electron as kinetic energy. This process also releases a characteristic photon when an outer shell electron fills the hole of the ejected electron. Due to the low binding energies of tissues, the resulting characteristic photon will not travel far before being absorbed in the tissue. The probability of the photoelectric effect is proportional to the cubic proton density, Z^3 of the material and inversely proportional to the cubic photon energy, E^3 [1-4].

Compton scattering refers to a photon that knocks an electron out of its orbit but does not transfer all of its energy to the electron. The photon is deflected at an angle much like a billiard ball and continues on to interact with the tissue. The probability of Compton scattering depends on the electron density of the material and the energy of the photon. **Figure 1.2.2** depicts the relative contributions of Compton scattering and the photoelectric effect in water [1].

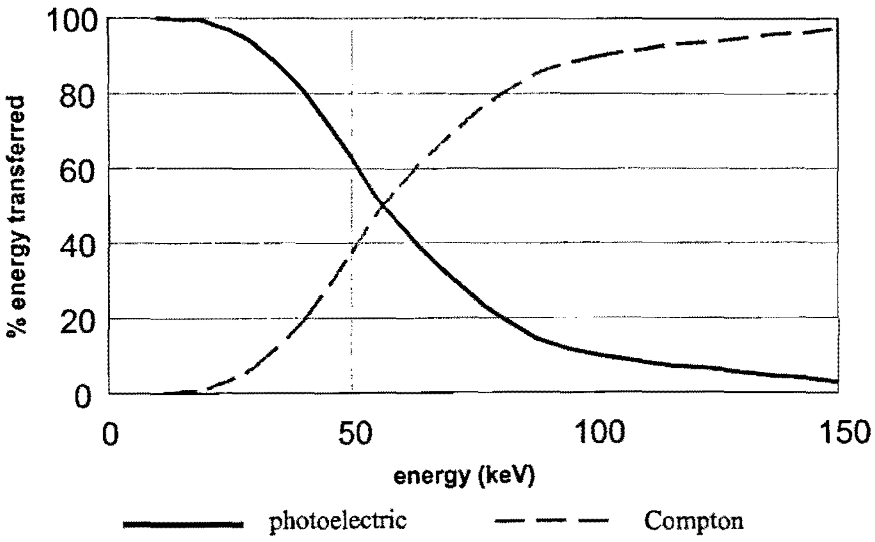


Figure 1.2.2: This graph shows the relative contributions of Compton interactions and the photoelectric effect as a function of photon energy. Plotting percent energy transfer gives a more descriptive representation as opposed to the percent of the total interactions as it includes a measure of the impact of those interactions. This plot was obtained using photons incident on water [1].

Both of these interactions contribute to the attenuation of photons from the source to the detector. Because the probability of both these interactions depends on the tissue, it is intuitive that by examining the out-coming photon intensity one could make some inferences about the material that has been traversed. Mathematically, the intensity of a monochromatic photon beam traveling through a homogeneous medium, can be described by Lambert-Beer's law as an exponential decay [1,2]:

(1.2.1)
$$I = I_o e^{-(\tau+\sigma)L}$$

Where ***I*** is the out-coming photon intensity (as measured by the detector), ***I_o*** is the input photon intensity, τ and σ are the attenuation coefficients (which is a representation of the probability of an interaction occurring per cm) of the photoelectric effect and Compton scattering respectively, and *L* is the length of the material that has been traversed. The attenuation coefficients are commonly represented as a sum, called the linear attenuation coefficient, μ. Keep in mind that this coefficient is a function of both the photon energy and the material. At 100 KeV, μ_{fat}= 0.154 cm^{-1}, μ_{muscle}= 0.176 cm^{-1}, μ_{bone}= 0.298 cm^{-1}, and μ_{lead}= 4.076 cm^{-1} [4].

If a photon beam passes through a non-homogeneous region, each infinitesimal region will have its corresponding μ value and the output intensity will be a function of the integral sum of the linear attenuation coefficients along the beam's path. Therefore equation (1.2.1) can be represented in the well-known form [1,2]:

(1.2.2)
$$I = I_o e^{-\int_L \mu(x)dx}$$

Where μ is now a function of the depth, which in general is not homogeneous. By taking the natural logarithmic ratio of ***I*** and ***I_o*** one can solve for the integral sum of all μ values along the photon path:

(1.2.3)
$$P = -\ln\left(\frac{I}{I_o}\right) = \int_L \mu(x)dx$$

Here, P is defined as the projection value due to a monochromatic photon beam incident on any material. It is using these projections that CT image reconstruction algorithms attempt to find all values of μ as a function of the three spatial dimensions. The μ values can be graphically represented to form an image. To improve image contrast what is graphically represented is the CT number, which is a rescaling of μ expressed in Hounsfield Units (HU):

$$CT\_number = \frac{\mu - \mu_{water}}{\mu_{water}} 1000 \qquad (1.2.4)$$

Where μ_{water} is the attenuation coefficient of water (this thesis does not rely on Hounsfield units; rather, linear attenuation coefficients or density are used interchangeably throughout this paper). Typical tissue ranges from -100 to 100 HU and bone is above 1000 HU facilitating high contrast [4].

Much like a simple x-ray image, it is not possible to deduce axial resolution from one projection. As equation (1.2.3) suggests, one can only know the sum of the attenuation coefficients along the beam path. However, collecting projection data from a full range of angles allows for the possibility of resolving the μ values along any axis. Of course, the process of resolving those μ values from the projection data is the topic of this thesis.

One issue with equation (1.2.3) is that, as mentioned before, an x-ray beam is not monochromatic. The simple exponential decay does not hold because the attenuation coefficients are a function of the photon energy. Furthermore, equation (1.2.1) describes the attenuation of ballistic photons, while in reality scattered photons can continue to interact through the medium creating noise in the projections. These issues are complex and beyond the scope of this discussion, but it should be noted that the equations outlined here only apply to ideal conditions. In practice, additional processing and compensation is required to interpret the projection information accurately [1,2]. In this research, an ideal monochromatic model is assumed for simplicity. Now that we have examined the x-ray source as well as the propagation of x-rays through tissue, let us examine the final step of data acquisition, namely, the detectors.

X-ray detectors

The function of a detector is to deduce the exiting intensity of photons. This can only be achieved indirectly by correlating some signal to photon count. There has been much development regarding detectors in CT devices over the last 30 years. Early CT machines rely on detectors filled with pressurized xenon gas that functions much like an ionization chamber. Each detector in the array is constructed with two thin tungsten plates, one that is connected to a high-voltage supply and one that is at zero volts. As a photon enters the detector, it may interact with the gas through the photoelectric effect thus causing ionization. The positive xenon ions will be attracted to the lower potential while the electrons will be attracted to the positive potential creating a current. If the bias voltage across the plates is set properly, the current will be proportional to the total energy of the photons that entered the detector. This measurement can then be related to the photon intensity. While this type of detector is inexpensive and was useful in older CT applications, a major problem is that a photon may pass through the xenon gas without interacting due to the low density of the gas [1,3,4,6].

A more efficient detector type called a solid-state detector makes use of a scintillating material such as $CdWO_4$, Gd_2O_2S or HiLight. As photons enter the detector they interact with the scintillating material though the photoelectric effect releasing photoelectrons. As these electrons interact with other electrons in the material, characteristic radiation is emitted in the form of light. This light is then detected by photodiodes located at the bottom of the detectors, and it is this signal that is related to the intensity of the photons entering the detectors. This type of detector is the basis of many CT machines today, some using amorphous silicon-based flat panel detectors to cover a large detector region with high spatial resolution (each detector pixel is about 100-200 microns) [1,3,4,6].

There are many factors that play a role in the effectiveness of detectors including linearity with respect to energy and intensity of the photon flux, linearity with respect to ambient temperature, the potential of radiation damage to the detectors, inter-detector (between two detectors) and intra-detector (inside a detector) uniformity, as well as the electronics associated with the measurements. Another crucial factor is the spacing between adjacent detectors in the array and the individual detector width. Both of these parameters affect the sampling rate of the projection measurements, and can have significant effects on image quality [1,2]. Now that we have discussed some of the basic components of CT machines, let us examine how all these components are utilized in commercial devices.

CT source-detector geometry

Of much interest to this thesis concerns the geometry involved in data acquisition. As CT has matured since its inception some 40 years ago, there have been a number of major developments in CT source-detector geometry for the main purpose of reducing scan time. The first clinical CT machine built in 1971 is aptly called first-generation. In this setup, the source emits a narrow pencil beam a few millimeters wide that traverses the patient and is detected at the opposite side by a single detector. The source and detector are then incrementally linearly translated to make single measurements of the subject **(figure 1.2.3)**. Following this, the source and detector are rotated one degree and the process is repeated for 180 degrees to obtain information for one slice (after 180 degrees the information obtained is simply a mirror representation of that angle minus 180 degrees). A projection function is defined by complete set of measurements at an angle. The projection function is discretized due to the detector width and spacing between adjacent measurements [1,3].

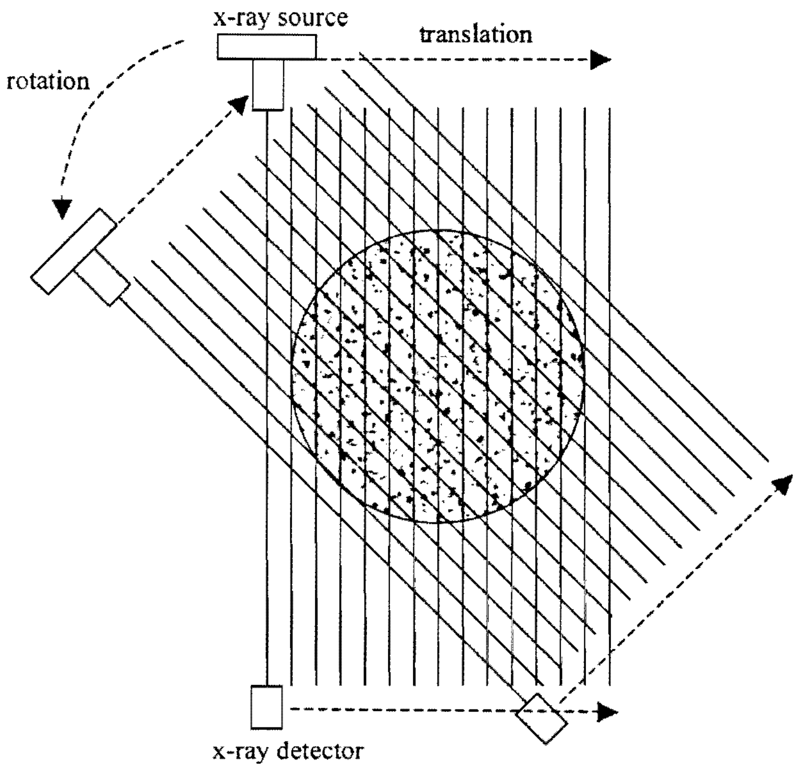


Figure 1.2.3: First generation CT geometry. A pencil beam and a single detector are translated across the object to collect projection information at one angle. The source and detector are then rotated and the process is repeated.

The first generation scanners have a conceptually simple geometry as the various beam paths can simply be described by a rotating coordinate system. As well, since there is only one detector, the issue of photon scatter plays less of a role. The major disadvantage of first generation scanner is that lengthy scan times (about four and a half minutes) translate into serious degradation in image quality due to patient motion [1,3].

Second generation scanners still rely on both translational and rotational motion, but in order to decrease scan time the beam is collimated into a fan shape (**figure 1.2.4**). Second generation scanners make use of an array of detectors instead of just one detector. This type of geometry can scan a complete slice in about 20 seconds [1,3].

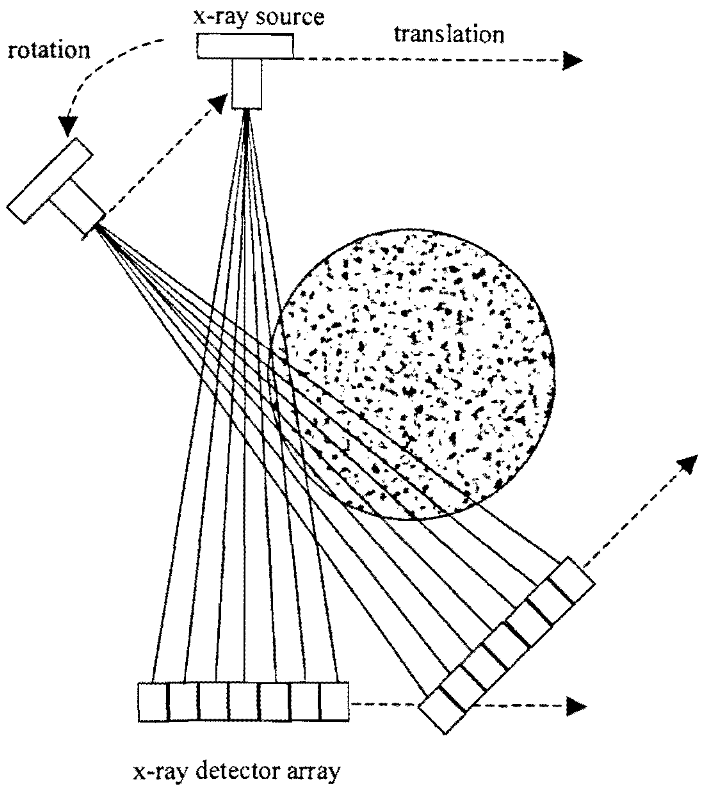


Figure 1.2.4: Second generation CT geometry. A fan beam and an array of detectors are translated across the object to collect projection information at one angle. The source and detector are then rotated and the process is repeated.

Third and fourth generation CT incorporates broader beams of either fan shape (for slice by slice) or cone shape (to scan multiple slices at a time) in conjunction with multiple detectors in an array. In third generation scanners, the fan beam width covers the entire object (or patient) eliminating the translational motion required in first and second-generation scanners. Fourth generation scanners incorporate a stationary ring of detectors, and a rotating source (**figure 1.2.5**). The novelty of fourth generation scanners is that each detector defines a unique set of projections collected over time, in contrast to a projection being defined by a set of adjacent detectors at an instant. The spacing between samples in a projection function is not determined by the detector spacing, but rather the sampling rate at which measurements are taken from a single detector [1,3].

While these developments have shortened scan time, to less than half a second per gantry rotation, issues of x-ray scatter, and reconstruction geometry have complicated image reconstruction and have introduced distinctive image artifacts.

The research in this thesis uses first generation scanner geometry as the developing and testing environment for the novel reconstruction algorithms, as it is the most simple to implement. An extension to more complex geometries is discussed where appropriate.

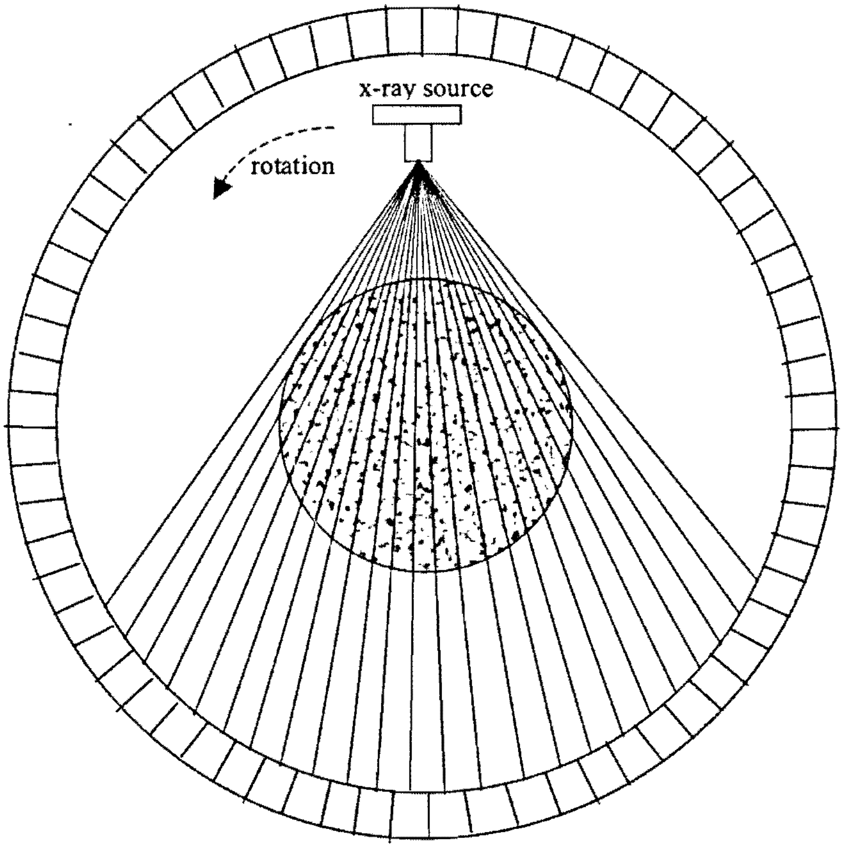


Figure 1.2.5: Fourth generation CT geometry. A diverging beam rotates around the patient, while a fixed detector ring measures out coming radiation. The sampling rate of the detector measurements determines the spacing between samples in a projection.

It is useful here to define a concept that will be referred to throughout the rest of this paper. Using first generation scanner geometry, one can say that the projections at any given angle stem from a common conceptual 'parallel beam' (while in reality there is no parallel beam, only a pencil beam that has been translated across the object), while each individual projection is made by a 'beamlet'. Just as a projection function is discretized by the detector array, each parallel beam can be thought of as consisting of discrete beamlets having a width equal to the detector width. The idea of a beamlet is commonly referred to as a *ray* in the literature. However a ray is usually defined as having no width [3]. The terms beamlet and detector are used somewhat interchangeably in this paper. They are defined to organize the CT information when discussing reconstruction algorithms.

1.3 Image Reconstruction Algorithms

Now that we have examined the basics of data acquisition and the physics of CT imaging, we can begin to explore the ideas behind image reconstruction. The conceptual goal of image reconstruction is to obtain a two-dimensional image slice from a series of one-dimensional projections. The simplest example of this is the two-dimensional point function. Any projection of the two-dimensional point function is a one-dimensional point function located at the position corresponding to the image **(figure 1.3.1)** [1,2].

Intuitively the reconstruction seems simple, but as the complexity of the function increases, the problem quickly loses its intuitive nature. The question then becomes: Through what mathematical process can an image be obtained from its projections? As it turns out the answer to this question is not trivial, and it has been the source of much research and development over

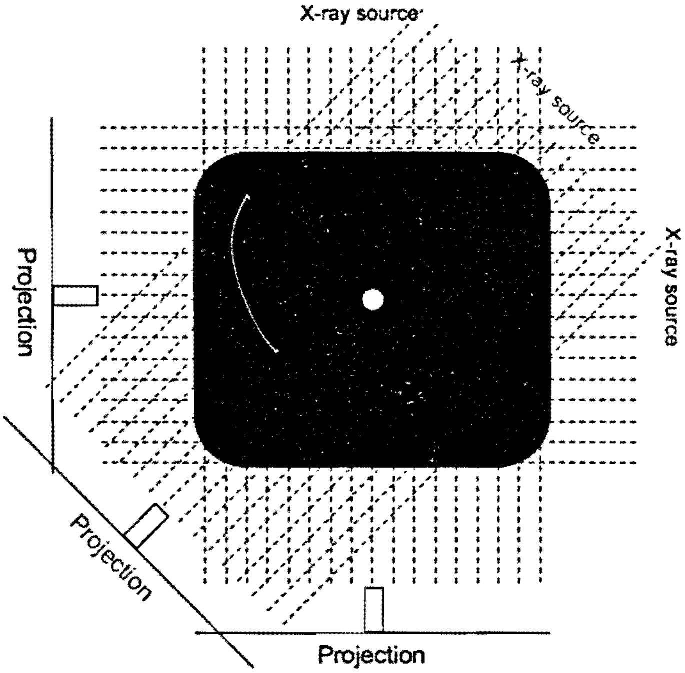


Figure 1.3.1: Parallel beam projections of a two-dimensional point function produces one dimensional point functions.

the last few decades. Presented here are various reconstruction approaches that have been investigated in the literature. For the sake of simplicity and elegance we will consider only one slice using 1st generation CT geometry.

Algebraic Reconstruction Algorithm

Let us begin with the most intuitive approach called the Algebraic Reconstruction Algorithm (ARA). First imagine a grid of four unknown attenuation coefficients μ_i. Suppose one wishes to image this unknown region using a CT scanner by taking five projection measurements each gathering unique information as shown in the (**figure 1.3.2**) [1,2].

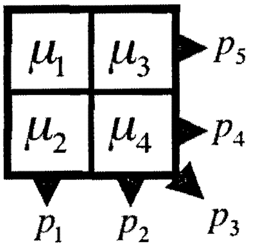


Figure 1.3.2: Theoretical projections of a region of four unknown densities

It should be noted that these measurements are not in general independent as is readily seen in this simple example ($P_1 = P_5 + P_4 - P_2$). From the definition of a projection above (equation 1.2.3), each projection is an integral sum of all the attenuation coefficients that are in that beamlet's path. Now supposing that every path-length in this example is two units long (though in reality it is not), corresponding to the two pixels that each beamlet has traversed, one obtains five equations:

$$(1.3.1)\quad \begin{aligned} P_1 &= \mu_1 + \mu_2 \\ P_2 &= \mu_3 + \mu_4 \\ P_3 &= \mu_1 + \mu_4 \\ P_4 &= \mu_2 + \mu_4 \\ P_5 &= \mu_1 + \mu_3 \end{aligned} \qquad \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix}$$

By solving the matrix equation **Ax=b** one could theoretically solve the unknowns and thus reconstruct the image. While this seems like a straightforward approach, ARA suffers from many limitations and difficulties. First of all, the matrix equation is highly unstable. When the system is underdetermined or over-determined solutions are often not possible and solving the equations results in non-physical or negative values. This issue is further pronounced when considering noise, radiation scatter, and measurement errors that cause inconsistencies in the matrix equations. In fact, Steve Webb, in his book *The Physics of Three-Dimensional Radiation Therapy,* describes the "impossibility of true inverse computed tomography" alluding to these issues [8]. Another limitation to ARA is its size constraint. ARA solves a matrix equation **Ax=b**, where **b** is a vector of projection values and **A** is a matrix of reconstruction pixel indices. The number of columns in **A** is determined by the number of image pixels in a slice (the desired reconstruction resolution), and the number of rows is determined by the number of detector measurements. In order to have a well-behaved system the number of rows should be of the same magnitude as the number of columns. Due to its theoretical simplicity, ARA was used on one of the first CT scanners in 1967. However, as the need for higher spatial resolution increased, huge computational difficulties arose given the very large size needed for array **A**. For example, for a reconstructed image resolution of 256×256 pixels, **A** would be an array of about 65,000×65,000 elements [1,2].

One final point should be noted regarding the previous example describing ARA. That example, taken from Jiang Hsieh's book, *Computed Tomography principles, designs, artifacts, and recent advances*, was simplified in an unreasonable way according to this researcher. Notice that any angled projection (which necessarily has a finite non zero width) must traverse partial pixels. The result of this is that the matrix **A** will contain non-integer values ranging from 0 to 1. The impact of these fractional contributions to a projection may be a highly overlooked source of error that will be discussed in the body of this paper. For now, let us look at alternatives to ARA that have been developed in order to offer a more practical reconstruction algorithm [1,2].

Iterative Reconstruction Techniques

Given the same example described above, another logical approach to the reconstruction involves iterative approximations. These techniques including ART (Algebraic Reconstruction Techniques) and SART (Simultaneous ART) make an initial guess at the true image, determine the projections of that image, and compare those projections with the true projection values. The difference between the true projections and the projections of a guess image is optimized in iterative steps. This process can best be described by a visual example (**figure 1.3.3**). It is easy

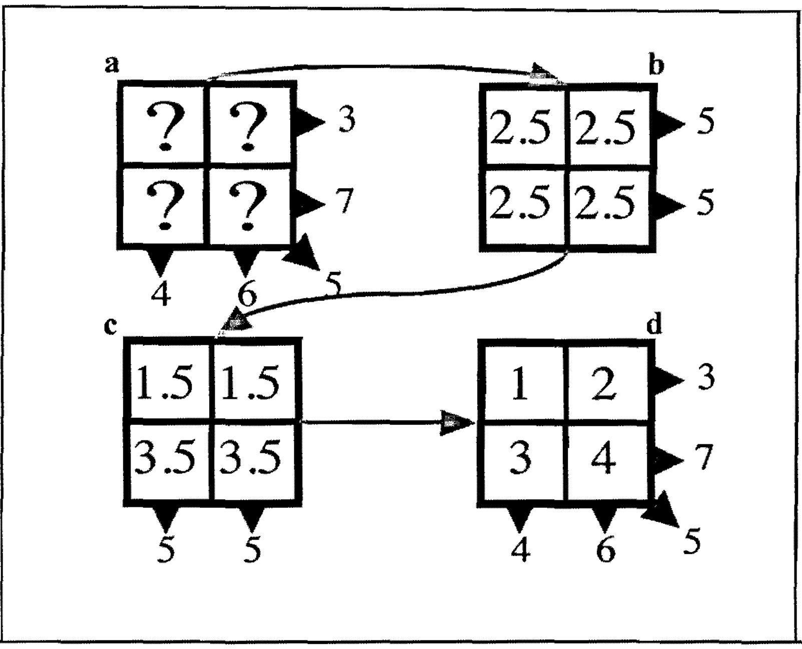


Figure 1.3.3: **a)** A region of four unknown densities is scanned producing five unique projections. **b)** The first iterative guess at the densities is based on the average value of the projections divided by the number of unknowns that each beamlet has passed through. This assumes that all unknowns have the same density. Following this, two theoretical projections are made from the estimated region and compared to the true projections. **c)** In the second iteration, the image is updated by adding or subtracting the difference between the theoretical projections and the true projections. Notice that the difference is applied evenly over the pixels that contribute to the projection in question. Following this, another set of theoretical projections in taken from the newly updated estimated image. **d)** After making the appropriate changes the theoretical projections of the third iteration match the true projections, and one can say that the image has been optimized [1].

to see that as the region becomes larger and more scans are taken, the complexity of the problem increases and these techniques often suffer a trade off between noise and convergence [1,2,9-12].

From the iterative example, one can understand an important concept in image reconstruction. This is the concept of backprojection. Backprojection is the process of applying a projection evenly, back along the beamlet's path. Given projections from only one angle of a region, backprojection represents the best possible estimate of the image. But what happens when one applies the backprojection algorithm for full set of projection data? **Figure 1.3.4** shows the backprojection process in steps attempting to reconstruct a point function located in the center of the region. There appears to be radial blurring of the point function, which represents the impulse response function of the backprojection algorithm. In order to recover the optimal estimate of the image the inverse of the impulse would have to be deconvolved from the result of the backprojection. This is a mathematical process whereby one filters the impulse response out of the backprojection. Together, the entire process has been aptly termed backprojection-filtering. However, this elegant concept is quite difficult to translate to application as the impulse

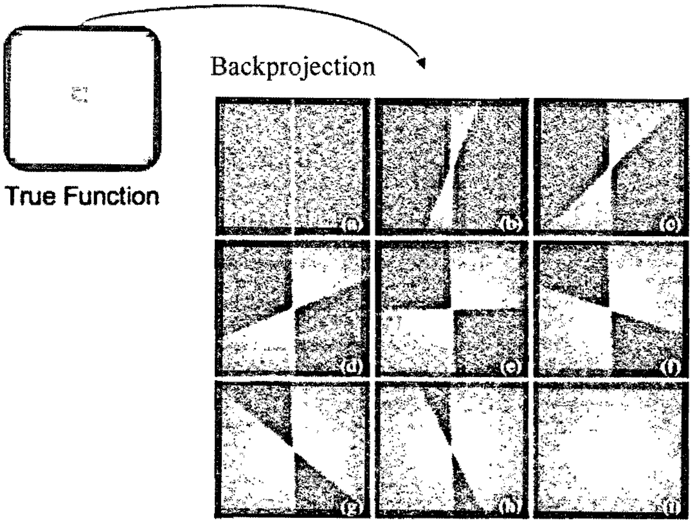


Figure 1.3.4: Using the backprojection method to reconstruct a two-dimensional point function from a full range of parallel projections. Images **a** through **i** depicts this process in steps. The result is an image with radial blurring [1].

response function is not easily definable, and the deconvolution involves non-trivial calculations in the space domain [1,2,13].

While this approach is not practically applied, it is a perfect introduction to discuss the most commonly used algorithm called filtered backprojection, which is based on the Fourier slice theorem. As the name implies the filtering takes place before the backprojection, which while being more conceptually difficult than backprojection-filtering, is more easily implemented. Before describing the filtered backprojection, it is important to understand the Fourier slice theorem.

The Fourier slice theorem

The Fourier slice theorem (for parallel beam geometry) states that the Fourier transform S_θ, of a projection $P_\theta(t)$, from a function $f(x,y)$, is equal to a slice of the two-dimensional Fourier transform $F(u,v)$, of that image subtending an angle θ **(figure 1.3.5)** [2].

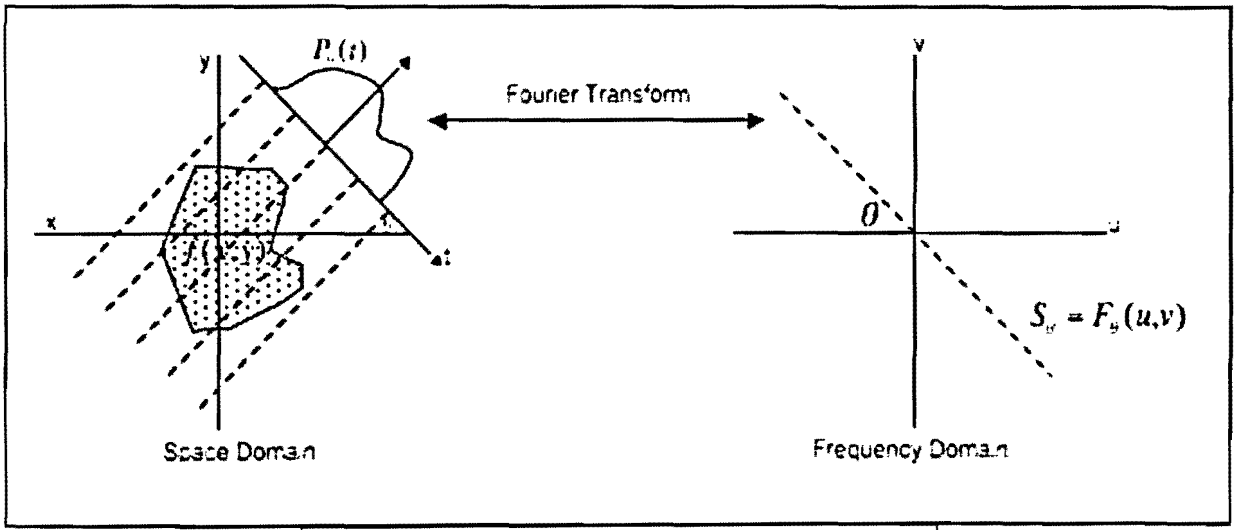


Figure 1.3.5: A depiction of the Fourier slice theorem.

In order to gain more insight into the Fourier Slice theorem a derivation is presented for a beam angle, θ=0. A more rigorous derivation can be found in textbooks on signal processing [2].

Beginning with an unknown two-dimensional function of x and y, $f(x,y)$, one can take a one-dimensional projection from any angle $p(t)_\theta$. Taking a projection at θ=0 gives:

$$p(x)_{\theta=0} = \int_{-\infty}^{\infty} f(x,y)dy \tag{1.3.2}$$

Taking the Fourier transform with respect to x on both sides gives:

$$\int_{-\infty}^{\infty} p(x)_{\theta=0} e^{-j2\pi xu} dx = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi xu} dxdy \tag{1.3.3}$$

Looking at the right side of the equation, it appears very similar to the two dimensional Fourier transform of $f(x,y)$. In fact:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(xu+yv)} dxdy \tag{1.3.4}$$

Evaluating $F(u,v)$ at v=0 gives:

$$F(u,v)\big|_{v=0} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi xu} dxdy \tag{1.3.5}$$

Thus it has been shown:

$$\int_{-\infty}^{\infty} p(x)_{\theta=0} e^{-j2\pi xu} dx = F(u,v)\big|_{v=0} \tag{1.3.6}$$

The Fourier slice theorem suggests that image reconstruction can be accomplished by building up the two-dimensional Fourier transform of $f(x,y)$ from the one-dimensional transforms of $p(t)_\theta$. Once the frequency space has been filled, an inverse Fourier transform should recover the reconstructed image [1,2].

While the Fourier slice theorem seems to be a theoretically sound approach to image reconstruction, it like other reconstruction methods, suffers from practical problems. A major difficulty arises when building the frequency space from the series of slices (each slice produced by an individual set of $p(\mathrm{t})_\theta$, contains a series of points, or samples corresponding to the detectors in the array). These slices form a pattern that can be described by a polar coordinate system (**figure 1.3.6**). However, in order to perform the two-dimensional inverse transform to recover the image, the samples must be interpolated onto a Cartesian grid. Interpolation in frequency space cause global artifacts in the reconstructed image. This is because each point in frequency space represents a frequency rather than a physical position. Furthermore, due to the polar orientation of the slices, the density of sample points decreases moving further from the origin. This causes the interpolation error to be greater for the higher frequency components. These as well as other issues do not allow the Fourier slice theorem to be applied for practical image reconstruction [1].

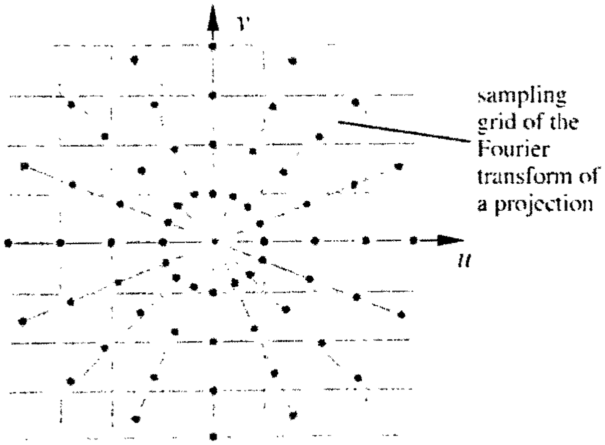


Figure 1.3.6: Sampling pattern from parallel beam projections in Fourier space based on the Fourier slice theorem [1].

Filtered backprojection

This brings us to the filtered backprojection algorithm, which is the most widely used algorithm employed today. The idea behind filtered back projection is to avoid summing the projections in the frequency domain. Instead, each projection is Fourier transformed, multiplied by a filter in the frequency domain, inverse transformed, and finally, backprojected on to the reconstruction grid. This process is done for each projection $p(t)_\theta$, and all the filtered backprojected images are summed up to form the final image. Interpolation and smoothing can then be performed on the image itself. The reason that a filter is needed is due to the polar coordinate sampling pattern. Looking at **figure 1.3.7 a)** if each transformed projection was a pie shape piece, then no filtering would be needed. However, due to the fact that each signal is shaped like a strip as **shown in b),** the lower frequencies are over sampled compared to the higher frequencies. It is thus necessary to approximate the pie shape with a weighting function that is shown **in c)** [1,2,13-19].

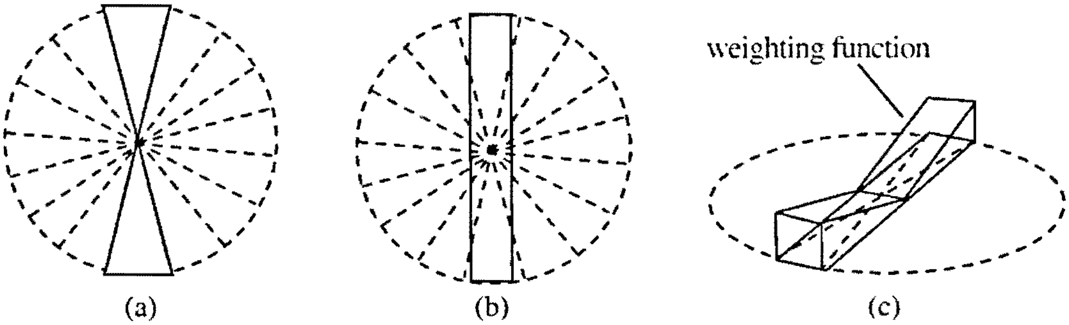


Figure 1.3.7: Illustration of filtered backprojection concept. **a)** Ideal frequency data from one projection. **b)** Actual frequency data from one projection. **c)** Weighting function in the frequency domain needed to approximate an ideal condition [1].

Notice that the filter is non-zero only in the region containing information from the given projection. The filter for each transformed projection assumes that all other frequency components are zero. In theory the filter would be a wedge function called the cropped Ram-Lak or ramp filter defined by $|f|$ where f is the frequency. However, due its sharp cutoff in the frequency domain, and its sensitivity to noise in the projections, this filter is typically modified through multiplication of a sine or cosine function to smooth it out [1,2,15].

The filtered backprojection algorithm has many advantages compared to other approaches that have been discussed. Instead of performing a large two-dimensional inverse Fourier transform, as is the case in applying the Fourier slice theorem directly, the filtered backprojection computes multiple inverse transforms of much less complexity (each one being essentially one-dimensional). With fast Fourier transform algorithms this can be accomplished quite fast [20]. The speed is increased further when considering that each backprojection is essentially independent and the filtered backprojection algorithm can begin reconstruction upon receiving the first projection measurements; it need not wait for all the CT data to be acquired. Filtered backprojection also solves the issue previously mentioned about backprojection filtering. It is well known that a convolution in the space domain can be performed more simply as a multiplication in the frequency domain. This is exactly the case in filtered backprojection; each $p(t)_\theta$ is filtered in the frequency by multiplication with a wedge filter before it is backprojected. It also solves the problems associated with the size constraints of ARA as the filtered backprojection algorithm stores only a small amount of information at a time. It is due to its effectiveness and efficiency that the filtered backprojection algorithm has become the method of choice for image reconstruction [1,2,13-19].

This is not to say that there are no problems associated with this method. The issues with filtered backprojection are a result of its reliance on the frequency domain and its inherent approximations. A major concern is that in order to properly reconstruct an image, the frequency domain must be sufficiently sampled. This precludes the possibility of a small number of angle views (ie: a full set of projection data is needed spanning all angles). As well, along the same lines is the necessity of a frequency cutoff of the wedge filter and the Fourier transformed projection data, which can have detrimental effects on the image. These issues and others will be examined in more depth in the next section on image artifacts [21].

1.4 Sources of Image Artifacts

The term image artifact connotes some degradation in image quality that hinders the image as a tool for its intended purpose. This somewhat qualitative definition is needed because no reconstructed image can perfectly represent the subject being imaged, therefore defining an artifact as a deviation between the image and the true object is difficult. There are many causes of image artifacts, each creating a unique type of image degradation such as streaking, shading, ringing, and bands. By examining the sources of these artifacts certain steps can be taken to reduce their effects. Some of these artifacts are inherent in the CT scanners themselves, or the scanning process (such as patient motion) while others are a function of the algorithms used for reconstruction. Although this research is aimed at reducing image artifacts due specifically to the reconstruction algorithms, presented here are many of the common artifacts found in CT. While textbooks have clearly defined the sources of many of these artifacts as stemming from the CT system design, most of this type of artifact actually stem from contributions from both the system design as well as limitations in the reconstruction algorithm [1,2,4].

Artifacts stemming from measurement errors

During the scanning process radiation scatter, patient motion, or mechanical malfunction can cause errors in isolated measurements. If the projection measurement is unrepresentative of the imaged region this will result in an unrepresentative reconstructed image. While this type of error cannot be totally avoided in the reconstruction, the graphical result will vary between different algorithms. The filtered backprojection algorithm causes streaking, blurring and ring artifacts in these cases due to the nature of the reconstruction filter. Radiation scatter affects the signal in the following way. Through Compton scattering, a photon may be deflected off the straight beam path and may enter another detector. The results of scatter are a lower signal to noise ratio and reduced contrast. The effects of scatter can be reduced with the use of collimators located on the detector that only allows entry of photons from a certain range of angles. Alternatively, the use of an air gap located between the patient and the detector allows some scattered photons to miss the detector array.

The effects of internal organ motion can be reduced by gating the x-ray beam in consonance with the breathing cycle. Alternatively, the real time visualization of organ motion, called 4D CT, has been enabled by the fast acquisition times of modern CT machines.

If a detector in the array has malfunctioned resulting in no measurement, the resulting image will contain artifacts characteristic of the CT geometry used. In third generation CT the result is a ring artifact [1,2,4].

Beam hardening

As mentioned above, Lambert-Beer's law only holds for a monoenergetic x-ray source. In reality, the x-ray tube produces a beam comprising a spectrum of energies (**figure 1.2.1**). Tissue attenuates lower energy photons more readily than higher energy photons, translating to energy dependant attenuation coefficients. Beam hardening refers to the phenomenon that as an x-ray beam passes through a subject, the lower energy photons are attenuated quickly, and the beam becomes proportionally more energetic as it travels further through the medium. This causes the measured projection to be less than it should be under ideal conditions. Beam hardening occurs in a non-linear fashion and the error that it causes increases with the beam path length. The characteristic artifacts of beam hardening are unwanted shading, blurring of boundaries, and streaks that are especially pronounced around sharp attenuation gradients. The effects of beam hardening can be reduced with the use of filters in the CT machine. By placing a metal filter (made of aluminum, copper, or brass) between the x-ray source and the patient, the beam can be hardened before it reaches the patient, resulting in a more narrow beam spectrum. Alternatively, computational methods such as two-pass beam hardening correction algorithms can be employed to reduce the effects of beam hardening. A two-pass algorithm refers to first reconstructing an image and then using the information of the positions of bone and soft tissue from that image, to create an accurate beam hardening correction for the final reconstruction [1,2,4].

Photon starvation

Photon starvation can occur in the presence of highly attenuating metal objects, dense bone, contrast agents, or obese patients. As the beam passes though this region, a large percent of the beam is attenuated producing a very small signal in the detector, translating to a very large projection measurement. Due to the logarithmic definition of a projection measurement (equation 1.2.3), a small error in the detector measurement in these circumstances results in a very large error in the projection measurement. Photon starvation can result in a low contrast high noise image, and can be responsible for severe streaking artifacts [1,2,4].

Aliasing artifacts

There are some artifacts, commonly referred to as aliasing artifacts that occur as a direct result of the filtered backprojection algorithm. The most fundamental source of these artifacts stems from fact that the actual imaged region is not bandlimited and thus exceeds the highest frequency that can be sampled. Specifically, Nyquist theorem suggests that errors in reconstruction will result if the projection data, $P_\theta(t)$, is sampled less than twice for its highest frequency component. This sampling is mainly a function of the CT system design set by the number of detectors in an array and the spacing between each detector. The characteristic artifact for under sampled data is streaking. Along the same lines, insufficient views (projection angles) will under-represent the frequency domain and will result in artifacts known as Moiré patterns

[2]. Sharp gradients present in the imaged region, such as metal inserts emphasize these aliasing artifacts. In many cases, the presence of a metal inserts severely limits the effectiveness of CT for treatment planning and diagnostic use [1,4,21].

It should be noted that Tretiak has derived an algorithm-independent lower bound for the mean squared error in a reconstructed image based on intrinsic noise in CT [22]. He has argued that the error estimated by Brooks and Dichiro [23] for filtered backprojection algorithms are close to this lower bound, leading him to conclude that no improvement can be made to reconstruction. However, the contradiction between the simple concept of reconstructing a point function, compared to the complex and unsatisfactory reconstruction methods currently available, has motivated this researcher to investigate new alternatives to CT image reconstruction.

1.5 The Point Function:

The research contained in this thesis began by examining the simple two-dimensional point function (**figure 1.5.1**). What was noticed was the intuitiveness of its reconstruction from its corresponding projections. With as few as two projections the human mind can quite easily determine that there is a single area of density in the center of the region and zero everywhere else. To reconstruct the image using the filtered backprojection much more information is required, and it still cannot truly represent the point function. Granted most regions are more complex than the point function, but the 'point' remains clear; the example of the point function begs for a reconstruction algorithm that can truly optimize the reconstructed images given a set of projections.

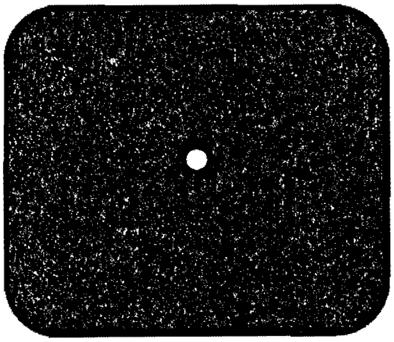


Figure 1.5.1: Two-dimensional point function

1.6. Hypothesis and Specific Aims

The goal of this thesis was to develop an image reconstruction algorithm that did not rely on traditional approaches to do the reconstruction. Specifically, the aim was to avoid Fourier transforms with all its inherent limitations. In the course of this research two unique algorithms were derived (and a third that was an extension to one of these novel algorithms) that could theoretically optimize the quality of the reconstructed image. The hypothesis was that developing and practically implementing these conceptually new methods of image reconstruction could optimize the images obtained from a CT scan. It was proposed that these methods could minimize image artifacts for situations of sharp density gradients, random noise, and limited angle views. To test this hypothesis, the algorithms were coded in a simulated CT environment to adapt to realistic situations such as, added noise in the detectors, and other non-idealities that are discussed in the methods section. The results of these algorithms were compared to currently accepted algorithms in order to validate the hypothesis. The specific objectives of this project were to:

a. Create a realistic simulated CT environment in Matlab in order to test the novel algorithms.

b. Fully develop each image algorithm in Matlab to optimize computational efficiency and to adapt to realistic concerns. Because these algorithms were not based on any previous reconstruction algorithms, and because of the intrinsic nature of these new reconstruction techniques, systematic errors were a concern.

c. Determine the feasibility of combining the novel algorithms to reconstruct an image. While both algorithms may be implemented independently, theoretical evidence had shown the possibility of decreasing computational effort by adapting the two algorithms into one process.

d. Determine if these newly developed algorithms could be adapted into clinical use, based on factors such as image quality and computational effort. These factors were judged relative to the currently accepted filtered backprojection algorithm. It was therefore necessary to include the filtered backprojection algorithm in the simulations.

CHAPTER II
THEORY

2.1 The Parallel Between External Beam Radiation Therapy and CT Imaging

To develop a novel approach to image reconstruction, optimization solutions to problems that contained parallels to image reconstruction were investigated. Specifically, there exists an interesting parallel between image reconstruction and the challenge of beam optimization for intensity modulated external beam radiotherapy. It may be helpful here to gain a conceptual understanding of this parallel, as the first novel reconstruction approach developed here was based on a beam optimization algorithm called the Fast Inverse Dose Optimization (FIDO) [24].

In external beam radiation therapy, a patient is prescribed a dose of radiation to a tumor volume. The oncologist also frequently prescribes a maximum allowed dose to other structures called organs at risk (these include the spine, rectum, and optic nerve just to name a few possible examples). Furthermore, radiation to all structures outside the defined targeted region should be kept to a minimum in accordance with the ALARA principle (as low as reasonably achievable). Of course it is impossible to deposit dose exclusively to the tumor region because external beams must traverse other tissues and bodily structures. In order to meet the oncologist's prescriptions, a treatment plan must be developed defining the trajectories and intensities of multiple beams incident on the patient. These beams will each interact with the patient and deposit their dose according to the physics of the interaction of radiation with matter. The dose deposition of each of these beams combine and overlap to create a high dose region at the defined tumor volume while maintaining the prescribed standards in the other regions. Finding the optimum angles and intensities of these beams constitutes a complex optimization problem that has been an area of much research. But what does this have to do with image reconstruction? The answer provides a beautiful insight into both these areas.

What the Oncologist prescribes can be thought of as an "image". The prescription consists of areas of relative high and low dose regions, which can be visualized as contrast of an image. The dose in beam optimization is the parallel to the unknown μ values in image reconstruction. In beam optimization one begins with the 'dose image' and what is sought after is the beams that best define that 'dose image'. Image reconstruction begins with the beam information (the angles and the input and output intensities) and seeks the image that best fits that information. So it becomes clear that image reconstruction is just the inverse problem with respect to beam optimization. Using these parallels, and the innovative FIDO algorithm, development began on a new image reconstruction algorithm called AIR (Algebraic Image Reconstruction).

2.2 *Algebraic Image Reconstruction (AIR) Theory*

In order to derive the equations of AIR it is necessary to define some terms. These terms will be used throughout this paper both for AIR and the other novel algorithms. **Figure 2.2.1** shows a hypothetical 2-dimensional domain being imaged. This region consists of a grid of unknown pixels each with its own absorption coefficient. The x-ray and detector array rotate around this region resulting in a projection q_m, for beam m, where m is the index of the beam angle (ranging from 1 to the total number of angles that are being used). q_m is further divided by the individual detectors in the array. It is helpful to think of each beam as being subdivided into smaller beams, called rays or *beamlets*. Let λ be the index of a beamlet in a beam. Let $q_{m,\lambda}$ be defined as the projection value due to beamlet λ in beam m. It is also necessary to define the weighting term $h_\ell^{m,\lambda}$, where ℓ is the index of the image pixel. $h_\ell^{m,\lambda}$ represents the fractional area (or volume in 3-D) of pixel ℓ traversed by beamlet λ of beam m. In a simplistic implementation of this algorithm $h_\ell^{m,\lambda}$ could be given a value of 1 if the beamlet has passed through the center of pixel ℓ, and zeros otherwise.

μ_ℓ^*	Actual absorption density at pixel index ℓ
μ_ℓ	Estimated absorption density at pixel index ℓ
m	Index of beam angle
λ	Index of beamlet in beam m
$q_{m\lambda}$	Measured projection of beamlet λ in beam m
$p_{m\lambda}$	Estimated projection of beamlet λ in beam m
$h_\ell^{m\lambda}$	Weighting of pixel index ℓ contributing to the projection of beamlet λ in beam m

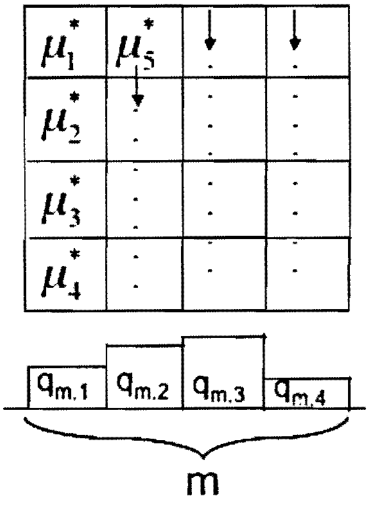


Figure 2.2.1: Defining the terms necessary to derive AIR

The derivation of AIR begins by defining μ_ℓ as an estimate to the true value of pixel index ℓ. The goal is to find an optimal set of μ_ℓ that most closely resembles the true values μ^*_ℓ. Given an estimated solution consisting of a set of μ_ℓ, one can define its corresponding set of projection values as:

$$p_{m,\lambda} = \sum_\ell \mu_\ell h_\ell^{m,\lambda} \tag{2.2.1}$$

Where $p_{m,\lambda}$ is the theoretical projection value from beamlet λ of beam m.

In order to find the set of μ_ℓ which most closely approximates the true values μ^*_ℓ, we require the difference between $p_{m,\lambda}$, the estimated projections, and $q_{m,\lambda}$, the measured projections, to be a minimum. AIR is at its essence an optimization algorithm that seeks to minimize this difference between $p_{m,\lambda}$ and $q_{m,\lambda}$. An objective function, can be defined, which is minimized to zero under such conditions.

$$\sigma_m^{CT} = \sum_{\lambda \in m} \left[p_{m,\lambda} - q_{m,\lambda} \right]^2 \tag{2.2.2}$$

Where σ_m^{CT} is the objective function for one beam angle.

The objective function is summed over all beamlets in a beam, and the term inside the summation is squared to prevent the possible solution of negative and positive values canceling out.

Substituting in the definition of $p_{m,\lambda}$ from (2.2.1) into (2.2.2) gives:

$$\sigma_m^{CT} = \sum_{\lambda \in m} \left[\sum_\ell \mu_\ell h_\ell^{m,\lambda} - q_{m,\lambda} \right]^2 \tag{2.2.3}$$

Now the expression contains the unknowns, μ_ℓ. Summing over all beam angles to obtain an objective function, which considers all beamlets in all beams:

(2.2.4) $$\sigma^{CT} = \sum_{m}\sum_{\lambda\in m}\left[\sum_{\ell}\mu_{\ell}h_{\ell}^{m,\lambda} - q_{m,\lambda}\right]^{2}$$

This is called the CT term objective function. The set of values μ_k that minimize σ^{CT} represents the optimal estimate of the true imaged region. The solution is found by setting the derivative of σ^{CT} with respect to μ_k equal to zero:

(2.2.5) $$\frac{\partial\sigma^{CT}}{\partial\mu_k} = 0$$

Substituting (2.2.4) into (2.2.5) gives:

(2.2.6) $$\sum_{m}\sum_{\lambda\in m}\frac{\partial}{\partial\mu_k}\left[\sum_{\ell}\mu_{\ell}h_{\ell}^{m,\lambda} - q_{m,\lambda}\right]^{2} = 0$$

Where the derivative operator does not depend on *m* or λ. Applying the operator through the brackets gives:

(2.2.7) $$2\sum_{m}\sum_{\lambda\in m}\left[\sum_{\ell}\mu_{\ell}h_{\ell}^{m,\lambda} - q_{m,\lambda}\right]h_{k}^{m,\lambda} = 0$$

Where $h_k^{m,\lambda}$ comes out as a result of the operator choosing the index *k* from all ℓ. Further simplifying:

(2.2.8) $$2\sum_{m}\sum_{\lambda\in m}\sum_{\ell}\mu_{\ell}h_{k}^{m,\lambda}h_{\ell}^{m,\lambda} - 2\sum_{m}\sum_{\lambda\in m}q_{m,\lambda}h_{k}^{m,\lambda} = 0$$

Because $\sum_{\ell}\mu_{\ell}$ does not depend on *m* or λ it can be brought to the front of the first term in (2.2.8):

(2.2.9) $$2\sum_{\ell}\mu_{\ell}\sum_{m}\sum_{\lambda\in m}h_{k}^{m,\lambda}h_{\ell}^{m,\lambda} - 2\sum_{m}\sum_{\lambda\in m}q_{m,\lambda}h_{k}^{m,\lambda} = 0$$

Equation (9) can be written more compactly as:

(2.2.10) i) $\sum_{\ell}\alpha_{k\ell}^{CT}\mu_{\ell} = \beta_{k}^{CT}$ where, ii) $\alpha_{k\ell}^{CT} = 2\sum_{m}\sum_{\lambda\in m}h_{k}^{m,\lambda}h_{\ell}^{m,\lambda}$ and iii) $\beta_{k}^{CT} = 2\sum_{m}\sum_{\lambda\in m}h_{k}^{m,\lambda}q_{m,\lambda}$

Equation (2.2.10) is an algebraic system where α^{CT} is a matrix containing an amount of rows and columns each equal to the number of unknowns, and β^{CT} is a column vector of length equal to the number of unknowns. The optimum solution set of μ_k can be found by inverting α^{CT}:

(2.2.11)
$$\mu_k = \sum_n \left(\alpha_{kn}^{CT}\right)^{-1} \beta_n^{CT}$$

AIR results in a matrix equation that is fundamentally different from ARA (ARA was described in section 1.3). Let us examine the physical meaning of $\alpha_{k\ell}^{CT}$ and β_k^{CT} to gain a more intuitive understanding of AIR.

In ARA the columns of **A** represent the pixel index, and each row identifies the pixels that contributed to a given projection from a given beamlet. **B** is a vector of projection values corresponding to each detector/beamlet. AIR is not formulated this way. Examining the expression for $\alpha_{k\ell}^{CT}$ (2.2.10ii), and assuming h values of either zero or one, the value assigned to the *kth* row and ℓth columns represents the number of beamlets that have passed through both pixels μ_k and μ_ℓ in the imaged region. The values along the diagonal α_{kk}^{CT}, simply counts the number of beamlets that have passed through pixel *k*. The matrix α^{CT} is completely independent of the projection values. This means that for a given beam configuration, α^{CT} and its inverse need only be initialized once, allowing for fast reconstruction. Furthermore α^{CT} is symmetric allowing for efficient memory storage. Examining the expression for β_k^{CT} (2.2.10iii) the k^{th} entry of β^{CT} is the sum of all projections from beamlets that have passed through μ_k.

In order to actualize AIR, two pieces of information are necessary from the CT scan. The first is the projection values from every detector, and the other is the set of pixels (or fractions of pixels) in the reconstruction grid that each beamlet has passed through, $h^{m,\lambda}$, which can be determined geometrically.

Solving equation (2.2.11) results in a global minimum only if the system is determined (i.e. if there is an adequate number of beam angles). However, some reconstructions may be underdetermined due to a lack of projection values, or may be undetermined due to detector noise or the stochastic nature of radiation. Considering these factors, solving equation (2.2.11) may yield non-physical or negative absorption densities. It is for this reason that another term, called the *weighted average* term was developed and added to the objective function, which effectively imposes a physical constraint in order to break possible degeneracy. Conceptually, this constraint assumes a homogenous or constant absorption density along the path of any beamlet when h is zero or one. Mathematically, this can be expressed as:

(2.2.12)
$$\sigma_{m,\lambda}^{av} = \sum_{\ell}\left[\mu_\ell h_\ell^{m,\lambda} - \tfrac{q_{m,\lambda}}{N_{m,\lambda}}\right]^2$$

Where $N_{m,\lambda}$ is the sum of $h_\ell^{m,\lambda}$ values (or geometric area) that beamlet λ of beam m has traversed. In the simple case that $h_\ell^{m,\lambda}$ is assigned values of 0 and 1, $N_{m,\lambda}$ can be defined as the number of pixels beamlet λ of beam m has passed through. $\sigma_{m,\lambda}^{av}$ is the *average term* objective function for beamlet λ of beam m.

Upon minimization to zero, equation (2.2.12) forces all pixels along the path of a beamlet to be the same value. Summing over all beamlets in a beam, and over all beams equation (2.2.12) becomes:

(2.2.13)
$$\sigma^{av} = \sum_{m}\sum_{\lambda\in m}\sum_{\ell}\left[\mu_\ell h_\ell^{m,\lambda} - \tfrac{q_{m,\lambda}}{N_{m,\lambda}}\right]^2$$

Upon minimization to zero, equation (2.2.13) will result in each μ_ℓ being an average of every projection that it has contributed to. Equation (2.2.13) is essentially backprojection formalized as a matrix equation, smearing back each projection along the respective beamlet's path, and averaging all the contributions to form an image. It should be noted that when h is given fractional values this definition does not hold, but its degeneracy breaking ability remains.

It may not be intuitive to assume that μ_ℓ should be the average of all projections that it has contributed to; rather, it may be more logical to attribute a larger weight in this averaging to the beamlets that passed through fewer pixels. For example, if a beamlet has passed through only one pixel, then the projection value would be equal to the μ_ℓ that that beamlet has traversed. Therefore, a weighting factor was incorporated into equation (2.2.13), which is a function of the number of pixels that a given beamlet has traversed (a beamlet passing through fewer pixels, is assigned a higher weight):

(2.2.14i) $$\sigma^{wav} = \sum_{m}\sum_{\lambda\in m} w_{m,\lambda}\sum_{\ell}\left[\mu_\ell h_\ell^{m,\lambda} - \tfrac{q_{m,\lambda}}{N_{m,\lambda}}\right]^2$$

Where *wav* stands for weighted averaging. $w_{m,\lambda}$ is the normalized weighting factor for beamlet λ belonging to beam m

(2.2.14ii) $w_{m,\lambda} = \dfrac{\tilde{w}_{m,\lambda}}{\sum_{m}\sum_{\lambda\in m}\tilde{w}_{m,\lambda}}$ a possible expression for $\tilde{w}_{m,\lambda}$: (14iii) $\tilde{w}_{m,\lambda} = \dfrac{1}{(N_{m,\lambda} - b)^a}$

Where a can be any positive integer whose value determines the magnitude of the weight. In the single pixel case, for a=2, the weight $w_{m,\lambda}$ will be 100 times larger than the weight for a beamlet containing 2 pixels, while the beamlet containing two pixels will be four times larger than the weight for a beamlet with three pixels (During preliminary testing, the weighting factor displayed limited effectiveness and thus a was set to zero for the main tests). b can be varied from zero to 1 but should never be set larger than the minimum value of $N_{m,\lambda}$.

Setting $\dfrac{\partial\sigma^{wav}}{\partial\mu_k} = 0$ to find the optimum set of μ_k satisfying this minimum gives:

(2.2.15i) $$\frac{\partial\sigma^{wav}}{\partial\mu_k} = 2\sum_m \sum_{\lambda\in m} w_{m,\lambda} \sum_\ell \left[\mu_\ell h_\ell^{m,\lambda} - \frac{q_{m,\lambda}}{N_{m,\lambda}}\right] h_k^{m,\lambda} \delta_{\ell k} = 0$$

Defining (2.2.15ii) $\beta_k^{wav} = 2\sum_m \sum_{\lambda\in m} w_{m,\lambda} h_k^{m,\lambda} \frac{q_{m,\lambda}}{N_{m,\lambda}}$ and (2.2.15iii) $\alpha_{k\ell}^{wav} = 2\sum_m \sum_{\lambda\in m} w_{m,\lambda} h_k^{m,\lambda} h_\ell^{m,\lambda} \delta_{lk}$

Using these definitions, equation (2.2.15i) becomes a linear system much like (2.2.11). Because both (2.2.15) and (2.2.11) are linear systems, they can be added to obtain another linear system, which represents the complete AIR objective function, that is:

(2.2.16) $$\sigma = c^{CT}\sigma^{CT} + c^{wav}\sigma^{wav}$$

Where c^{CT} and c^{wav} are the respective importance coefficients of the *CT* object term and weighted averaging objective term.

Equation (2.2.16) shows that the objective functions are not simply added, but have attributed to them respective importance coefficients. This is done because the desired solution results from σ^{CT}, while σ^{wav} is needed only to break the degeneracy in an under determined system. For this reason c^{CT} can be set to one and c^{wav} can be made relatively small. While the precise values of the importance coefficients are not crucial, they need to be determined empirically from the simulations.

Upon optimization, equation (2.2.16) becomes:

(2.2.17) i) $\sum_\ell \alpha_{k\ell}\mu_\ell = \beta_k$ where ii) $\alpha_{k\ell} = c^{CT}\alpha_{k\ell}^{CT} + c^{wav}\alpha_{k\ell}^{wav}$ and iii) $\beta_k = c^{CT}\beta_k^{CT} + c^{wav}\beta_k^{wav}$

Other terms were also developed in order to break degeneracy and avoid negative densities. The relevance of each is examined in the results section of this thesis. One significant term called the *zero averaging* term is formulated exactly the same as the weighted averaging term except for one difference. The zero averaging term is applied only for projections that are equal to zero (in reality this would mean below a set threshold in order to account for noise and attenuation through air), and its importance coefficient should be made relatively high. The result is to force all pixels that contributed to a zero projection to be equal to zero.

Other terms that were developed are the *global* and *local averaging* or *smoothing* terms. Again their function is to break degeneracy and avoid negative densities. The global smoothing term was designed to smooth out large differences in density over the whole image slice. Conceptually it seeks to minimize the difference between all pixels. If given a large enough weight within the objectivity function, the result will be a constant density over the whole slice, so the intention is to assign only a small importance to this term.

(2.2.18) $$\sigma^{global} = \sum_i \sum_j (\mu_i - \mu_j)^2$$

The optimum value can be found by:

(2.2.19) $$0 = \frac{\partial \sigma^{global}}{\partial \mu_k} = 2\sum_i \sum_j (\mu_i - \mu_j)(\delta_{ik} - \delta_{jk})$$

By calling N_{cells} the number of pixels in the image slice we obtain the α and β array:

(2.2.20) $$\text{i) } \alpha_{kl}^{global} = 4(\delta_{kl} - \tfrac{1}{N_{cells}}) \quad \text{ii) } \beta_k^{global} = 0$$

The local term is very similar in concept to the global term, however the local term seeks to minimize the difference between nearest neighbor pixels. The total objective function can theoretically be a sum of all these terms described above, each with its own importance coefficient:

(2.2.21) $$\text{i) } \sum_\ell \alpha_{k\ell} \mu_\ell = \beta_k$$

$$\text{ii) } \alpha_{k\ell} = c^{CT}\alpha_{k\ell}^{CT} + c^{wav}\alpha_{k\ell}^{wav} + c^{zeroav}\alpha_{k\ell}^{zeroav} + c^{global}\alpha_{k\ell}^{global} + c^{local}\alpha_{k\ell}^{local}$$

$$\text{iii) } \beta_k = c^{CT}\beta_k^{CT} + c^{wav}\beta_k^{wav} + c^{zeroav}\beta_{k\ell}^{zeroav} + c^{global}\beta_{k\ell}^{global} + c^{local}\beta_{k\ell}^{local}$$

By making use of the various AIR terms it is hypothesized that AIR can solve the optimal image reconstruction while avoiding negative attenuation coefficients in the image.

2.3 Parsed Algebraic Image Reconstruction (PAIR) Theory

While AIR presents a sound theoretical solution to image reconstruction, it is subject to similar size constraints associated with ARA. This constraint confines algebraic reconstruction algorithms to relatively low image resolution. The premise behind PAIR (the Parsed Algebraic Image Reconstruction) in reconstructing an image to a desired resolution is to instead solve, using the theory outlined in AIR, multiple but independent lower resolution images that still cover the entire domain. Once the images have been acquired one can use these independent solutions to gain added spatial resolution. Higher resolution can be achieved through a number of lower resolution images. Let us explore how this can be achieved as we examine the theory behind PAIR.

Suppose one wishes to reconstruct an image with a resolution of m x n cells. As explained above, AIR would require an mn x mn sized alpha matrix. One can instead reconstruct a lower resolution image using AIR by assuming, for example, that every group of four adjacent absorption densities is actually one homogeneous value of μ. This grouping can be achieved in a number of different configurations as shown in **figure 2.3.1.**

Using the solutions (as obtained using AIR) from a number of independently oriented lower resolution grids one can obtain the resolution of the desired grid. This is the meaning of the name **Parsed** Algebraic Image Reconstruction.

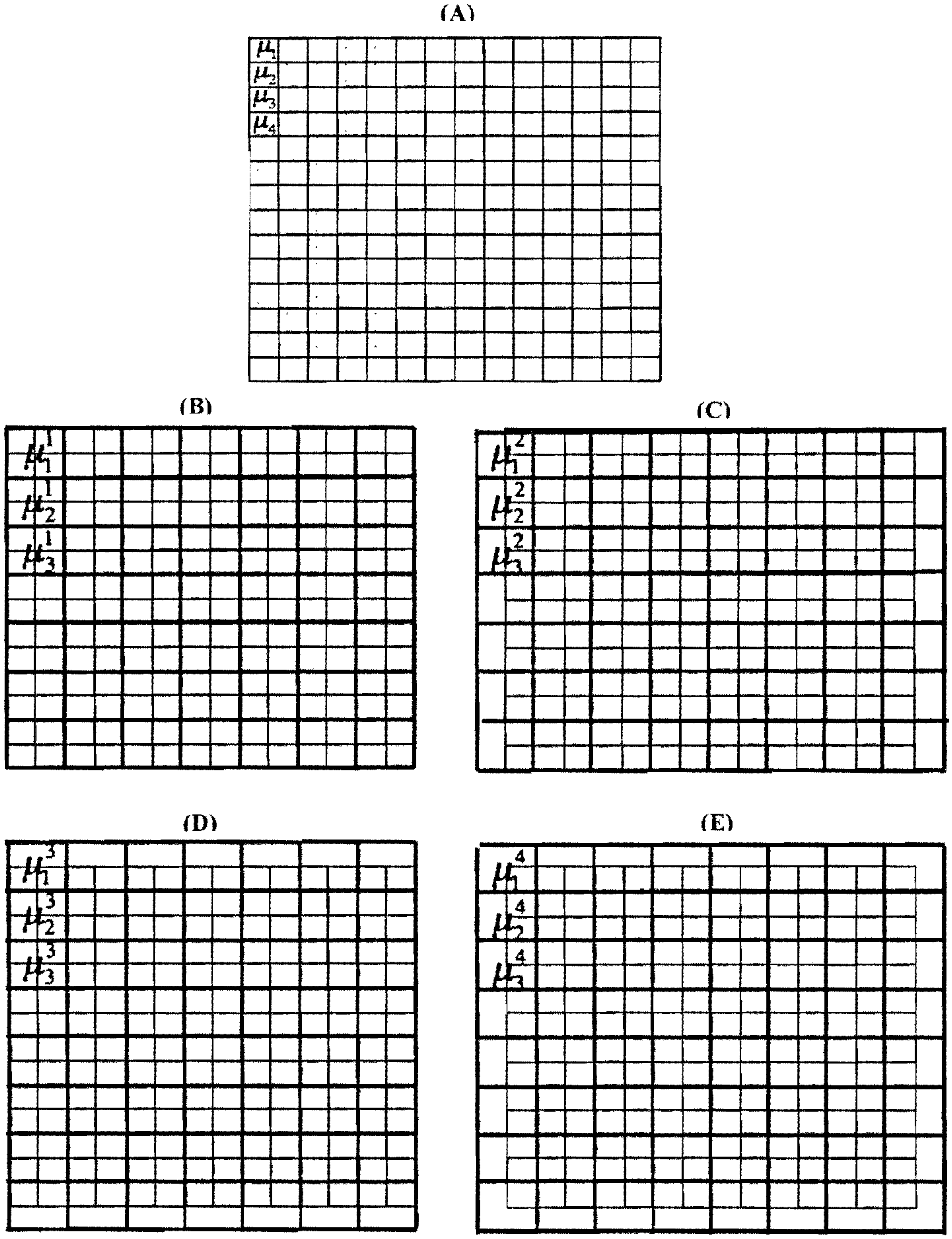


Figure 2.3.1: **(A)** Shows a grid of 14x14 unknowns μ_ℓ, which is an example of a desired resolution for image reconstruction. Figures **(B)**, **(C)**, **(D)**, and **(E)** show four independent configurations that can be used to group the unknowns in (A) in to a lower resolution grid. Here μ^n connotes the set of solved absorption densities for the *n*th configuration.

Obtaining the higher resolution from the lower resolution grids is straightforward. Referring to **Figure 2.3.1** for definitions of absorption densities belonging to specific grids, the method of obtaining the desired absorption densities μ_ℓ, can be accomplished in steps as follows (the superscript index denotes a reconstruction orientation and the subscript denotes a pixel in that grid):

(2.3.1) i) $\mu_1 = \mu_1^4$

ii) $\mu_2 = \mu_1^2 - \mu_1$

iii) $\mu_3 = \mu_2^4 - \mu_2$

$$\vdots$$

Where equation (2.3.1) utilizes the independence of the four grids to recover the underlying higher resolution image.

From (2.3.1) it is clear that the solution begins at the corner of grid **(E)** (**figure 2.3.1**) whose corner values are equal to the corner absorption densities of the desired grid. The solution then progresses inwards, using the solved pixels to determine the next adjacent pixel. It is expected that, due to noise, there will not be one unique solution to the underlying grid's absorption densities. In fact, beginning at different corners of the problem, may give different solutions. To minimize propagation of errors, and to represent the higher resolution grid more accurately, it is expected that by solving equation (2.3.1) four times (one solution from each corner) and then averaging, may yield a desirable solution.

An alternative approach to obtain the final image while avoiding the propagation of noise is to simply map the parsed solutions onto the higher resolution grid and then average those solutions. Error propagation is avoided because each solved pixel does not depend on the others.

The downside is that this method cannot truly resolve the desired resolution, and may exhibit unwanted smoothing of the image.

The theory of PAIR can be extended to any user defined PAIR pixel sizes. Suppose a very fine reconstruction resolution is desired; it may be necessary to employ more than four parsed solutions. One could then solve nine independent PAIR images to achieve the desired resolution. Each PAIR reconstruction pixel will be the size of nine AIR reconstruction pixels, and the solution could be determined using a similar process to that in equation (2.3.1). Because PAIR solves multiple independent solutions, its implementation is in theory easily parallelizable. Increasing the number of AIR pixels per PAIR pixels would decrease computational concerns, but it must be determined if this process can be practically applied for any number of parsed solutions.

Geometric Image Reconstruction Algorithm (GIRA) Theory

The motivation behind GIRA (Geometric Image Reconstruction Algorithm) was to develop a reconstruction method that does not rely on constructing a matrix of equations, or Fourier transforms, but instead solves the reconstruction one unknown at a time. In this respect it would only be necessary to store a matrix of unknowns μ_ℓ, and fill in individual pixels as the solution progresses. The idea behind GIRA was to exploit the beamlets that pass through only a small geometric area in the imaged region. The projection values of these 'special' case detectors give definitive information about that small region that has been traversed. Why not consider that area to be solved (its μ_ℓ value equal to the projection value scaled by the size of the region) and remove it from the set of unknowns? By repeating this process, GIRA could shave away the problem set and possibly solve the entire image reconstruction. Each time another small region is solved and removed from the set of unknowns, it may open up more possible solutions. A corollary of GIRA is that the reconstruction resolution is no longer user defined, nor is it made up of equally size square pixels, but it is an outcome of the beam geometries and each solved region will in general be an irregular shape. The resolution can then be mapped onto pixels or voxels for display. The hypothesis is that the resolution inherent in GIRA will be higher than other reconstruction techniques. There is also no theoretical need for averaging terms or for terms that avoid negative densities because the solutions are solved individually. GIRA has the added advantage of being able to begin reconstruction with the first set of projections, unlike matrix solutions, which depend on all the projection information to reconstruct the image. This suggests speedy reconstruction times.

The theory of GIRA has thus been described in brief. A more detailed description will be given in the methods section when examining the practical implementation of GIRA.

CHAPTER III
METHODS

3.1 Simulating CT Projections in Matlab

In developing each algorithm, there was much feedback from practical testing. A theoretical concept was derived, it was then implemented in Matlab, and from those results the theories and concepts were revised. This section examines the practical implementation of the algorithms in Matlab, referring to progressive generations of the CT simulation to gain perspective of the developing process and the various simulation models used for testing. The progressive generations represent added complexity in the CT simulation. As is the case in much scientific research, theories were initially tested in over simplified scenarios before graduating to less ideal, or more complex testing environments.

The first step was to create a model of the CT scanning process. This was necessary in order to test the algorithms in controlled and simplified scenarios and allowed for the controlling of parameters such as beam angles, detector size, noise, and to define the subject being image. In creating the simulation it was first necessary to define a matrix of values, which would represent the densities of the subject being imaged. Matlab has a built in function called *phantom* that produces the Shepp and Logan head phantom of any user defined resolution. The Shepp and Logan phantom is a commonly used phantom in CT image testing and quality assurance. The center of each pixel of the phantom matrix was then mapped onto a Cartesian coordinate system so that each pixel in the matrix occupied a unit position on the two-dimensional grid.

To imitate the CT scanning process, a method was devised to effectively sum the pixel values of the phantom matrix along different paths creating the individual projections. Given a beam's angle and the size of each detector (initially set to a unit length) in the array, the path of each beamlet can be determined with the respect to the Cartesian coordinate system. At any given angle the detector array can be thought of as being positioned at an arbitrary distance from the phantom grid, with the center of the detector array on axis with the center of the phantom grid. The cross-sectional length of the phantom changes depending on the beam angle, so the detector array was initialized to fit the maximum length (occurring at 45 degrees relative to the phantom grid). Because of this, there are instances when some beamlets do not traverse any of the pixels, and the projection value is trivially zero.

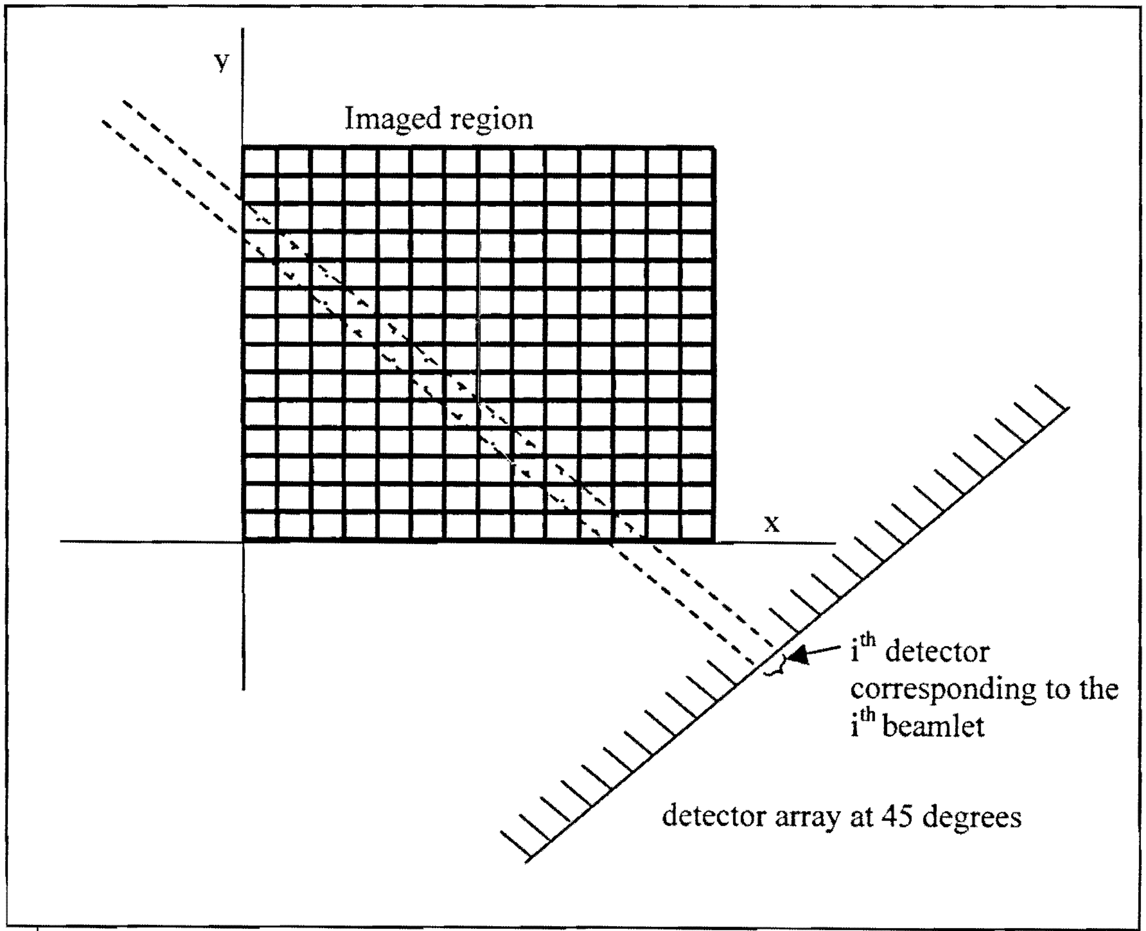

y
Imaged region
x
ith detector corresponding to the ith beamlet
detector array at 45 degrees

Figure 3.1.1: The projection value measured by detector *i* is simulated in Matlab by determining which pixel's centers lie within the path of that beamlet. Knowing the start and end of each detector, a for loop checks if the coordinate of a pixel is within those boundaries.

With the positions of the detector boundaries (the start and end of each detector) known within the Cartesian coordinate system, determining which pixels lie within a given beamlet path is straightforward **(figure 3.1.1)**. To accomplish this in Matlab, a *for* loop was coded, which checks each pixel in the phantom matrix to determine if its center lies within the path of a given beamlet. If the center is within the beamlets path, it is considered to be completely within that beamlet, and it is summed into the projection measurement corresponding to that detector.

The *for* loop checks all pixels for every beamlet and stores two crucial output parameters for each beamlet:

1 The projection value corresponding to that detector

2 The set of pixel indices that contributed to that projection.

Notice that referring to equation (2.2.1), values of *h* are either one if the pixel's center is contained in the beamlet path, or zero if it is not. There can be no partial pixel contributions. This aspect was subsequently modified to allow for partial contributions. The issue of partial contributions will be discussed in more detail shortly.

Due to the fact that in the simulation each detector in an array begins where the other ends, it may happen that a pixel contributes to two adjacent detector measurements (i.e. a beamlet boundary lies directly on the center of a pixel). In this case the program assigns it to only one of the two beamlets and its corresponding projection value maintaining consistency in the projections.

The issue of noise was approached in a simplistic way. While in reality CT noise can be broken down into many different aspects, each depending on many factors, for the sake of this research it was sufficient to incorporate a superficial noise function. The purpose was to test how the algorithms handle inconsistent projection information. Included in the simulation were two independent noise functions; one that simulates a random background noise in the detectors, and one that simulates noise due to radiation scatter. The background noise function adds a user defined background level noise to each projection value. Before it is added to each projection, the background level noise is multiplied by a random number (ranging from 0 to 1). Each projection is thus perturbed by a different amount ranging from 0 to the maximum user defined value. The second noise function, simulating radiation scatter is specified by the user as a percent of the projection value. The user-defined percent of each projection multiplied by a random number is added to each projection value. In this case, more noise will be added to higher projection values. It must be noted that these functions do not represent the reality of noise in CT imaging. They served only to test our algorithms under varying conditions. A more realistic treatment of noise would be required for advanced simulation models.

The process of obtaining CT projections can be described as performing Radon transforms on a function from various angles. A Radon transform in two dimensions, is the integral transform consisting of the integral of a function over straight lines. Matlab contains a function called *radon*, which can perform the CT projections but with some differences compared to the simulation developed here. It is worth describing this method and why it was not used for the testing environment.

Using Radon transforms, a projection at any angle consists of discrete points rather than finite sized detectors **(figure 3.1.2).** For this reason, an interpolation process is employed to ensure accurate projections. The *radon* function projects the coordinates of the pixel onto the detector line. It will fall somewhere on this line, but not in general directly on a point demarcated as a detector. The value of that pixel is then distributed to the two nearest detectors relative to the distance from the detectors. This is the process of Radon transforms with linear interpolation. The result is projections that can have partial or fractional contributions from pixels.

It was decided that it was necessary to code an alternative CT simulation to allow for total control over the simulation parameters, and the ability to test the novel algorithms in situations with varying degrees of complexity. Let us now examine the various simulation types that were developed in order to test the algorithms in progressively more complex and less ideal conditions. As the complexity increases, one can say that the simulation more closely represents a subset of the non-idealities of real CT imaging.

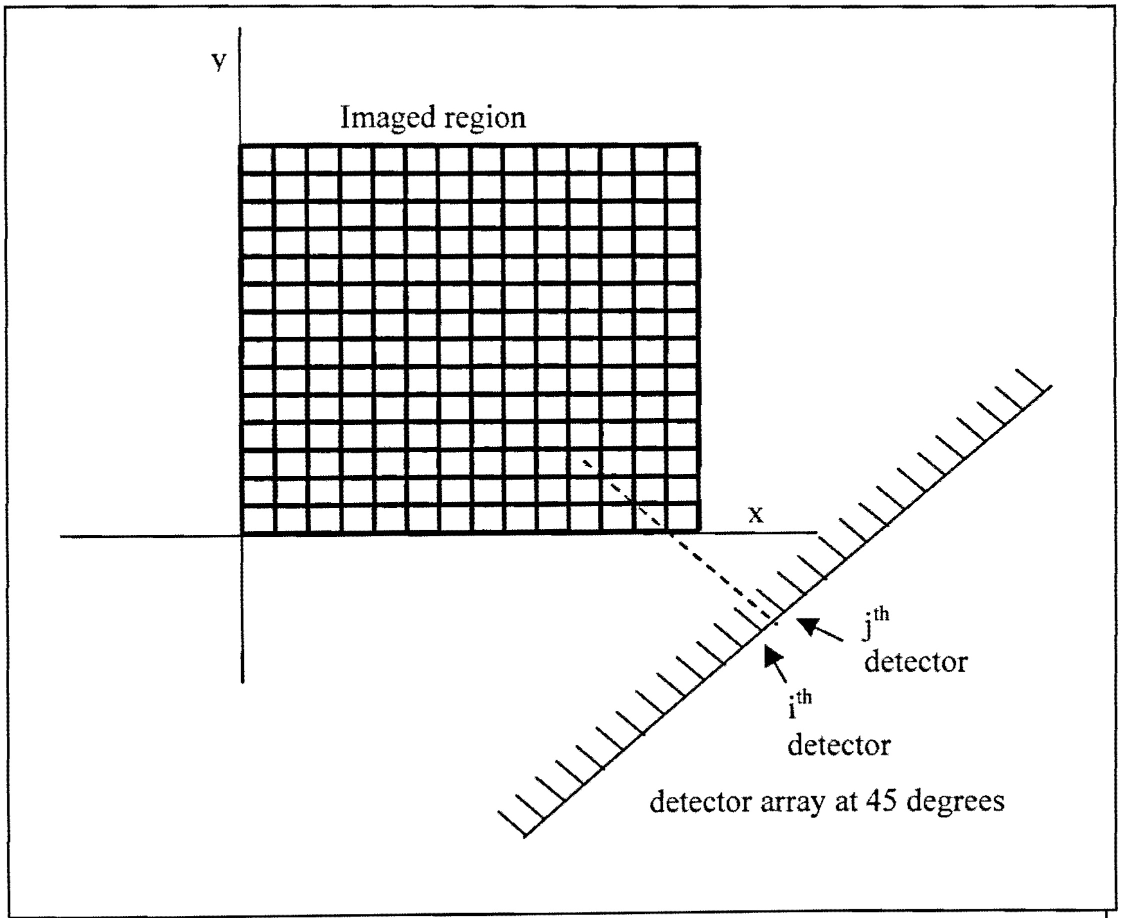


Figure 3.1.2: An example of the *Radon* transform algorithm in Matlab. For each beam angle, the coordinates of each pixel are projected on to the detector line. In contrast to the simulation developed in this work, the detector array consists of point detectors. It is therefore necessary to interpolate the projections. In this example, the projection lies directly between two detectors, and the value of the pixel will be distributed half to the ith detector and half to the jth detector.

3.2 Developing a Realistic CT Simulation Model

The initial simulation that was developed has certain simplifications that do not extend to more realistic circumstances in a trivial way. The main issue concerned representing the object being imaged by a finite grid. Specifically, two aspects of the simulation seemed to be unrepresentative of the reality of CT imaging. The first aspect is the lack of partial contributions of pixels to the projection values. The second simplification is that the reconstruction grid is set to have the same resolution as the phantom matrix, which is not possible in reality because the actual imaged region is not pixilated. This simplification implies that the phantom matrix is homogeneous within the defined reconstruction pixel size, which is not generally true. The assumption leads to a much more self-consistent system of equations. The simulation with the mentioned assumptions refers to the most ideal or simplified model that was used to test the algorithms.

In order to allow for partial pixel contributions, the values of the phantom matrix can be mapped onto a higher resolution grid before simulating the projections. The higher resolution grid is then used to simulate the projection measurements. This allows for partial contributions as can be seen in the example in **figure 3.2.1**. In this example a beamlet may traverse ¼, ½, ¾, or the full area of any given reconstruction pixel.

However, the reconstruction of the image is carried out using the original resolution (indicated in red in **3.2.1**). The original pixel size is defined as a squared unit in dimensions, and the width of each detector in the array is a unit in width. This method was expected to produce projections very similar to the Matlab radon transform function, as this process acts as an interpolation function. In fact, the Matlab function *radon* does this exact same process to improve the accuracy of its projections. When taking projections, it divides each pixel into four fractions. The difference is that in the simulation developed in this thesis, one can specify any number of small pixels per larger pixel (which we will refer to as fractions per phantom pixel). By increasing the number of fractions, one can approximate the true paths taken by the beamlets. The addition of partial pixel contributions constituted the next progressively complex model that was used in testing the algorithms.

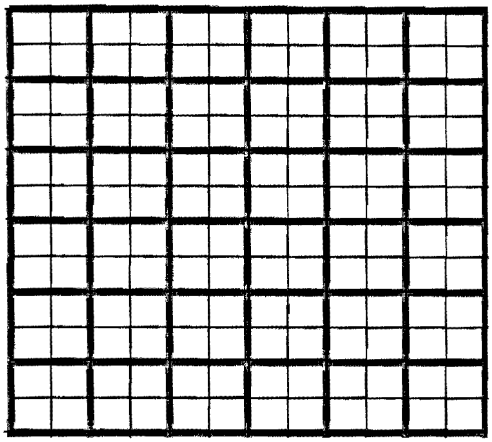


Figure 3.2.1: The grid in bold red is the resolution of the phantom being scanned. Each pixel in the red grid is a unit area and contains an attenuation coefficient that is homogeneous within the pixel. The black lines are an example of a remapping of the phantom matrix. The attenuation coefficients from the red grid are divided by the number of small black cells within one red cell. Note that each red pixel still has a homogeneous attenuation coefficient.

While this approach solved the first concern of partial contributions, it did not correct the second issue of the unrealistic homogeneity within the reconstruction pixels. In other words, the simulation model at this point approximates the case of an infinitely fine reconstruction resolution (so fine that the object being imaged is homogeneous within the predefined reconstruction pixels).

To gain more insight into exactly why this is not representative of reality, it must be understood that in real CT reconstruction there is no phantom grid. The definition of a pixel is required only for the reconstruction. The geometric coordinates of any given pixel in the reconstruction grid will likely encompass many inhomogeneities in the true medium. This means that while many beamlet paths along different angles may pass through the same reconstruction pixel in the course of a scan, they will each sample a unique geometry of that pixel and thus record inconsistent measurements.

The subsequent incarnation of the simulation addressed the reality of inhomogeneity within the regions of the phantom matrix demarcated as reconstruction pixels. This was achieved by defining the reconstruction resolution independent of the phantom resolution. Referring to **figure 3.2.2**, first a phantom resolution is chosen (indicated by red) whose pixels define a unit dimension and then, as described above, the phantom matrix is mapped onto a higher resolution grid in preparation for taking the projections. Each detector is a unit in width.

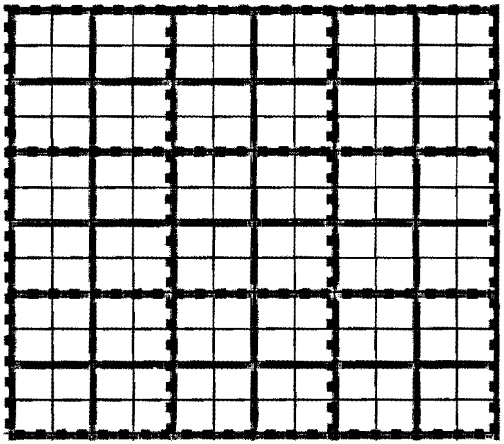


Figure 3.2.2: The grid in bold red is the resolution of the phantom being scanned. The black lines are an example of a remapping of the phantom matrix onto a finer resolution for the sake of taking projections. The dotted blue grid is the pixel resolution of the reconstructed image. This model has an additional level of complexity in that the reconstruction pixel may contain inhomogeneous regions. The relative resolutions of these grids can be changed and tested for convergence within the simulations.

After simulating the projection measurements, the program is left with the projection values for each detector, and the set of pixel indices that contributed to each projection. At this point, the set of pixel indices represent the highest resolution grid in **figure 3.2.2**. At that resolution there are no partial pixel contributions, so all h weights will be either zero or one. In order to prepare the information for reconstruction and to maintain a level of 'blindness' with respect to the phantom, the small pixel indices that were recorded are remapped to conform to the lower resolution reconstruction grid (blue). That is, all little pixels that are contained in one big blue reconstruction pixel are assigned the same pixel index. The *h* weights are updated to represent a fraction of the reconstruction pixel. In the example in **figure 3.2.2**, every little pixel is given a 1/16th fraction of the reconstruction pixels. There are four unique phantom pixels per reconstruction pixel and each phantom pixel is subdivided into an additional four fractions. From this example it is clear that the reconstruction pixels may contain inhomogeneities with regard to the phantom matrix. It is the job of the reconstruction algorithm to average those inhomogeneities into a single value to be represented in the reconstruction resolution. This simulation model marked another, less ideal test for the reconstruction algorithms.

It is important to realize that this paradigm of relating to varying resolutions was an after thought of the theoretical idea behind PAIR. The example just described is essentially the same process of obtaining one of the PAIR solutions necessary to reconstruct a finer resolution. However, in PAIR reconstruction, additional lower resolution grids are defined and from those solutions PAIR seeks to resolve the resolution of the blue grid.

The final incarnation of the simulation, allows for a change in the detector width relative to the phantom resolution. This added yet another degree of complexity, and allowed for more control over the simulation parameters.

3.3 An Alternative Method of Representing Pixels

The shift towards a more complex simulation model was expected to impact AIR and PAIR's ability to accurately reconstruct an image. In solving a low-resolution image (lower than the phantom resolution) AIR must assume that each reconstruction pixel is composed of a homogeneous density. This assumption is mathematically represented in the theory of AIR in equation (2.2.1). Recall that equation (2.2.1) states that a projection is defined retrospectively, as the sum of the reconstruction pixels' fractions that it has passed through. If a beamlet has passed through a partial reconstruction pixel, that pixel's contribution is multiplied by a weighting factor equal to the fractional area of the pixel that was contained in the beamlet's path. Therein lies the problem. The weighting factor *h,* is a geometric weight while in reality it should represent the fraction of density contained in the pixel. These two definitions are equivalent only when the region being imaged is homogeneous within each reconstruction pixel.

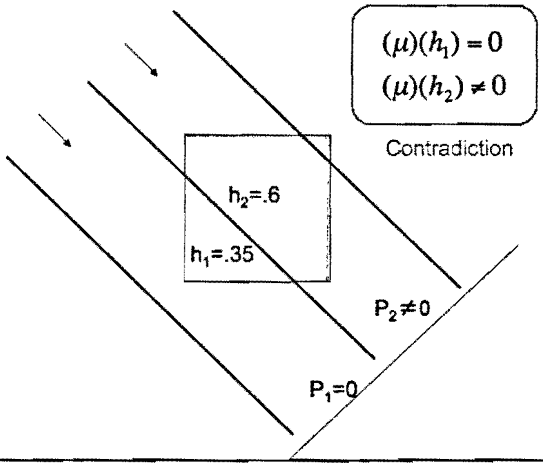


Figure 3.3.1: if a single reconstruction pixel containing inhomogeneous density (blue indicates some μ value and white is zero) is scanned by two beamlets, their respective projections will create conflicting equations as shown in the figure. These equations build the matrices of AIR and PAIR.

In the less ideal simulations there may be varying attenuation coefficients within a reconstruction pixel. This leads to inconsistencies in the matrix equation (**figure 3.3.1**). To overcome these inconsistencies an alternative formulation of equation (2.21) was developed. It is important to note that the CT simulation itself (including the calculation of projection values) does not change from the previously described model. This new development occurs at the level of retroactively representing the beamlets' paths in preparation for the matrix equations.

The idea behind this development stemmed from the desire to avoid the partial contribution of pixels in the matrix equations. In the original simulation the angled beams' paths were imposed on to the coordinate system of a grid that remained un-rotated. Suppose instead, defining a rotated coordinate system in which a grid of pixels is parallel to the incident beam. This grid does not represent the reconstruction grid, but is a theoretical construct that covers the same geometric region as the reconstruction grid (except for the corners which will lie outside the region of interest) **(figure 3.3.2).**

Each beam angle has its own respective grid in which no partial pixels are traversed, meaning that all *h* values are by definition equal to either zero or one. Here no assumption has been made

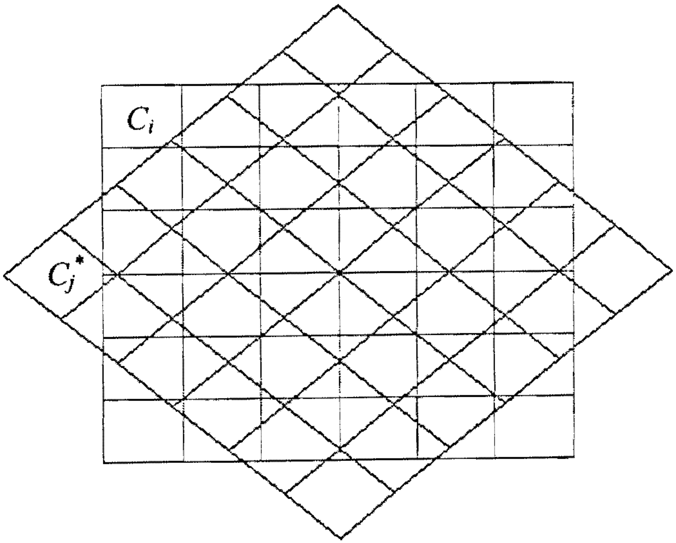


Figure 3.3.2: Supposing a beam angle at 45 degrees, the blue grid is a theoretical grid of unknown attenuation coefficients (index C_i^*) oriented parallel to the beam angle. Beamlets pass through the blue pixels with a weight (*h* value) of either zero or one. A similar grid can be designed for each beam angle. The reconstruction pixel grid in indicated in red with index C_i.

as to the homogeneity of the pixels. One can then map each angled grid back on to the reconstruction grid, relating their respective geometric overlap. For a given beam angle, each pixel's area in the reconstruction grid is covered by some combination of the geometric areas of the pixels in the rotated grid. This relationship can be expressed in a matrix equation as follows:

$$\begin{bmatrix} \omega_{1,1} & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ \vdots & \cdot & & & & \\ \vdots & & \cdot & & & \\ \vdots & & & \omega_{i,j} & & \\ \vdots & & & & \cdot & \\ \vdots & & & & & \cdot \end{bmatrix} \times \begin{bmatrix} c_1^* \\ c_2^* \\ c_3^* \\ c_j^* \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_i \\ \vdots \\ \vdots \end{bmatrix} \tag{3.3.1}$$

Where C_i is the geometric area of i^{th} pixel in the reconstruction grid, C_j^* is the geometric area of j^{th} pixel in the rotated grid, and $\omega_{i,j}$ is the fraction of C_j^* that geometrically overlaps C_i. Except for the rows and columns of the matrix corresponding to corner pixels, each row and each column should add up to one representing the fact that each pixel is overlapped entirely by pixels in the other grid (a discussion the issue of corner pixels will be presented shortly).

Upon solving for the set of geometric areas, C_j^* in terms of C_i equation (3.3.1) becomes:

$$\begin{bmatrix} c_1^* \\ c_2^* \\ c_3^* \\ c_j^* \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} \omega_{1,1} & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ \vdots & \cdot & & & & \\ \vdots & & \cdot & & & \\ \vdots & & & \omega_{i,j} & & \\ \vdots & & & & \cdot & \\ \vdots & & & & & \cdot \end{bmatrix}^{-1} \times \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_i \\ \vdots \\ \vdots \end{bmatrix} \tag{3.3.2}$$

Where the matrix of $\omega_{i,j}$ weights has been inverted.

Intuition may suggest that the inverse of the matrix should be nothing more than the transpose. The rows of the $\omega_{i,j}$ matrix relate the geometric area of C_i in terms of C^*, while the columns relate the geometric area of C_j^* in terms of C. This is most clearly recognizable in the case of a beam angle at 0 degrees. The $\omega_{i,j}$ matrix will be the identity matrix and its inverse would be equal to its transpose. Nevertheless, at angles other than 0 or 90 degrees, the matrix inversion results in a non-trivial relationship between the two grids.

After the inversion, the C_j^* areas are expressed in terms of fractions of C_i in an unintuitive way, which no longer represents the simple geometric overlap. Knowing that each beamlet traverses a set of C_j^* (rotated grid pixels) with no partial contributions, one can instead express that set in terms of C_i using the fractions obtained from the inversion.

From this method ones arrives at an altered equation (2.2.1):

$$p_{m,\lambda} = \sum_{\ell} \mu_{\ell} \tilde{h}_{\ell}^{m,\lambda} \tag{3.3.3}$$

Where $\tilde{h}_{\ell}^{m,\lambda}$ is the modified h value for the pixel ℓ contributing to beamlet λ in beam m.

There were a number of practical issues to overcome in the implementation of this method. One challenge was to accurately determine the $\omega_{i,j}$ values. The premise relies on the determination of the shared area of two squares in separate coordinate systems. The overlapping area was approximated in following way. Because each reconstruction pixel can consist of a number of smaller pixels determined by the choice of phantom resolution, it can be determined if the centers of those small phantom pixels lie inside a square defined by four corner coordinates using the Matlab function *inpolygon.* The small pixels approximate the fractional area with an accuracy corresponding to the simulated projections (**figure 3.3.3**). It should be noted this method of determining the shared areas is not the only possible approach.

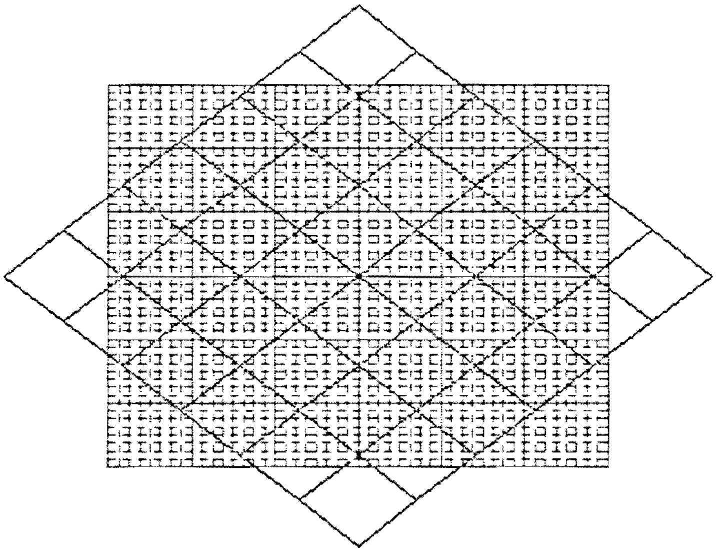


Figure 3.3.3: The overlapping areas are approximated by counting the number of small pixels (indicated by a dotted red line) in a blue pixel using the Matlab function *inpolygon*.

Due to the fact that the number of small pixels represents only an approximate area, it can happen that a given pixel in the rotated grid may seemingly contain more than a unit area. For this reason, this method of representing pixels was only implemented in cases where the number of phantom pixels per reconstruction pixel was made large enough to minimize this error.

The second issue was that in order to ensure that the matrix of weights had a stable inverse, a square matrix was desired. This requires the number of pixels of the rotated grid be equal to the number of reconstruction pixels. There are two problems associated with the requirement that the number pixels in both grids be equal. The first problem is that the imaged region will not be fully covered by the rotated grid at angles other than 0 and 90 degrees (see **figure 3.3.3**). The second difficulty is that the stability of inverting the matrix may depend on a one-to-one relationship between the two grids, and this relationship will not generally exist as the reconstruction grid and the rotated grid do not occupy the same domain. To overcome the first problem that the imaged region will not be covered, trivial rows and columns of pixels can be added to the problem set, keeping the imaged region in the center, ensuring that the corners that are not covered are not of crucial importance. The extra rows and columns can be discarded after the inversion, as they do not represent unknown densities.

However, the issue of corner cells still posed a problem in setting up and inverting the system of equations. There will always be rows and columns of the matrix in equation (3.3.1) that do not add up to one. To correct for this, the corners had to be 'cut', literally. Before explaining this procedure, note that these corner pixels do not represent any part of the imaged region, and are outside of the reconstruction domain. The overhanging corners can therefore be manipulated to conform to a neat set of equations. Looking at **figure 3.3.4**, and considering arguments of symmetry, it is clear that any exposed corner area of the reconstruction grid will have a corresponding corner in the rotated grid. In order to ensure that both grids occupy the same domain and thus maintain a one-to-one relationship, the unshared area of the corner pixels in the rotated grid can be assigned to the corresponding pixels of the reconstruction grid. Presented here is a simple 2x2 pixel example of this procedure for clarity:

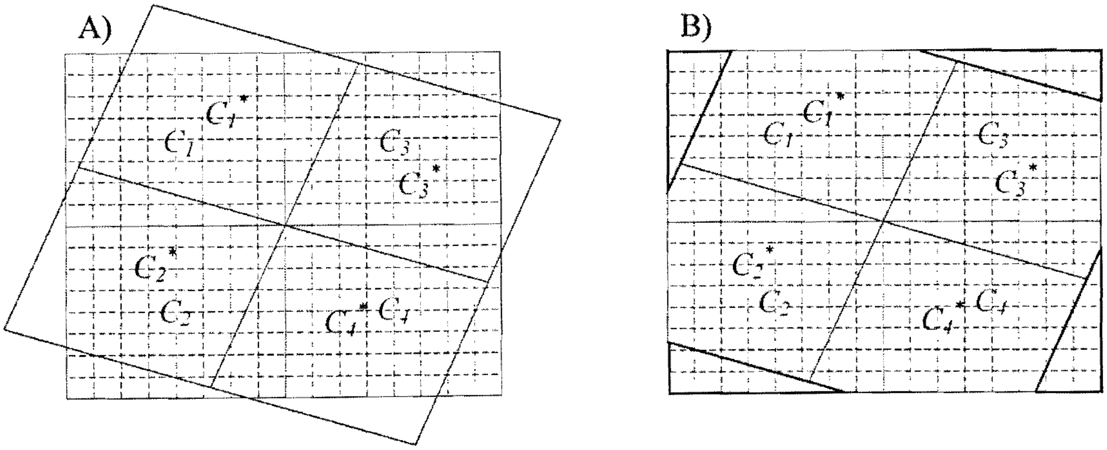


Figure 3.3.4: A) a beam angle at 20 degrees represented by a 2x2 pixel grid in a rotated coordinate system cannot cover the original grid as the corners are exposed. **B)** The corners of the rotated grid can be assigned to the exposed area of the original grid maintaining a one-to-one relationship between the two grids.

Referring to **figure 3.3.4,** the matrix equation relating the two grids in **A)** is:

(3.3.4)
$$\begin{bmatrix} 0.7031 & 0.1875 & 0 & 0 \\ 0 & 0.7031 & 0 & 0.1875 \\ 0.1875 & 0 & 0.7031 & 0 \\ 0 & 0 & 0.1875 & 0.7031 \end{bmatrix} \times \begin{bmatrix} c_1^* \\ c_2^* \\ c_3^* \\ c_4^* \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}$$

Notice that the rows and columns do not add up to one because the corners are not included in the domain. Assigning the corners of the rotated grid to the corresponding corners of the original grid as shown in **figure 3.3.4 B)**, gives:

(3.3.5)
$$\begin{bmatrix} 0.7031 & 0.1875 & 0.1094 & 0 \\ 0 & 0.7031 & 0 & 0.2969 \\ 0.2969 & 0 & 0.7031 & 0 \\ 0 & 0.1094 & 0.1875 & 0.7031 \end{bmatrix} \times \begin{bmatrix} c_1^* \\ c_2^* \\ c_3^* \\ c_4^* \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}$$

Now each row and column adds up to one representing the fact that the entire domain is shared by both grids. Although in this simple case both equations (3.3.4) and (3.3.5) are solvable, the later formulation is more stable especially in larger domains.

Due to all the above-mentioned issues, this method was only carried out and tested in this thesis under specific cases. To summarize, they were:

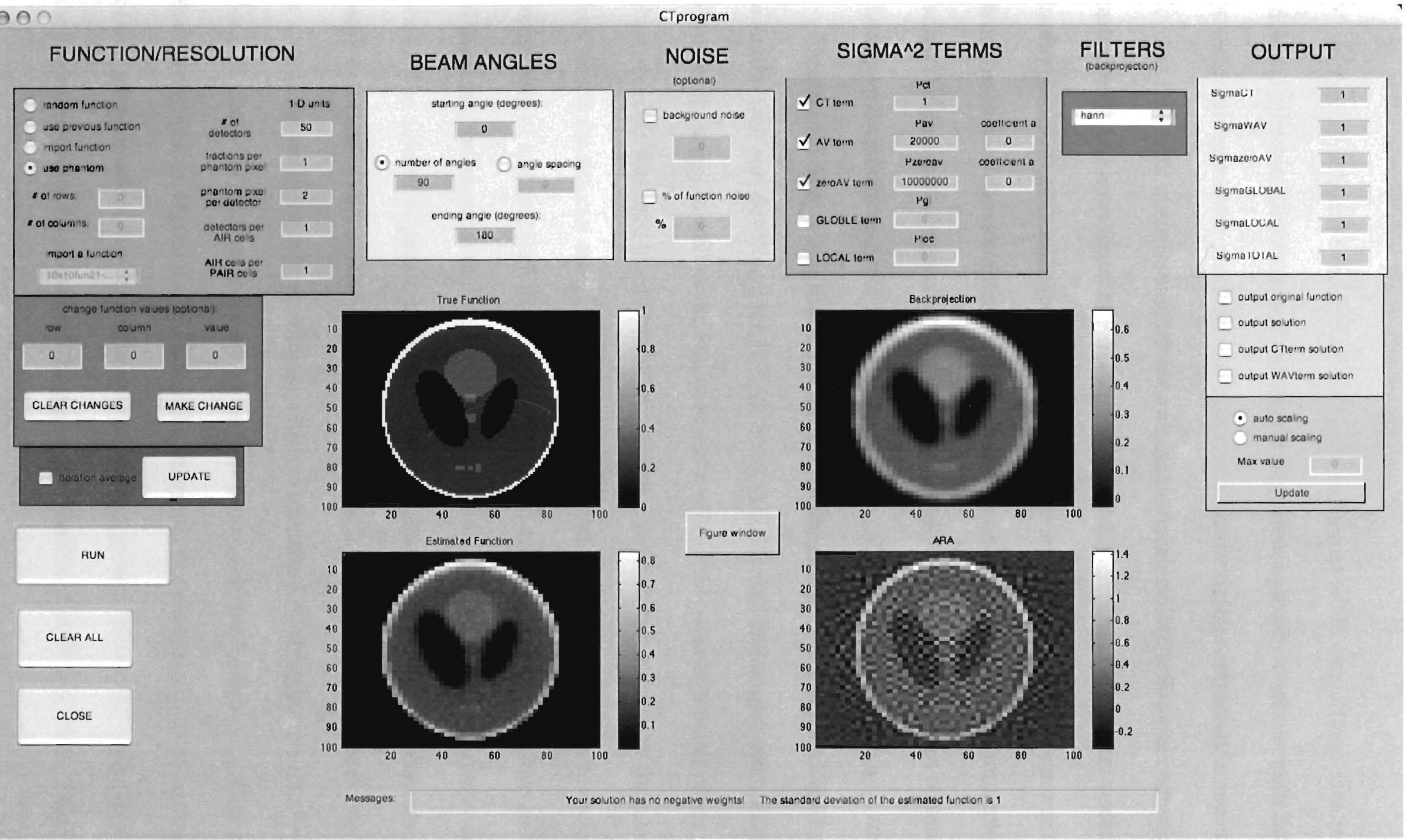
- A larger number of phantom pixels per reconstruction pixel (low reconstruction resolution relative to the imaged region) in order to ensure an accurate approximation of the shared areas of the grids.
- The number of detectors in the array was set equal to the reconstruction resolution (i.e.: if 50 detectors were used, the reconstruction resolution would be 50x50 pixels), in order to maintain a square matrix in equation (3.3.5).

This method of representing pixels will be referred to as the *inversion method* as opposed to the original method. When no mention of the method is made, the original method is assumed.

3.4 The Graphical User Interface (GUI)

Due to the need to test the reconstruction algorithms in many situations including, for example, changes to the phantom matrix and varied beam angle views, it was necessary to develop a simulation environment where changes to such parameters could be made without going into the code and manually making changes. As well, the output would need to be displayed in an organized fashion. For these reasons a graphical user interface was created through which all the simulations were performed. Matlab contains all the graphics necessary for the interface, all that was needed was to design the layout, and program each button, window, checkbox, and textbox.

Figure 3.4.1 shows the GUI for the CT simulation along with AIR and PAIR reconstruction (a similar GUI was developed for GIRA). Starting on the left there is a choice of the type of phantom to image, as well as the ability to make value changes to individual pixels in the phantom. This was used to add 'metal' type gradients. There is control over the number of detectors in the detector array. The number of fractions per phantom pixel allows for the added complexity of splitting each pixel of the phantom matrix into fractions. By increasing the number of fractions, the phantom matrix is remapped onto a higher resolution grid as was described above. The number of phantom pixels per detector effectively defines the phantom matrix resolution relative to the number of detectors. There is also, as discussed previously, the ability to choose the reconstruction resolution, which is controlled by changing the number of detectors per AIR pixel. To the right there are parameters that control the beam angles. One can specify the starting and ending angle, and either the increment or the total number of angles. To the right of that are parameters that control noise such as background 'electronic' noise, or a percent of the projection noise. Next over are the parameters specific to AIR and PAIR. There is control over the importance coefficients of the terms that comprise the objective function. After that there is a list of filters to choose from pertaining to the filtered backprojection algorithm, which was used as a control algorithm against which to compare the novel reconstruction algorithms. To the far right, the output displays a measure of the error in the reconstruction. As well as controls that allow rescaling of the images for visual purposes. The images are displayed in the four plots in the center. One can include the true phantom, AIR reconstruction, the difference between the phantom and the AIR image, as well as the filtered backprojection reconstruction for comparison, or one can display the basic ARA reconstruction. The center button entitled 'figure window' opens up a new window containing the images for quick and easy saving. The very bottom of the GUI displays crucial messages, such as the number of negative pixels in the reconstruction or other appropriate information.



○ random function
○ use previous function
○ import function
● use phantom

1-D units

# of detectors	50
fractions per phantom pixel	1
phantom pixel per detector	2
detectors per AIR cells	1
AIR cells per PAIR cells	1

# of rows: 0

# of columns: 0

import a function: 10x10fun21-...

change function values (optional):

row	column	value
0	0	0

□ iteration average

starting angle (degrees): 0

● number of angles: 90 ○ angle spacing: 0

ending angle (degrees): 180

□ background noise: 0

□ % of function noise

%: 0

☑ CT term	Pct: 1	
☑ AV term	Pav: 20000	coefficient a: 0
☑ zeroAV term	Pzeroav: 10000000	coefficient a: 0
□ GLOBLE term	Pg: 0	
□ LOCAL term	Ploc: 0	

hann

SigmaCT	1
SigmaWAV	1
SigmazeroAV	1
SigmaGLOBAL	1
SigmaLOCAL	1
SigmaTOTAL	1

□ output original function
□ output solution
□ output CTterm solution
□ output WAVterm solution

● auto scaling
○ manual scaling

Max value: 0

Messages: Your solution has no negative weights! The standard deviation of the estimated function is 1

Figure 3.4.1: The GUI for the CT simulation for AIR and PAIR (a similar one was created for GIRA). It allows for quick parameter changes such as the type of phantom to image. making individual changes to values in the phantom matrix, the number of detectors in the detector array, the reconstruction resolution relative to the phantom resolution, beam angle spacing, random noise in the projection. and rescaling of the reconstructed images, just to name some.

3.5 Implementing AIR and PAIR

While the theory of AIR and PAIR presented straightforward methods for image reconstruction, their implementation was made difficult due to computational effort and memory concerns. In approaching the issue of storing data, it was decided that the CT information could be stored most efficiently with the use of structures. There were two possibilities of how to orient these structures. The information could be stored as a subset of the detectors (or beamlets) or as the subset of the reconstruction pixels. If the information were to be stored as a subset of the pixels, each pixel index would contain the beamlet indices that it had contributed to as well as the respective fraction that is had contributed. By storing the information as a subset of the beamlets, each beamlet would contain the fractions and pixel indices that had contributed to that given detector. It was decided that organizing the information as a subset of beamlets/detectors was more practical. The structure called 'Beamlet' was organized as follows (where i is the index of all beamlets in the CT scan):

Beamlet(i).projection = The projection value of detector i.

Beamlet(i).cellspot = A vector of image pixel indices that contributed to the i^{th} projection.

Beamlet(i).h = A vector of values of the same size as the corresponding 'cellspot' vector, representing the fraction (or *h* value) of each pixel index that contributed to the i^{th} projection.

Beamlet(i).lengths = The length of vector 'cellspot'.

Note that the pixel indices and their respective fractions would at first be representative of the highest resolution used in the calculation of the projections. After the simulation they are updated to represent the user defined reconstruction grid resolution, which is not generally the same as the phantom resolution. This stage is critical to maintain 'blindness' with respect to the subject being imaged.

This structure called 'Beamlet' is the only information required to implement AIR. Presented here is the Matlab code to produce the Alpha and Beta arrays for the CT term objective function. (Equations (2.2.10) i, and ii):

```
alphaCT=zeros(res12,res12);          % initializing the Alpha matrix to the appropriate size
  for i= 1:length(beamlets)          % this loop runs through each beamlet index
     if Beamlet(i).lengths~=0        % continues if beamlet contains at least one pixel
        C=Beamlet(i).cellspot;
        H=Beamlet(i).h;
        for j=1:beamlet(i).lengths;
           for k= 1:j
             alphaCT(C(j),C(k))=alphaCT(C(j),C(k))+2*H(j)*H(k);
           end
        end
     end
  end
```

Notice that the code constructs only the lower triangular and the diagonal elements of the Alpha matrix. Because the matrix is symmetric, the upper triangular portion can be constructed easily using the transpose of the lower triangular portion.

```
betaCT= zeros(res12,1);              % initializing the Beta vector to the appropriate size
for i=1:length(beamlets)             % this loop runs through each beamlet index
     if Beamlet(i).lengths~=0;       % continues if beamlet contains at least one pixel
           pr= Beamlet(i).projection;
           C= Beamlet(i).cellspot;
           H= Beamlet(i).h;
           for j= 1: Beamlet(i).lengths
           betaCT(C(j))= betaCT(C(j))+2*pr*H(j);
           end
     end
end
```

For every detector, the code calls up the pixel indices that were traversed by that beamlet, and adds the appropriate value to the Alpha and Beta array index. As was discussed in the theory of AIR, the Alpha matrix is not constructed with the projection values, and can be initialized prior to the CT scan. The Alpha and Beta arrays for the other terms are constructed separately and then added together in proportion to their importance coefficients. The final values of the image pixels μ_k are then recovered by using the backslash function (essentially performing matrix left division) given by:

$$\text{Mews} = \text{Alpha}\backslash\text{Beta} \tag{3.4.1}$$

Where Mews is a vector containing the solution values μ_k, which can be reshaped into a square grid for visualization.

During the testing of the reconstruction algorithms, while experimenting with the values of the importance coefficients, there was a need to make multiple attempts to reconstruct an image from each CT simulation. For this reason it was not efficient to perform each CT simulation repeatedly and to construct the various matrices each time. To avoid this, after each CT simulation the structure 'Beamlet' and the Alpha and Beta matrices for each objective function were saved to the computer's hard drive. This allowed for fast testing where applicable. When a change was made to the phantom matrix, like the addition of a region of high density, this would cause a change in the projection values. In this case both the CT simulation (the projection measurements) and the Beta vectors would need to be recalculated. However the Alpha matrices would not be affected. The cases that would necessitate the construction of new Alpha matrix are either a change in the CT beam angles or a change in the desired reconstruction resolution.

Coding for PAIR not only implied that the above-mentioned procedure be carried out multiple times, but PAIR introduced additional levels of complexity. In theory the user defines the number of AIR pixels per PAIR pixel in order to meet computational needs. However, in this research only the first degree of PAIR was implemented. That is, four AIR pixels per PAIR pixel, or in other words, four PAIR images used to recover the desired resolution. To obtain each independent PAIR solution the structure 'Beamlet' needs to be modified. After the simulated CT scan the structure is configured for the AIR resolution, but it must be reconfigured for each PAIR orientation. Specifically, this can be achieved as follows:

Beamlet(i).projection - These values remain the same. The information here is not dependant on the choice of reconstruction resolution.

Beamlet(i).cellspot - These indices are remapped to conform with the PAIR orientation and resolution (**figure 2.3.1**).

Beamlet(i).h - These value need to be updated because *h* must represent the fraction of the PAIR pixel that is contained in the given beamlet.

Beamlet(i).lengths - These values are updated to equal the new length of vector 'cellspot'.

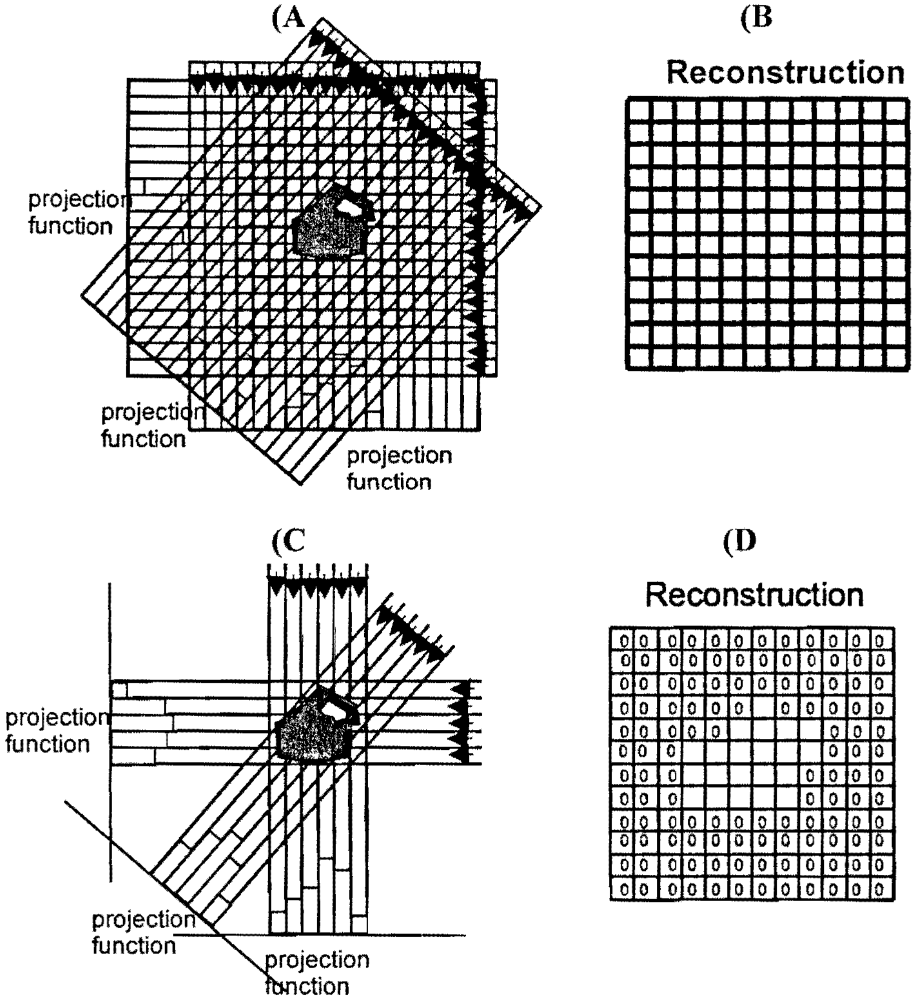
This structure is redefined for each PAIR solution. After the four independent low-resolution solutions have been calculated, they are mapped back onto the desired resolution grid in preparation to solve the final μ values. Now they can either be simply averaged together, or solved using equation (2.3.1). In this research, implementation of equation (2.3.1) to resolve the higher resolution image was not met with initial success and therefore the simple averaging technique (described in the theory of PAIR) was used. The reason for this will be discussed in the discussions section.

3.6 Implementing GIRA

GIRA requires almost the same information as AIR and PAIR, namely, the projection values from each beamlet $q_{m,\lambda}$, and the geometric region that each beamlet has traversed $h^{m,\lambda}$. However, whereas AIR and PAIR only require values for $h^{m,\lambda}$, GIRA requires the geometric locations or coordinates of all $h^{m,\lambda}$ values. This is necessary because GIRA may solve any geometric region of the imaged domain (recall that in theory, the reconstruction resolution need not be user defined in GIRA but is a function of the geometric regions traversed by beamlets). In this research testing of GIRA was conducted using the most simplified model of the simulation, namely setting the reconstruction resolution equal to the phantom resolution necessitating all $h^{m,\lambda}$ values to be equal to 0 or 1. The extension to less ideal situations will be discussed subsequently.

GIRA is split up into a number a distinct stages or functions, but does not progress through these stages linearly. Rather, GIRA moves back and forth between these stages each triggered by certain characteristics of the reconstruction.

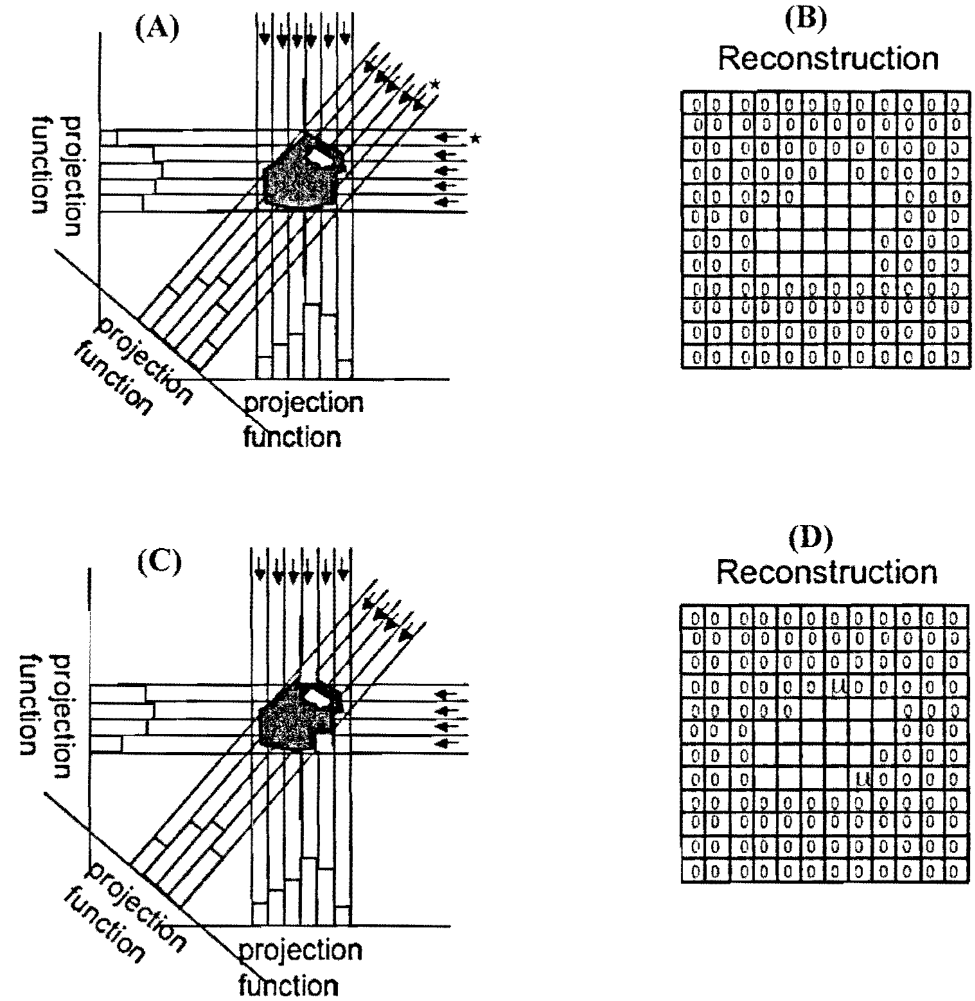
The first function of GIRA identifies all $q_{m,\lambda} = 0$. This special case is of much importance because it implies that all unknowns corresponding to $h_\ell^{m,\lambda} = 1$ must satisfy $\mu_\ell = 0$. In other words, if GIRA finds a projection value of zero, it can be assumed that the pixels that contributed to this projection must also be equal to zero. GIRA then fills in the reconstruction grid with the now known values, eliminates the zero projections from the computers memory, and finally, searches through all beamlets and sets all $h_\ell^{m,\lambda} = 0$ corresponding to pixels that have been solved. This effectively erases all solved pixels from the problem set. A graphical depiction of the first stage of GIRA, called "GIRA ZEROS", is illustrated in **figure 3.6.1** below. Once all zero projections have been exhausted (and the set of $q_{m,\lambda} = 0$ removed), and all known pixels have been eliminated from the problem set (by updating the set of h), GIRA moves on to the next stage.



0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0		0	0	0	0	0
0	0	0	0	0					0	0	0
0	0	0							0	0	0
0	0	0						0	0	0	0
0	0	0						0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.6.1: **(A)** an unknown region is imaged using a number of different angles (the arrows represent individual beamlets. **(B)** GIRA begins by initializing a solution grid, but no pixels have been solved. **(C)** All projections that are equal to zero are eliminated and all pixels that contributed to these zero projections are removed from the set of unknowns. **(D)** The reconstructed image is filled in with zeros where applicable.

The next function of GIRA searches for beamlets that have only passed through one unknown (ie, $\sum_{\ell} h_{\ell}^{m,\lambda} = 1$). This condition implies that the unknown pixel's density is equal to the projection value of that beamlet ($q_{m,\lambda} = \mu_k$, where k is the index of the unknown). The k^{th} pixel is filled into the reconstruction grid, and it must also be removed from the problem set. To do this, GIRA finds all beamlets that have passed through pixel k, (beamlets satisfying $h_k^{m,\lambda} = 1$) and eliminates this pixel from the beamlets memory by setting $h_k^{m,\lambda} = 0$. GIRA then subtracts the value of μ_k from all projections that pixel k has contributed to. This process of searching for beamlets that pass through one pixel, called "GIRA ONES" is repeated until there are no more beamlets that have passed through only one pixel. Once GIRA cannot perform the ONES function, it returns to the ZEROS function, as it is possible that subsequent to subtracting from the projections q, there may be some zero projections. For a graphical illustration of GIRA ONES, see **figure 3.6.2.**



(B) Reconstruction

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0		0	0	0	0	0
0	0	0	0	0					0	0	0
0	0	0							0	0	0
0	0	0						0	0	0	0
0	0	0						0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

(D) Reconstruction

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	μ	0	0	0	0	0
0	0	0	0	0					0	0	0
0	0	0							0	0	0
0	0	0						0	0	0	0
0	0	0					μ	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.6.2: (A) The problem space after the ZEROS function. The red stars indicate a beamlet that has only passed through one pixel. **(B)** The reconstruction grid subsequent to GIRA ZEROS. **(C)** The beamlets (and their corresponding projections) passing through only pixel are eliminated and the solution begins to be 'carved' out. Notice the other reduced projection values as a result of the eliminated pixel. **(D)** The reconstruction grid is filled in with solved densities.

GIRA passes between the ZEROS function and the ONES function until one of two conditions is met:

1) The reconstruction grid is completely filled and thus the reconstruction is complete.
2) There are no more zero projections and no more beamlets passing through one pixel.

If the second condition arises, GIRA will proceed to the next stage of reconstruction. In this stage called "GIRA TWOS", GIRA searches for two beamlets that pass through the same pixels, with one of the beamlets passing through one extra pixel. This implies that the difference between their projection values will be equal to the absorption density of the pixel that is exclusive to one beamlet. The value of that pixel is filled into the solution grid and all $q_{m,\lambda}$ and $h_k^{m,\lambda}$ are updated to remove that pixel and its absorption density from the problem. Once GIRA TWOS finds the solution to one pixel, GIRA goes back to looking for zero projections and beamlets that pass through one cell.

GIRA passes between the ZEROS function, the ONES function and the TWOS function until one of two conditions is met:

1) The reconstruction grid is completely filled and thus the reconstruction is complete.
2) There are no more zero projections, no more beamlets passing through one pixel, and no more sets of two beamlets that satisfy the necessary conditions to execute GIRA TWOS.

If the second condition arises, GIRA will proceed to the next function. In this next stage, no pixels are solved; rather, the information in $q_{m,\lambda}$ and $h_k^{m,\lambda}$ is reorganized to allow more pixels to be solved using ZEROS, ONES, and TWOS. This function, called GIRA MEWS, removes small groups of pixels from beamlets in order to uncover more solutions. GIRA MEWS works as follows. Beginning with the beamlet that has the smallest number of contributing pixels, these pixels are for the moment considered as one region with absorption density equal to the corresponding projection value. This group of pixels is then removed from all other beamlets containing the whole group. This is achieved by setting all $h_k^{m,\lambda} = 0$ for all the pixels in the group and subtracting their total absorption density from the beamlet's projection value. It is important to note that if another beamlet contains only some of the pixels in the group but not all, it is not modified. This process is carried out for all beamlets, starting from the beamlets passing through the smallest number of pixels.

GIRA MEWS is in reality a way to reorganize the projection information to uncover additional trivial solutions. Once MEWS has completed this task, it is likely that the other GIRA functions will be able to solve additional pixels. GIRA passes through all four functions in the way described above until the grid is solved.

As the projection values are updated throughout GIRA by subtracting pixel values from them, it may occur that a projection value becomes less than zero. This can occur if various projections contain inconsistent measurements of the same region, or it can result from a propagation of round-off error. If this happens, GIRA will immediately remove that beamlet from the problem set assigning any pixels contained in that beamlet a value of zero.

As mentioned before, it is foreseen that GIRA could be implemented without a predefined reconstruction pixel size. Consider for example GIRA ZEROS. When a projection value equals zero, it can be concluded that the entire geometric area the contributed to that projection has a density of zero. Instead of GIRA ONES searching for a beamlet containing only one pixel, it can be defined as a beamlet containing the smallest geometric area. In the simulations and testing that were conducted in this thesis however, a predefined reconstruction grid was defined.

There may be cases when GIRA may not be able to fully reconstruct an image, for example an underdetermined system. To deal with these cases, a number of approaches could be proposed. The first is to combine AIR with GIRA by using GIRA to reconstruct the image until it cannot solve more pixels, and then using AIR to solve the remaining system. Another way to combine the two algorithms is to find small groups of cells that can be solved by AIR within the GIRA algorithm. A third idea could be to add additional terms to the GIRA algorithm that would group pixels and assign them an average value based on a projection. These ideas have not been explored in this thesis but constitute an interesting direction of future research.

It is interesting to note that GIRA can be thought of as a method of solving a special case of a matrix equation. GIRA searches out the matrix rows that have trivial solutions, either because the right hand side (projection value) is equal to zero, or because there is only one unknown in the row. Once trivial solutions have been exhausted, GIRA performs minimal matrix algebra (similar to row reducing) to uncover other trivial solutions. This process continues until all unknowns are solved. The reason it can be called a 'special case' of a matrix problem is because the geometry of the beams leads to a system of equations with many trivial solutions, and it is precisely this type of system that GIRA can exploit.

3.7 Implementing ARA and the Filtered Backprojection

One of the most essential aspects of scientific research is the inclusion of a control. A control in the context of this research was the use of an established algorithm such as ARA and filtered Backprojection, against which to compare AIR, PAIR, and GIRA. For research based on simulations the employment of a control had especially pronounced importance. Implementing ARA and filtered backprojection on the simulated data provided insight into any bugs in the CT simulation itself.

Matlab comes equipped with a very easy to use filtered backprojection algorithm called *iradon*, but it is based on the projections created using the Matlab function *radon*, which was described above. It would not represent a control to implement filtered backprojection using the *radon/iradon* functions while employing the newly developed CT simulation for the novel algorithms. A control by definition needs to be subjected to the same conditions as the experimental treatment. It was therefore necessary to write the code for filtered backprojection as well as ARA from scratch. The code would need to take as an input the structure 'Beamlet' from the CT simulation, and output the reconstructed images from the filtered backprojection and ARA.

Implementation of ARA in Matlab was trivial as the structure 'Beamlet' is configured quite well for construction of the ARA arrays. Beamlet(i).projection is the value of the ith index in the Beta vector. Beamlet(i).cellspot represents the column indices of Alpha matrix, and Beamlet(i).h are the values assigned to those indices

In implementing the filtered backprojection, it was necessary to reorganize the information from the CT simulation. The projection values from Beamlet(i).projection were rearranged into a matrix, called 'Proj', with the number of rows equal to the number of beamlets in a beam (in other words the number of detectors at any angle) and the number of columns equal to the number of beam angles. With this organization, the Matlab function for the fast Fourier transform, *fft* can be applied to each column in one command. However, before the projection information could be transformed to the frequency domain it must be zero padded. This was achieved by adding additional rows of zeros to the bottom of the matrix 'Proj' (approximately doubling the number of rows). After transforming the projections to the frequency domain using *fft*, the data was multiplied by a filter. The filter designs and their respective Matlab codes were taken from the *irdaon* function. Included was the basic cropped Ram-Lak filter, and various manipulations of this filter through multiplication of cosine or sine functions such as cosine ramps and hamming windows. Included in the GUI was the ability to change the filter type with ease. In preliminary testing it was determined that the Ram-Lak filter multiplied by a Hann window provided the optimal filter, so all the filtered backprojection results in this research used this filter.

After filtering the transformed data, the function *ifft* was used to obtain the inverse transform of each column resulting in a matrix, called 'filterproj' with columns representing the filtered projection data for each beam angle. Before backprojection, the additional rows that were added to zero pad the data were removed. Each column in the array 'filterproj' then represented the filtered projection for each beam angle. The values in the 'filterproj' array were then backprojected one by one, summing onto the image pixel grid. The code for the backprojection after filtering is written out below for clarity.

```
Image=zeros(res1,res2);          % initializing size of  image grid
for i=1:length(beamlets)         %  loop runs through each beamlet
   p=filterproj(i);              % the ith filtered projection value corresponding to the ith detector
   Place = Beamlet (i).cellspot;    % vector of pixel indices the were traversed by the ith beamlet
   if Place~=0;                  % continues if beamlet traversed at least one pixel
      w= Beamlet (i).h;          % vector of fractions of the pixels that were traversed
      for L=1:length(Place)      % for each pixel index
   % the next line of code adds a fraction of the filtered projection to the appropriate pixel
           Image(Place(L)) =Image(Place(L)) + p*w(L);
      end
   end
end
```

Finally the image needed to be scaled by the appropriate scaling factors. There was a scaling factor corresponding to the differential angle spacing; that is, the image pixels were multiplied by the range of angles used (in radians) divided by the number of beam angles used. Additional scaling corresponded to the image resolution relative to the resolution of the function imaged.

With the CT simulation complete, and all reconstruction algorithms coded, testing of the hypothesis began.

3.8 A Note on Computational Constraints

All simulations and testing were carried out on a Macintosh with a 2 GHz Intel Core Duo processor, and 2 GB of RAM. The computer's memory limitation constrained the domain of the testing. The largest solvable matrices in AIR were about 4,900 x 4,900 (corresponding to a reconstruction resolution of 70x70).

CHAPTER IV

TESTING AND RESULTS

4.1 Validation of the CT simulation

Before testing the algorithms, it was necessary to validate the CT simulation. To achieve this the images produced using the built-in filtered backprojection algorithm, *iradon*, from projections obtained with the built in *radon* transform function were compared against the images obtained using the newly developed CT simulation, reconstructed with the coding of filtered backprojection presented in section 3.7. Both reconstruction algorithms used a hann filter. The initial results of the validation seemed to suggest some error in the simulation or the coding of the filtered backprojection, as the resulting image appeared more coarse and grainy compared to the image produced with *radon/iradon* (**figure 4.1.1**). This could have been due to one or two factors. The first was the absence of interpolation in the CT simulation. Interpolation tends to act like a smoothing function for filtered backprojection. The second was that the *radon* function splits each pixel in the phantom matrix into four fractions before taking the projections. This also contributes to the overall smoothness of the resulting images. By increasing the number of fractions per phantom pixel (recall that this allows for partial cell contributions) a result nearly matching the image from *radon/iradon* was achieved. This validated the simulation and it was time to test AIR, PAIR, and GIRA. Testing of the reconstruction algorithms began with the most simple simulation models and increased in simulation complexity throughout the testing.

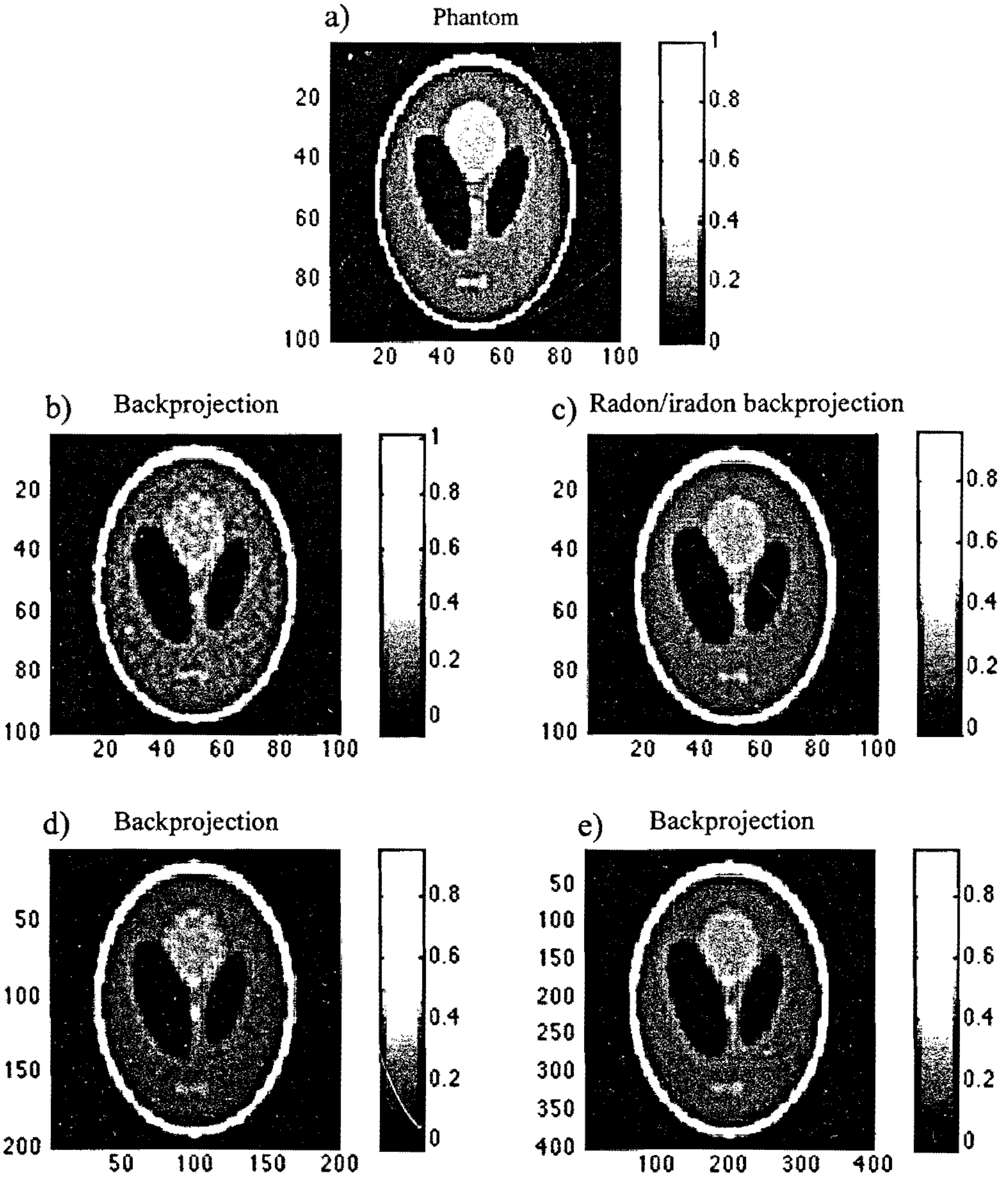


Figure 4.1.1: **a)** Shepp and Logan head phantom at 100x100 resolution. **b)** Using our CT simulation and reconstructing the image with our adaptation of the filtered backprojection algorithm, the image appears grainy and coarse. **c)** Using the 'radon' function to produce the CT projections and 'iradon' for the reconstruction, a much smoother image is obtained. This is because *radon* uses a process of linear interpolation in the backprojection algorithm, while the image in **b** is produced assuming whole contributions from pixels. **d)** By increasing the number of fractions in our simulation (essentially splitting each pixel into fractions) we can approximate the effect of linear interpolation. The image in **d** was obtained using 4 fractions per phantom pixel. Notice that the number of pixels on the grid has been multiplied by four. This does not represent a higher resolution. The solution is just mapped onto a finer grid to indicate that we have used 4 fractions per phantom pixel. **e)** Using 16 fractions per pixel the image now almost exactly matches the image in **c**. This convergence demonstrates the validity of our simulation. Each image was produced from 180 beam angles spanning 180 degrees. Each reconstruction uses the same Hann filter.

4.2 Testing AIR and PAIR

For a straightforward description of the simulation parameters for each test, the following notation is used:

P = The square root of the resolution of the phantom being imaged. (If the phantom is 60x60, **P** will be 60).

F = The square root of the number of fractions dividing each phantom pixel. (As **F** is increased the simulation becomes more realistic with respect to pixels' fractional contributions to the projections). When displaying the phantom and the reconstructed images, this grid will be used as a reference (i.e. the phantom and reconstructed images will be displayed on a **(PFxPF)** size grid. If **F** is not specified, assume **F**=1.

D = The number of detectors in the array per beam angle.

R = The square root of the resolution of the reconstructed image. (As R is decreased relative to **P**, the simulation becomes more realistic with respect to inhomogeneities within the reconstruction pixel size).

M = The size and μ coefficient of an added metal gradient in the imaged region. If **M** is not specified assume no metal insert.

A = The range of beam angles is displayed as follows (in degrees):
Start angle : angle spacing : end angle
Or in an alternative representation, which may at times be more useful:
(Start angle | end angle | number of angles)

Pnoise = The maximum percent noise perturbing each projection value. If **Pnoise** is not specified assume no noise.

$\mathbf{C}^{ct}$ = The value of the CT term importance coefficient (see equation 2.2.21). If $\mathbf{C}^{ct}$ is not specified, assume $\mathbf{C}^{ct}$=1. A similar term will be used for each importance coefficient. If the other terms are not specified, assume their importance equal to zero.

It must be stated that it would neither be organized nor systematic to include all test results. The results displayed in this paper are the ones deemed most relevant to the hypothesis. In order to orient the reader to the organization of the results, a general description and motivation for each test is now presented.

Test descriptions: a road map of the AIR/PAIR tests

The first set of tests comprising **figures 4.2.1, 4.2.2,** and **4.2.3** are intended to demonstrate a number of relevant results. **Figure 4.2.1** demonstrates AIR's proof of principle in reconstructing an image from a sufficient number of projections in the most simple simulation model (i.e. a determined system with consistent equations). Following this the number of beam angles is reduced in **figures 4.2.2** and **4.2.3** creating underdetermined systems to demonstrate how the various averaging terms are employed in AIR to reconstruct an optimal image.

The second test also reduces the beam angle views, but here the beam angle range is limited to 90 degrees. This test again demonstrates that by employing the AIR averaging terms an optimal solution is obtained.

The third test includes two difficulties for image reconstruction; limited beam angle views and a sharp gradient in the imaged region. This test also observes the effects of simulating projections when the phantom pixels are divided into fractions indicated by the variable **F**. The simulation is therefore performed twice. Once for **F**=1 and once for **F**=4.

The fourth test introduces simulated noise in the projection measurements. This simulation is also performed twice. Once for **F**=1 and once for **F**=4.

The simulations in tests one through four are ideal in that the phantom resolution and reconstruction resolution are the same (i.e. **P**=**R**). In the following tests (five through eleven), this ideality is removed from the simulations. Many of these tests parallel the earlier tests, only with this added non-ideality.

The fifth test obtains projections using a high-resolution phantom but reconstruction is performed using a relatively lower resolution. This test also demonstrates the proof of principle of the inversion method (described in section 3.3).

In the tests that follow, the reconstruction resolution is increased from the previous tests. Given the computational constraints (see section 3.8), this precludes the use of ARA and necessitates the use of PAIR (see section 2.3) to obtain the AIR solutions.

The sixth and seventh tests attempt reconstruction at a higher resolution than that dictated by the number of detectors in the detector array (i.e. **D**<**R**).

The eighth test attempts reconstruction from an insufficient number of beam views - a parallel to the first test.

The ninth test attempts reconstruction with the presence of a sharp gradient in the imaged region.

The tenth test simulates a limited beam angle range with the more complex simulation model (compared to the second test).

The eleventh test includes two difficulties in image reconstruction; limited beam angle views and random noise in the projection functions.

Displaying the result: a road map to the AIR/PAIR figures

Due to the nature of this research, many figures are presented in order to compare AIR with the original phantom, filtered backprojection, and ARA. For organization, the images in a figure are titled and are often labeled alphabetically as follows: **Phantom (a)** depicts the phantom being imaged. **Backprojection (b)** displays the filtered backprojection reconstruction. When ARA is present, it is given the title **ARA (d)**, but when it is not included, the label **(d)** may be used for AIR. There are often multiple **AIR** or **PAIR** reconstructions in a figure. This is necessary to examine the effects of varying the importance coefficients of the objective terms in AIR (see equation 2.2.21). This is certainly the case in the first few tests (one to five), which specifically examine the effects of changing the importance coefficients. From tests six and on, once PAIR is included, for organization, only one AIR result is displayed (one choice of importance coefficients) representing the best estimate of the reconstruction (as judged visually by this researcher).

The range of values in any reconstruction varies with the reconstruction method. The figures are displayed in the results *each with its own relative contrast.* That is, if a reconstructed image has values between -0.5 and 0.5, this is the range of values depicted. Presented as such, one can observe the accuracy of the reconstruction in terms of representative attenuation coefficients. If every image was displayed from 0 to 1, negative attenuation values and other deviations will not be visible. However, when displaying the images in the tests with a sharp gradient, the maximum value is set to one in order to visualize the image (no lower limit is set). For the sake of comparison, each set of results is displayed with value range from 0 to 1 on the adjacent page. These rescaled results are labeled with the letter **i**. For example, **Figure 4.2.1** are the results of the first test with relative contrast, and **Figure 4.2.1i** are the same results rescaled from 0 to 1.

Test one: Proof of principle of AIR

The first set of results obtained from the ideal simulations reveals the proof of principle of AIR. **Figure 4.2.1** is an example of a determined system with consistent equations – an ideal situation. In this case both AIR **(c)** and ARA **(d)** recover the exact image. Here AIR does not require any averaging terms. The similarity of the ARA and AIR images reflects the similarity of the two algorithms. As was discussed previously, AIR reformulates the same information into a square matrix equation. This has obvious advantages even in this first test. In this test AIR inverted an array of size (3600 x 3600) to reconstruct the image while ARA inverted an array of size (13696 x 3600). The extra rows reflect the number of beamlets/detectors in the scan. Because the filtered backprojection method is based on approximations, an exact solution is not possible **(b)**. By increasing the number of fractions per phantom pixel (shown later in **figure 4.2.5**) the filter backprojection image can be enhanced, but it cannot be exact.

The first set of tests were conducted with the following parameters:

P = 60; **D** = 60; **R** = 60; **A** = 0: S :180. Varying **S** (the angle spacing) from 1 to 18. When **S** =1 the simulation is performed with 180 beam angles. When **S** =18 the simulation is performed with a mere 10 beam angles.

S = 1

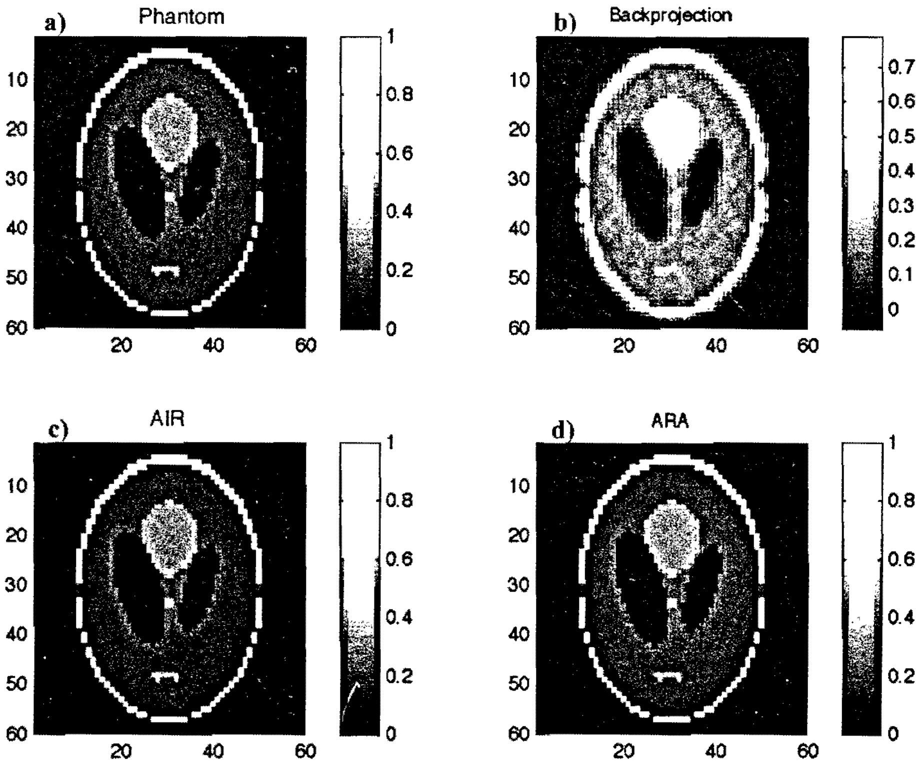


Figure 4.2.1: a) The phantom being imaged. **b)** Image reconstruction using filtered backprojection and a Hann filter. For an explanation of the degraded image quality see validation. **c)** Image reconstruction using AIR (only the *CT* objective term) produces exact reconstruction. **d)** Image reconstruction using ARA also produces an exact solution. This is possible because the system is determined with 180 beam angles, and no noise in the projections.

S = 1

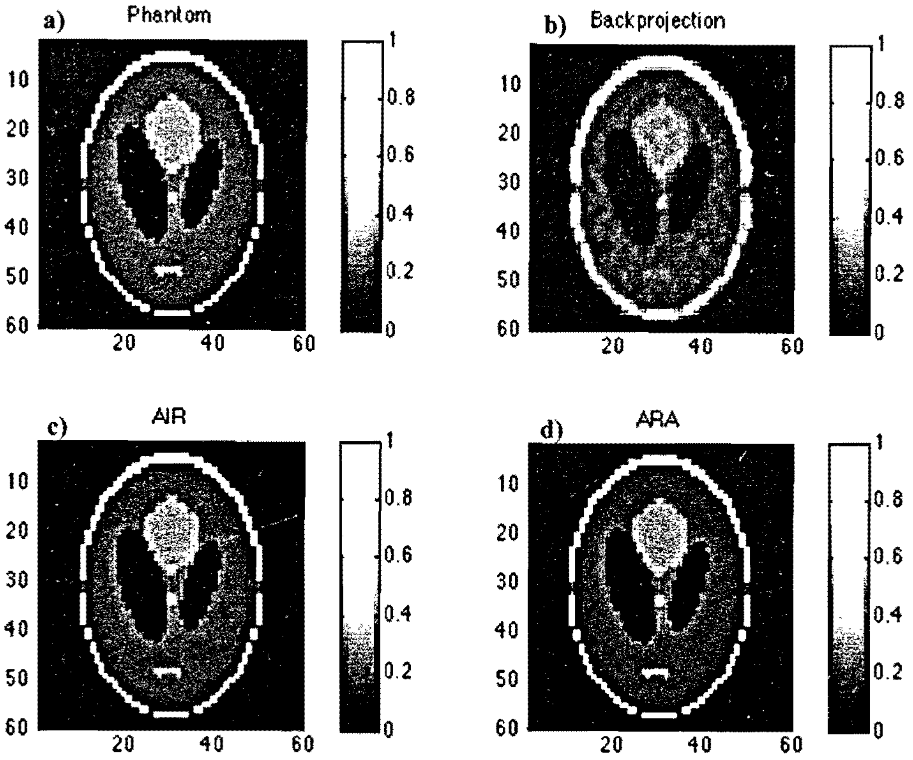


Figure 4.2.1i rescaled: a) The phantom being imaged. **b)** Image reconstruction using filtered backprojection and a Hann filter. For an explanation of the degraded image quality see validation. **c)** Image reconstruction using AIR (only the *CT* objective term) produces exact reconstruction. **d)** Image reconstruction using ARA also produces an exact solution. This is possible because the system is determined with 180 beam angles, and no noise in the projections.

Averaging terms of AIR used to break degeneracy

The following figures (**figure 4.2.2** images **c, e, g,** and **h)** depict how the main objective functions in AIR interact to reconstruct the image. By decreasing the number of beam angle views (the spacing, **S** is increased to 4 degrees between beam angle views) the system becomes underdetermined. However, at this point in the testing there are no inconsistencies in the projection data. When AIR relies only on the *CT* term objective function, the result is a non-physical solution much like ARA (compare images **c** and **d**). From the projection information of an underdetermined system, there may be an infinite number of solutions.

Images **e** and **f** depict the similarity between the backprojection method without filtering and AIR's weighted averaging term. Both methods smear the projection data back along the beamlets' paths. By employing the weighted averaging and the zero averaging terms to break degeneracy, AIR is able to recover the exact image (**h**). In this case, the main term is the *CT* objective function. The exact values of importance coefficients of the averaging terms are not crucial here. They can even vary in order of magnitude and will not affect the result, as long as they are not given much more importance than the *CT* term. Image **g** shows the result of employing a small amount of averaging.

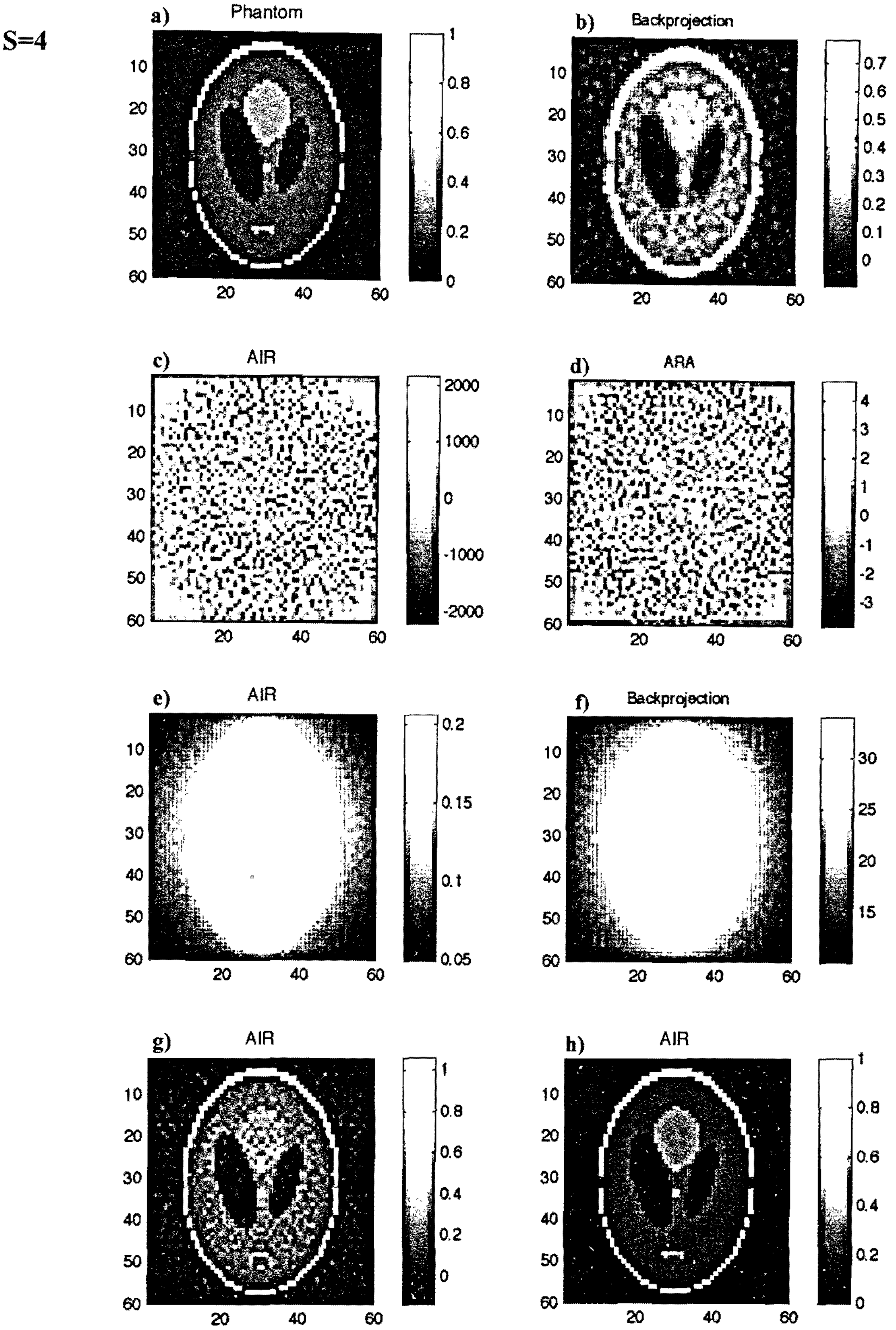


Figure 4.2.2: a) The phantom being imaged. **b)** Filtered backprojection with a Hann filter. **c)** AIR with $C^{ct}=1$. **d)** ARA cannot reconstruct the image as the system is underdetermined. **e)** AIR with $C^{ct}=0$ and $C^{wav}=1$. **f)** Backprojection algorithm with no filtering. Notice the similarity between image **f** and image **e**. Both algorithms are based on the concept of smearing a projection back along the beamlet's path. **g)** AIR with $C^{ct}=1$ and $C^{wav}=1.0\text{x}10^{-6}$. **h)** AIR with $C^{ct}=1$, $C^{wav}=1.0\text{x}10^{-6}$, $C^{zeroav}=1$. Image **h** is an exact reconstruction of the phantom, even though there were only 45 projections taken.

S=4

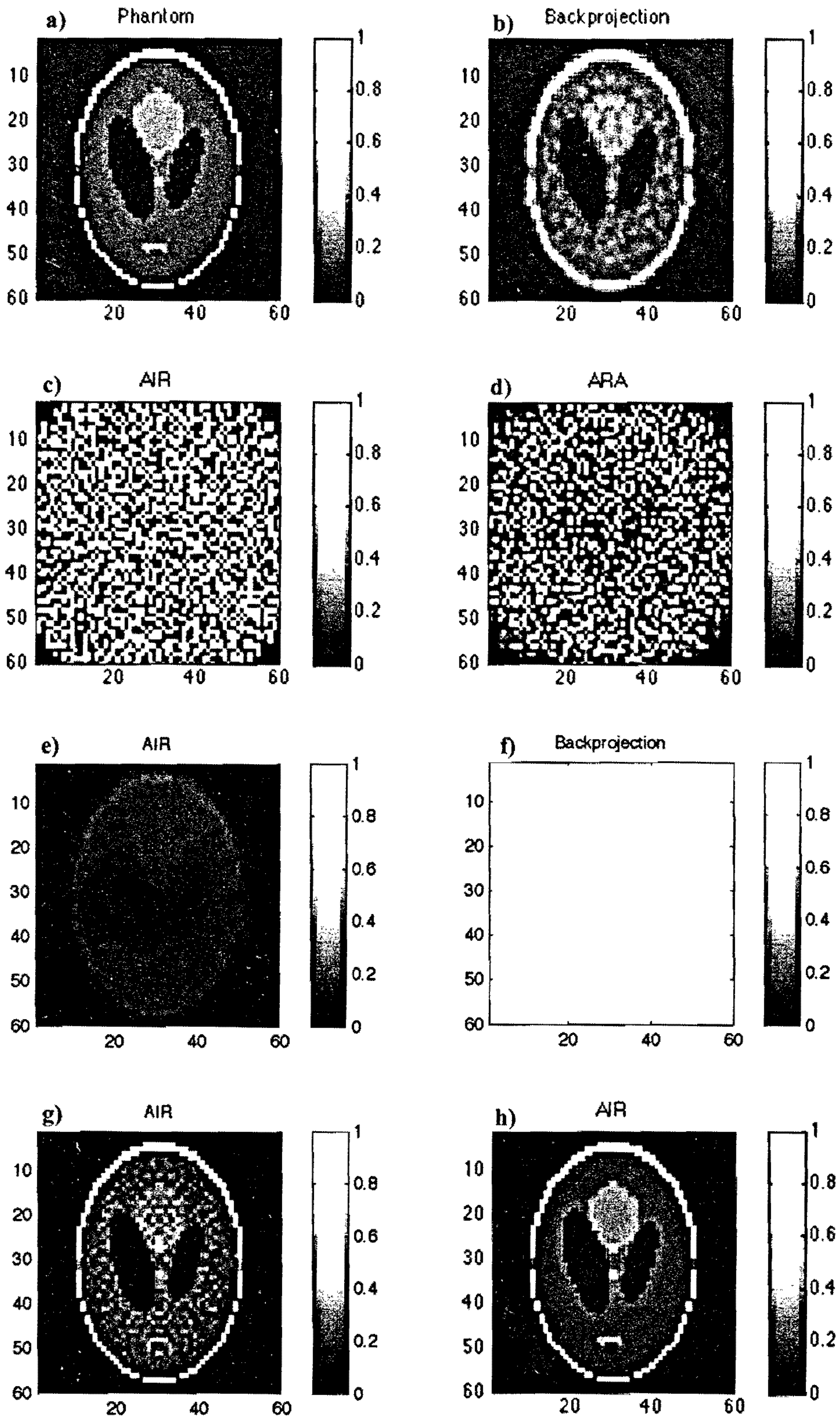


Figure 4.2.2i rescaled: a) The phantom being imaged. **b)** Filtered backprojection with a Hann filter. **c)** AIR with $C^{ct}=1$. **d)** ARA cannot reconstruct the image as the system is underdetermined. **e)** AIR with $C^{ct}=0$ and $C^{wav}=1$. **f)** Backprojection algorithm with no filtering. Notice the similarity between image **f** and image **e**. Both algorithms are based on the concept of smearing a projection back along the beamlet's path. **g)** AIR with $C^{ct}=1$ and $C^{wav}=1.0\times10^{-6}$. **h)** AIR with $C^{ct}=1$, $C^{wav}=1.0\times10^{-6}$, $C^{zeroav}=1$. Image **h** is an exact reconstruction of the phantom, even though there were only 45 projections taken.

Reducing beam angle views

As the number of beam angle views continues to drop **(figure 4.2.3)**, the exact solution cannot be recovered due to a lack of information. The weighted averaging and zero averaging terms begin to play more of a central role in image reconstruction and in avoiding negative attenuation values. The values of the importance coefficients seem to depend on the number of projections taken, and the reconstruction resolution. However, there has not been an effort made to develop an automatic way of determining these coefficients. This is a necessary development before AIR can be considered a practical reconstruction tool.

The images on the right show the extreme sensitivity of filtered backprojection to the number of beam angle views. The large number of negative attenuation values, streaking artifacts, noise, and low contrast render the images unrepresentative.

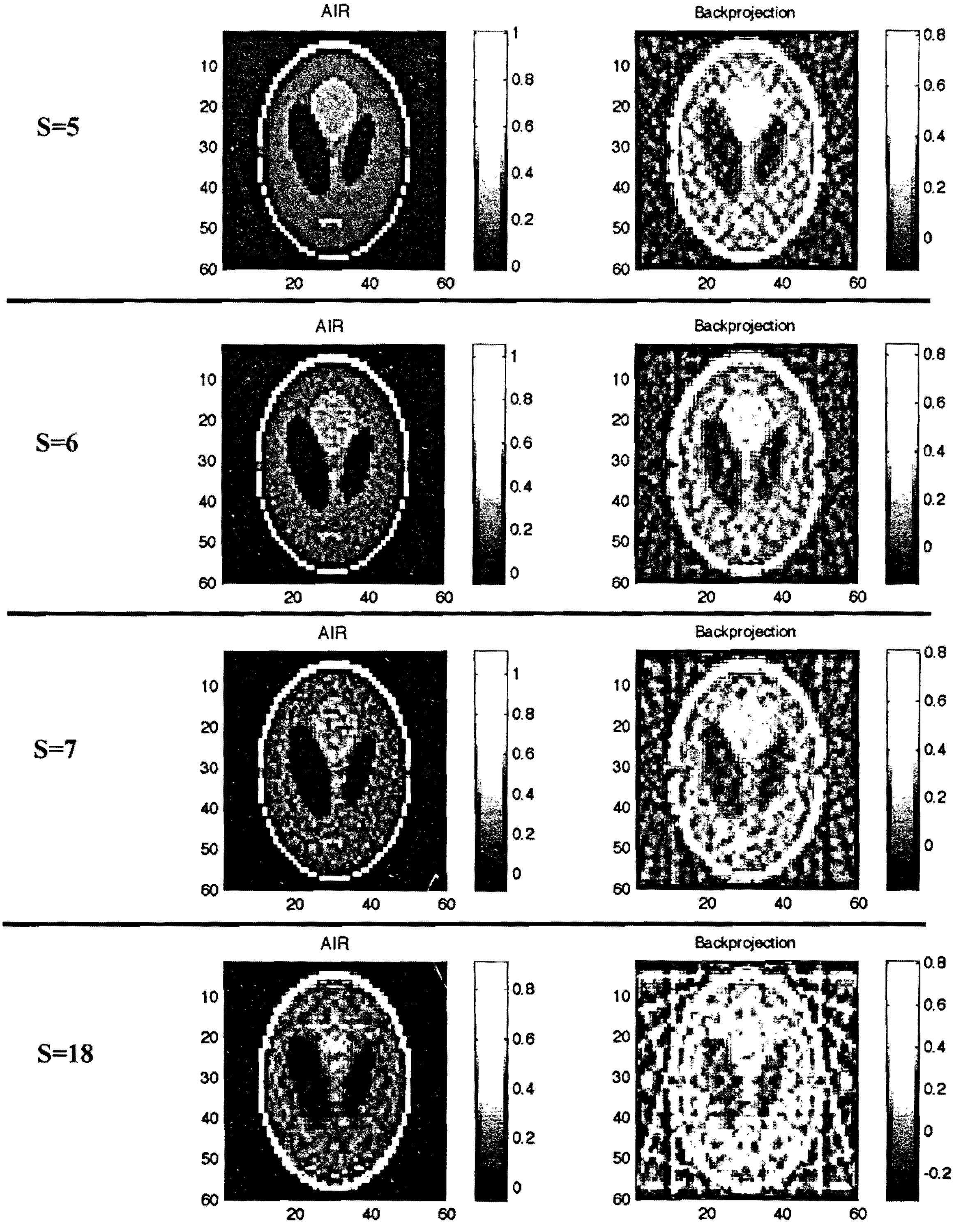


Figure 4.2.3: A comparison of AIR and filtered backprojection as the number of beam angles drops (the spacing between angles increases). ARA is not included because all of these simulations lead to underdetermined systems. Images on the left are AIR reconstructions. These images were obtained with $C^{ct}=1$, $C^{wav}=1.0\text{x}10^{-6}$, $C^{zeroav}=1$, except the last image which was obtained with $C^{ct}=1$, $C^{wav}=2000$, $C^{zeroav}=1000000$. Images on the right were all obtained using filtered backprojection with a Hann filter.

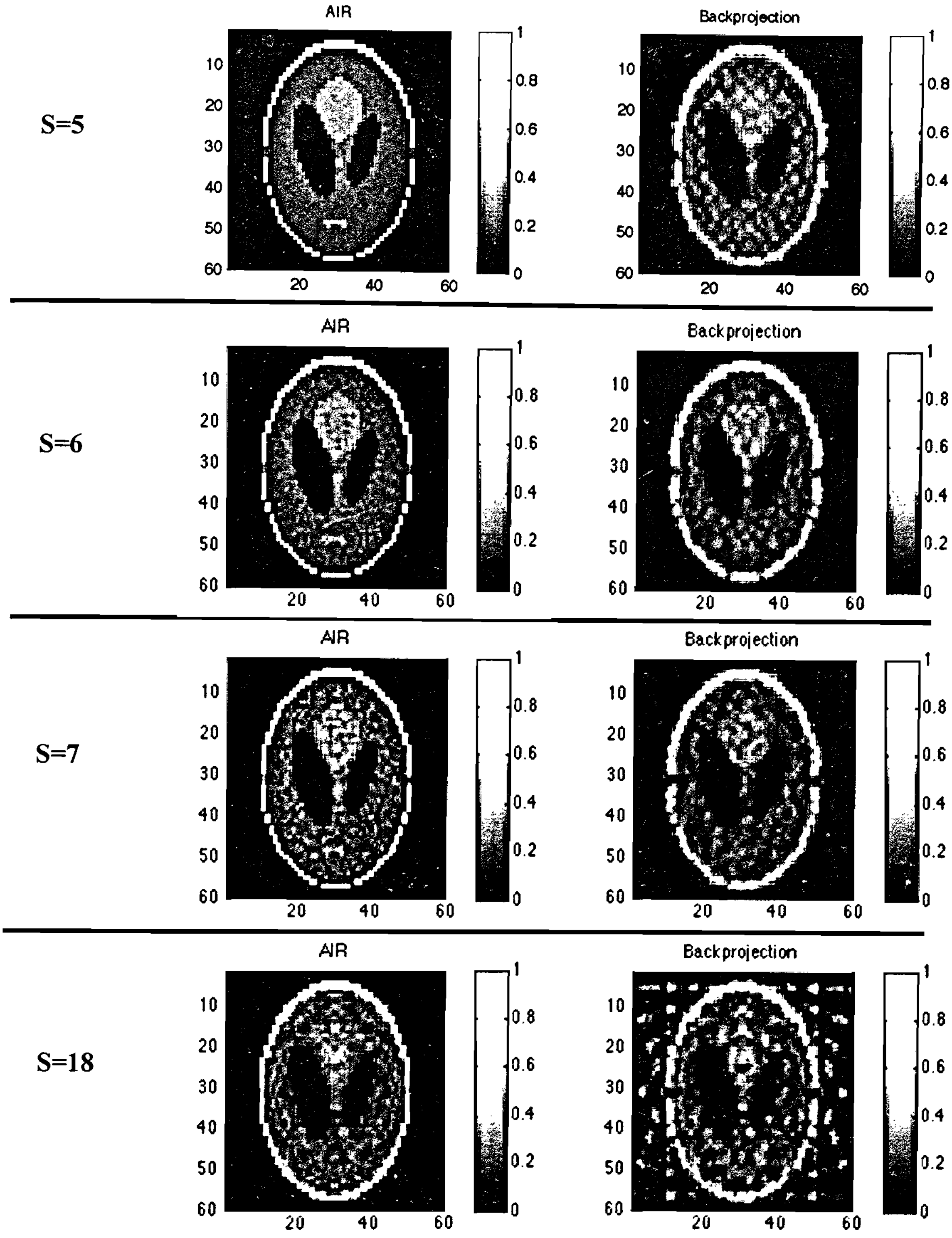


Figure 4.2.3i rescaled: A comparison of AIR and filtered backprojection as the number of beam angles drops (the spacing between angles increases). ARA is not included because all of these simulations lead to underdetermined systems. Images on the left are AIR reconstructions. These images were obtained with $C^{ct}=1$, $C^{wav}=1.0\text{x}10^{-6}$, $C^{zeroav}=1$, except the last image which was obtained with $C^{ct}=1$, $C^{wav}=2000$, $C^{zeroav}=1000000$. Images on the right were all obtained using filtered backprojection with a Hann filter.

Test two: Limited beam angle range

In the second test (**figure 4.2.4**), the CT simulation scans only a 90-degree range, using only 45 beam angle views. The filtered backprojection algorithm relies on a sufficient and even sampling in the frequency domain, and thus cannot reconstruct the image without serious degradation (image **b**). Images **c** and **d** again depict the similarity of ARA and the *CT* objective term. The system is degenerate and cannot be solved. By adding a small amount of averaging AIR begins to recover the structure of the image (image **e**). By employing the zero averaging term along with the weighted averaging term, AIR collapses to the exact solution (image **f**). The main point to realize is that while the weighted average term is expected to perform in a manner similar to backprojection, and thus may degrade the image quality, the crucial term is the *CT* term and the others are employed only to break degeneracy.

The second set of tests were conducted with the following parameters:

P = 60; **D**=60; **R** = 60; **A** = (0|90|45). That is, scanning only a 90-degree range using 45 beam angles.

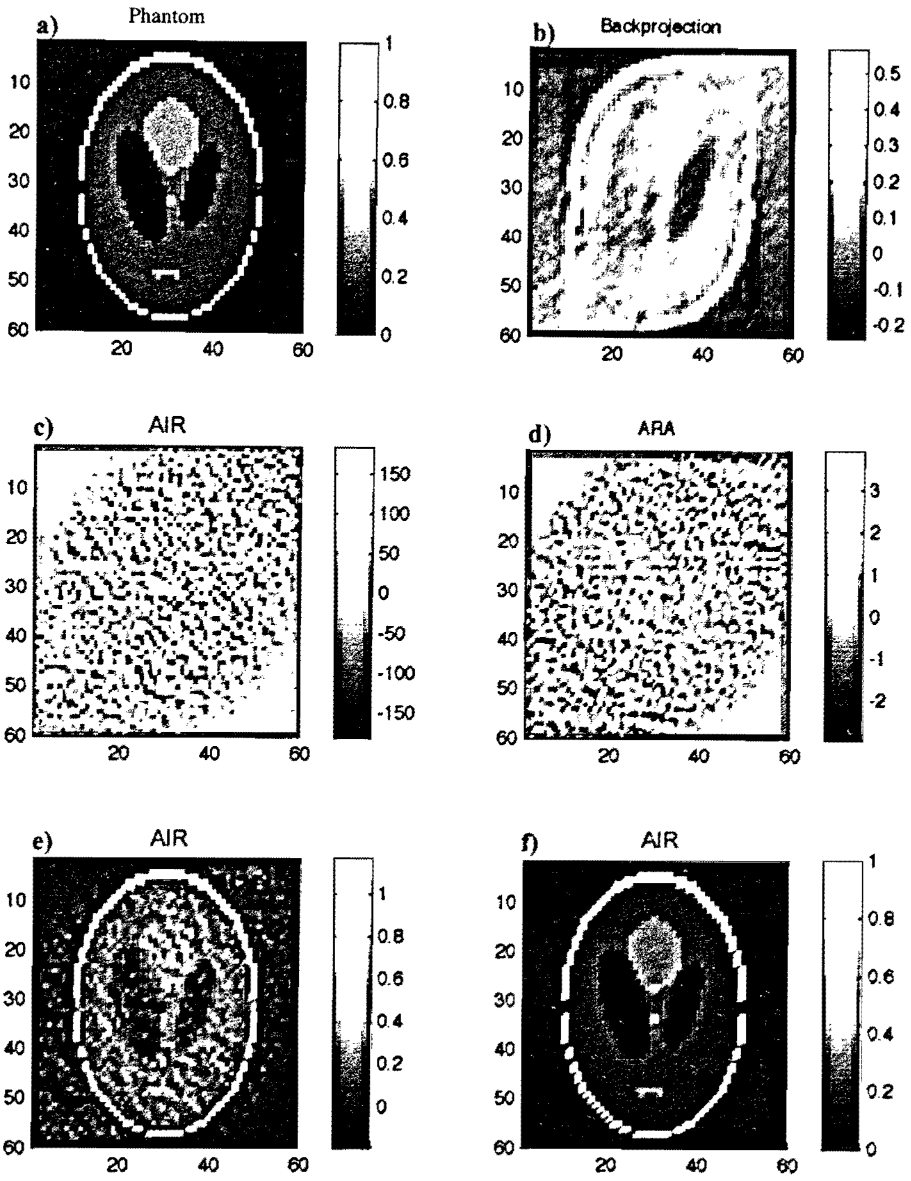


Figure 4.2.4: a) The phantom being imaged. **b)** Backprojection cannot reconstruct the image without serious degradation. **c)** AIR with $\mathbf{C}^{ct}$=1. **d)** ARA cannot reconstruct the image, as the system is underdetermined. **e)** AIR with $\mathbf{C}^{ct}$=1, $\mathbf{C}^{wav}$=1.0x10^{-6}. **f)** AIR with $\mathbf{C}^{ct}$=1, $\mathbf{C}^{wav}$=1.0x10^{-6}, $\mathbf{C}^{zeroav}$=1. Image **f** is an exact reconstruction of the phantom, even though there were only 45 projections taken spanning 90 degrees.

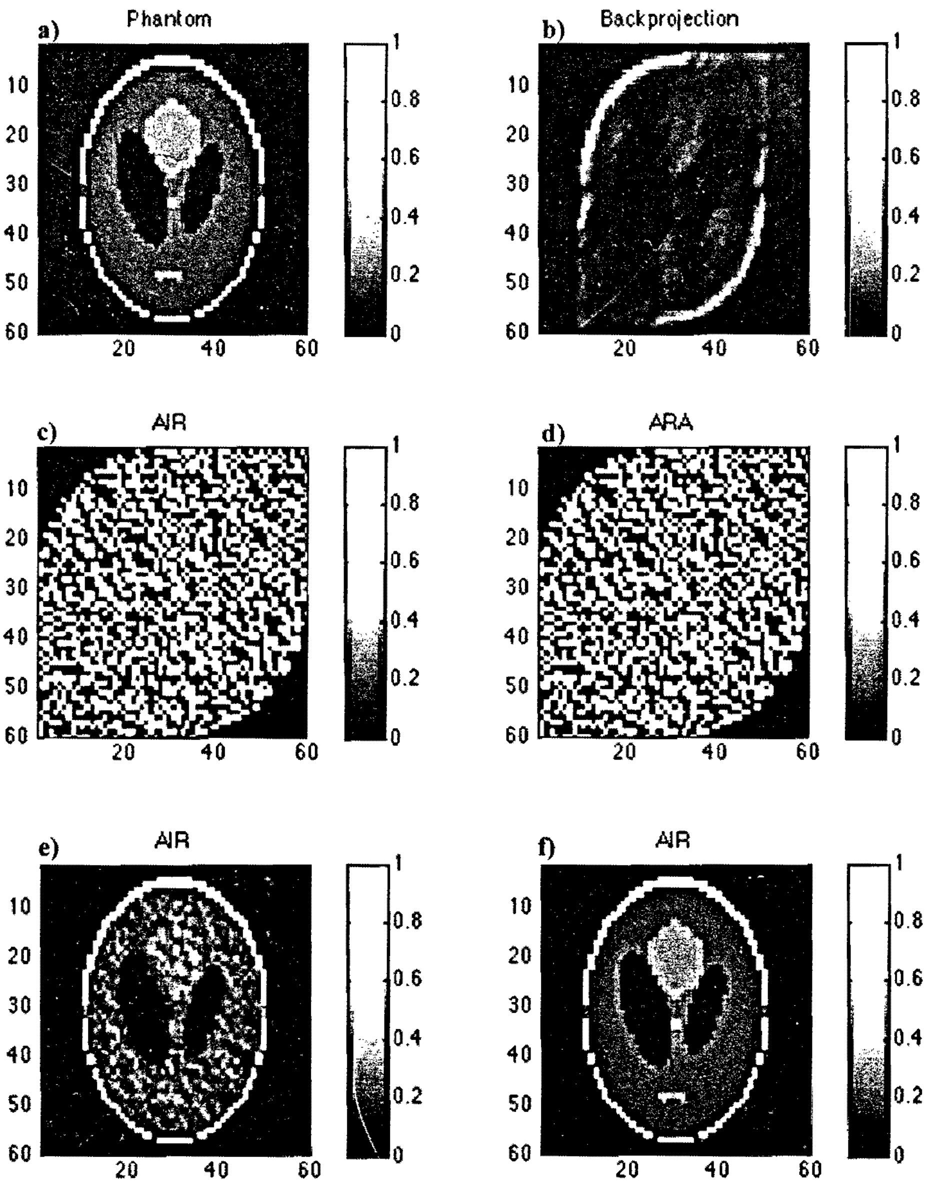


Figure 4.2.4i rescaled: a) The phantom being imaged. **b)** Backprojection cannot reconstruct the image without serious degradation. **c)** AIR with C^{ct}=1. **d)** ARA cannot reconstruct the image, as the system is underdetermined. **e)** AIR with C^{ct}=1, C^{wav}=1.0x10^{-6}. **f)** AIR with C^{ct}=1, C^{wav}=1.0x10^{-6}, C^{zeroav}=1. Image **f** is an exact reconstruction of the phantom, even though there were only 45 projections taken spanning 90 degrees.

Test three: Metal gradient

It is well known that the filtered backprojection introduces ringing or streaking in the presence of sharp density gradients. In the third test (**figure 4.2.5**), a single pixel in the phantom matrix is replaced with a value of 30 (the maximum value of the phantom is otherwise 1). This simulation model also begins to increase complexity with the introduction of pixel fractions (images **e** and **f**).

The filtered backprojection images are greatly enhanced by this additional factor. The fractions help smooth the image when the filtered backprojected data is being summed together in the image grid (compare images **b** and **f**). Notice the large number of negative attenuation values.

Here again ARA cannot reconstruct the underdetermined system (image **d**). It should be noted that in this test ARA's inability to reconstruct is a function of the limited beam angles and not the metal gradient.

AIR with the degeneracy breaking averaging terms reconstructs an exact image (images **c** and **e**). While the addition of the fractions alters the projection information, it does not introduce inconsistencies and thus does not have a significant effect on AIR at this point.

The third test was conducted with the following parameters:

P = 60; **F** = 1 or 4; **D** = 60; **R** = 60; **M**= (1x1), μ=30; **A** = (0 | 180 | 45)

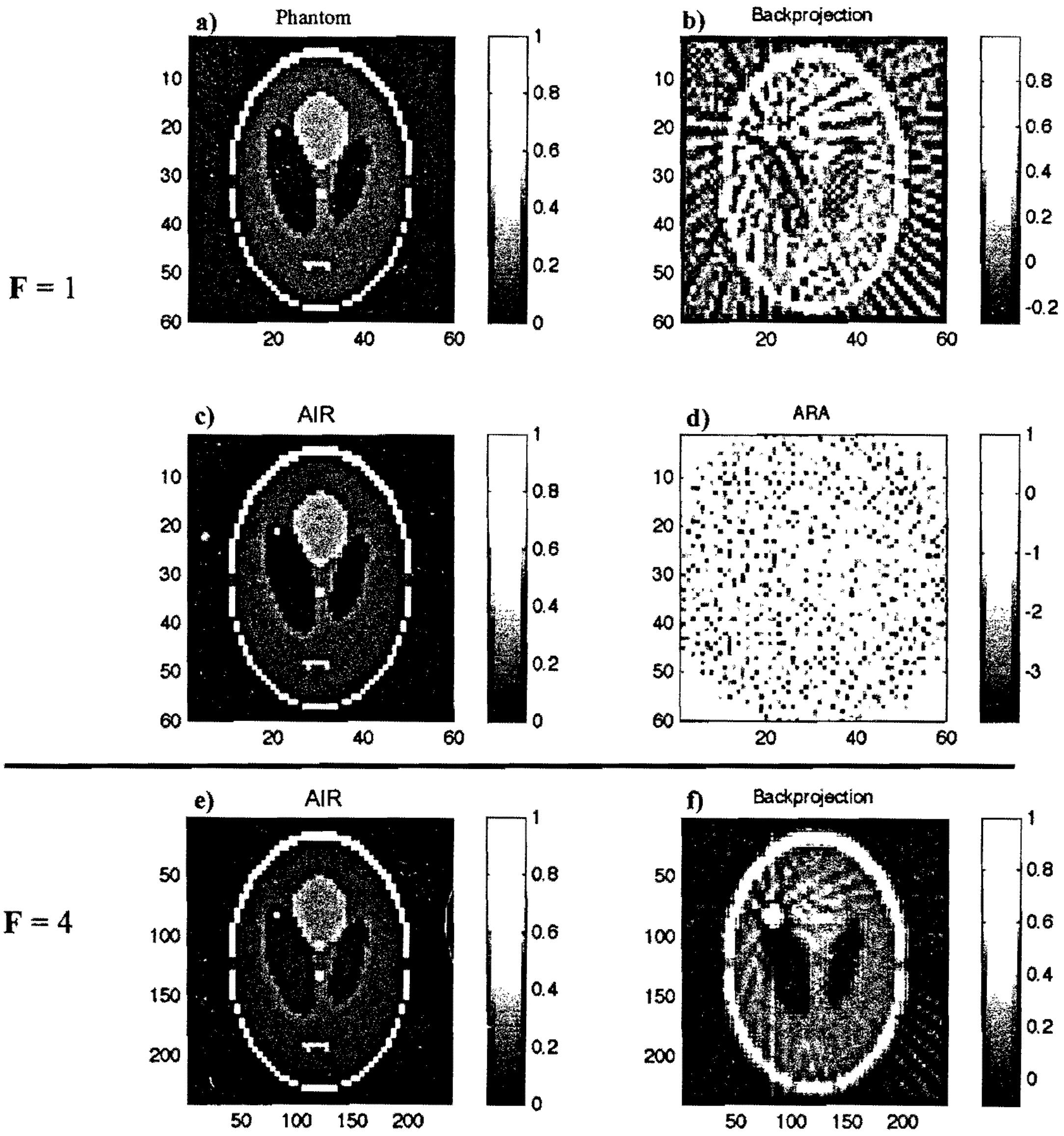


Figure 4.2.5: All images have been rescaled to a maximum value of one in order to display the results. **F=1: a)** The phantom being imaged. **b)** Filtered backprojection introduces ringing artifacts in the presence of sharp density gradients. **c)** AIR with C^{ct}=1, C^{wav}=1.0x10^{-6}, C^{zeroav}=1 produces an exact solution. **d)** ARA cannot solve the underdetermined system. **F=4:** Performing the same simulation only now the phantom pixels and reconstruction pixels are split up into 16 fractions each, in order to interpolate the projection data. Notice that the number of pixels has been multiplied by 16. This does not represent a finer reconstruction resolution. The solution has been mapped on to a higher resolution grid to indicate that we have used 16 fractions per phantom pixel. **e)** AIR with C^{ct}=1, C^{wav}=1.0x10^{-6}, C^{zeroav}=1 produces the same solution as in **c. f)** The filtered backprojection reconstruction is enhanced by the addition of pixel fractions as the backprojected data is averaged from multiple beamlet contributions. Ringing artifacts are still prevalent, and densities are underrepresented.

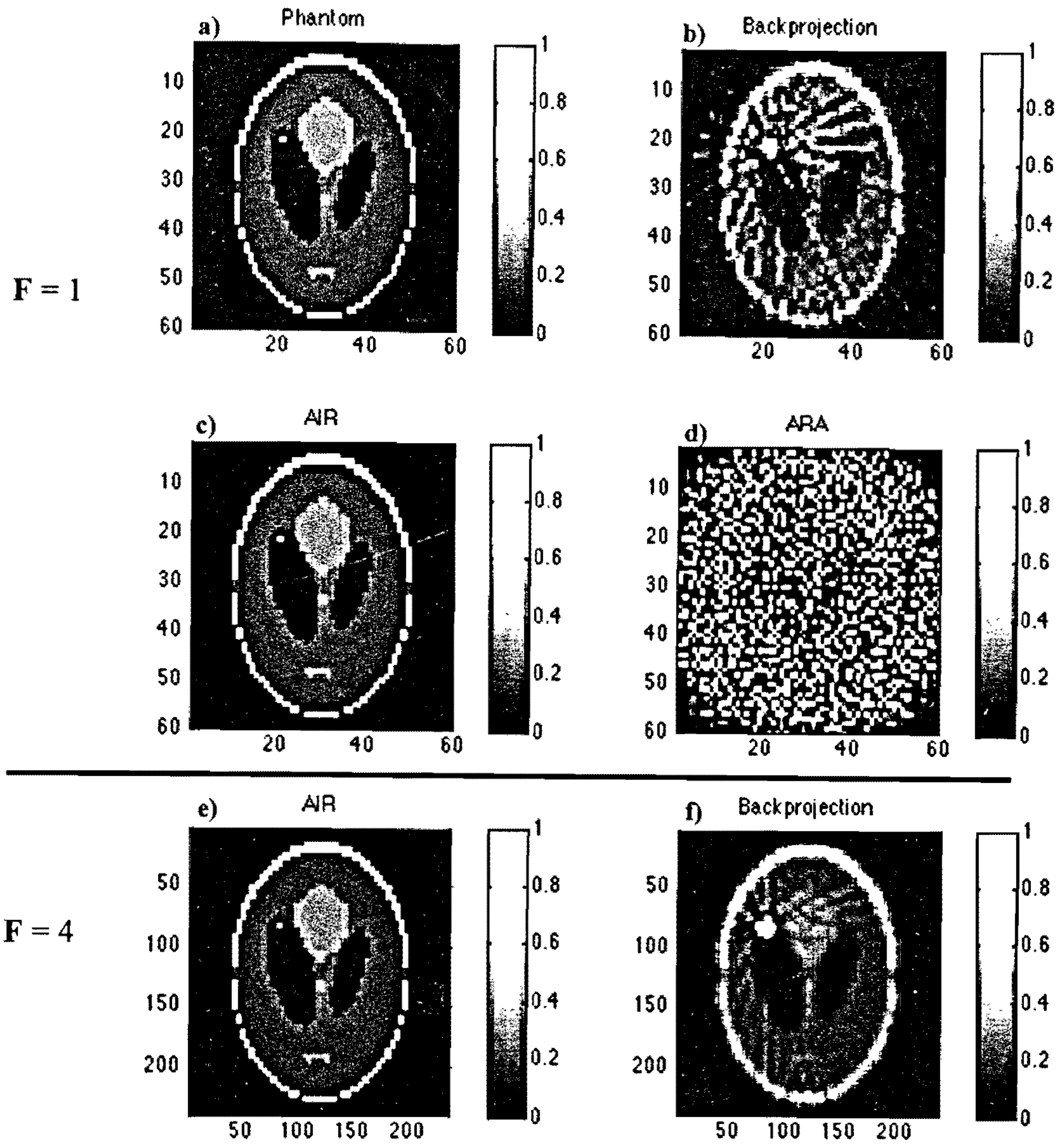


Figure 4.2.5i rescaled:
F=1: a) The phantom being imaged. **b)** Filtered backprojection introduces ringing artifacts in the presence of sharp density gradients. **c)** AIR with $C^{ct}=1$, $C^{wav}=1.0\times10^{-6}$, $C^{zeroav}=1$ produces an exact solution. **d)** ARA cannot solve the underdetermined system.
F=4: Performing the same simulation only now the phantom pixels and reconstruction pixels are split up into 16 fractions each, in order to interpolate the projection data. Notice that the number of pixels has been multiplied by 16. This does not represent a finer reconstruction resolution. The solution has been mapped on to a higher resolution grid to indicate that we have used 16 fractions per phantom pixel. **e)** AIR with $C^{ct}=1$, $C^{wav}=1.0\times10^{-6}$, $C^{zeroav}=1$ produces the same solution as in **c. f)** Filtered backprojection reconstruction is enhanced by the addition of pixel fractions as the backprojected data is averaged from multiple beamlet contributions. Ringing artifacts are still prevalent, and densities are underrepresented.

Test four: Adding noise to the projections

The fourth test examines reconstruction in the presence of noise (**figure 4.2.6** and **4.2.7**). With the addition of noise, comes a fundamentally distinct issue that was not present in the previous tests. While the number of equations lends to a determined system, the projection information is no longer self-consistent, and an exact solution is no longer possible. ARA reconstructs an image with many negative and unrepresentative attenuation values (image **d** of **figure 4.2.6** and **4.2.7**). Filtered backprojection is smoothed out by the addition of pixel fractions (compare image **b** of **figure 4.2.6** with image **b** of **figure 4.2.7**).

AIR with small importance values for the averaging terms results in an image much like the ARA reconstruction (compare images **c** and **d** of **figure 4.2.6**). In this test the averaging terms in AIR begin to take on a more central role (This was certainly the case later, in the more realistic simulations). **Figure 4.2.6** (images **e**, **f** and **g**) highlights the manual choice of importance coefficients, and the solution's sensitivity to those values. It is not clear whether image **f** or **g** is the optimal solution (**figure 4.2.6**). As the importance of the weighted averaging term is increased, contrast seems to be in tradeoff with the overall smoothness of the image. Image **h** depicts a smeared image when too much importance is attributed to the weighted averaging term. Image **h** also highlights the relevance of the *CT* term. While it may use a smaller importance coefficient in more realistic simulations, if the *CT* term's importance is made too small relative to the other terms, the result is image **h**.

AIR, ARA and filtered backprojection images are smoothed and enhanced with the addition of pixel fractions in **figure 4.2.7**. Again, images **c**, **e** and **f** illustrate the difficulty in defining the term optimal in situation when an exact solution is not possible.

The fourth test was conducted with the following parameters:

P = 60; **F** = 1 or 4; **D** = 60; **R** = 60; **A** = (0 | 180 | 180); **Pnoise** = 10%

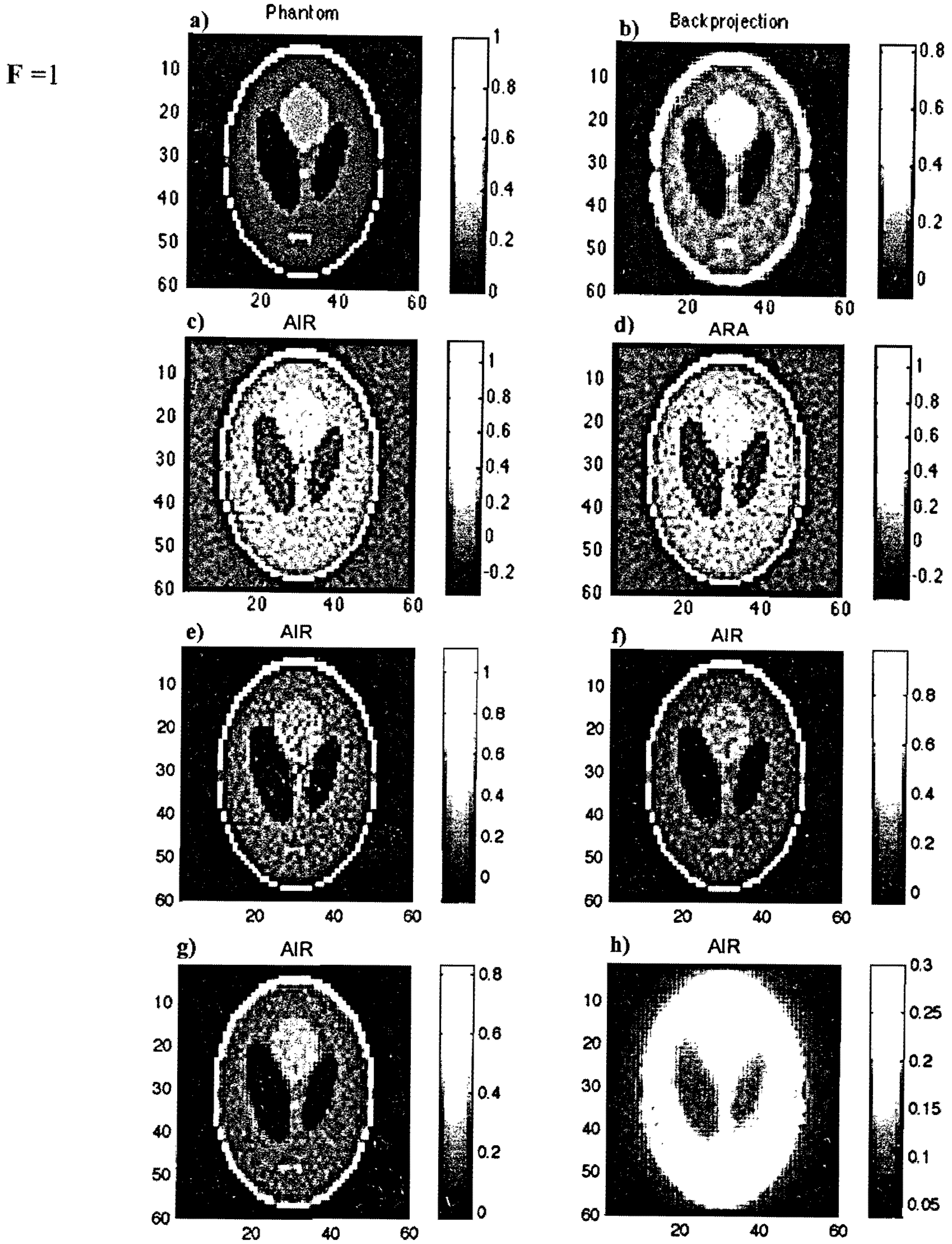


Figure 4.2.6: **a)** The phantom being imaged. **b)** Filtered backprojection seems to average out random noise. **c)** AIR with $C^{ct}=1$, $C^{wav}=1.0x10^{-6}$, $C^{zeroav}=1$. **d)** ARA cannot reconstruct the image due to the inconsistencies in the matrix equations. **e)** AIR with $C^{ct}=1$, $C^{wav}=100$, $C^{zeroav}=1000$. **f)** AIR with $C^{ct}=1$, $C^{wav}=1.0x10^{4}$, $C^{zeroav}=1x10^{6}$. **g)** AIR with $C^{ct}=1$, $C^{wav}=3.0x10^{4}$, $C^{zeroav}=1x10^{6}$. **h)** AIR with $C^{ct}=1$, $C^{wav}=1.0x10^{6}$, $C^{zeroav}=1x10^{6}$. These images depict the sensitivity to the choice of importance coefficients. The zero averaging term must be given more importance than the weighted averaging.

F =1

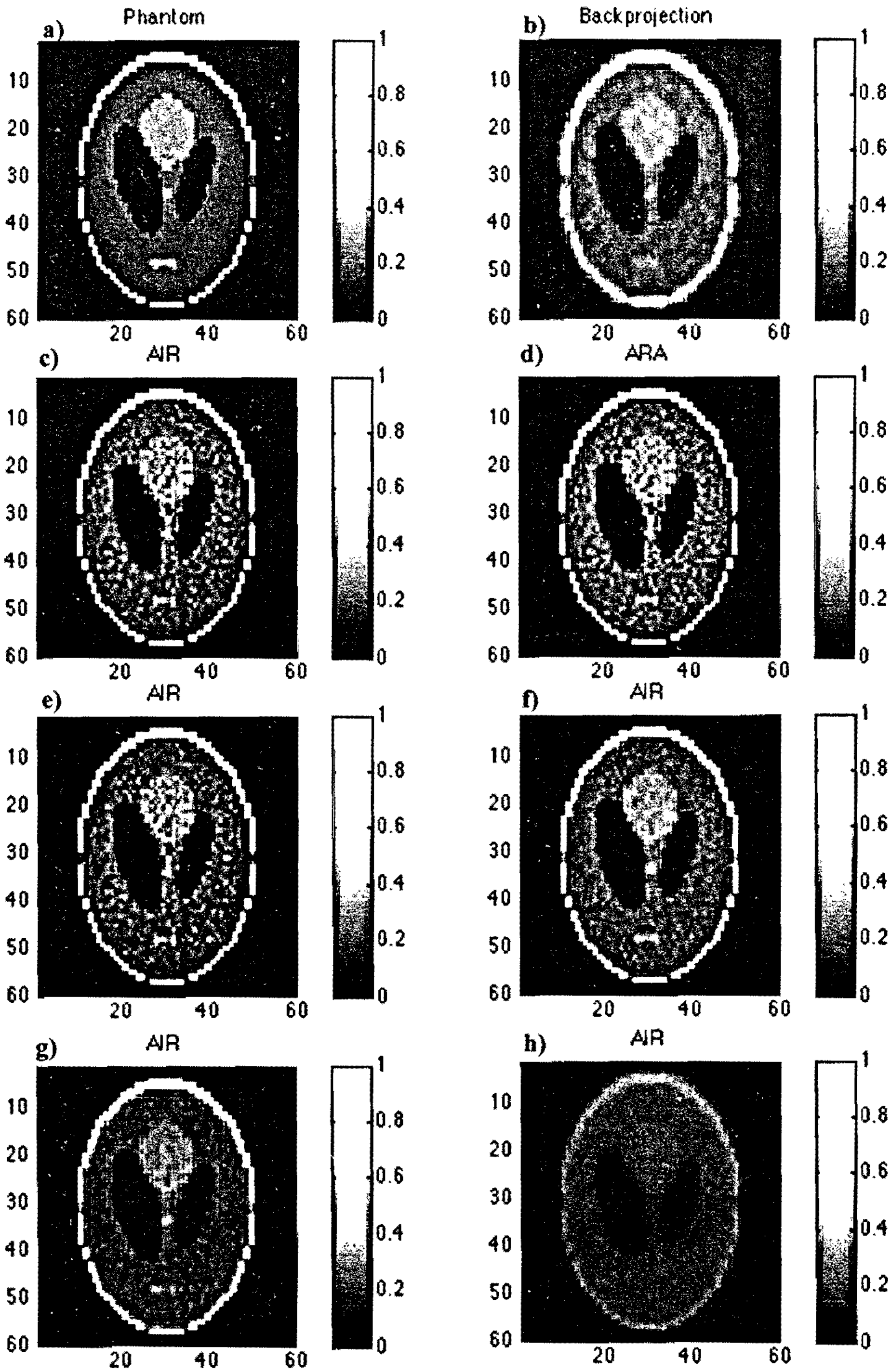


Figure 4.2.6i rescaled: a) The phantom being imaged. **b)** Filtered backprojection seems to average out random noise. **c)** AIR with $C^{ct}=1$, $C^{wav}=1.0\text{x}10^{-6}$, $C^{zeroav}=1$. **d)** ARA cannot reconstruct the image due to the inconsistencies in the matrix equations. **e)** AIR with $C^{ct}=1$, $C^{wav}=100$, $C^{zeroav}=1000$. **f)** AIR with $C^{ct}=1$, $C^{wav}=1.0\text{x}10^{4}$, $C^{zeroav}=1\text{x}10^{6}$. **g)** AIR with $C^{ct}=1$, $C^{wav}=3.0\text{x}10^{4}$, $C^{zeroav}=1\text{x}10^{6}$. **h)** AIR with $C^{ct}=1$, $C^{wav}=1.0\text{x}10^{6}$, $C^{zeroav}=1\text{x}10^{6}$. These images depict the sensitivity to the choice of importance coefficients. The zero averaging term must be given more importance than the weighted averaging.

F=4

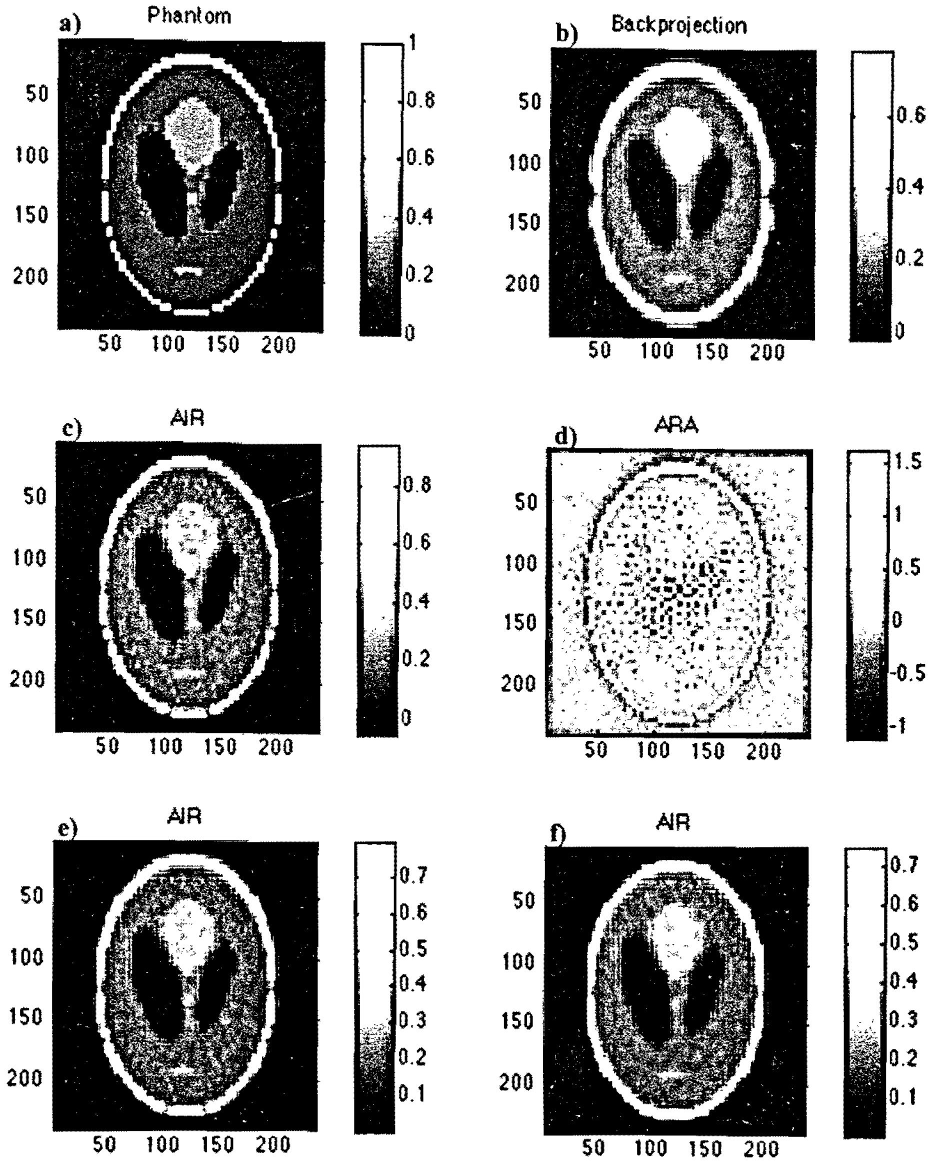


Figure 4.2.7:
a) The phantom being imaged. **b)** Filtered backprojection reconstruction. **c)** AIR with $C^{ct}=1$, $C^{wav}=2.0\text{x}10^4$, $C^{zeroav}=1\text{x}10^6$. **d)** ARA reconstruction. **e)** AIR with $C^{ct}=1$, $C^{wav}=5.0\text{x}10^4$, $C^{zeroav}=1\text{x}10^6$. **f)** AIR with $C^{ct}=1$, $C^{wav}=8.0\text{x}10^4$, $C^{zeroav}=8\text{x}10^6$.

F=4

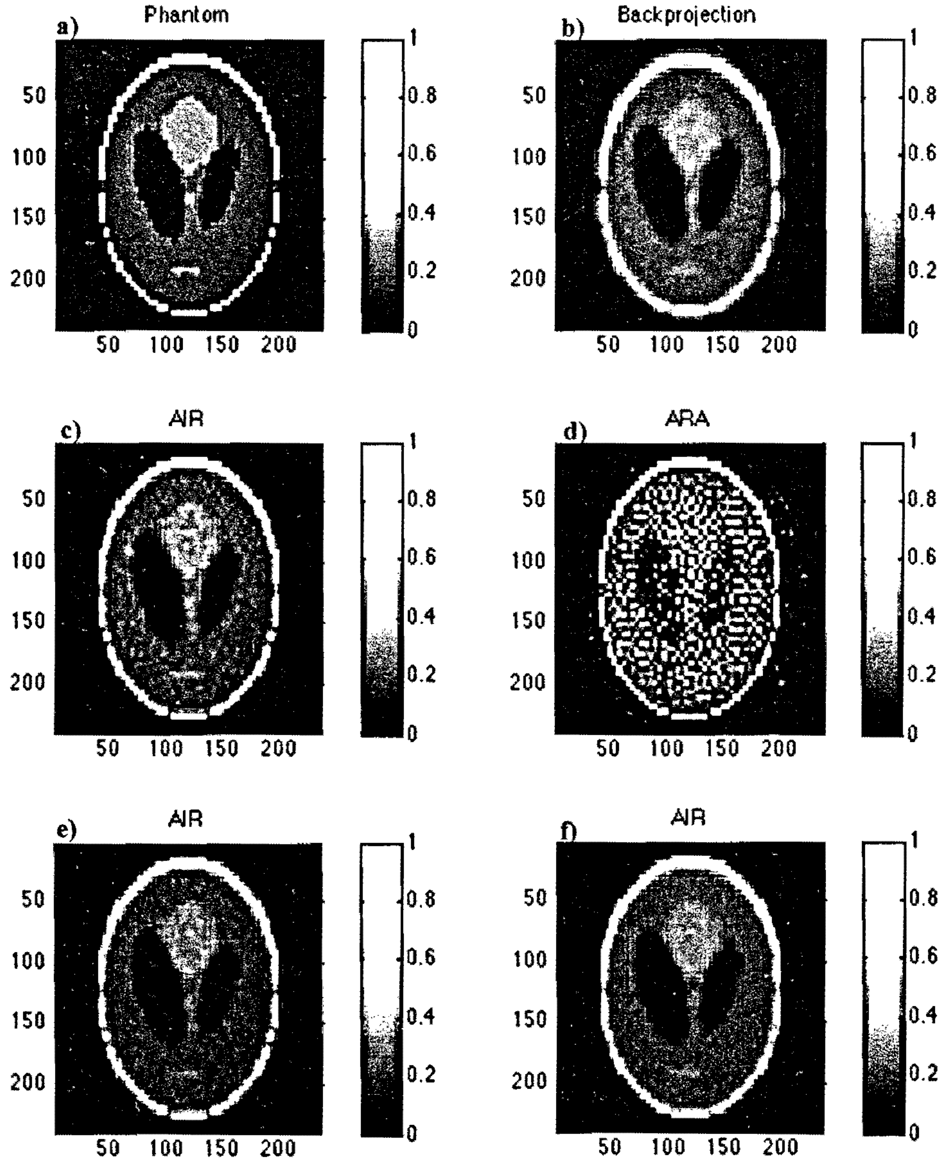


Figure 4.2.7i rescaled:
a) The phantom being imaged. **b)** Filtered backprojection reconstruction. **c)** AIR with $C^{ct}=1$, $C^{wav}=2.0\text{x}10^4$, $C^{zeroav}=1\text{x}10^6$. **d)** ARA reconstruction. **e)** AIR with $C^{ct}=1$, $C^{wav}=5.0\text{x}10^4$, $C^{zeroav}=1\text{x}10^6$. **f)** AIR with $C^{ct}=1$, $C^{wav}=8.0\text{x}10^4$, $C^{zeroav}=8\text{x}10^6$.

Test five: High-resolution phantom reconstructed on a lower resolution grid

All the tests of AIR and PAIR from this point on were conducted using the more complex simulation model that was developed. By imaging a 540 x 540 resolution phantom and attempting to reconstruct the image on a 60 x 60 grid, there are many inconsistent projections. Inconsistent - meaning that many beamlets may pass through the same reconstruction pixel area, but will sample unique phantom pixels within that region, leading to inconsistent measurements. This situation has some parallels to using the homogeneous pixel fractions, but with simulated noise. It is for this reason that the last test preceded this one.

The fifth test also includes reconstruction using the inversion method. As mentioned in the section describing the inversion method, it was only implemented in cases with a large number of phantom pixels per reconstruction pixel, and an equal number of detectors in the array (60) compared to the reconstruction resolution (60x60). In the initial test, the inversion method could not calculate the altered *h* weights for the beam angles at 45 and 135 degrees. The reason for this will be discussed in a separate section devoted to an in depth discussion of the inversion method. Nevertheless, in this test those two angles were removed from the simulation in order to compare the results.

Image **b** shows that even though the projections contain inconsistent information, filtered backprojection is smoothed by the fractioned contributions. The contrast is degraded and the image is blurred.

Images **c** and **d** are the results of AIR using the original method of representing pixels. AIR requires the averaging terms to produce an optimal image. Even though the importance coefficients of both the weighted averaging and zero averaging terms are much large than the *CT* term, the *CT* term is still a fundamental component to AIR. Without it, image contrast would be smeared (as was depicted in **figure 4.2.6** image **h**). With the proper choice of importance coefficients, AIR produces image **d**. This image has better contrast and sharpness compared to the filtered backprojection's reconstruction, but there seems to be a bit of noise. As was mentioned before, by increasing the weighted averaging term a trade off is made between contrast and overall smoothness.

It should be noted that the ARA reconstruction was not included in these figures. ARA reconstruction in this simulation results in an unrepresentative image much like AIR without the degeneracy breaking average terms (image **c**).

Image **e** shows results of AIR with the inversion method of representing pixels, and using only the *CT* term. The initial result seems much more representative than the original method producing image **c**. Although the image is far from perfect, the initial result gives credit to the theory behind the inversion method. The issue is that the averaging terms in AIR are formulated specifically for the original method of representing pixels. By imposing the inversion method on the averaging terms, the result is a non-physical solution (that has not been shown). Image **f** is the result of attempting to use the inversion method of representing pixels for the *CT* term function, but the original method for the averaging terms. Though the image is somewhat improved by this strategy, the result is far from optimal.

The fifth test included the inversion method and was conducted with the following parameters: **P** = 540; **F** = 1; **D** = 60; **R** = 60; **A** = (0 | 180 | 178) (Initially 180 angles were desired, but when using the inversion method, the inversion at 45 and 135 degrees resulted in an unusable singular matrix. To avoid this the projection information from those two angles was removed from the simulation).

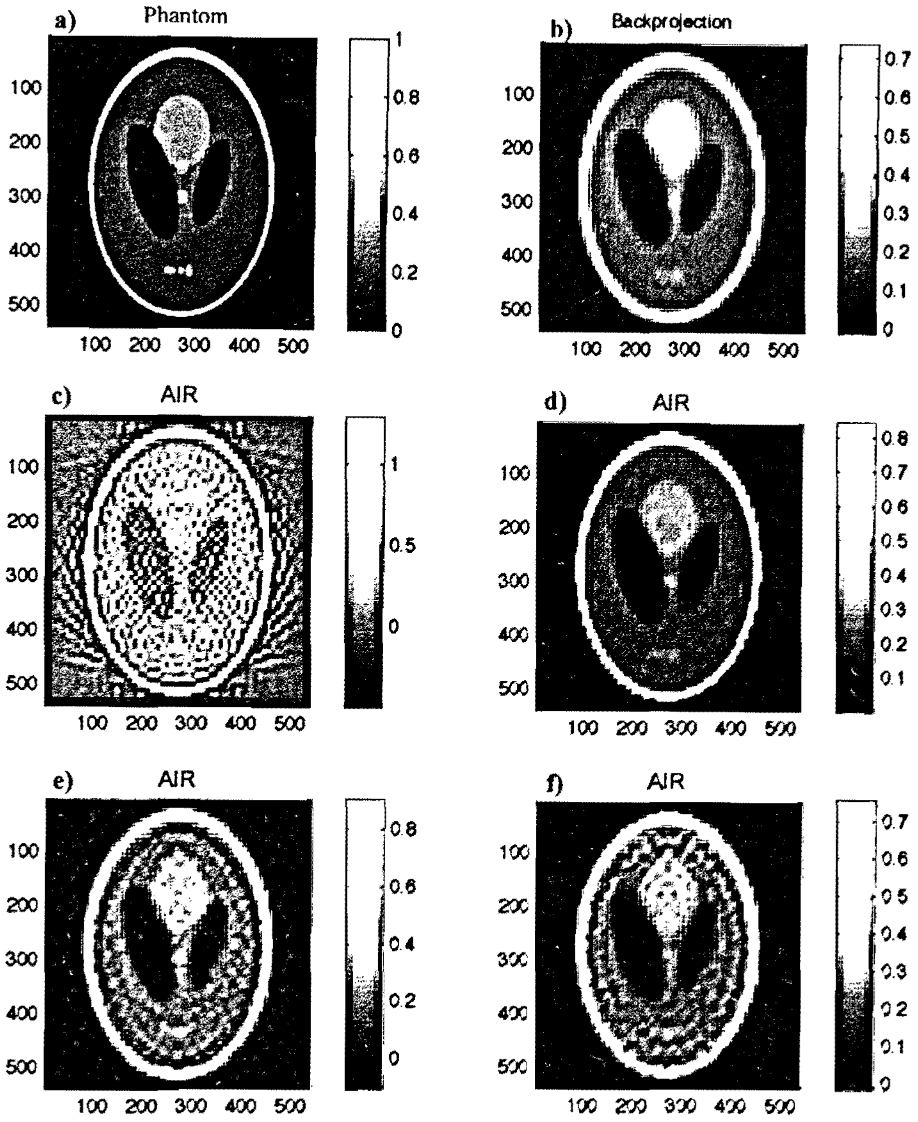


Figure 4.2.8: a) The phantom being imaged. **b)** Filtered backprojection reconstruction. The reconstruction resolution is 60x60 but it has been displayed on a higher resolution grid. **c)** AIR with C^{ct}=1. **d)** AIR with C^{ct}=1, C^{wav}=5.0x10^4, C^{zeroav}=1.0x10^7. These importance coefficients produce an optimal image. **e)** AIR with C^{ct}=1, using the inversion method to represent pixels. The initial result seems much more representative than the original method producing image **c**. **f)** AIR with C^{ct}=1, C^{wav}=1.0x10^5, C^{zeroav}=5.0x10^7, using the inversion method to represent pixels for the CT term, and the original method to represent pixels for the weighted averaging and zero averaging terms. Employing the inversion method for the weighted averaging and zero averaging terms results in unrepresentative images (not shown here).

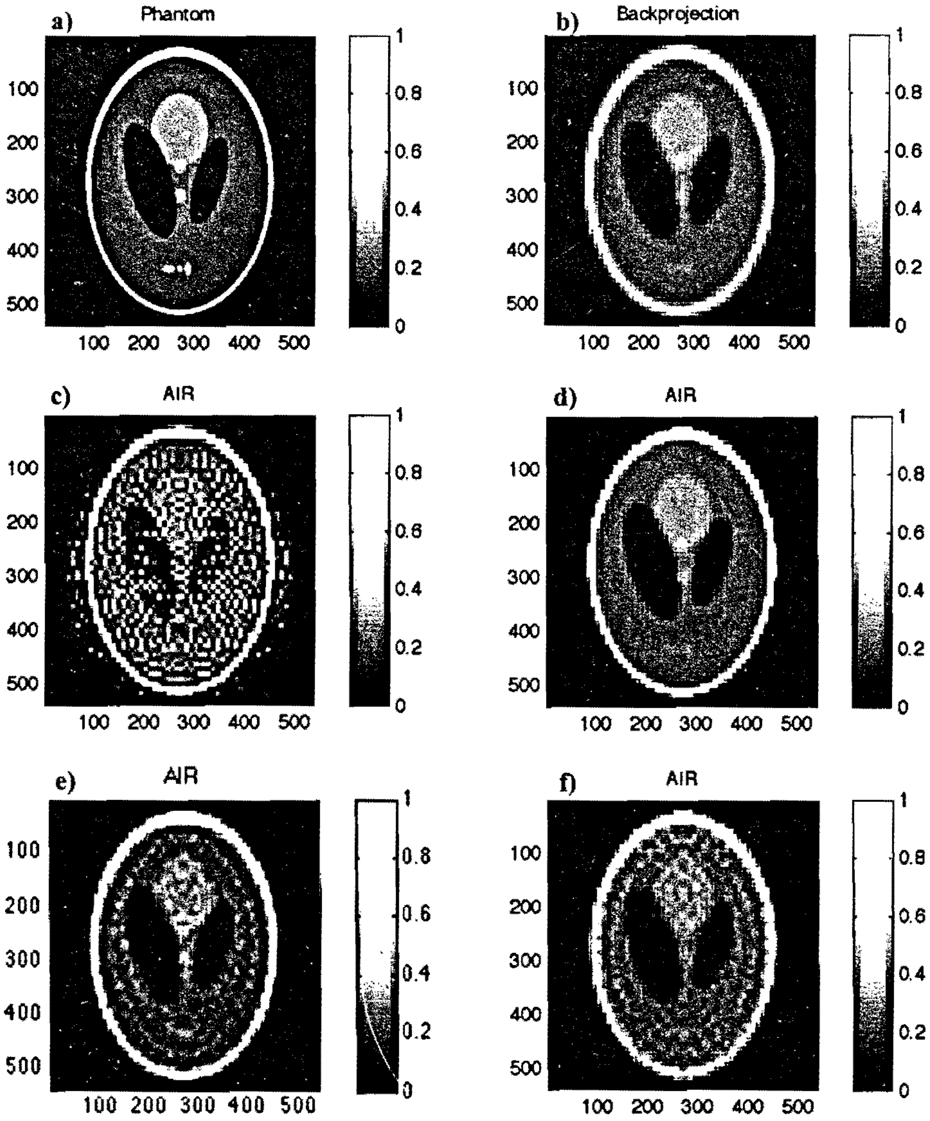


Figure 4.2.8i rescaled: a) The phantom being imaged. **b)** Filtered backprojection reconstruction. The reconstruction resolution is 60x60 but it has been displayed on a higher resolution grid. **c)** AIR with $C^{ct}=1$. **d)** AIR with $C^{ct}=1$, $C^{wav}=5.0x10^4$, $C^{zeroav}=1.0x10^7$. These importance coefficients produce an optimal image. **e)** AIR with $C^{ct}=1$, using the inversion method to represent pixels. The initial result seems much more representative than the original method producing image **c. f)** AIR with $C^{ct}=1$, $C^{wav}=1.0x10^5$, $C^{zeroav}=5.0x10^7$, using the inversion method to represent pixels for the CT term, and the original method to represent pixels for the weighted averaging and zero averaging terms. Employing the inversion method for the weighted averaging and zero averaging terms results in unrepresentative images (not shown here).

Test six and seven: Reconstruction resolution set higher than detector resolution

The sixth and seventh tests (**figures 4.2.9** and **4.2.10**) represent the special case of reconstruction resolution beyond the seeming limits dictated by the number of detectors. In the sixth test (**Figure 4.2.9**), the reconstruction resolution is set equal to the phantom resolution, but the detector width is twice the size of the phantom pixel. Though the reconstruction resolution was set to 120x120, the filtered backprojection algorithm is inherently constrained by the width of the detectors and thus reconstructs an image that appears to conform to a 60x60 grid (image **b**). Furthermore, the image appears blurred and contains subtle streaking artifacts. ARA cannot be included in all the tests that follow because the matrix equation exceeds computational ability (see section 3.8).

Normally, this situation would present AIR with a simple underdetermined system comprising consistent projection information, and an exact solution could be found. However, due to the relatively high reconstruction resolution, PAIR is used to reconstruct the image. For each PAIR orientation, the reconstruction pixels may contain inhomogeneous phantom pixels leading to an inconsistent set of projections. In the seventh test (**Figure 4.2.10**), by increasing the imaged phantom resolution, the simulation becomes less ideal by incorporating more inhomogeneities in a reconstruction pixel.

The four images of **c** (**figures 4.2.10** and **4.2.9**) are the four independently oriented PAIR images using the *CT* term, the weighted averaging term, the zero averaging term, and the local averaging term. The local averaging term contributes to the overall smoothness of the image, but a tradeoff exists between smoothness and contrast.

The final AIR result is depicted in **d**. This image is obtained by first mapping each PAIR solution onto a 120x120 grid, and then averaging all four images onto the final AIR grid. This averaging process itself smoothes out random fluctuations in the image. Though the result is not exact, it represents the true function with more accuracy than the filtered backprojection image. Interestingly, both the AIR and filtered backprojection images are enhanced in the simulation using a higher phantom resolution (**Figure 4.2.10**).

The sixth test included PAIR and was conducted with the following parameters:

P = 120; **F** = 1; **D** = 60; **R** = 120 (each PAIR solution used a 60x60 resolution; **A**=(0 | 180 | 180).

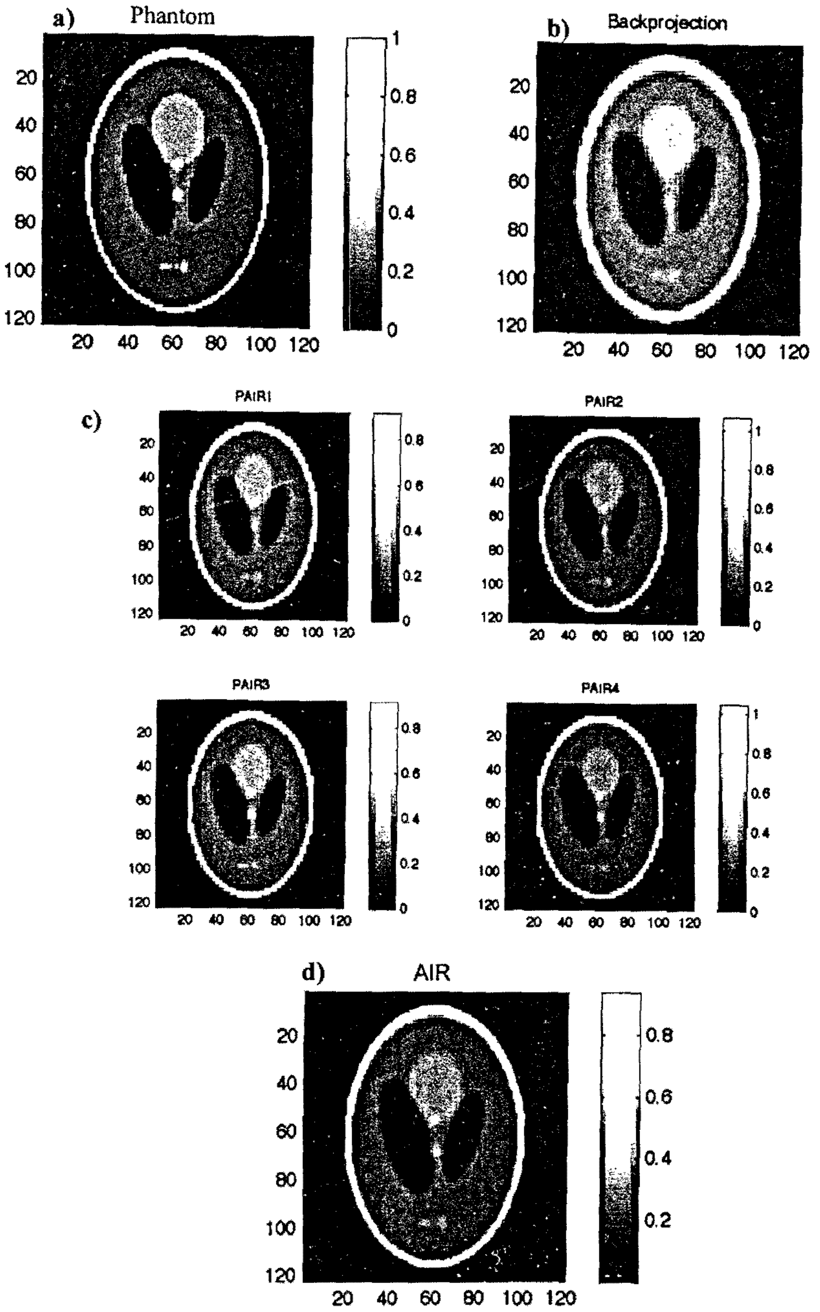


Figure 4.2.9:
a) The phantom being imaged. **b)** Filtered backprojection reconstruction on a 120x120 grid has blurring and slight streaking artifacts. **c)** PAIR with C^{ct}=1, C^{wav}=1.0x10^4, C^{zeroav}=1.0x10^7 C^{local}=100. Each PAIR solution uses a reconstruction grid of 60x60. The images are displayed on a higher resolution grid. **d)** The final AIR image results from averaging four 60x60 PAIR solutions onto a 120x120 grid. The result is of noticeably higher quality than the backprojected image. ARA is not included because the system of equations exceeds computational ability.

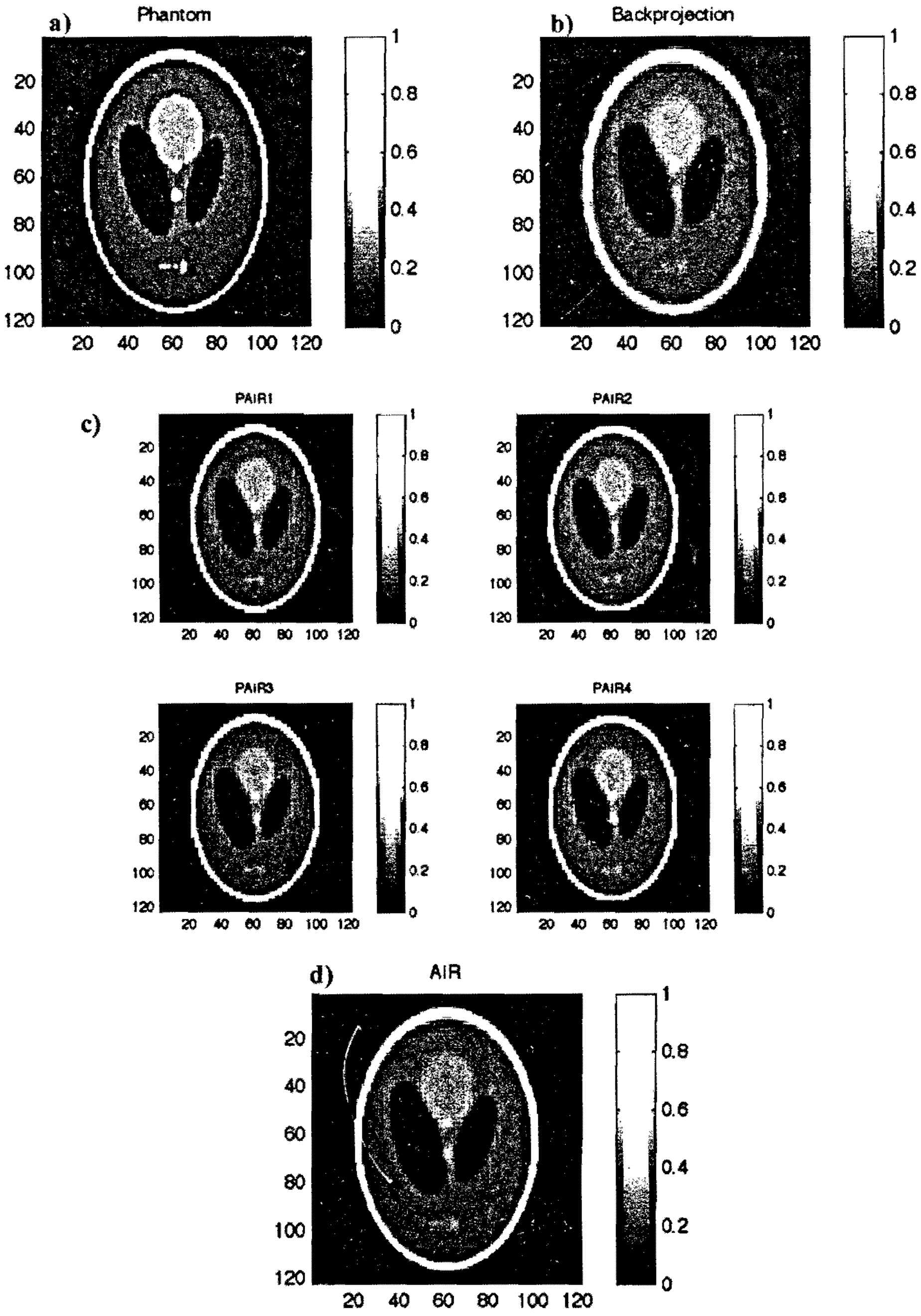


Figure 4.2.9i rescaled:
a) The phantom being imaged. **b)** Filtered backprojection reconstruction on a 120x120 grid has blurring and slight streaking artifacts. **c)** PAIR with $\mathbf{C^{ct}}$=1, $\mathbf{C^{wav}}$=1.0x10^4, $\mathbf{C^{zeroav}}$=1.0x10^7 $\mathbf{C^{local}}$=100. Each PAIR solution uses a reconstruction grid of 60x60. The images are displayed on a higher resolution grid. **d)** The final AIR image results from averaging four 60x60 PAIR solutions onto a 120x120 grid. The result is of noticeably higher quality than the backprojected image. ARA is not included because the system of equations exceeds computational ability.

The seventh test included PAIR and was conducted with the following parameters:

P = 240; **F** = 1; **D** = 60; **R** = 120 (each PAIR solution used a 60x60 resolution; **A**=(0 | 180 | 180).

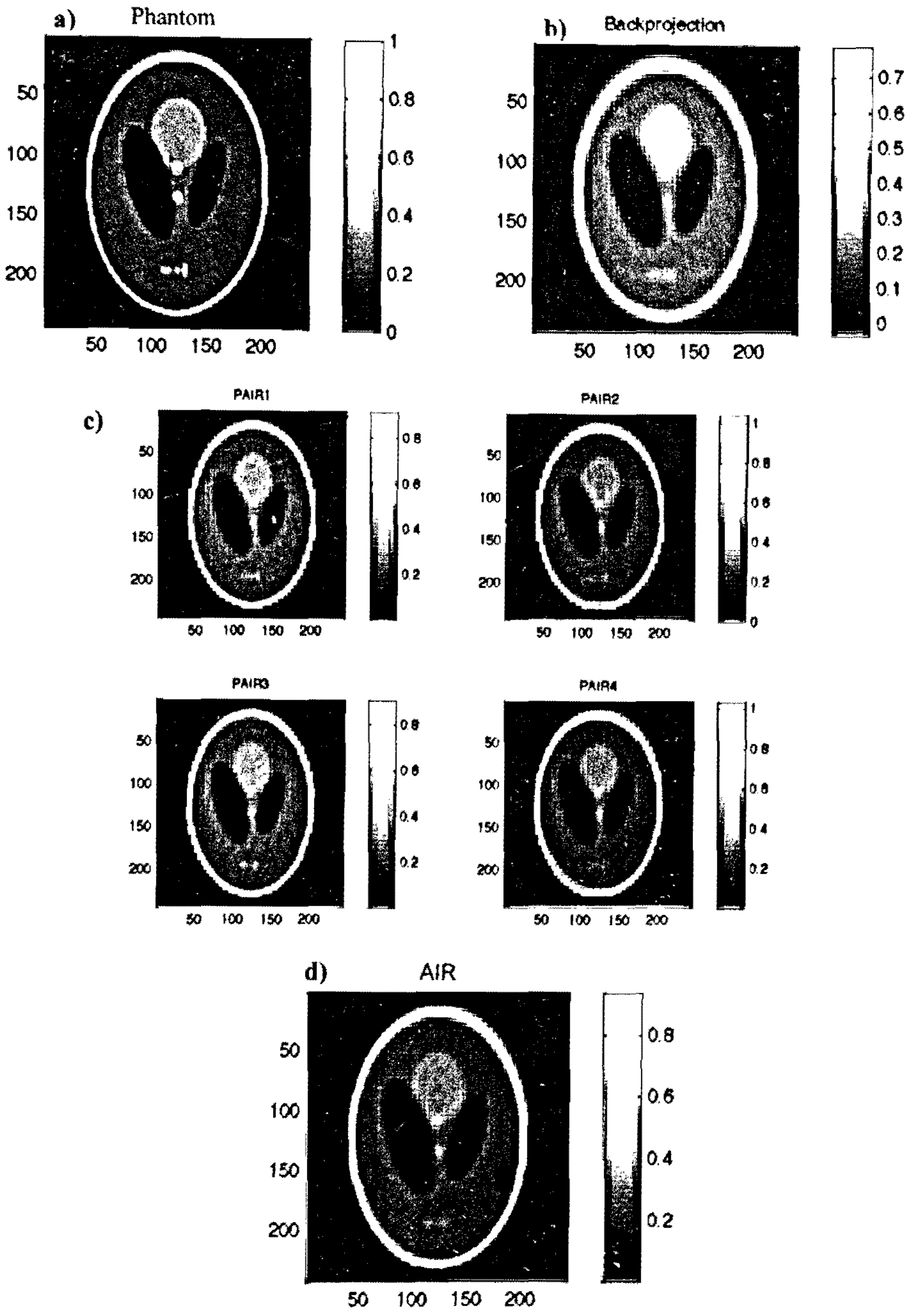


Figure 4.2.10:
a) The phantom being imaged. **b)** Filtered backprojection reconstruction on a 120x120 grid. The image is blurred. **c)** PAIR with C^{ct}=1, C^{wav}=1.0x10^4, C^{zeroav}=1.0x10^7 C^{local}=100. Each PAIR solution uses a reconstruction grid of 60x60. The images are displayed on a higher resolution grid. **d)** The final AIR image results from averaging four 60x60 PAIR solutions onto a 120x120 grid. The result is of noticeably higher quality than the backprojected image. ARA is not included because the system of equations exceeds computational ability.

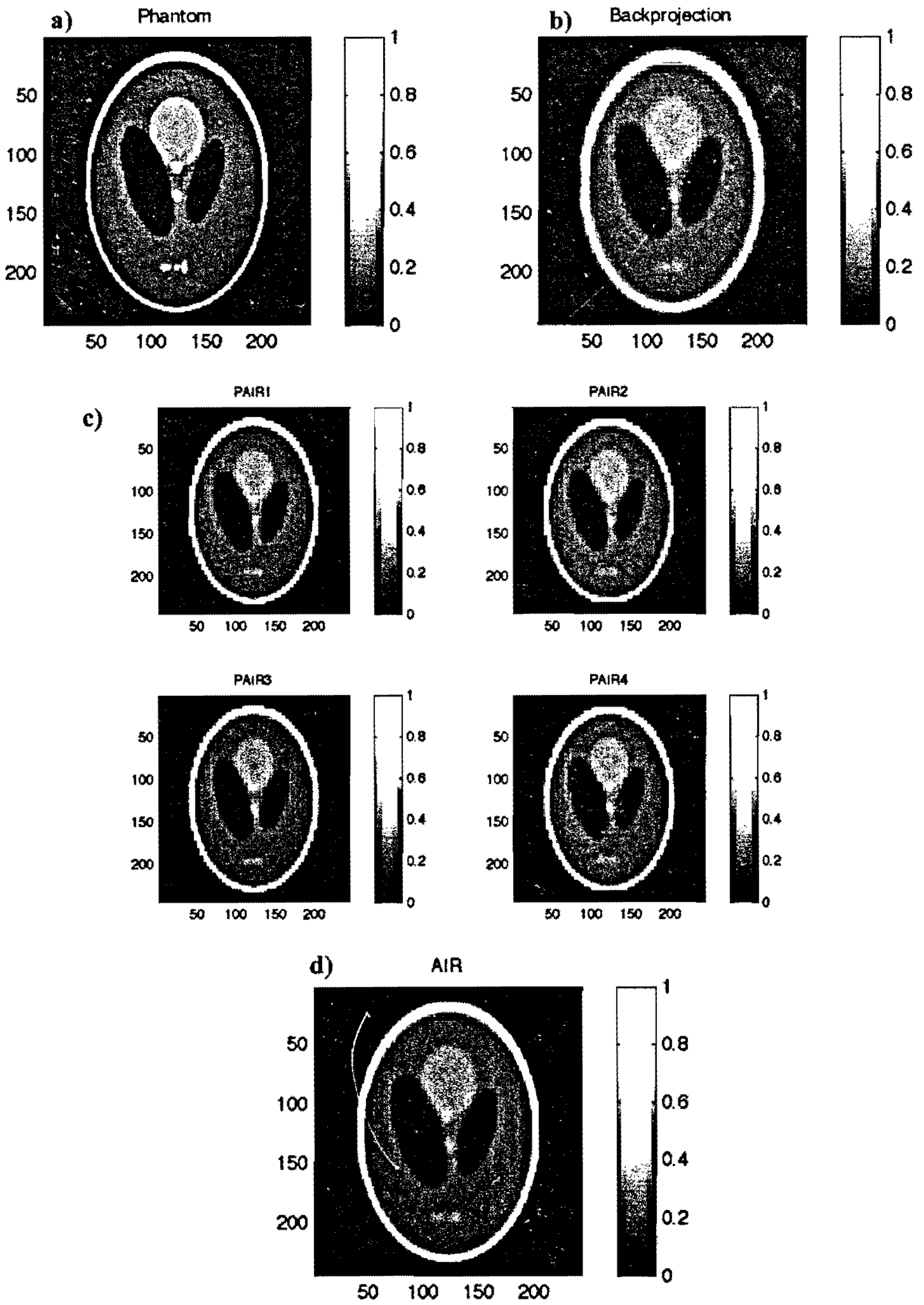


Figure 4.2.10i rescaled:
a) The phantom being imaged. **b)** Filtered backprojection reconstruction on a 120x120 grid. The image is blurred. **c)** PAIR with $C^{ct}=1$, $C^{wav}=1.0x10^4$, $C^{zeroav}=1.0x10^7$ $C^{local}=100$. Each PAIR solution uses a reconstruction grid of 60x60. The images are displayed on a higher resolution grid. **d)** The final AIR image results from averaging four 60x60 PAIR solutions onto a 120x120 grid. The result is of noticeably higher quality than the backprojected image. ARA is not included because the system of equations exceeds computational ability.

Test eight: Limited beam angle views in the less ideal simulation model

The eighth test (**figure 4.2.11**) examines the effect of minimal angle views in the less ideal simulation model (a more realistic parallel to the first set of tests). The filtered backprojection is under sampled in the frequency domain and therefore has pronounced streaking artifacts (image **b**). Notice the representation of non-physical negative attenuation values in the image.

The PAIR images avoid these streaks mainly due to the large value assigned to the importance coefficient of the zero averaging term (image **c**). The final AIR image is smooth and avoids negative values, and large artifacts, but the contrast is degraded (image **d**). A range of results can be obtained with higher contrast but more overall image noise.

The eighth test included PAIR and was conducted with the following parameters:

P = 240; **F** = 2; **D** = 120; **R** = 120 (each PAIR solution used 60x60 resolution; **A** = (0 | 180 | 22).

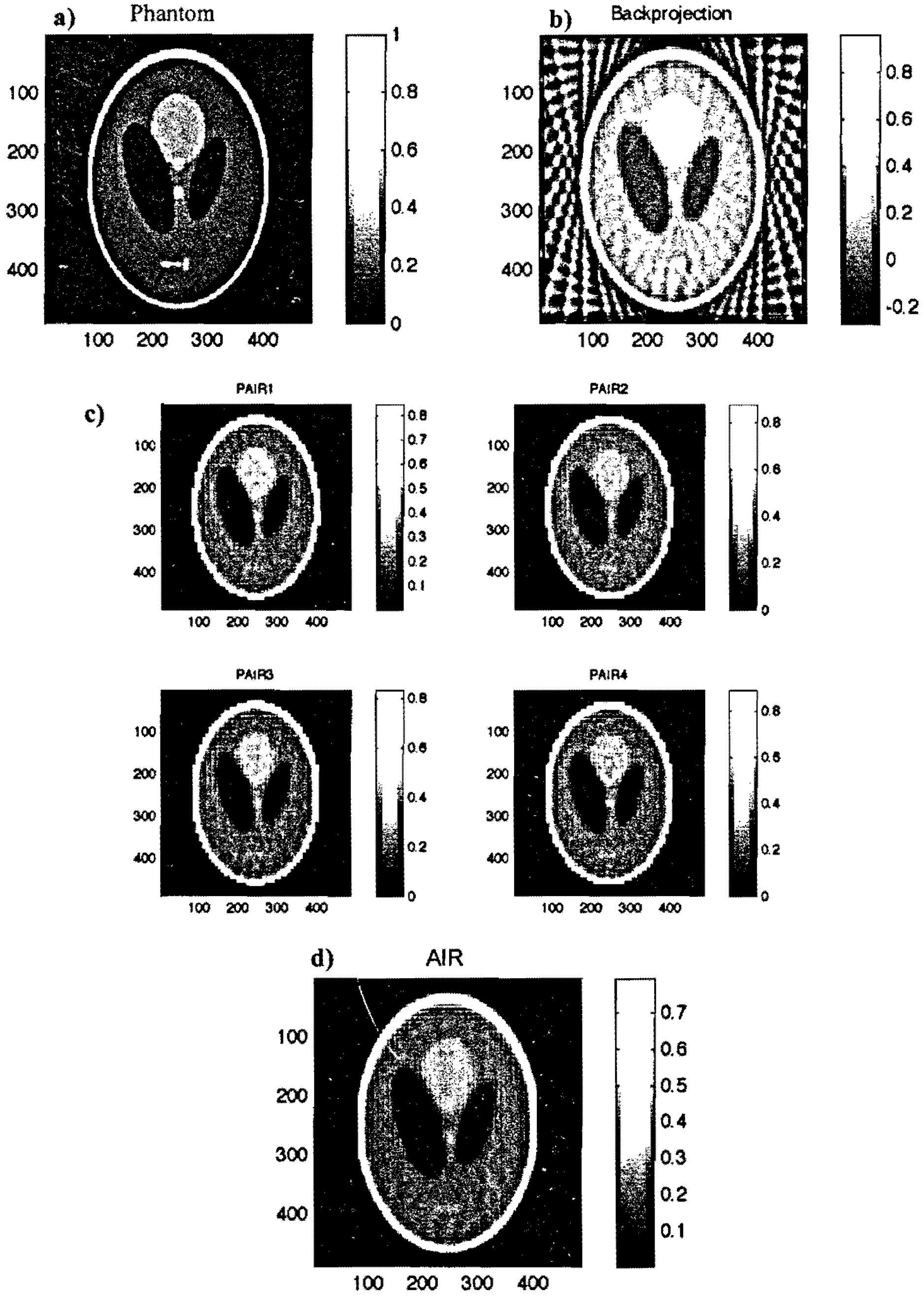


Figure 4.2.11:
a) The phantom being imaged. **b)** Filtered backprojection reconstruction on a 120x120 grid has streaking artifacts. **c)** PAIR with $\mathbf{C}^{ct}$=1, $\mathbf{C}^{wav}$=1.8x10^{4}, $\mathbf{C}^{zeroav}$=2.0x10^{7}, $\mathbf{C}^{local}$=1. Each PAIR solution uses a reconstruction grid of 60x60. The images are displayed on a higher resolution grid. **d)** The final AIR image results from averaging four 60x60 PAIR solutions onto a 120x120 grid. The result is of noticeably higher quality than the backprojected image. ARA is not included because the system of equations exceeds computational ability.

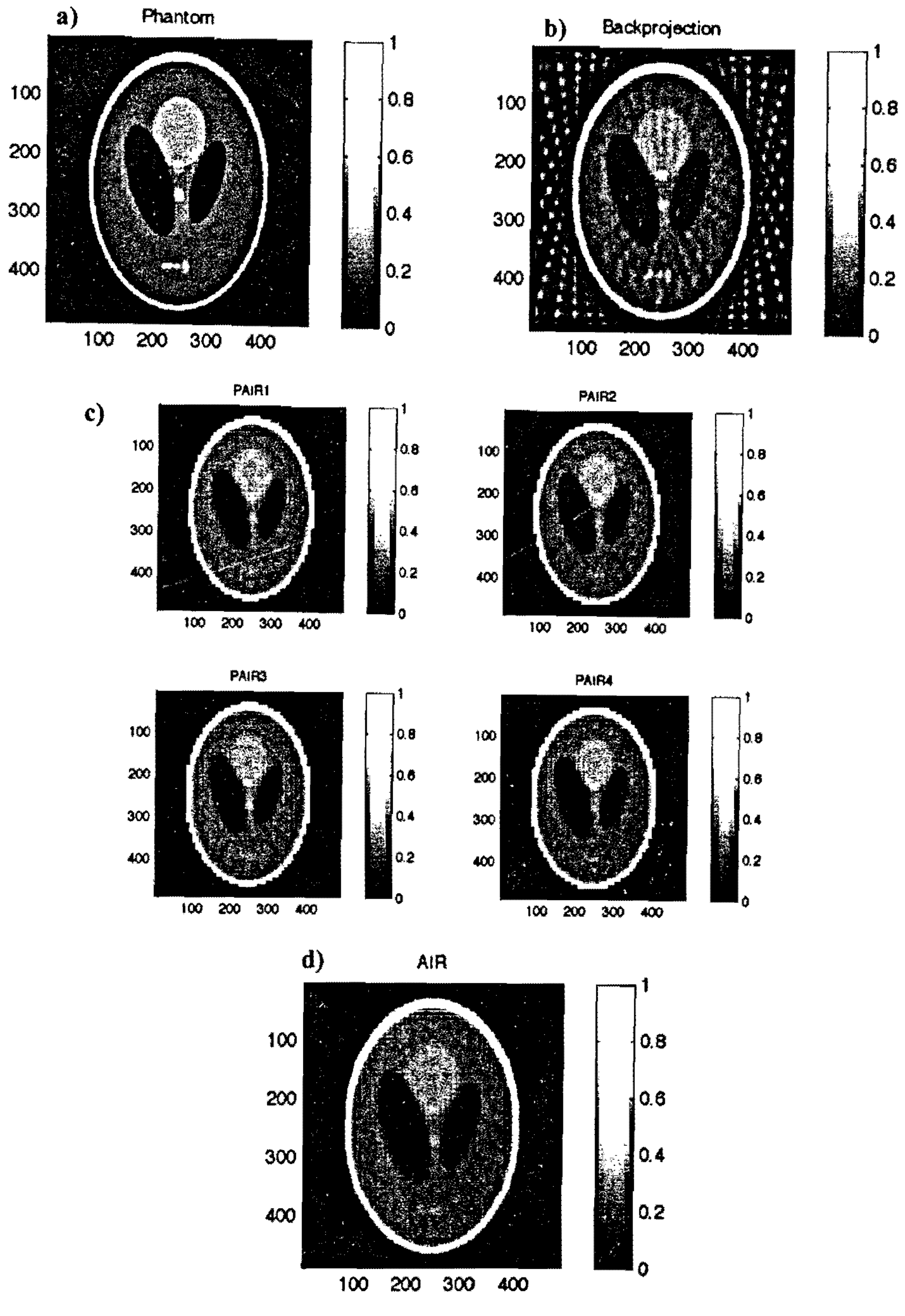


Figure 4.2.11i rescaled:
a) The phantom being imaged. **b)** Filtered backprojection reconstruction on a 120x120 grid has streaking artifacts. **c)** PAIR with $C^{ct}=1$, $C^{wav}=1.8\text{x}10^4$, $C^{zeroav}=2.0\text{x}10^7$, $C^{local}=1$. Each PAIR solution uses a reconstruction grid of 60x60. The images are displayed on a higher resolution grid. **d)** The final AIR image results from averaging four 60x60 PAIR solutions onto a 120x120 grid. ARA is not included because the system of equations exceeds computational ability.

Test nine: Metal gradient in the less ideal simulation model

The ninth test (**figure 4.2.12**) revisits the sharp gradient scenario in the less ideal simulation model. All the images have been rescaled to a max value of one for visualization. The filtered backprojection image depicts aliasing or ringing artifacts, and has a significant number of negative values (image **b**).

The PAIR results seem to smooth out the sharp gradient but avoids the artifacts present in backprojection, as well as avoiding negative attenuation values (image **c**). Though it is not pictured here, assigning a large importance to the weighted averaging term results in streaks very similar to filtered backprojection image. The similarities can be seen faintly in the final AIR solution. The final AIR image smoothes out general noise that is present in the PAIR images though the contrast is degraded (image **d**).

The ninth test included PAIR and was conducted with the following parameters:

P = 240; **F** = 1; **D** = 120; **R** = 120; **M**= (8x8), μ=30; **A** = (0 | 180 | 180).

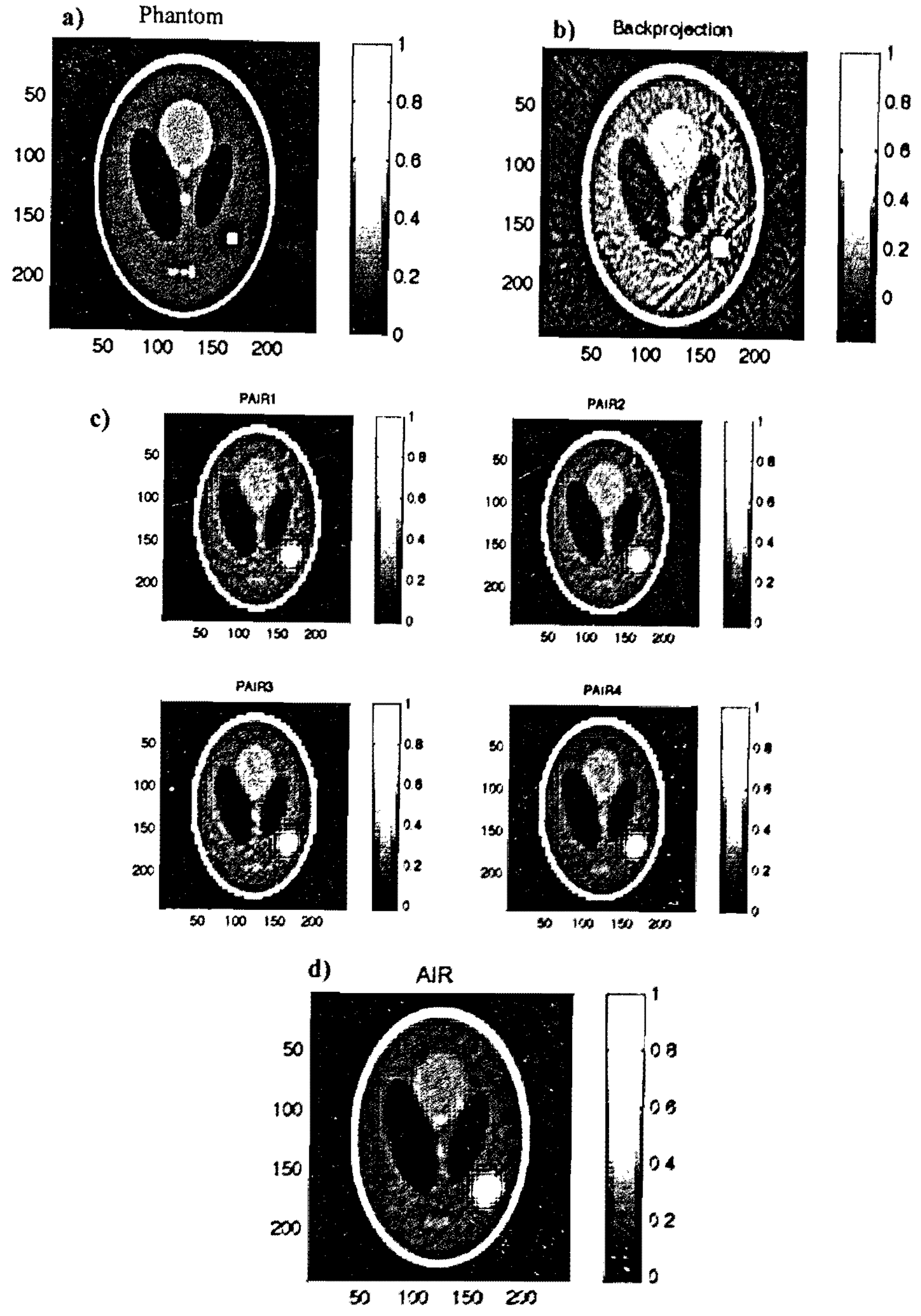


Figure 4.2.12:
a) The phantom being imaged. **b)** Filtered backprojection reconstruction has streaking artifacts. **c)** PAIR with $C^{ct}=1$, $C^{wav}=7.0x10^4$, $C^{zerowav}=1.0x10^3$, $C^{local}=1$. Each PAIR solution uses a reconstruction grid of 60x60. The images are displayed on a higher resolution grid. **d)** The final AIR image results from averaging four 60x60 PAIR solutions onto a 120x120 grid. The result is of noticeably higher quality than the backprojected image. ARA is not included because the system of equations exceeds computational ability.

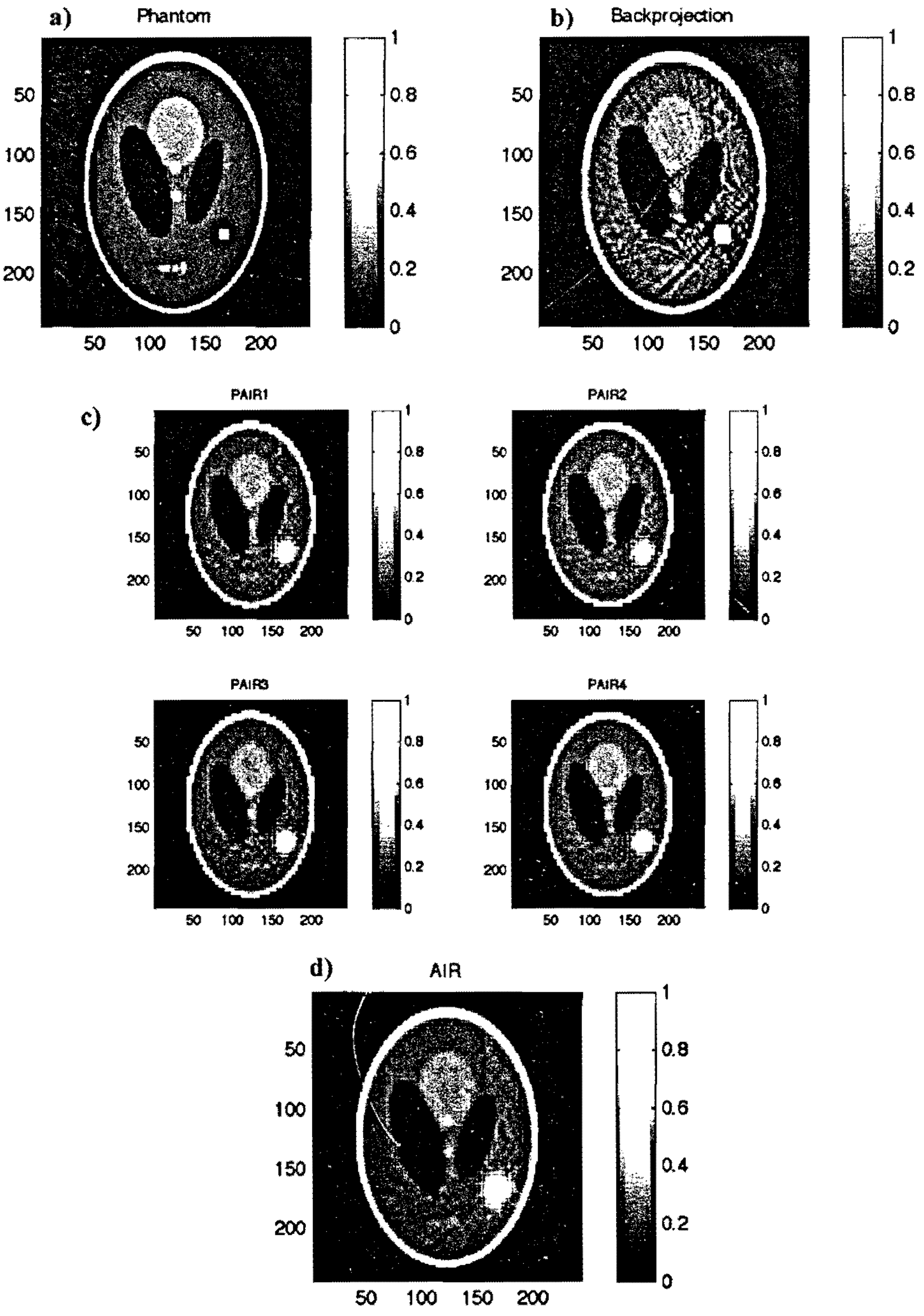


Figure 4.2.12i rescaled:
a) The phantom being imaged. **b)** Filtered backprojection reconstruction has streaking artifacts. **c)** PAIR with $C^{ct}=1$, $C^{wav}=7.0\text{x}10^4$, $C^{zeroav}=1.0\text{x}10^8$, $C^{local}=1$. Each PAIR solution uses a reconstruction grid of 60x60. The images are displayed on a higher resolution grid. **d)** The final AIR image results from averaging four 60x60 PAIR solutions onto a 120x120 grid. The result is of noticeably higher quality than the backprojected image. ARA is not included because the system of equations exceeds computational ability.

Test ten: Limited beam angle range in the less ideal simulation model

The tenth test (**figure 4.2.13**) examines a limited range of angle views in the less ideal simulation model. The results highlight the limits of AIR when it depends to heavily on the averaging terms (images **c** and **d**). In the more ideal simulation (the second test **figure 4.2.4**), only a small importance assigned to the averaging terms was enough to break degeneracy and find the exact solution. In the more complex simulations an exact reconstruction does not exist and thus the averaging terms are needed with greater importance. The filtered backprojection image has poor contrast, many negative values, and image blurring (image **b**). While the final AIR image is far from exact, it avoids negative values, improves contrast, and reduces blurring.

The tenth test included PAIR and was conducted with the following parameters:

P = 240; **F** = 1; **D** = 120; **R** = 120; **A** = (0 | 90 | 180).

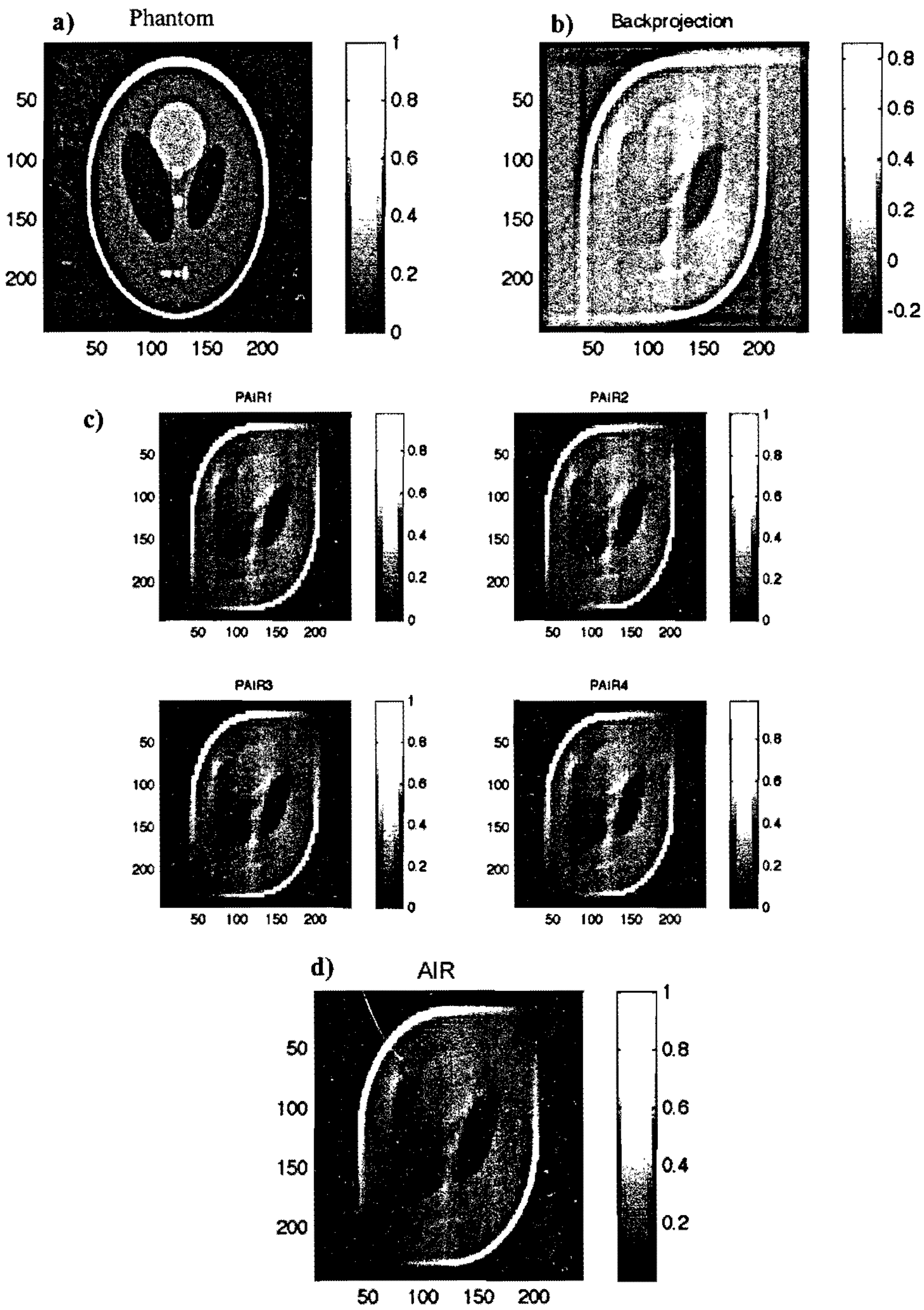


Figure 4.2.13: **a)** The phantom being imaged. **b)** Filtered backprojection reconstruction requires a full range of angles. **c)** PAIR with C^{ct}=1, C^{wav}=1.0x10^3, C^{zeroav}=5.0x10^7, C^{local}=100. In the more complex simulations PAIR and AIR require a more complete set of projections. **d)** The final AIR image results from averaging four 60x60 PAIR solutions onto a 120x120 grid. The result is of higher quality than the backprojected image, but depicts the limitations of AIR. ARA is not included because the system of equations exceeds computational ability.

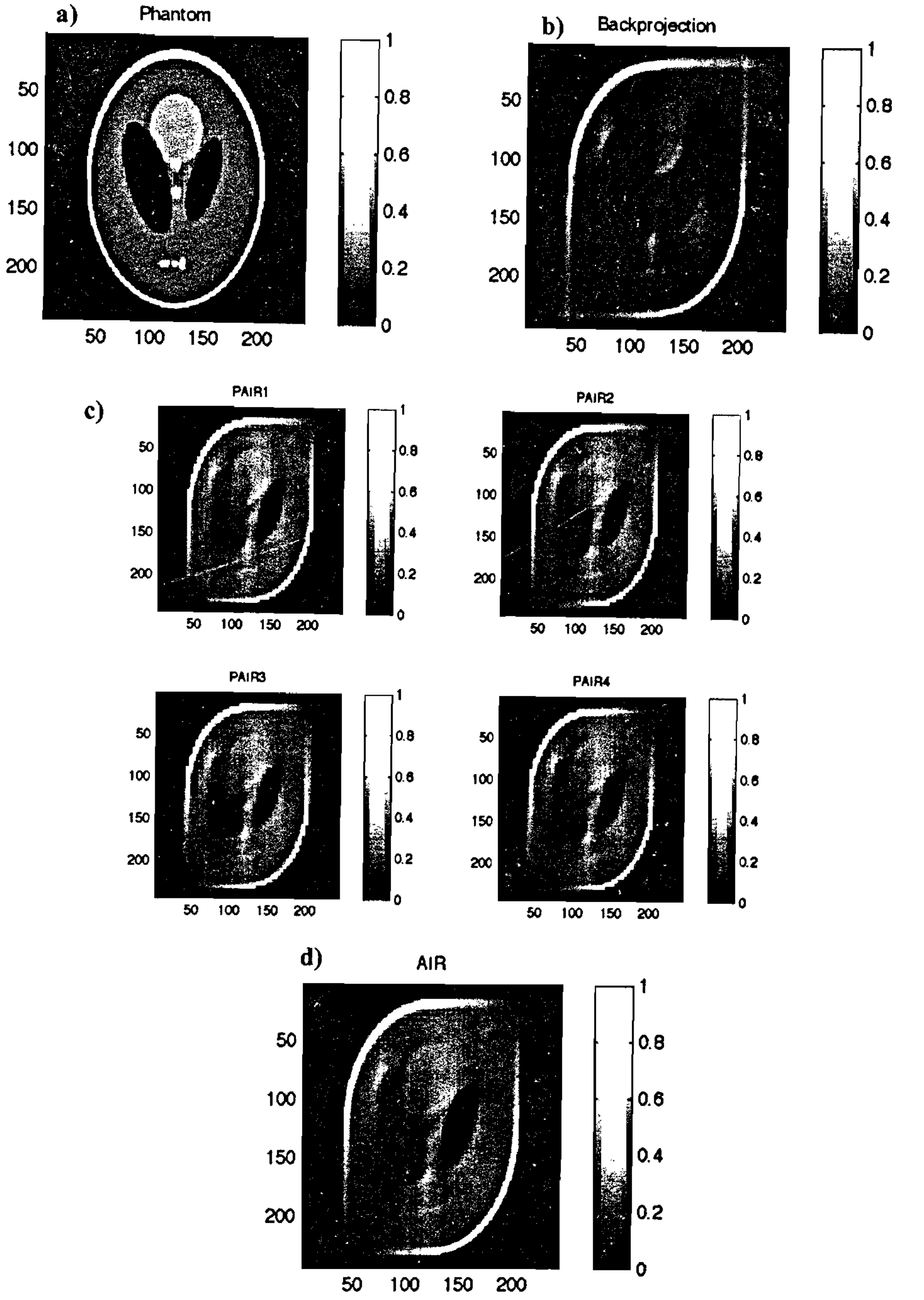


Figure 4.2.13i rescaled: a) The phantom being imaged. **b)** Filtered backprojection reconstruction requires a full range of angles. **c)** PAIR with $C^{ct}=1$, $C^{wav}=1.0x10^3$, $C^{zeroav}=5.0x10^7$, $C^{local}=100$. In the more complex simulations PAIR and AIR require a more complete set of projections. **d)** The final AIR image results from averaging four 60x60 PAIR solutions onto a 120x120 grid. The result is of higher quality than the backprojected image, but depicts the limitations of AIR. ARA is not included because the system of equations exceeds computational ability.

Test eleven: Limited beam angle views and projection noise in the less ideal simulation

The eleventh test (**figure 4.2.14**) imposes two difficulties to reconstruction: limited beam angle views and random noise. This final test was simulated with 1220 x 1220 resolution phantom and reconstruction was set to be 140 x 140. The filtered backprojection image is degraded by streaking, negative values, poor image contrast, and noise (image **b**). The final AIR result is more representative of the true phantom, but still suffers contrast degradation and image blurring (image **d**). This test highlights the advantages of AIR over the filtered backprojection algorithm in a situation involving many non-idealities.

The eleventh test included PAIR and was conducted with the following parameters: **P** = 1220; **F** = 1; **D** = 280; **R** = 140; **A** = (0 | 180 | 20); **Pnoise** = 10%.

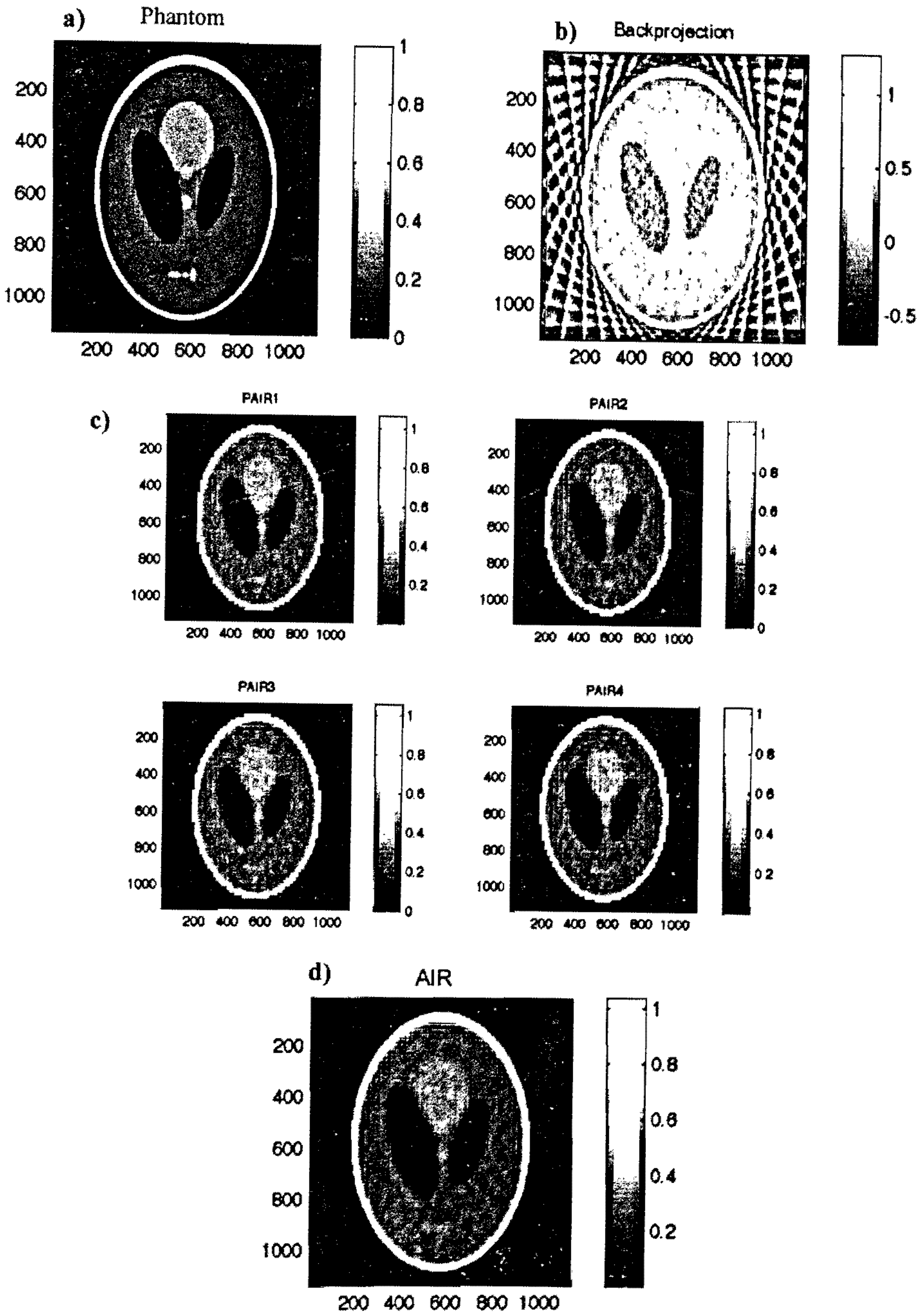


Figure 4.2.14: a) The phantom being imaged. **b)** Filtered backprojection reconstruction requires a full range of angles. **c)** PAIR with C^{ct}=1, C^{wav}=4.2x10^4, C^{zeroav}=1.0x10^8. **d)** The final AIR image results from averaging four 70x70 PAIR solutions onto a 140x140 grid. The result is of noticeably higher quality than the backprojected image. ARA is not included because the system of equations exceeds computational ability.

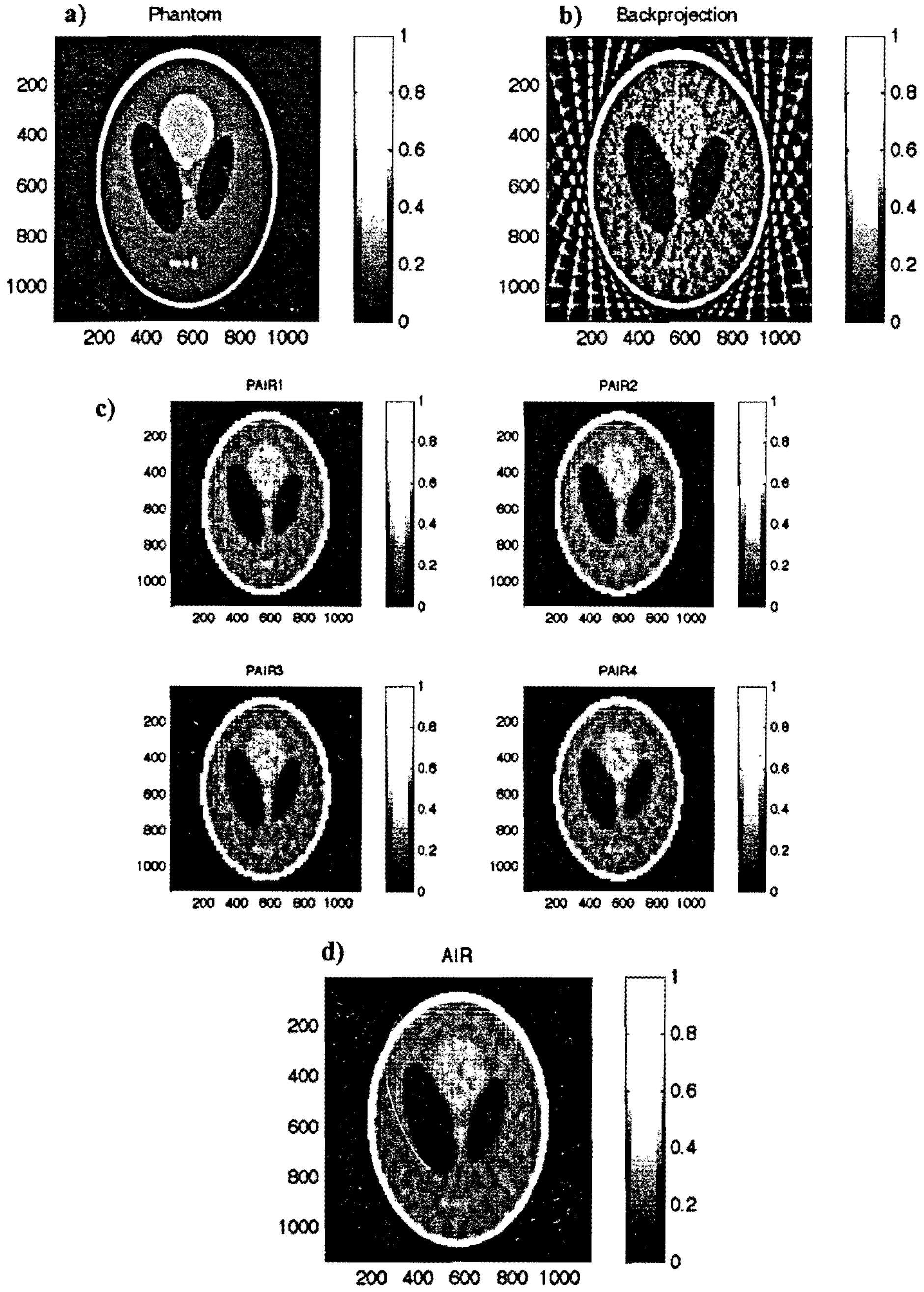


Figure 4.2.14i rescaled: a) The phantom being imaged. **b)** Filtered backprojection reconstruction requires a full range of angles. **c)** PAIR with $C^{ct}=1$, $C^{wav}=4.2\text{x}10^4$, $C^{zeroav}=1.0\text{x}10^8$. **d)** The final AIR image results from averaging four 70x70 PAIR solutions onto a 140x140 grid. The result is of noticeably higher quality than the backprojected image. ARA is not included because the system of equations exceeds computational ability.

4.3 Testing GIRA

GIRA was tested using only the most ideal simulation model (i.e. the reconstruction resolution is set to be equal to the resolution of the phantom being imaged). The tests presented here explore GIRA's ability to reconstruct an image from a limited number or range of beam angle views as well as the effects of increasing reconstruction resolution.

In the first few tests (one, two and part of three) the results include GIRA reconstruction with only the ZEROS and ONES function (image **b**), GIRA reconstruction with the ZEROS, ONES, and TWOS function (image **c**), and the complete GIRA reconstruction employing all the GIRA functions (image **d**) as developed in section 3.6 (pg 71). Unless otherwise stated, all images are displayed each with its own relative contrast.

Test one: Proof of principle of GIRA

The first GIRA test (**figure 4.3.1**) simulates a limited number beam angle views. The figure depicts the progressive nature of GIRA. In image **b**, using only the ZEROS and ONES function, GIRA is able to solve the pixels around the phantom, and a few pixels of the phantom itself (the image does not differentiate in contrast between unsolved pixels and solved zero pixels). Given the 180-degree beam angle range, these two functions will tend to solve the image in a circular fashion working inwards. Image **c** shows the progress of the ZEROS, ONES, and TWOS functions. The pattern of the solution depends on the geometric relationship between the various beamlets. Using all GIRA functions the reconstruction is complete and produces an exact solution as shown in image **d**. The first result demonstrates the proof of principle of GIRA.

The first test was performed using the following parameters:

P = 50; **D**=50; **R** = 50; **A** = (0|180|68).

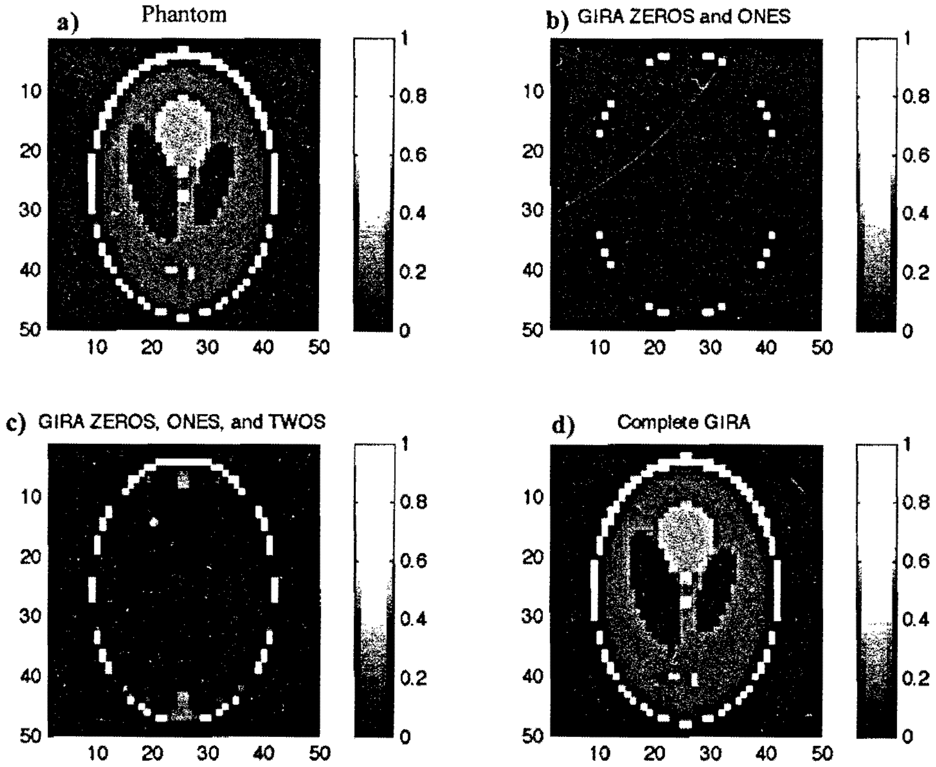


Figure 4.3.1: a) The phantom being imaged. **b)** The image is reconstructed using only the ZEROS and ONES function. It is clear that the image is not complete **c)** Using the additional function TWOS, GIRA can reconstruct more of the image. **d)** Using all GIRA functions the reconstruction is complete and produces an exact solution.

Test two: Limited range of beam angles

The second test (**Figure 4.3.2**) reduces both the range of angles and the number of angles in the simulation. It is clear that the solution pattern is much different compared to the previous test. An interesting point is that fewer angles are required to solve this scenario then the previous one. The reason being that here the beam angles are closer together and thus have much more geometric interaction. The final solution recovers the exact solution.

The second test was performed using the following parameters:
P = 50; **D**=50; **R** = 50; **A** = (0|90|60).

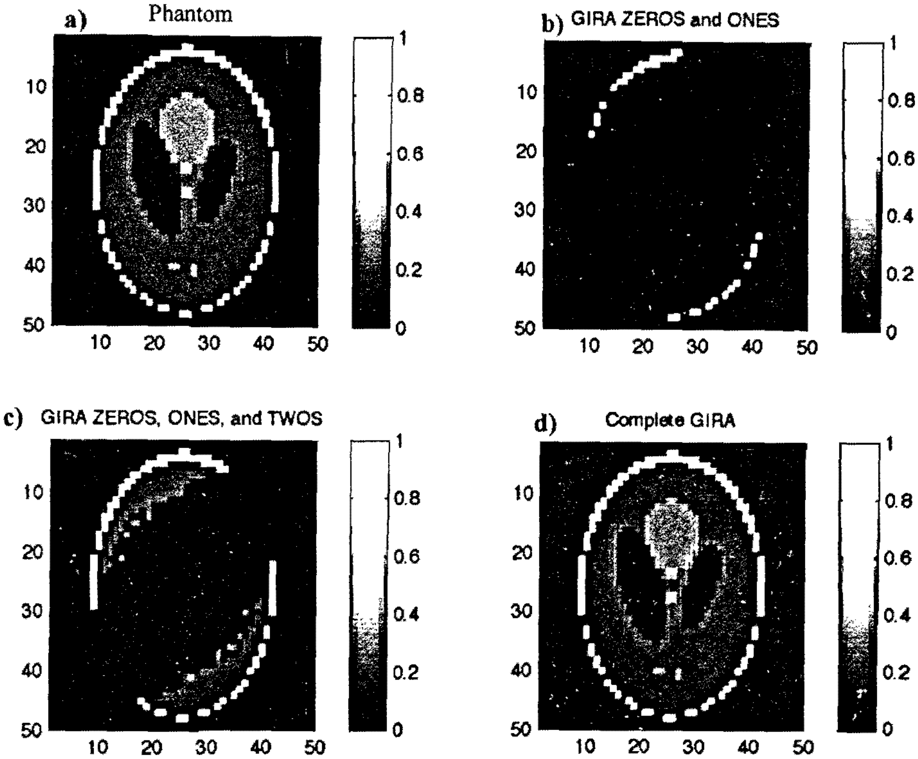


Figure 4.3.2: a) The phantom being imaged. **b)** The image is reconstructed using only the ZEROS and ONES function. It is clear that the image is not complete **c)** Using the additional function TWOS, GIRA can reconstruct more of the image. **d)** Using all GIRA functions the reconstruction is complete and produces an exact solution

Test three: Reconstructing an underdetermined system

In the third test (**figures 4.3.3 and 4.3.4**), as the phantom resolution and reconstruction resolution are increased, a larger number of beam angles is required to reconstruct the entire image. **Figures 4.3.3** depicts the incomplete reconstruction using only 90 beam angles. GIRA has not been developed to deal with underdetermined systems. By increasing the number of beam angles to 120, GIRA is able to complete the reconstruction (**Figures 4.3.4**).

The third test was performed using the following parameters:
P = 70; **D**=70; **R** = 70; **A** = (0|180|N). Where **N**, the number of beam angles, was varied.

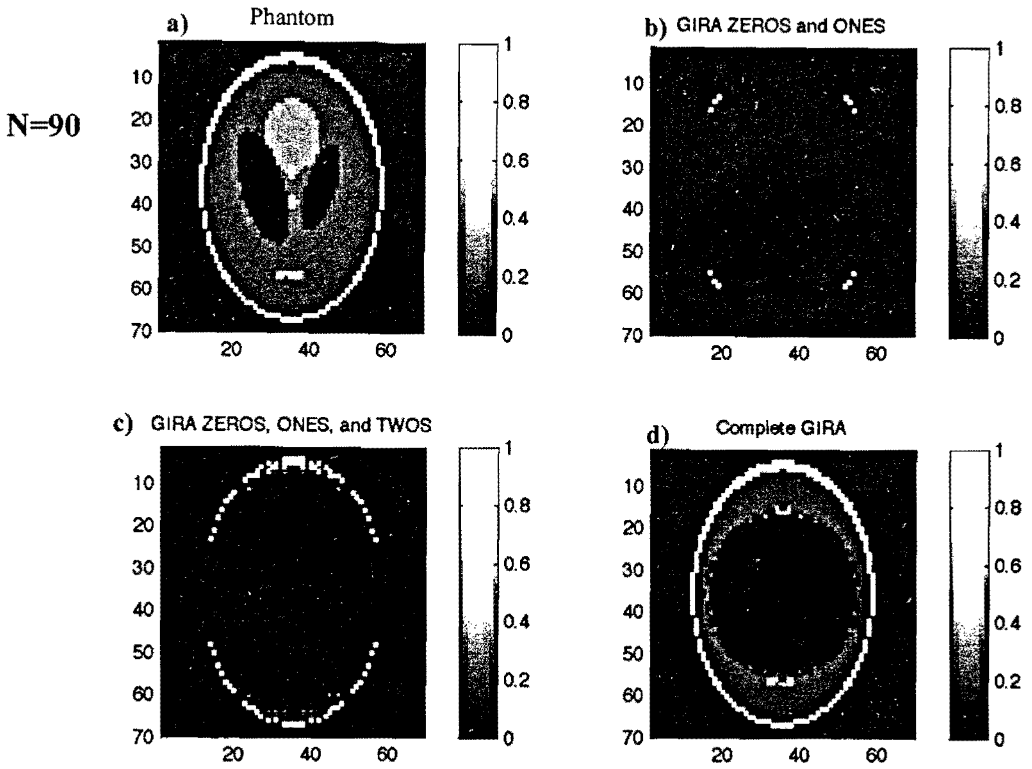


Figure 4.3.3: a) The phantom being imaged. **b)** The image is not fully reconstructed using only the ZEROS and ONES function. **c)** Using the additional function TWOS, GIRA can reconstruct more of the image. **d)** Using all GIRA functions the reconstruction is still not complete. There are not enough beamlets for GIRA to finish the reconstruction.

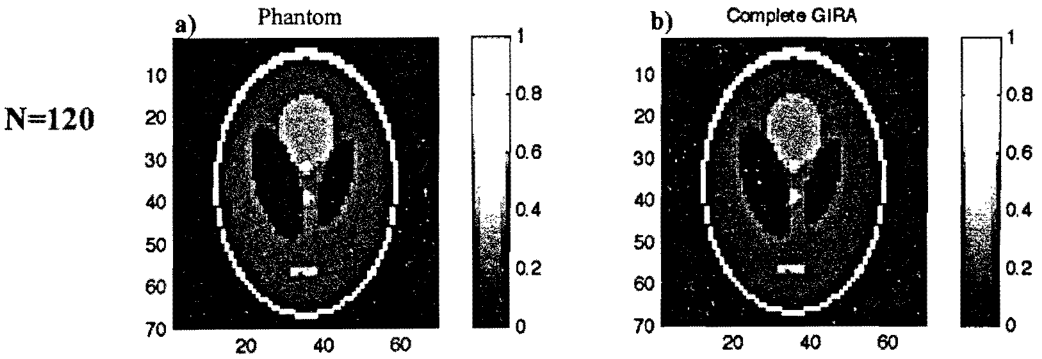


Figure 4.3.4: a) The phantom being imaged. **b)** With enough beam angles GIRA can fully reconstruct the image.

Test four: Increasing reconstruction resolution

In the fourth test (**figure 4.3.5**) the resolution is again increased. Even though these simplified tests represent ideal conditions GIRA runs into a difficulty. Because the determination of pixels at the center of the image depends on the determination of the outer pixels, the accumulation of errors may have a large effect on the solution. Although these ideal simulations do not present inconsistent projection information, and errors should not be present, it seems as though the image degradation was caused by the accumulation of round-off error.

Image **d** shows this issue beginning to surface. There is slight misrepresentation at the center of the image. Image **g** depicts the continued trend of the error. Moving towards the center the error increases. Image **h** has been rescaled for visualization of the error. Because GIRA works successively, with each pixels value depending on the previously solved pixels, the algorithm is inherently subject to such propagation errors.

The fourth tests were performed using the following parameters:

P = **D** = **R** = **G** where **G** was varied; **A** = (0|180|180)

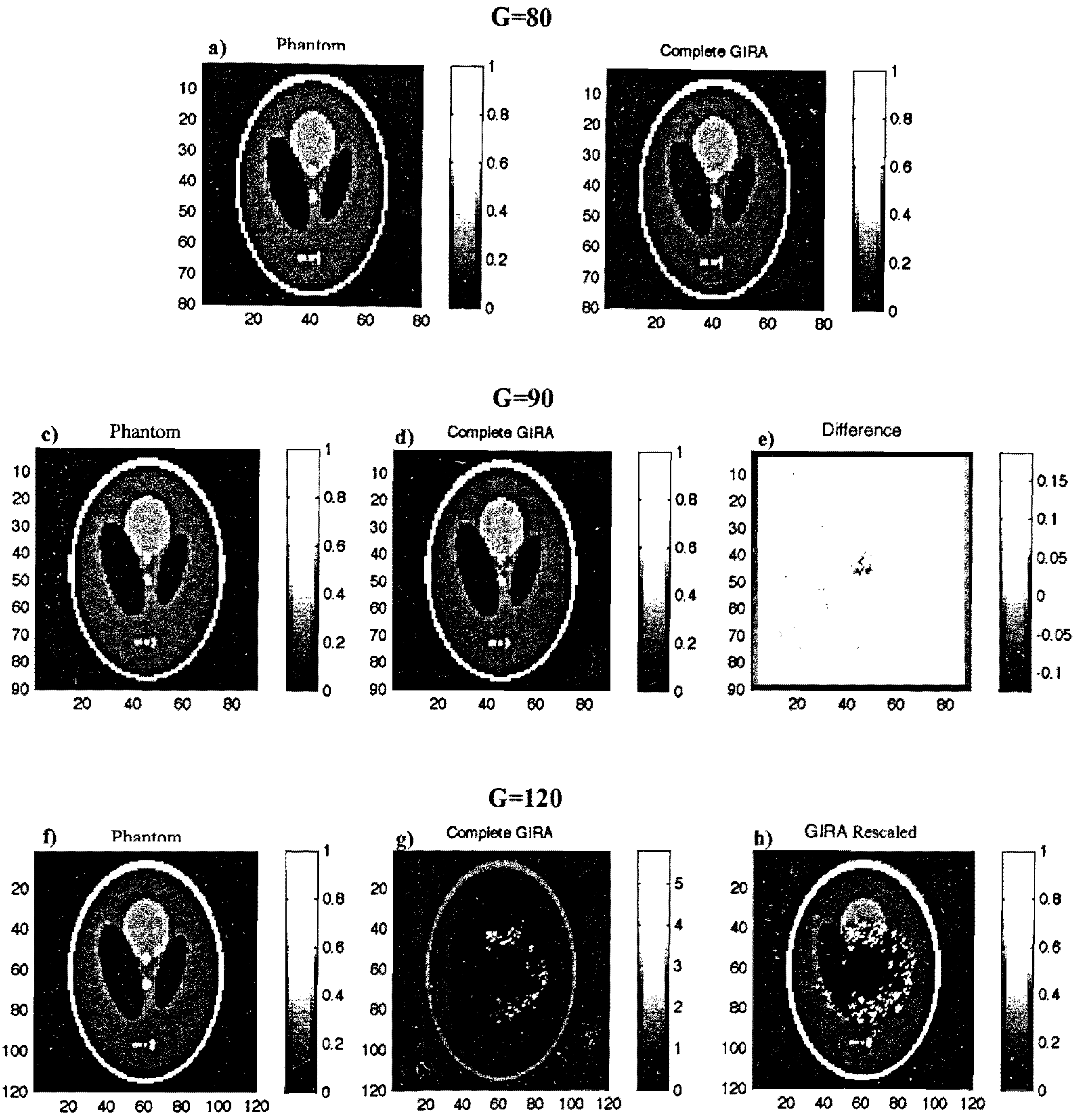


Figure 4.3.5:
a) The phantom being imaged. **b)** GIRA image reconstruction. There is a small amount of degradation in the center of the image.
c) The Phantom being imaged. **d)** GIRA image reconstruction. The error increases towards the center. **e)** The difference between the true phantom and the reconstructed image
f) The Phantom being imaged. **g)** GIRA image reconstruction. The error increases towards the center. **h)** A rescaling of the image in **f** from 0-1.

CHAPTER V

DISCUSSION AND CONCLUSIONS

5.1 Discussion of AIR and PAIR

The results of AIR have demonstrated the proof of principle of this reconstruction method. A record of invention has been issued and a patent is being pursued. In ideal cases, AIR reconstructed the images exactly (exactly-meaning that the largest absolute difference between an image pixel and a corresponding phantom pixel was on the order of 10^{-8}). In these ideal cases, the role of the averaging terms was simply to break degeneracy in order for the *CT* term to recover the true solution. As the complexity of the simulation model was increased, the averaging terms became more central to the reconstruction. The reason for this is that in the realistic simulations, there is not one true solution, but an infinite number of approximate solutions. By changing the importance coefficients of the various terms, one can obtain various approximations; some with better contrast, or structural definition, or overall smoothness.

The weighted averaging term seemed to perform much like the backprojection algorithm, and became a crucial term in the reconstruction. The zero averaging term removed any artifacts around the phantom. The concept of the zero averaging term can theoretically be applied to any reconstruction algorithm and should have profound effects on image quality. The local averaging term was not as central to reconstruction. It had overall smoothing effects but at the high cost of contrast. The global averaging term was not used at all in the depicted results. Its effect was similar to the local averaging term, but with a more profound and global smoothing effect at an even higher cost of contrast then the local averaging term.

The averaging terms that were developed do not represent all possible formulations. It is likely that other averaging terms could be derived and employed with effective results. While this was beyond the scope of this research, it can become a topic of future investigation. The point is that the performance of the averaging terms formulated here should not necessarily be considered a limiting factor in the reconstruction.

For AIR to become a plausible reconstruction tool it is necessary that the values of importance coefficients be built into the optimization. Changes to the importance coefficients in the tests conducted in this thesis relied on knowledge of the phantom that was imaged. Throughout the testing this researcher gained an intuitive approach to determining the values of the importance coefficients. The importance coefficients for the averaging terms seemed to depend on the reconstruction resolution, the number of detectors in the array, the number of beam angles, and certain characteristics of the imaged region (for example, the presence of a sharp gradient necessitated higher importance values for the averaging terms). Without this knowledge it may still be possible to find the optimum coefficients, but it may require a second iteration of AIR. This crucial aspect has not been fully investigated, and constitutes an important topic of future work. It should be noted that a similar difficulty exists in filtered backprojection, in determining the filter function that is used. In fact, one can argue that the ability to obtain multiple reconstruction images - some with better contrast or reduced noise - may be an advantage in the reconstruction method. Depending on the application, one can choose the importance coefficients accordingly.

The implementation of PAIR in this testing was not the ideal implementation that was described in the theory of PAIR. Initially the goal was to fully resolve the higher resolution from the lower resolution images through equation (2.3.1). Because the PAIR solutions were only approximations, equation (2.3.1) became very sensitive to error propagation and could not be used. The implementation in this thesis resorted to simply averaging the PAIR solutions together on a higher resolution grid. While this may not represent the most representative solution, the results were generally of higher quality than the corresponding filtered backprojection image. The results were quite notable considering that the AIR solution was comprised of four PAIR solutions, each at half the resolution of the corresponding filtered backprojection image. It should be noted that the process of averaging the four PAIR solutions has the consequential benefit of smoothing out noise in the image, and it may be useful in conjunction with the other resolving methods that were described.

AIR and PAIR have demonstrated their ability to reconstruct images under difficult conditions, one being limited angle views (figures **4.2.11** and **4.2.14**). This scenario may have important implications on imaging. By reducing number of angle views, it may be possible to lower dose to patients complying with the ALARA principle. AIR's ability to reconstruct images in the presence of metal inserts may also have important applications to imaging. The existence of sharp gradients in the imaged region is a major concern in imaging, and often limits the effectiveness of computed tomography.

Another question that remains is the issue of size. Due to computational constraints in this research, an upper limit to the reconstruction resolution possible for AIR and PAIR could not be defined. In current image reconstruction, where multiple slices are imaged together, the number of unknowns is multiplied by the number of slices that contribute to common projections. This may limit the application of AIR and PAIR to earlier generations of CT machines.

Although AIR and PAIR have shown positive preliminary results, further development into the mentioned issues and investigation under increasingly complex scenarios are essential in order to determine the feasibility of these methods.

5.2 Discussion of the Inversion Method

To recap, the inversion method in this thesis refers to first expressing a beamlet's path by a grid oriented at the same angle as that beam. This rotated grid is subsequently expressed in terms of the reconstruction grid by a matrix inversion. The intermediate step of using a rotated grid resulted in a non-trivial relationship between the beamlet's path and the reconstruction grid. The results of the inversion (referring to the example in equation 3.3.5) were unintuitive, yielding in most cases, partial contribution (*h* values) from most or all reconstruction pixels to a given projection. Some of the contributions appeared negative in weight - something that was not expected. The inversion method demonstrated a proof of principle in the fifth test when AIR reconstructed the image with the employment of this method (**Figure 4.2.8**). This method was of course not without concerns.

For one, the remapping required a solution to a matrix equation whose size was determined by the desired resolution. This matrix equation needed to be solved once for each angle and thus constituted a large set of computations. Of course, as this is strictly a geometric problem, these matrix equations can be solved once for a given geometry and kept in computed hard drive memory ready to use.

The second problem with the method was in its practical representation. As was discussed in the methods section, the remapping required a one-to-one relationship with the original reconstruction grid and the rotated grid. In approximating the area of the grid in terms of smaller squares, this one to one relationship was not obtained in general. Though this problem was corrected in most cases, the difficulties in the remapping have not been completely eradicated. The issue seemed to resurface at angles close to 45 and 135 degrees where the corners of the rotated grid were far away from the domain of the original grid. The result was a relationship that could not be inverted. It is suspected that the solution to this problem lies in the correct representation of the area of the grid instead of relying on approximations.

It should be stated that the steps that were taken to produce a stable relationship between the two grids should not be considered the only approach. For instance, a square matrix (the same number of pixels in the rotated grid and reconstruction grid) is not a theoretical necessity. Likewise, the method used to determine the overlap between the two grids does not constitute the only approach. The motivation here was only to demonstrate the proof of principle of this novel concept.

The other concern with the inversion method was its incompatibility with the averaging terms (**Figure 4.2.8** image **f**). This raises the question of whether other averaging terms specific to the inversion method could be developed.

The development of the inversion method represented a fascinating tangent to this research, but due to computational constraints, was not be given the same rigorous treatment as the reconstruction methods. Though the preliminary tests displayed promising results, a more in depth examination of this method is required to determine its full implications towards reconstruction. Future investigation may involve an analytic solution to the overlap between the two grids, or the development of averaging terms designed to be compatible with the inversion representation of pixel fractions.

5.3 Discussion of GIRA

Using logical functions to reconstruct images, GIRA has demonstrated its ability to solve reconstructions that would otherwise cause computational concerns if formulated as a matrix equation. While the theory of GIRA presented a fast and computationally efficient method of reconstruction, the issue of error propagation posed a real concern (**Figure 4.3.5**). In the presence of noise this issue is expected to magnify. There are many possible solutions to this problem. One such idea is to use information from multiple projections to determine the value of a pixel. Suppose there are multiple beamlets that satisfy the GIRA ONES condition for the same pixel. Due to random noise, their projection values may not be equal to each other. To solve the single pixel, an average value could be used to dampen the effect of the noise. A similar averaging concept can be applied to the other GIRA terms. This idea presents an interesting direction of future research.

Another potential area of future research is the development of additional terms in GIRA to complete the reconstruction in cases of limited projection information (as was the case in the third GIRA test). A possible additional function could seek to solve small sets of pixels that, due to the geometry of the problem, form equations isolated from the rest of the unknowns. Amongst other terms that could be developed, a most intriguing idea alluded to in the hypothesis, is to combine AIR and GIRA to reconstruct an image. There are many plausible approaches to combine these algorithms. GIRA can be used to reduce the problem set to a smaller number of unknowns allowing AIR to complete the reconstruction. Alternatively they can be combined in a more dynamic way using AIR to solve small sets of pixels forming independent equations and employing GIRA to solve the trivial solutions. The drawback of these proposed approaches is that they preclude the initialization of the AIR matrices.

While this research has shown a proof of the basic principles of GIRA in ideal simulations, additional developments and a more thorough investigation is required to determine the feasibility of this approach towards practical image reconstruction.

5.4 Conclusions

The developments in this research could be split into two distinct but overlapping areas. One area was the development of a CT simulation model, which could accurately represent a subset of the complexities in acquiring CT projections. The main focus in this area was to maintain a degree of blindness to the object being imaged. To this end, the reconstruction resolution, the detector resolution, and the reconstruction resolution were all defined separately. However, future developments to the CT simulation model should include, amongst other complexities, current CT source-detector geometries, a polychromatic x-ray source, a true representation of x-ray scatter, and object motion during the simulation.

The second area was the development and testing of AIR, PAIR, and GIRA. These algorithms have each demonstrated preliminary abilities to optimize image reconstruction compared to the filtered backprojection algorithm and ARA under various conditions. Specifically, these novel algorithms showed particularly promising results in the presence of sharp density gradients and in under determined systems involving limited beam angle views. However, there are many pending questions that need to be addressed. These questions relate to additional terms for each algorithm, the inversion method of representing pixels, increasing the reconstruction resolution, increased complexity in CT the simulation, and the feasibility of combining the novel algorithms into one reconstruction method. The work presented here constitutes the preliminary research and development of these novel approaches to CT image reconstruction.

One definition that was never clarified was the meaning of an optimal image. For the ideal simulations where an exact solution could be recovered, the definition was trivial. However, in the tests that involved reconstruction on a lower resolution grid, an exact solution was not possible. In these scenarios, which were more representative of the reality of image reconstruction, only approximate solutions existed. The difficulty is in defining which approximation represents the optimal solution. The results presented in this research were qualitative in comparison and analysis, and a metric through which to determine the optimal solution was not defined. Although many of the comparative results presented here have visually obvious conclusions, the need for a strict metric of determining the optimal image would be required for further testing.

Referring to the objectives mentioned in the hypothesis, the majority of the aims have been reached. The development of a realistic CT simulation in Matlab was one of the greatest challenges of this project. In adapting each algorithm into Matlab many practical issues were confronted, for example, the representation of inhomogeneous pixel fractions.

The topic of optimizing the reconstruction speed and computational efficiency of the novel algorithms was, for the most part, left for future research due to the constraints of the computers that were used. Furthermore, it was decided that this investigation should be left until beta testing, and could not be given a full treatment in Matlab.

The initial question that sparked this research was, "Why Fourier?" A major goal of this paper was to substantiate this question as a valid contention to current CT image reconstruction. The overwhelming majority of research in image reconstruction does not bother asking this question. With the preliminary success of AIR, PAIR and GIRA, it is this researchers' hope that "why Fourier?" becomes a question that is contemplated more often.

REFERENCES

[1] Jiang Hsieh, *Computed Tomography: Principles, Design, Artifacts, and Recent Advances.* Spie Press, Washington, 2003 (referenced figures used with permission of the author).

[2] A. C. Kak and Malcolm Slaney, *Principles of Computerized Tomographic Imaging*, IEEE Press, 1988.

[3] Jerrold T. Bushberg, J. Anthony Seibert, Edwin M. Leidholt Jr., John M. Boone, *The Essential Physics of Medical Imaging Second Edition.* Lippincott Williams & Wilkins, Philadelphia, 2002.

[4] Harold Johns, John Cunningham, *The Physics of Radiology Fourth Edition.* Charles C Thomas, Illinois, 1983.

[5] R. A. Brooks and G. DiChiro, *Statistical limitations in x-ray reconstructive tomography.* Med. Phy., vol. 3, pp 237-240, 1976.

[6] Frank H. Attix, *Introduction to Radiological Physics and Radiation Dosimetry.* John Wiley & Sons, Inc., New York, 1986.

[7] Australian Radiation Protection and Nuclear Safety Agency *http://www.arpansa.gov.au/radiationprotection/basics/xrays.cfm.*

[8] Steve Webb, *The Physics of Three-Dimensional Radiation Therapy: Conformal Radiotherapy, Radiosurgery and Treatment Planning.* Institute of Physics Publishing, Bristol and Philadelphia, 1993.

[9] G. T. Herman and A. Lent, *Iterative reconstruction algorithms*, Comput. Biol. Med., vol. 6, pp 273-294, 1976.

[10] R. Gordon, R. Bender and G. T. Herman, *Algebraic reconstruction techniques (ART) for three dimensional electron microscopy and x-ray photography*, J. Theor, Biol., vol. 29, pp 471-481, 1970.

[11] A.H. Andersen and A. C. Kak, *Simultaneous algebraic reconstruction technique (SART): a superior implementation of the art algorithm.* Ultrason. Imaging, vol. 6, pp. 81-94, Jan. 1984.

[12] P. Gilbert, *Iterative methods for the reconstruction of three dimensional objects from their projections.* J. Theor. Biol., vol. 36, pp 105-117, 1972.

[13] A. C. Kak and B. Roberts, *Image reconstruction from projections.* Handbook of pattern recognition and image processing, T.Y. Young and K. S. Fu, Eds. New York, NY: Academic Press, 1986.

[14] R.C. Gonzalez, *Digital Image Processing.* Reading, MA: Addison-Wesley, 1997.

[15] N. Baba and K. Murata, *Filtering for image reconstruction from projections.* J. Opt. Soc. Amer., vol. 67 pp. 662-668, 1997.

[16] P. Dreike and D. P. Boyd, *Convolution reconstruction of fan-beam reconstructions.* Comp. Graph. Image Proc., vol. 5, pp 459-469, 1977.

[17] G.T. Herman and A. Naparstek, *Fast image reconstruction based on a Radon inversion formula appropriate for rapidly collected data.* SIAM J. Appl. Math., vol. 33, pp. 511-533, Nov. 1977.

[18] P. N. Keating, *More accurate interpolation using discrete Fourier transforms.* IEEE Trans. Acoust. Speech Signal Processing, vol. ASSP-26, pp. 368-369, 1978

[19] R.M. Lewitt and R.H.T. Bates, *Image reconstruction from projections*. Optik, vol 50, pp 19-33,1978.

[20] E.O. Brigham, *The Fast Fourier Transform*. Englewood Cliffs, NJ: Prentice-Hall, 1974.

[21] C.R. Crawford and A. C. Kak, *Aliasing artifacts in computerized tomography*. Appl. Opt., vol. 18, pp 3704-3711, 1979.

[22] O.J. Tretiak, *Noise limitations in x-ray computed tomography*. J. Comput. Assist. Tomog., vol 2, pp 477-480, Sept. 1978.

[23] R. A. Brooks and G. DiChiro, *Statistcal limitations in x-ray reconstructive tomography,* Med. Phys., vol. 3, pp. 237-240, 1976.

[24] S. P. Goldman, J. Z. Chen, J. J. Battista, *Feasibility of a fast inverse dose optimization algorithm for IMRT via matrix inversion without negative beamlet intensities*. Med. Phys. 2005 Sept ;32 (9):3007-16.