

Electronic Theses and Dissertations, 2004-2019

2013

Curvelets And The Radon Transform

Jill Dickerson
University of Central Florida

 Part of the [Mathematics Commons](#)
Find similar works at: <https://stars.library.ucf.edu/etd>
University of Central Florida Libraries <http://library.ucf.edu>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Dickerson, Jill, "Curvelets And The Radon Transform" (2013). *Electronic Theses and Dissertations, 2004-2019*. 2529.
<https://stars.library.ucf.edu/etd/2529>

CURVELETS AND THE RADON TRANSFORM

by

JILL DICKERSON

B.S. Florida Southern College, 2011

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Mathematics
in the College of Sciences
at the University of Central Florida
Orlando, Florida

Spring Term
2013

© 2013 Jill Dickerson

ABSTRACT

Computed Tomography (CT) is the standard in medical imaging field. In this study, we look at the curvelet transform in an attempt to use it as a basis for representing a function. In doing so, we seek a way to reconstruct a function from the Radon data that may produce clearer results. Using curvelet decomposition, any known function can be represented as a sum of curvelets with corresponding coefficients. It can be shown that these corresponding coefficients can be found using the Radon data, even if the function is unknown. The use of curvelets has the potential to solve partial or truncated Radon data problems.

As a result, using a curvelet representation to invert radon data allows the chance of higher quality images to be produced. This paper examines this method of reconstruction for computed tomography (CT). A brief history of CT, an introduction to the theory behind the method, and implementation details will be provided.

ACKNOWLEDGMENTS

This research would not be possible without the help and the support of my advisor, Alexander Katsevich. Thank you cannot begin to address the level of gratitude I hold towards him.

It is an honor for me to thank the other members of my committee, Brian Moore and Alexandru Tamasan. I am extremely grateful for their patience and flexibility.

I am deeply indebted to the entire math department for the tremendous level of understanding they displayed as I finished this process. Without the faculty and staff, I would not have finished in a timely manner.

Thank you to Ryan Dickerson, who supplied all of the non-Matlab graphics in this research paper. He was always open for a new project, even though he had his hands busy completing his degree in Graphic Design. He is available for hire, if someone needs a freelance graphic artist.

Lastly, I offer my regards to all those directly, as well as indirectly, involved with the completion of this process. Your support played a crucial role that I value beyond what words can express.

TABLE OF CONTENTS

LIST OF FIGURES	6
LIST OF EXAMPLES	8
LIST OF ACRONYMS or ABBREVIATIONS.....	9
CHAPTER ONE: INTRODUCTION.....	1
CHAPTER TWO: CONCEPTS.....	3
2.1 Computer Tomography Scanning.....	3
2.2 Radon Transform.....	6
2.3 Wavelet Transform.....	10
2.4 Curvelet Transform	10
2.5 Reconstructions.....	17
2.5.1 Filtered Back Projections	18
2.5.2 Iterative Reconstruction	19
CHAPTER THREE: METHODOLOGY	20
CHAPTER FOUR: RESULTS	27
CHAPTER FIVE: CONCLUSION.....	39
APPENDIX: CODE.....	41
LIST OF REFERENCES	48

LIST OF FIGURES

Figure 1: Detectors take measurements of the x-ray projections through the body.	5
Figure 2: From the projections, an image of a "slice" of the body is constructed.	5
Figure 3: The concept of the computation of the Radon transformation.	7
Figure 4: Graphical representation of a disc centered at the origin with radius one.	8
Figure 5: The fine-scale bases are "needles" with a specifically defined orientation.	11
Figure 6: Curvelets occur at dyadic lengths.	12
Figure 7: Supports for curvelets of scale $j=2$ at locations $l=7$ and $l=4$ (light gray), and curvelets of scale $j=4$ at locations $l=1$ and $l=7$ (dark gray).	15
Figure 8: Supports for curvelet functions expressed in Figure 7. Similar graphic in [11]	16
Figure 9: Pseudo-polar support in the frequency domain for discrete curvelet transforms, similar graphic can be found in [12].	17
Figure 10: Image reconstruction. Clockwise from top left: original image, 10% of curvelets retained, 50% of curvelets retained, 20% of curvelets retained.	28
Figure 11: Image reconstruction. Top: 50% of curvelets retained (left) and 30% of curvelets retained (right). Bottom: Original image.	29
Figure 12: A curvelet's coefficient becomes larger as more of the function edge it overlaps.	30
Figure 13: Curvelets at different scales; in order, $j=3$, $j=4$, $j=5$, $j=7$	32
Figure 14: Modified Shepp-Logan head phantom (left) and its Radon transform (right).	33
Figure 15: Test image to estimate curvelet coefficients (left) and corresponding Radon transform (right).	33

Figure 16: Curvelets of interest in test image	34
Figure 17: Image representation of example U_m	35
Figure 18: Curvelet e with aliasing prominent.	37

LIST OF EXAMPLES

<i>Example 1: Radon Transform of a Disc</i>	<i>7</i>
<i>Example 2: Support of a Curvelet.....</i>	<i>15</i>
<i>Example 3: Steps to do Iterative Reconstruction.....</i>	<i>19</i>
<i>Example 4: Formula for Curvelet Coefficient</i>	<i>20</i>
<i>Example 5: Finding Curvelet Coefficients using Radon Transform of a Function</i>	<i>24</i>
<i>Example 6: Comparing the Methods for Finding Curvelet Coefficients</i>	<i>36</i>

LIST OF ACRONYMS or ABBREVIATIONS

CT- computed tomography

DWT- discrete wavelet transforms

FBP-forward back projection

MRI -magnetic resonance imaging

PET- positron emission tomography

SPECT- single photon emission computed tomography

CHAPTER ONE: INTRODUCTION

CT is a standard medical imaging technique which first came into clinical use in the 1970's.

During the CT process, X-rays are transmitted through an object: most commonly an animal or human. The intensities of the outgoing x-rays, known as projections, are then measured. An algorithm takes the measurements of the projections and mathematically reconstructs the image from its line integrals [1].

The Curvelet transform is a higher dimensional generalization of the Wavelet transform that has become popular in the fields of image processing, signal processing and scientific computing.

Curvelets provide a multi-scale object representation of images or signals at different scales and different angles. Also, the curvelet transform provides an ideal mathematical methodology for representing objects which display smoothness apart from singularities along smooth curves or have a minimal scale length, for example, in cartoons or text. Two unique mathematical properties curvelets possess are; singularities along curves can be closely approximated with few coefficients in a non-adaptive manner and they differ from other directional wavelet transforms because the degree of localization in orientation varies with scale. The shape of the curvelet basis function at scale j is 2^j by $2^{j/2}$ so the fine-scale bases are highly directionally dependent at fine scales [2].

This paper will introduce computed tomography and curvelet theory, describe the implementation, and explore the numerical application. In particular, section 2.1 explains the details of the current standard practice of CT scans, while the reconstruction methods will be

described in the rest of chapter 2. Next, chapter 3 will discuss the methodology of analysis, clarify the algorithms used in the Matlab code, and derive numerical implementations of the Matlab code. Chapter 4 will provide numerical results, showing curvelet application for various situations. Finally, chapter 5 summarizes the conclusions of this paper. Matlab code for the algorithms is provided in the appendix.

CHAPTER TWO: CONCEPTS

In classical CT, a slice of the entire body is taken in order to reconstruct what is inside the patient's body. CT algorithms reconstruct two dimensional slices of an object using either the parallel-beam, or fan-beam, x-ray scanning [3]. There are many types of methods used for the reconstruction. The most popular algorithms are part of a family called the filtered back projection (FBP) family, which involve sequential one dimensional filtering then computing an integral. These algorithms are the most widely used because they are accurate and efficient [4]. Reconstruction can also be curvelet based. The following sections in this chapter introduce the theory and background information utilized in the experimental setup.

2.1 Computer Tomography Scanning

Alessandro Vallebona was an Italian radiologist who suggested a method to represent a single slice of the body. This suggested process was the birth of tomography. Soon after Bernard Ziedses des Plantes develops planigraphy. In doing so, he laid the ground work for the tomographic technique and was able to produce as a result geometrically perfect images for the first time. Since the early 1900s, many advancements have been made [5]. In 1972, Godfrey N.Hounsfield's development of computed tomography marks the beginning of a new era in diagnostic imaging. With the aid of computed tomography it was possible for the first time to produce non-superimposed images of an object slice. By 1989 the first spiral CT scans were operational [6].

Until the 1970's, if a patient had a problem with something going on inside his body, a doctor would either use conventional X-rays or have to operate in order to identify it. Conventional X-rays do not always help in identifying the problem. However, operation was clearly not a very safe method of diagnosis since any invasive procedure carries significant risk. This is especially true for medical problems involving the brain [7]. However, in more recent years, advances in science have made it possible to look inside the body without requiring surgery; a method known as medical imaging. Medical imaging uses sources from inside or outside of the body in order to detect what is going on in a patient. X-rays and ultrasound devices are often used in the medical imaging methods that require sources and detectors outside of the body to capture what is going on inside the body. One of the most common of which would be computerized axial tomography(CAT). Other imaging methods use a source inside the body. These include magnetic resonance imaging (MRI), positron emission tomography (PET) and single photon emission computed tomography (SPECT). The most common competitors to CT are MRI and ultrasound. These methods have certain advantages over CT in safety as X-rays can easily damage soft tissue [8]. However, MRI is slow and ultrasound has low spatial resolution, so CT will remain an essential diagnostic tool. For this reason, doctors would like to be able to localize the CT scan, to minimize the effect of the harmful X-rays. This paper will focus on CT, medical imaging using X-ray sources outside of the body. Wavelets have been used for reconstruction of the CT data, and because curvelets are an extension of the wavelet transform, curvelets may prove to be an effective means of reconstruction.

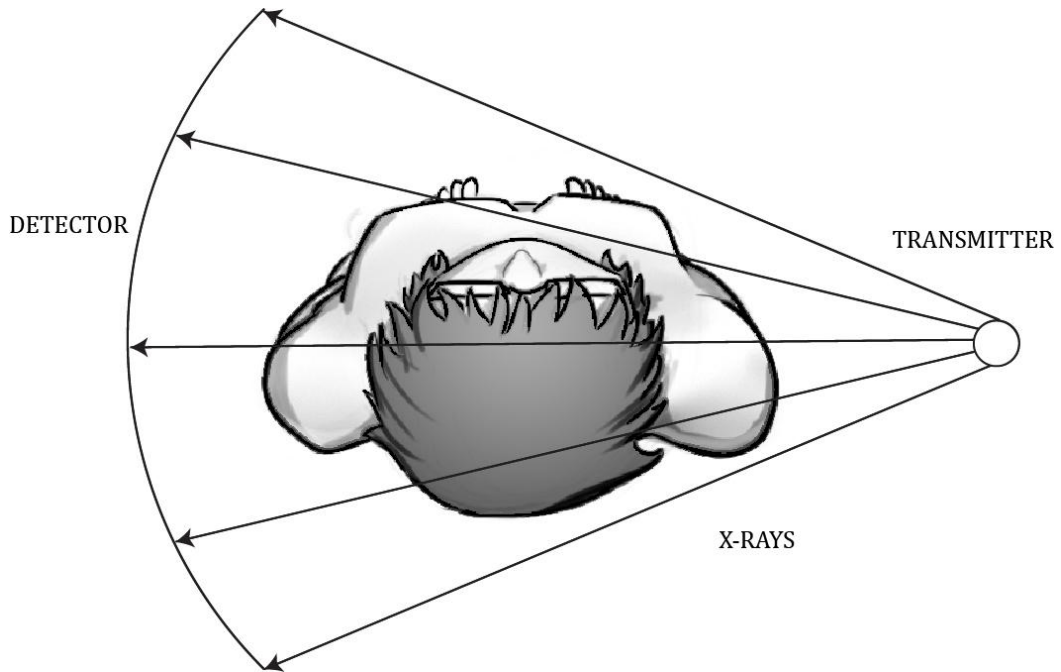


Figure 1: Detectors take measurements of the x-ray projections through the body.



Figure 2: From the projections, an image of a "slice" of the body is constructed.

2.2 Radon Transform

During the process of image reconstruction, the CT data is converted “back” into the image using the inverse Radon transform [9]. Without access to a CT scanner, the data needs to be computer-generated in order to run a test. The Radon transform is a process that can be used to simulate the data received by a CT scan.

The Austrian mathematician Johann Radon introduced what is now called the Radon transform in 1917. Since this paper focuses on two dimensional image reconstruction, this section will discuss the two dimensional Radon transform.

The Radon transform is an integral operator that involves the integral of a function over straight lines [10]. It is defined by

$$Rf(\theta, s) = \int_{L(\theta, s)} f(x) dS(x) \quad (1)$$

where $L(\theta, s) = \{(x, y) \in \mathbb{R}^2: x \cos \theta + y \sin \theta = s\}$. L can be understood as the line parameterized by $(x(t), y(t)) = (t \sin \theta + s \cos \theta, -t \cos \theta + s \sin \theta)$ where s is the distance of L from the origin and θ is the angle the normal vector to L makes with the x axis. This idea can be seen in the figure below.

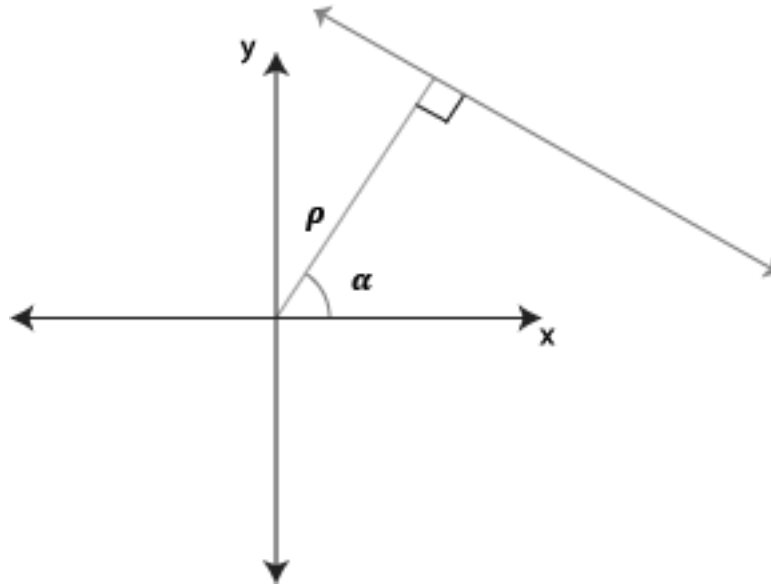


Figure 3: The concept of the computation of the Radon transformation.

Consider the following example.

Example 1: Radon Transform of a Disc

A disc centered at the origin with radius one has the formula

$$f(x, y) = \begin{cases} 1 & \text{if } x^2 + y^2 \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

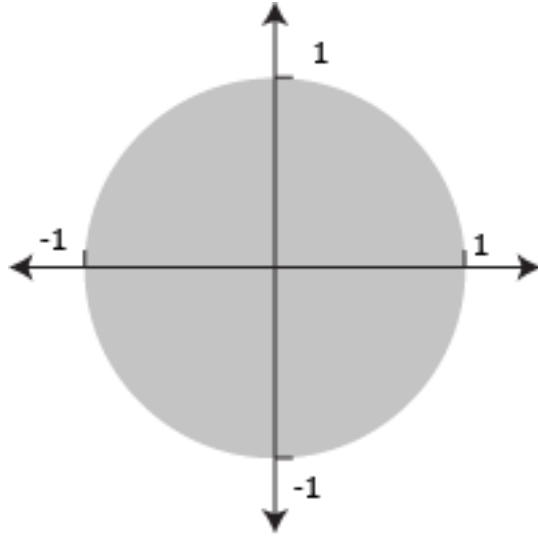


Figure 4: Graphical representation of a disc centered at the origin with radius one.

Then, the Radon transform of the function can be computed.

$$\begin{aligned}
Rf(\theta, s) &= \int_{L(\theta, s)} f(x, y) dS(x) \\
&= \int_{-\infty}^{\infty} f(t \sin \theta + s \cos \theta, -t \cos \theta + s \sin \theta) dt \\
&= \int_{-\infty}^{\infty} \begin{cases} 1 & \text{if } (t \sin \theta + s \cos \theta)^2 + (-t \cos \theta + s \sin \theta)^2 \leq 1 \\ 0 & \text{otherwise} \end{cases} dt \\
&= \int_{-\infty}^{\infty} \begin{cases} 1 & \text{if } t^2 \sin^2 \theta + 2st \cos \theta \sin \theta + s^2 \cos^2 \theta + t^2 \cos^2 \theta + s^2 \sin^2 \theta - 2st \sin \theta \cos \theta \leq 1 \\ 0 & \text{otherwise} \end{cases} dt \\
&= \int_{-\infty}^{\infty} \begin{cases} 1 & \text{if } t^2 (\sin^2 \theta + \cos^2 \theta) + s^2 (\cos^2 \theta + \sin^2 \theta) \leq 1 \\ 0 & \text{otherwise} \end{cases} dt \\
&= \int_{-\infty}^{\infty} \begin{cases} 1 & \text{if } t^2 + s^2 \leq 1 \\ 0 & \text{otherwise} \end{cases} dt \\
&= \int_{-\infty}^{\infty} \begin{cases} 1 & \text{if } |t| \leq \sqrt{1-s^2} \\ 0 & \text{otherwise} \end{cases} dt \\
&= \int_{-\sqrt{1-s^2}}^{\sqrt{1-s^2}} 1 dt \\
&= 2\sqrt{1-s^2} \tag{3}
\end{aligned}$$

Notice that the Radon transform of our function is independent of θ . Since this function is a disc centered at the origin, the value for θ does not change the value of the Radon transform. If the disc were not centered at the origin, then the value for θ would affect the Radon transform.

2.3 Wavelet Transform

Although this paper is primarily concerned with the curvelet transform, this section is devoted to the wavelet transform. The curvelet transform is an extension of the wavelet transform, so the wavelet transform will be described enough to help with understanding the curvelet transform origin. To help with comprehension, examples will primarily refer to the discrete wavelet transform (DWT).

Fourier transform gives the contents of the signal without specifying the location. The wavelet transform generalizes the Fourier transform by using a basis. This basis characterizes both location and spatial frequency. DWTs are wavelet transforms performed non-continuously, and are generally used in signal and image processing [11]. A wavelet is a mathematical function used to divide a given function into different scale components. Each scale component can then be studied with a resolution that matches its scale. Wavelets can accurately deconstruct and reconstruct a function with discontinuities, an image or signal. The major disadvantage of the wavelets is that it does not encode the edge information.

2.4 Curvelet Transform

Lines and curves commonly occur in nature. Discrete wavelet transform has the disadvantage of poor directionality. This causes DWT to be less applicable in certain situations. This issue gave birth to the family of directional wavelet transforms. Curvelets, a member of this family, were created in order to maintain directionality. They use basis functions that are localized in orientation. This means that directional wavelet transforms not only identify the content and

location, but also the orientation at that point. As the name suggests, curvelets are an edge(curve) preserving structure giving orientation. In particular, the basis functions are long, needle-like structures, as shown in the figure below. A curvelet transform varies from other directional wavelet transforms in that the degree of localization changes with scale [2].

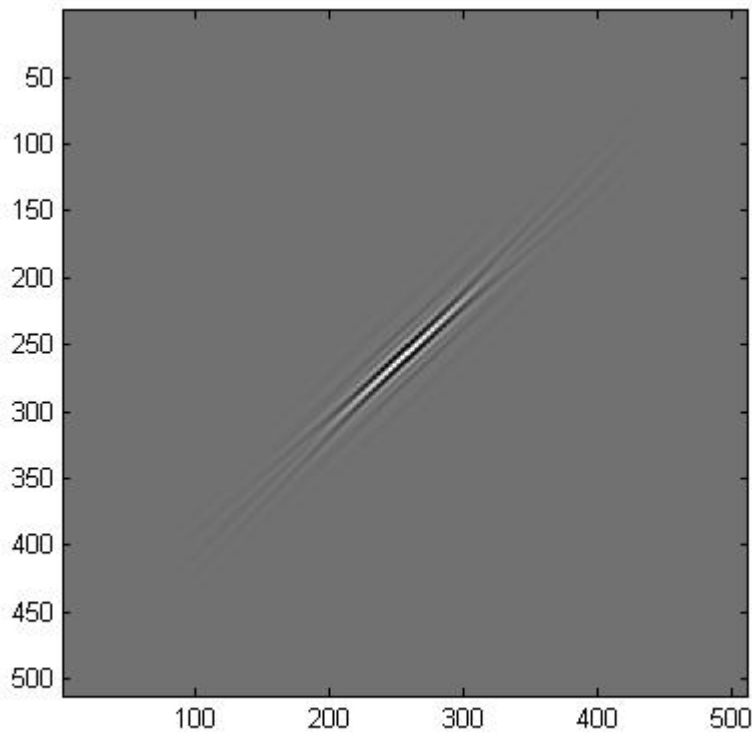


Figure 5: The fine-scale basis functions are “needles” with a specifically defined orientation.

Conceptually, the curvelet magnifies the object. Instead of magnifying everything uniformly in each direction, curvelets magnify what is necessary. It then creates a “container” to fit the edges of the object. This requires the curvelet to be directionally dependent. In the 2D image case, the curvelet goes in two directions.

Because curvelets have nice properties, they are becoming increasingly popular in image processing and scientific computing. Curvelets occur at all scales, locations, and orientations. What makes them unique is that a curvelet occurs at all dyadic lengths, as shown in Figure 6.

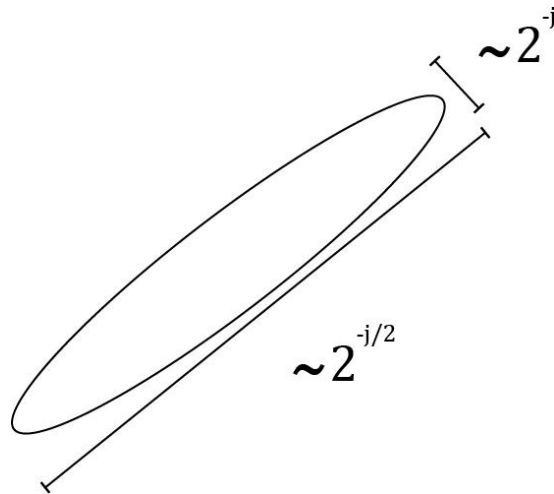


Figure 6: Curvelets occur at dyadic lengths

The curvelets are directionally dependent and become increasingly more so with decreasing scale. Another characteristic of curvelets is that they obey a scaling relation which says that the width of a curvelet element is about the square of its length. These needles are conceptually stretching the signal at various levels to find an area of significance.

The next thing to consider is when curvelets would be an appropriate basis. Curvelets are best used when representing piecewise smooth images or functions. These images can have singularities along the smooth curves. This property holds for cartoons and text. The edges of such images appear increasingly straight at small scales. Taking advantage of this property, the basis defines the higher resolution curvelets to be skinnier than the lower resolution curvelets.

Natural images usually have this property. Therefore, curvelets can provide a sparser representation than other wavelet transforms if the image is of the correct type [2].

For scale 2^{-j} , $j \in \mathbb{N}_0$, define, as in [12], the generating curvelet,

$$\psi_{j,0,0}(s, \omega) = 2^{-3j/4} \cdot W(2^{-j} \cdot s) \cdot V\left(\frac{2^{\lceil j/2 \rceil + 1}}{\pi} \cdot \omega\right) \quad (4)$$

in the frequency domain using polar coordinates (s, ω) , where $W(s)$ is a radial window and $V(\omega)$ is an angular window. Assume the windows W and V are real and smooth, and that they satisfy the following conditions,

$$\sum_{j=-\infty}^{\infty} W^2(2^j s) = 1, \quad s \in (3/4, 3/2), \quad (5)$$

$$\sum_{l=-\infty}^{\infty} V^2(\omega - l) = 1, \quad \omega \in (-1/2, 1/2). \quad (6)$$

There are many possible functions to satisfy these conditions, so the ones used in the experiments will be explicitly expressed in the methodology portion.

Translating and rotating the generating curvelets $\psi_{j,0,0}$ constructs the family of curvelets

$\{\psi_{j,l,k}(x)\}_{j,l,k}$. These are defined in [12] by

$$\psi_{j,l,k}(x) = \psi_{j,0,0}\left(R_{\theta_{j,l}}\left(x - b_k^{j,l}\right)\right), \quad (7)$$

where $R_{\theta_{j,l}}$ denotes the rotation matrix

$$R_{\eta} = \begin{pmatrix} \cos \eta & \sin \eta \\ -\sin \eta & \cos \eta \end{pmatrix} \quad (8)$$

with respect to $\theta_{j,l}$, the scale dependent rotation angles,

$$\theta_{j,l} = l \cdot \pi \cdot 2^{-\lceil j/2 \rceil - 1}, \quad -2^{\lceil j/2 \rceil - 1} \leq l < 2^{\lceil j/2 \rceil - 1} \quad (9)$$

where $b_k^{j,l}$ are the scale dependent locations defined by

$$b_k^{j,l} = R_{\theta_{j,l}}^{-1} \left(\frac{k_1}{2^j}, \frac{k_2}{2^{j/2}} \right)^T, \quad k = (k_1, k_2) \in \mathbb{Z}^2 \quad (10)$$

This means that all curvelets in the curvelet family are created from one curvelet through rotation, translation, and changing the scale. This implies that, in the frequency domain, each curvelet is defined

$$\psi_{j,l,k}(\xi) = e^{-i\langle b_k^{j,l}, \xi \rangle} 2^{-3j/4} W(2^{-j}s) V \left(\frac{\omega + \theta_{j,l}}{\theta_{j,l}} \right) \quad (11)$$

The curvelet functions are supported in the frequency domain inside a polar wedge with radius

$2^{j-1} \leq s \leq 2^{j+1}$ and angle $\frac{2^{-\lceil j/2 \rceil} \pi(-1-l)}{2} < \omega < \frac{2^{-\lceil j/2 \rceil} \pi(1-l)}{2}$ [2]. This means that the support of

the curvelet function $\hat{\psi}_{j,l,k}$ does not depend on the position $b_k^{j,l}$, so will not change with changes

in k . Support concepts are shown in *Example 2*.

Example 2: Support of a Curvelet

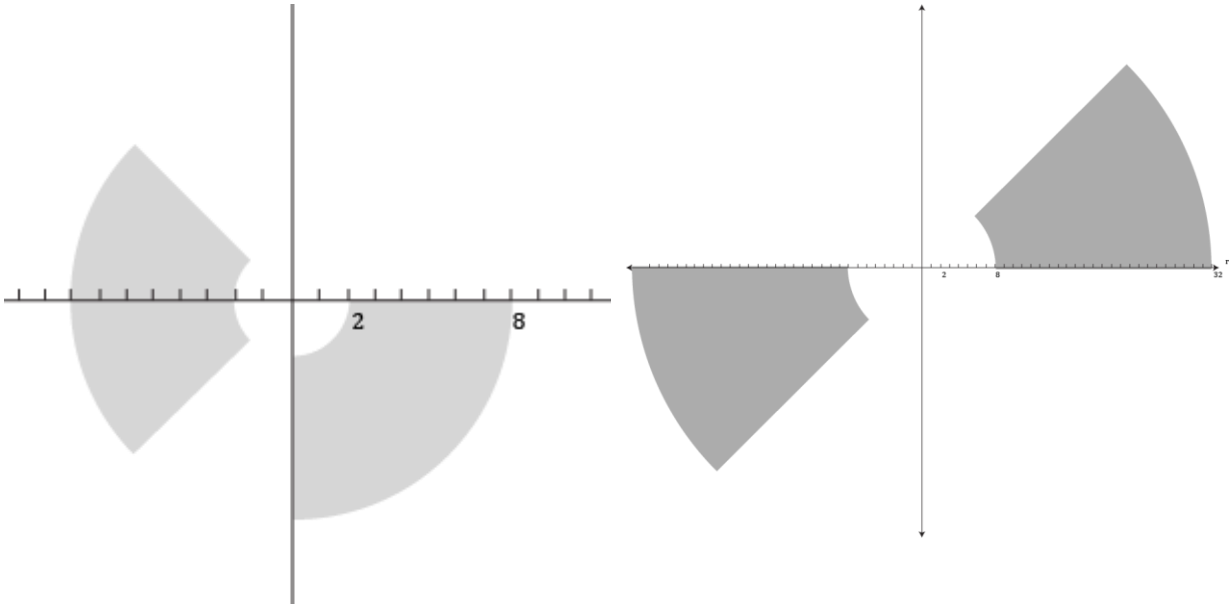


Figure 7: Supports for curvelets of scale $j=2$ at locations $l=7$ and $l=4$ (light gray), and curvelets of scale $j=4$ at locations $l=1$ and $l=7$ (dark gray).

The curvelet function in the frequency domain, $\psi_{2,7,k}(s, \omega)$, is supported in the wedge with angle $-2\pi < \omega < \frac{-3\pi}{4}$ and radius $2 \leq s \leq 8$. This support wedge is unaffected by the choice of value for k . Figure 7 shows the effects of differing values for scale j and location l , while the combination of each can be seen in Figure 8.

After discretization, the support alters slightly into a pseudo polar support in the frequency domain, as seen in Figure 9. Calculations with computers use this pseudo polar support. The tiling in the 2D Fourier domain corresponds to 5 scales. The angular wedges change with

increased scale, with 16 angle intervals at the second coarsest level and 64 angle intervals at the fifth and finest scale. Each angular wedge has a length proportional to 2^j , with a width proportional to $2^{j/2}$.

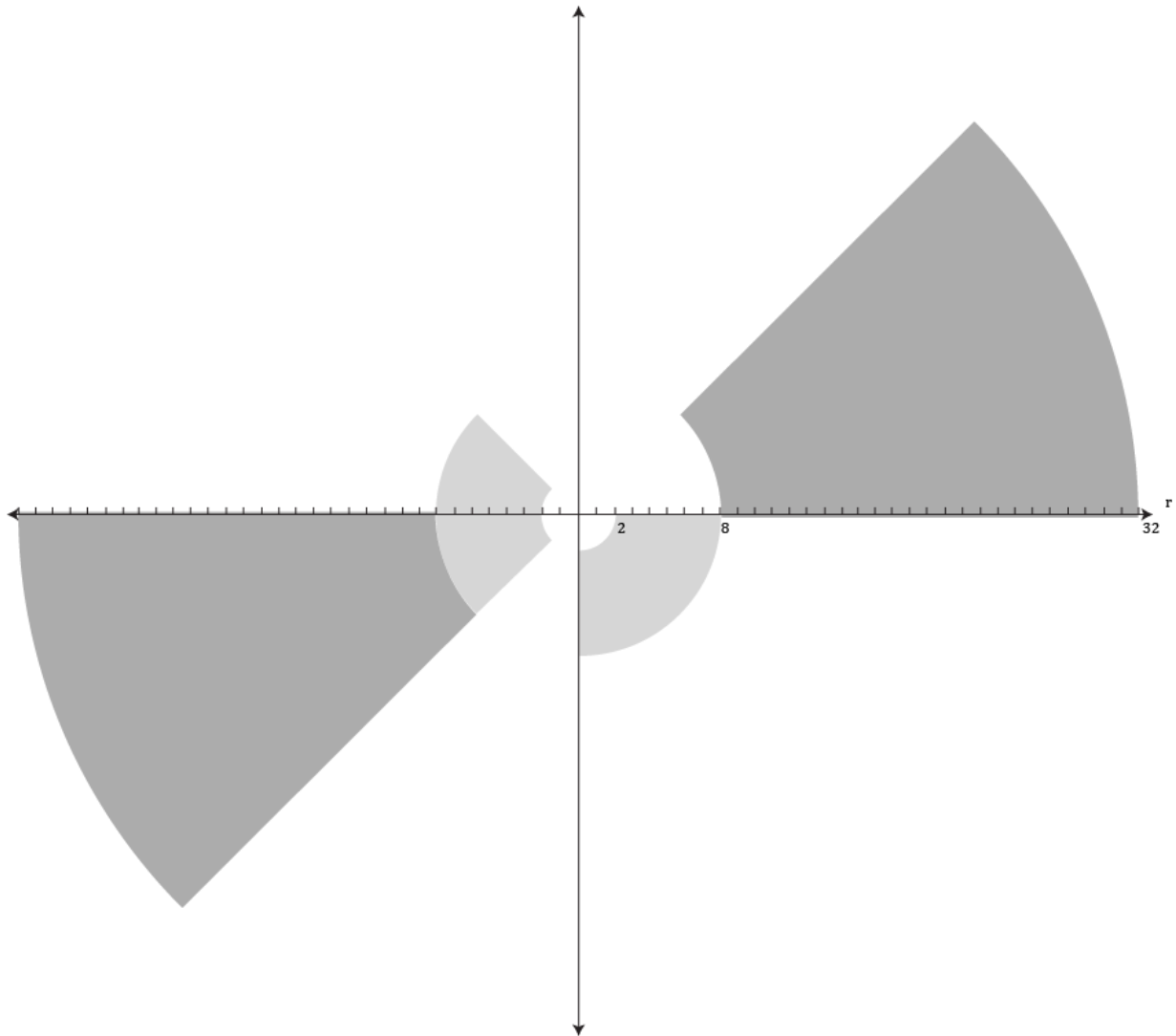


Figure 8: Supports for curvelet functions expressed in Figure 7. Similar graphic in [13]

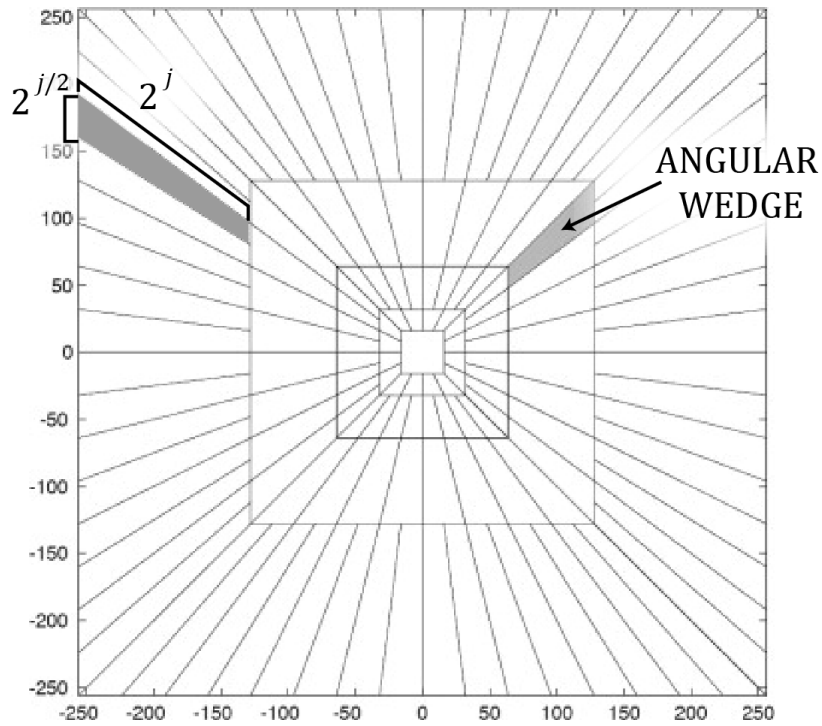


Figure 9: Pseudo-polar support in the frequency domain for discrete curvelet transforms, similar graphic can be found in [12].

2.5 Reconstructions

The computer uses algorithms in order to reconstruct an image from the CT partition data. This is an example of an “inverse problem”. A great deal of effort has been invested in looking for faster and more accurate reconstruction methods [1]. This section will give a brief background on the two most common methods.

2.5.1 Filtered Back Projections

Filtered back projection (FBP) is the most popular method of solving the current CT image reconstruction problems [3]. There must be a full angle range of non-truncated data in order to use this type of reconstruction. FBP uses two steps [11]. The first step is filtering. The formula used for filtering is the convolution defined by

$$g(\alpha, q) = \int_{\mathbb{R}} \frac{Rf_p(\alpha, p)}{q - p} dp \quad (12)$$

where $Rf_p(\alpha, p)$ is the partial derivative, with respect to p , of the Radon transform. Next, define the operator

$$(R^* \mu)(x) = \int_0^{2\pi} \mu(\alpha, \alpha \cdot x) d\alpha \quad (13)$$

This operator is the dual of the Radon transform, and satisfies the condition $\langle Rf, \mu \rangle = \langle f, R^* \mu \rangle$ where f is in the (x, y) domain, and μ is in the Radon, or (θ, s) domain. It takes a function from the Radon domain to the image domain. This operator helps define the back projection,

$$f(x) = \frac{1}{4\pi^2} (R^* g)(x) \quad (14)$$

which is the second step of FBP. When using Matlab, the command `iradon(x)`, which stands for inverse Radon transform, does FBP. The fact that Matlab has a command for FBP adds to the convenience of the method for reconstructing an image from Radon data.

2.5.2 Iterative Reconstruction

Even though FBP is the most popular method of image reconstruction, it often has limitations on the image quality produced. Noisy data produces a noisy image when using FBP, which leaves no room for improvements. With iterative reconstruction, adjustments are made at each step of forward projection in order to produce a better quality image [14].

Example 3: Steps to do Iterative Reconstruction

- An initial estimate of the activity distribution is made, usually assuming that all pixels have the same value.

- Using forward projection, an estimate is made of the counts that would be acquired in each projection.

- The estimated projections are compared with the actual measured projections acquired by the data.

- This comparison is then used to update the original estimate. The complete process is repeated until the differences between the measured projections and the estimate are negligible.

CHAPTER THREE: METHODOLOGY

This section will derive a few of the numerical implementations of the Matlab code.

Sections 2.1 and 2.2 explain the method of data collection. The next question to answer would be what to do with the data once it is received.

Curvelet decomposition states that any function can be represented as a summation of curvelets with corresponding coefficients.

$$f = \sum_{\alpha} c_{\alpha} \psi_{\alpha} \quad (15)$$

Furthermore, the coefficients can be found by evaluating the inner product of the function and the curvelet. In other words, every function can be represented as a summation of curvelets,

$$f = \sum_{\alpha} \langle f, \psi_{\alpha} \rangle \psi_{\alpha} \quad (16)$$

Example 4: Formula for Curvelet Coefficient

Let $\xi = (s, \omega)$ and the terms $c_{j,l,k}(f) := \langle f, \psi_{j,l,k} \rangle$ be the curvelet coefficients. Then, the curvelet coefficients can be found by

$$\begin{aligned}
c_{j,l,k}(f) &= \langle f, \psi_{j,l,k} \rangle \\
&= \int_{\mathbb{R}^2} f(x) \overline{\psi_{j,l,k}(x)} dx \\
&= \int_{\mathbb{R}^2} \widehat{f}(\xi) \overline{\widehat{\psi}_{j,l,k}(\xi)} d\xi \\
&= \int_{\mathbb{R}^2} \widehat{f}(\xi) e^{i\langle b_k^{j,l}, \xi \rangle} 2^{-3j/4} W(2^{-j}s) V\left(\frac{\omega + \theta_{j,l}}{\theta_{j,l}}\right) d\xi
\end{aligned} \tag{17}$$

Depending on the available information, it may be easier to calculate using the second row of (14). Using whichever equation is most appropriate, the value for the coefficient can be approximated using a summation.

The problem with direct curvelet decomposition is that, for problems such as CT scans, the function is unknown. Therefore, the coefficients to the curvelets would be impossible to calculate using this decomposition. The next issue to be discussed would be how to recreate a function using the Radon transform of the function. This section introduces new concepts to calculate these curvelet coefficients.

Let $\psi_{j,l,k}$ be a curvelet, with $b_k^{j,l}$ and $\theta_{j,l}$ as defined in section 2.4. Let $\theta(\omega) = (\cos \omega, \sin \omega)^T$

Then, the Radon transform of the curvelet given by

$$R\psi_{j,l,k}(\theta(\omega), s) = 2^{j/4} V\left(\frac{2^{\lceil j/2 \rceil}}{\pi}(\omega + \theta_{j,l})\right) \sqrt{2\pi} \widehat{W}\left(2^j \langle b_k^{j,l}, \theta(\omega + \theta_{j,l}) \rangle - 2^j s\right) \tag{18}$$

The proof of this is shown in [15].

Before moving forward, the window functions W and V need to be defined. It is important to satisfy the conditions [16]:

$$\sum_{l=-\infty}^{\infty} V^2(t-l) = 1, \quad t \in \mathbb{R} \quad (19)$$

$$\sum_{j=-\infty}^{\infty} W^2(2^j r) = 1, \quad r > 0. \quad (20)$$

This explicit example will represent the window functions used in this construction of the curvelet. These functions will use the definition of the scaled Meyer's windows [13].

$$V(t) = \begin{cases} 1 & |t| \leq 1/3 \\ \cos\left[\frac{\pi}{2}\nu(3|t|-1)\right] & 1/3 \leq |t| \leq 2/3 \\ 0 & \textit{else} \end{cases} \quad (21)$$

$$W(r) = \begin{cases} \cos\left[\frac{\pi}{2}\nu(5-6r)\right] & 2/3 \leq r \leq 5/6 \\ 1 & 5/6 \leq r \leq 4/3 \\ \cos\left[\frac{\pi}{2}\nu(3r-4)\right] & 4/3 \leq r \leq 5/3 \\ 0 & \textit{else} \end{cases} \quad (22)$$

where ν is a smooth function satisfying the condition

$$\nu(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x \geq 1 \end{cases} \quad \nu(x) + \nu(1-x) = 1, \quad x \in \mathbb{R} \quad (23)$$

Possibilities for such functions would be $\nu(x) = x$, or a smoother polynomial such as

$\nu(x) = 3x^2 - 2x^3$. The smooth window function leads to a faster decay of the curvelets. An

arbitrary smooth window using the example from [17] is given by

$$\nu(x) = \begin{cases} 0 & x \leq 0 \\ \frac{s(x-1)}{s(x-1) + s(x)} & 0 < x < 1 \\ 1 & x \geq 1 \end{cases} \quad \text{with} \quad s(x) = e^{-\left(\frac{1}{(1+x)^2} + \frac{1}{(1-x)^2}\right)} \quad (24)$$

Equation (21) will be the smooth function used in the experimentation. A plot of the V and W window functions are shown in Figure 10, a similar plot is shown in [13].

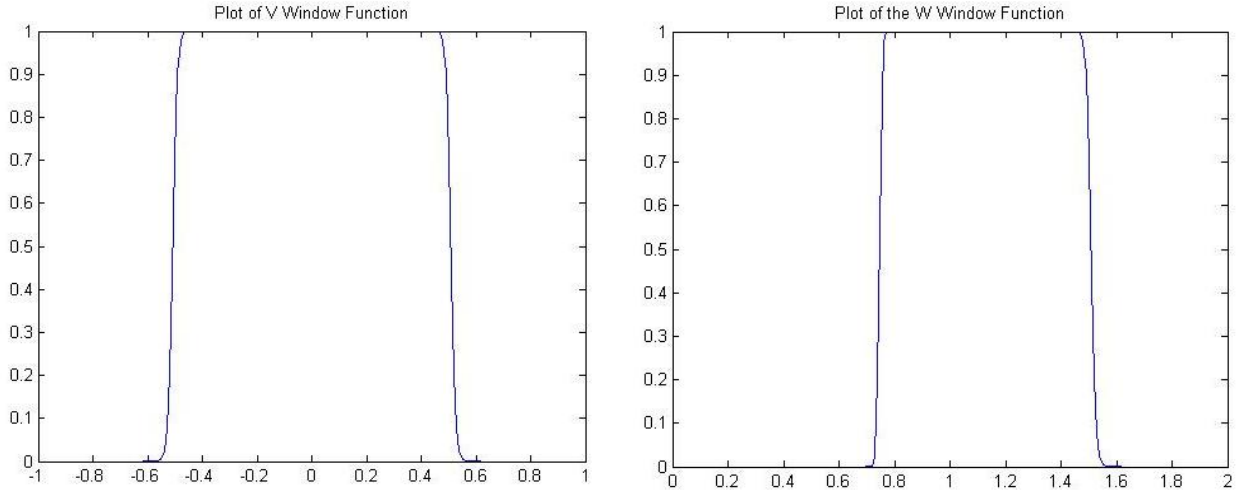


Figure 10: Window Functions.

Next, define the operator \square^α on one variable, as defined in [16], by

$$(\square^\alpha f)(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} |\omega|^{\frac{\alpha}{2}} \widehat{f}(\omega) e^{i\omega t} d\omega \quad (25)$$

where $\hat{f}(\omega)$ is the Fourier transform of the function. There are many variants of the Fourier transform. In order to be consistent with Matlab definitions, this paper will use the Fourier transform defined by

$$\hat{f}(t) = \int_{-\infty}^{\infty} f(\omega) e^{-i\omega t} d\omega \quad (26)$$

Therefore, the definition for the inverse transform is

$$f(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(t) e^{i\omega t} dt \quad (27)$$

Notice the relationship between the \square operator and the inverse Fourier transform. The reason for defining the \square operator is to construct a function such that

$$R^* U_m = 2^{-j} \psi_m \quad (28)$$

where R^* is the same as defined in (13). Using the result from [16], define such functions by

$$U_m(\theta, s) = 2^{-j} \cdot (\square^2 \otimes I) R \psi_m(\theta, s) \quad (29)$$

with $m = m(j, l, k)$, where \square^2 acts on the radial variable s , and I acts on the angular variable θ .

Example 5: Finding Curvelet Coefficients using Radon Transform of a Function

Using the property of the frames, $R^* U_m = 2^{-j} \psi_m$, a new method of finding the curvelet coefficients can be computed. Recall that regular curvelet decomposition has the curvelet coefficient $\langle f, \psi_m \rangle$. From this, the new curvelet coefficient can be calculated

$$\begin{aligned}
\langle f, \psi_m \rangle &= \langle f, 2^j \cdot R^* U_m \rangle \\
&= \overline{2^j} \langle f, R^* U_m \rangle \\
&= 2^j \langle Rf, U_m \rangle
\end{aligned} \tag{30}$$

This shows that the curvelet coefficients can be found with the Radon transform of the function.

Before plugging values into Matlab, there must be some calculations to create the code that computes U_m . The objective is to simplify the long formula for U_m into something that can more easily be converted into an algorithm using Matlab. With this in mind, this section will analyze calculations for U_m .

$$\begin{aligned}
U_m(s, \omega) &= 2^{-j} \cdot (\square^2 \otimes I) R\psi_m(s, \omega) \\
&= 2^{-j} \cdot (\square^2 \otimes I) 2^{j/4} V \left(\frac{2^{\lceil j/2 \rceil + 1}}{\pi} (\omega + \theta_{j,l}) \right) \sqrt{2\pi} \widehat{W} \left(2^j \langle b_k^{j,l}, \theta(\omega + \theta_{j,l}) \rangle - 2^j s \right) \\
&= 2^{-3j/4} V \left(\frac{2^{\lceil j/2 \rceil + 1}}{\pi} (\omega + \theta_{j,l}) \right) \sqrt{2\pi} \cdot \frac{1}{2\pi} \int_{-\infty}^{\infty} |t|^{\frac{2}{\alpha}} \widehat{W} \left(2^j \langle b_k^{j,l}, \theta(\omega + \theta_{j,l}) \rangle - 2^j t \right) e^{ist} dt \\
&= 2^{-3j/4} V \left(\frac{2^{\lceil j/2 \rceil + 1}}{\pi} (\omega + \theta_{j,l}) \right) \sqrt{2\pi} \cdot \frac{1}{\alpha} e^{is\beta/\alpha} \int_{-\infty}^{\infty} |t| W \left(\frac{-t}{\alpha} \right) e^{ist} dt
\end{aligned} \tag{31}$$

where $\widehat{W} \left(2^j \langle b_k^{j,l}, \theta(\omega + \theta_{j,l}) \rangle - 2^j t \right)$ can be found by the following calculation. Let $\alpha = -2^j$ and $\beta = 2^j \langle b_k^{j,l}, \theta(\omega + \theta_{j,l}) \rangle$, then

$$\begin{aligned}
\widehat{W}(\alpha s + \beta) &= \int_{-\infty}^{\infty} e^{-is\lambda} \widehat{W}(\alpha\lambda + \beta) d\lambda \\
&= \frac{1}{\alpha} e^{is\beta/\alpha} \int_{-\infty}^{\infty} e^{-i(\alpha\lambda + \beta)s/\alpha} \widehat{W}(\alpha\lambda + \beta) d(\alpha\lambda + \beta) \\
&= \frac{1}{\alpha} e^{is\beta/\alpha} \widehat{W}\left(\frac{s}{\alpha}\right) \\
&= \frac{2\pi}{\alpha} e^{is\beta/\alpha} W\left(\frac{-s}{\alpha}\right)
\end{aligned} \tag{32}$$

Since the window function W is an explicitly-defined function, and only nonzero inside an interval (18), the value for an integral can easily be approximated by a summation.

Using these equations, the corresponding coefficient for each curvelet can be approximated. It can be done regardless of whether the function is known or unknown. If the function is unknown, then the Radon transform of the function can be used.

This study's Matlab code creates an approximation for the curvelet coefficient using the Radon transform of a function.

CHAPTER FOUR: RESULTS

Numerical experiments will show the extension of the theoretical results in practice. All of the experiments were done in Matlab. Curvelab, a software available for download, has written Matlab code in order to do the curvelet transform on 2D or 3D graphics [18]. This paper will be examining the curvelet transform on two dimensional function.

Before doing experiments using the Radon data, consider an experiment of representing an image using curvelet decomposition. For the experiment, let f be the image to represent as a sum of curvelets.

Using Curvelab, code can be created to show how curvelet decomposition can be used for image reconstruction. In the experiments, depicted in Figure 11 and Figure 12, first the curvelet transform is taken of an image. The image must be a grayscale image. If it is not, then the image can easily be converted to one using the command `rgb2gray(x)`, where x is the colored image. To make the computations more efficient for Curvelab, the grayscale image is then resized to a 512 by 512 image matrix. Because the curvelets occur at dyadic lengths, having a 512 by 512 matrix simplifies the curvelet decomposition. Next, select a percentage of the largest coefficients in modulus, and set the others to zero. Lastly, the code uses the inverse curvelet transform to create a partial reconstruction of the original image.

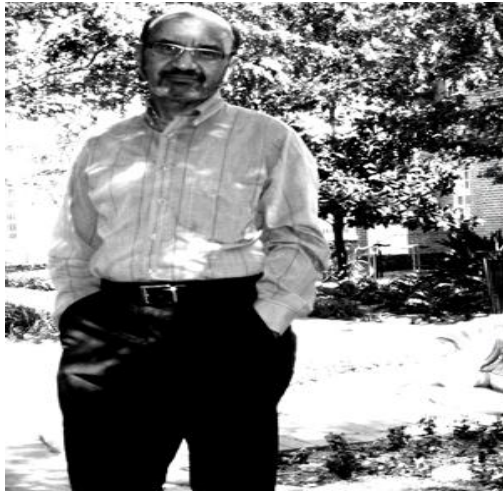


Figure 11: Image reconstruction. Clockwise from top left: original image, 10% of curvelets retained, 50% of curvelets retained, 20% of curvelets retained

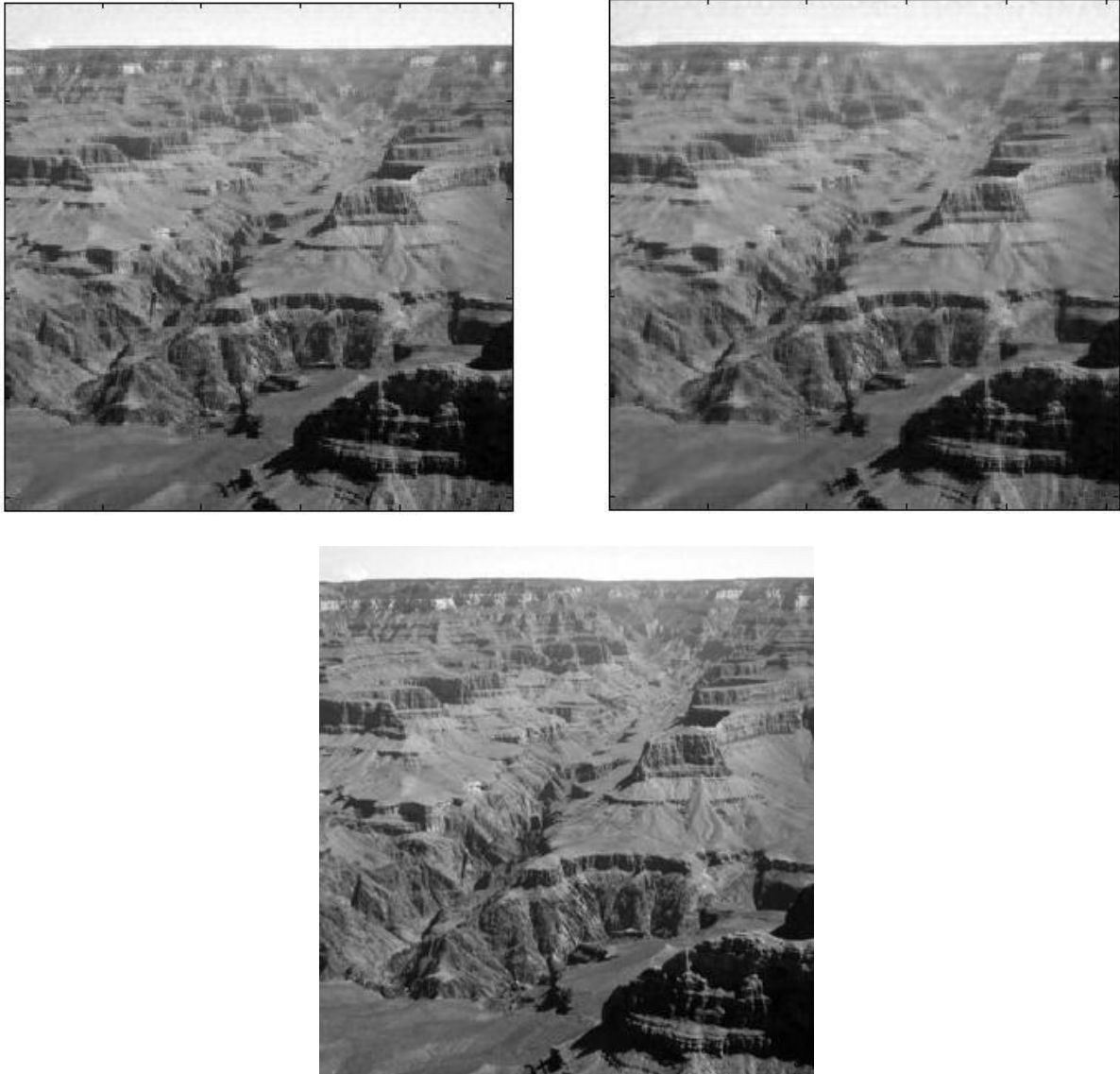


Figure 12: Image reconstruction. Top: 50% of curvelets retained (left) and 30% of curvelets retained(right). Bottom: Original image.

Figure 11 and Figure 12 show the benefit of using curvelets as a basis in a sparse representation for images. This gives evidence needed to pursue representing images in medical field. In the experiments doing partial reconstruction of a function, the function to represent is an image. For medical imaging, the function to represent is a slice of the body, showing what is inside to help diagnose a patient. The problem with representing the image of the slice of the body is that the slice of the body is unknown. The CT scan gives the detected X-ray projections for each line of transmitted X-rays, which is equivalent to the Radon transform of the body slice, Rf . The problem is to reconstruct the image of the function, the slice of the body, using the Radon transform of the function, the CT scan data. From the image reconstruction experiments, it is clear that curvelets can do well when used as a basis for image reconstruction. With this in mind, the next step is to do curvelet decomposition on a figure, and then see if those values can be found with the Radon data. The objective of curvelet decomposition is to find the corresponding curvelet coefficients for a function.

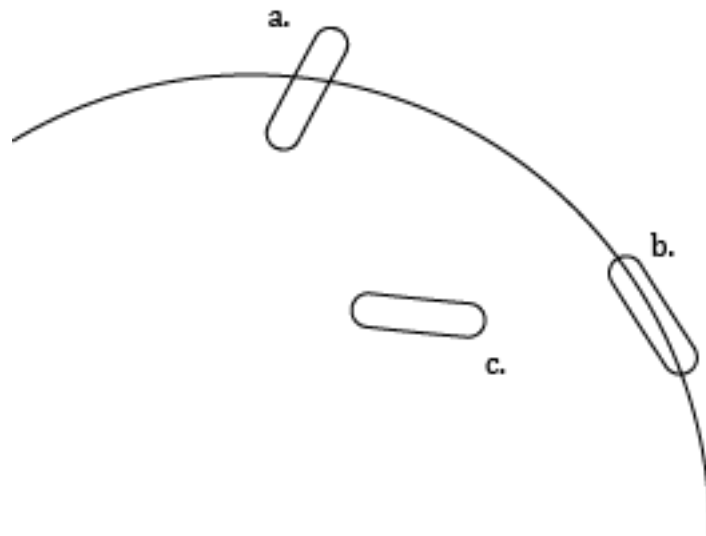


Figure 13: A curvelet's coefficient becomes larger as more of the function edge it overlaps.

As shown in Figure 13, there are different scenarios that can happen with curvelets. There is little overlap of the function and curvelet a, but it is not very much, so the corresponding coefficient is close to zero. Curvelet b is following the edge of the function. Thus, the corresponding coefficient for curvelet b is nonzero. For curvelet c, there is no overlap of the function edge, therefore the corresponding coefficient is zero.

The following experiments will work to compute the curvelet coefficients of a function using the Radon data. To do this, the first thing to inspect is the curvelet in the discrete space for function representation. Figure 14 shows the effects of changing the scale in the curvelets. Notice the localization in the curvelets as the values for j increases. The length of the curvelet is approximately 2^j . For large values of j , such as $j=7$, the curvelet is smaller and more localized than the other curvelets of smaller values for j . This means the smaller details of the image would use curvelets with higher j in reconstruction.

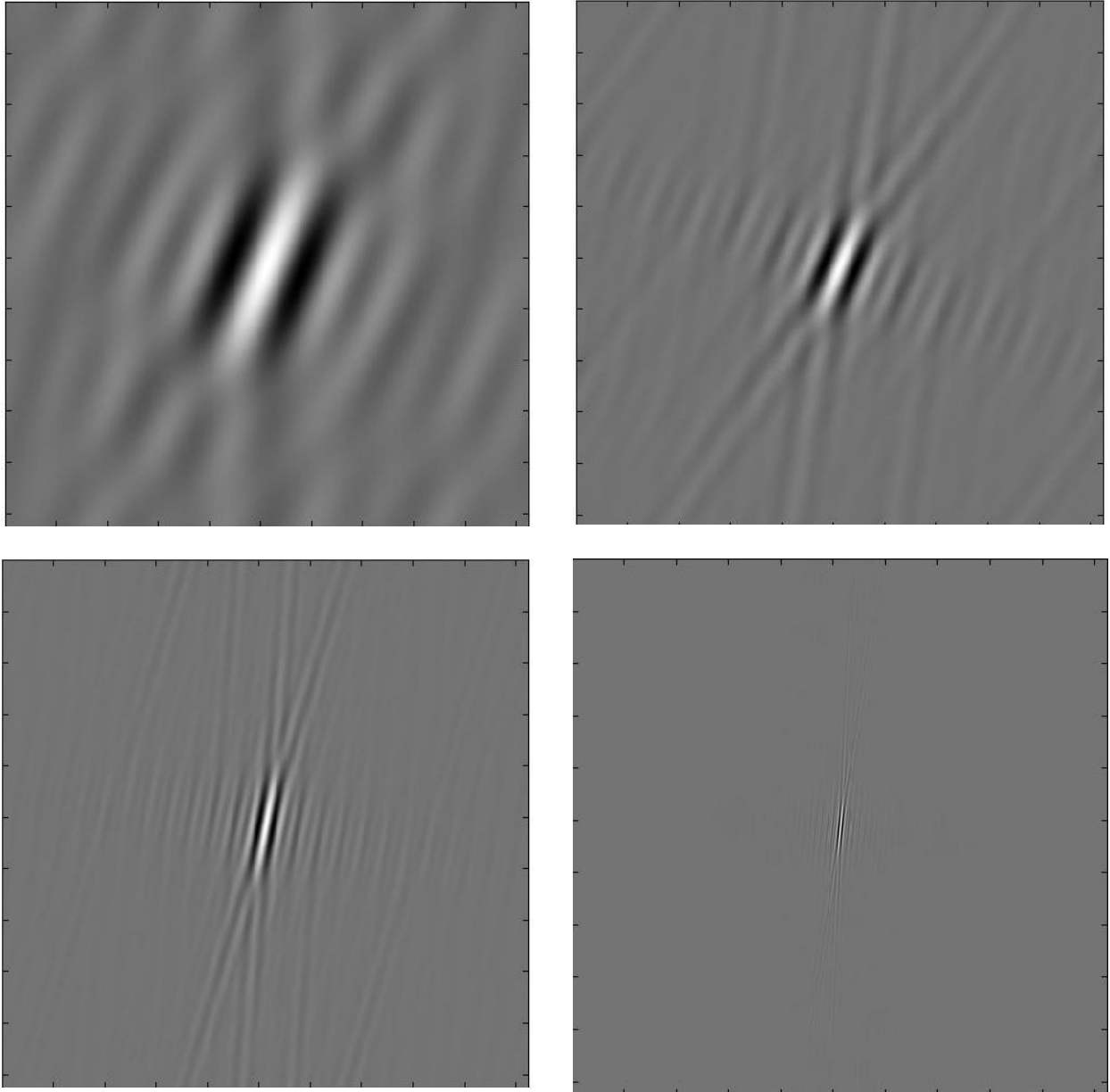


Figure 14: Curvelets at different scales; in order, $j=3, j=4, j=5, j=7$.

Next, consider the Radon data and the format of the acquired Radon data. On the original image, a line is made through the image with the equation $x \cos \theta + y \sin \theta = s$, where θ is the angle and s is the distance from the center. The integral of each line is a number that is put into a matrix whose columns represent θ and rows represent s . See Figure 15 for an example.

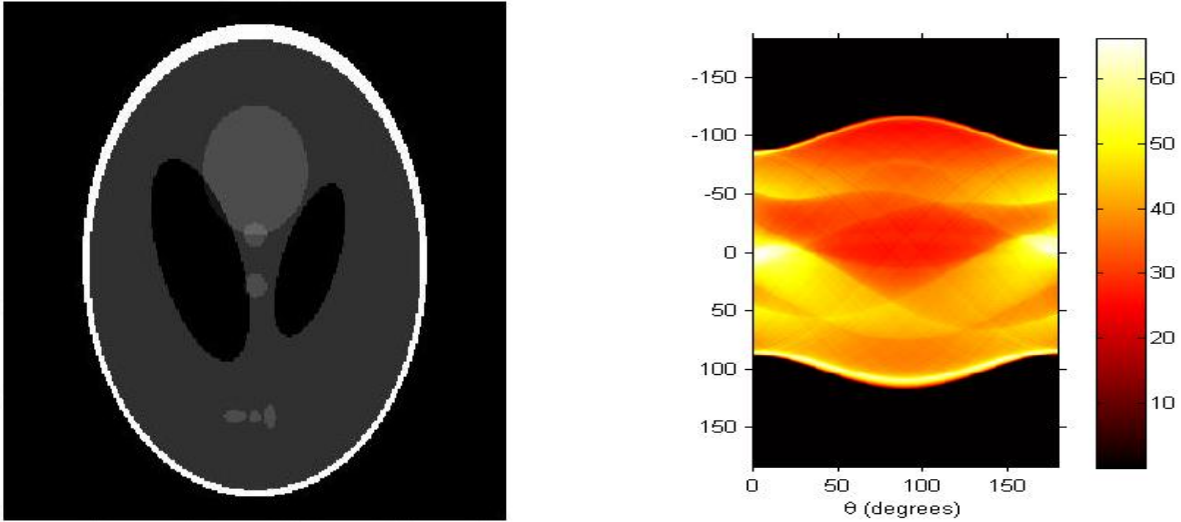


Figure 15: Modified Shepp-Logan head phantom (left) and its Radon transform (right).

After doing the Radon transform on an image, the experimental code will use the equations formulated in the methodology section to estimate the expected curvelet coefficient. Consider the image on which the experiment will be done, shown in Figure 16, along with the image of its Radon transform.

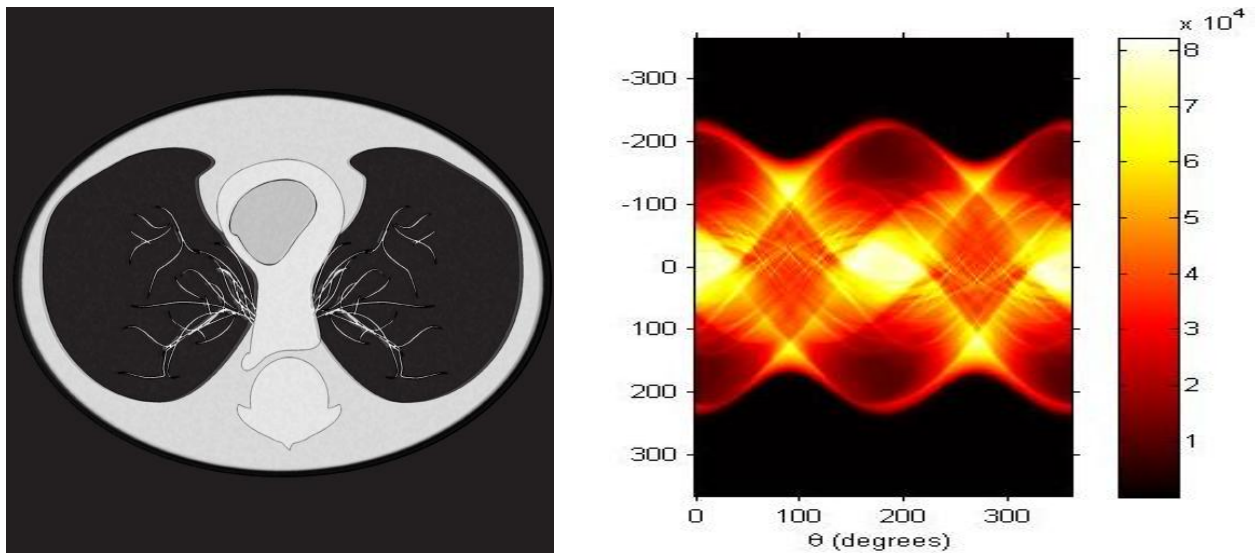


Figure 16: Test image to estimate curvelet coefficients(left) and corresponding Radon transform(right)

In order to compare the methods of curvelet coefficient extraction, the next thing is to choose which curvelets to inspect. Figure 17 shows five curvelets and the respective location in the test image.

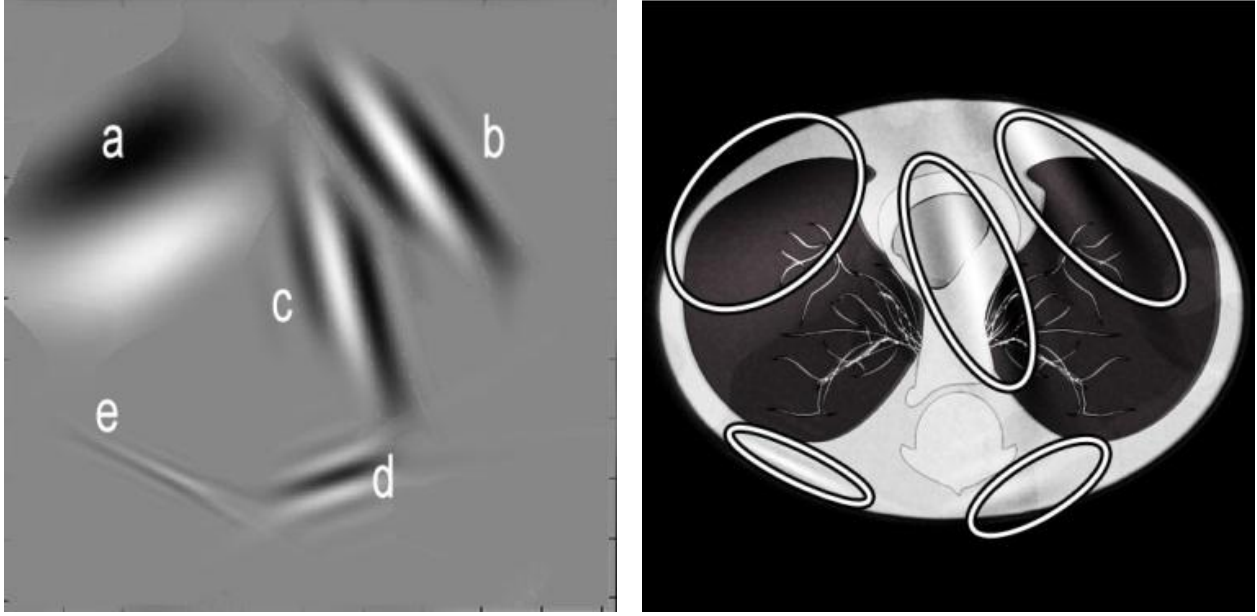


Figure 17: Curvelets of interest in test image

The test curvelets are in areas of interest in the image, and so would be best for comparing coefficient values. To calculate the coefficients, the curvelets will be estimated using a summation.

$$\sum_{x=-256}^{256} \sum_{y=-256}^{256} \overline{\psi_{j,k,l}(x,y)} * f(x,y) \Delta x \Delta y \quad (33)$$

Looking at the curvelet examples in Figure 14, one can notice a ripple effect. This ripple effect will occur because of the conversion from the Fourier domain. The ripples go on infinitely, but the magnitude of the ripples decreases as the distance from the curvelet increases. These ripples

make computation slow. To counteract this, a person could have the code set all small values to zero, which will make the computation more efficient.

Before computing the coefficients for each curvelet, the code will calculate estimations for U_m .

Those values are used with the Radon data to compute the coefficients.

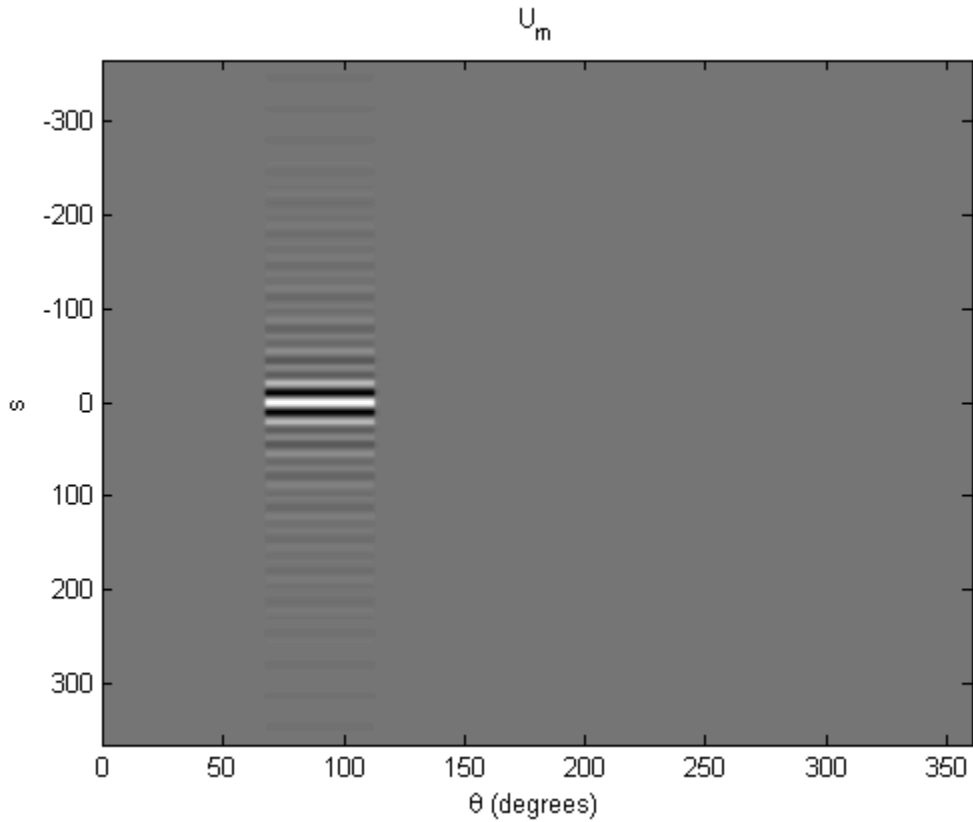


Figure 18: Image representation of example U_m

Figure 18 shows an image for the approximation of U_m . Notice the dimensions are the same as those of the Radon data in Figure 16, where the rows are the radial values and the columns are the angle values. Also, notice that U_m is nonzero only for a specific window with the angular variable. In addition, the U_m quantities have the same ripple effect as that of the curvelets, so the

values decay for large values of s . This gives hope toward localization using curvelets. From these numbers, the curvelet corresponding coefficients will be computed using the formula,

$$\sum_{s=-364}^{364} \sum_{t=0}^{360} U_{j,k,l} \left(s, \frac{t * \pi}{180} \right) * Rf \left(s, \frac{t * \pi}{180} \right) \Delta s \Delta t \quad (34)$$

Example 6: Comparing the Methods for Finding Curvelet Coefficients

Curvelet Found	Method		Relative Error
	$\langle f, \psi_m \rangle$	$2^j \langle Rf, U_m \rangle$	
A	7750590	7691364	.4%
B	2514301	2464984	1.9%
C	3262012	3227391	1.1%
D	1400844	1397639	.2%
E	2300461	2193825	4.6%

Comparing the values in the table, the first thing to notice is how large the numbers are. This is because, for this experiment, each pixel in the 512x512 image was set to a weight of one square unit. Therefore, this experiment is equivalent to a graph with the boundaries $-256 \leq x \leq 256$ and $-256 \leq y \leq 256$ with each unit square with a specific weight. This weight was determined by the grayscale shade in the image where 0 is black and 256 is white. For parts of this function, specifically the areas of interest, the intensity can get very large, so the corresponding coefficients are large.

The next detail to notice is the difference between the coefficients found with the method using the known function and the coefficients found using the Radon transform of the function. One

cause of this difference is the “ripple effect” seen in the approximated curvelets. The extra weight carried in the ripples will cause the difference to increase. This difference can also be attributed to the code approximating the coefficients using a summation. The error is compounded at each step of the summation. With this in mind, the error can improve by taking a tighter sum. Since the coefficients are calculated through integrals, the approximating sum will improve with smaller intervals of approximation.

The relative error in curvelet e is much higher than would be expected with this pattern. Figure 19 shows curvelet e . Notice that the curvelet ripple is wrapping around the image and appearing on the other side of the image. The aliasing effect causes the curvelet to “pick up” values from incorrect parts of the function. To improve the error in the curvelet coefficient calculation, the wrap around issue needs to be addressed.

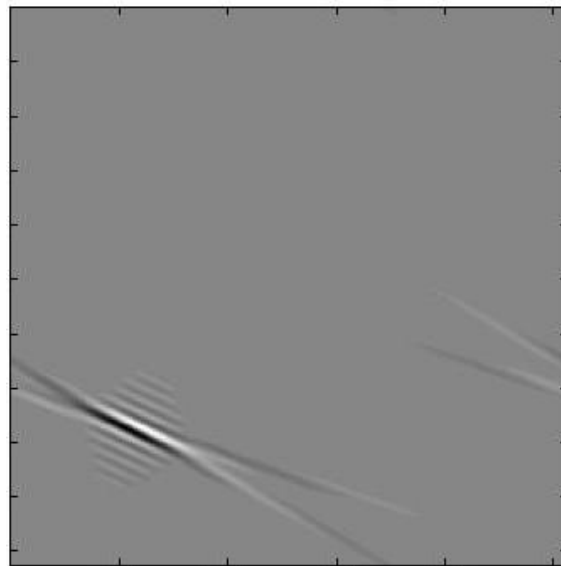


Figure 19: Curvelet e with aliasing prominent.

Many curvelets can overlap in one area when representing a function as a sum of curvelets. This makes the method extremely robust. However, finding each curvelet coefficient using the Radon data is not yet as efficient as FBP. This might be a small price to pay for the robustness of the method. On the other hand, in practical use, the patient would need to be able to be diagnosed as soon as possible.

CHAPTER FIVE: CONCLUSION

Computed tomography was created in the 1970's. With increased power in technology, more time and money is being spent each year to create more efficient and accurate methods for data analysis and image reconstruction. The results in this study show the possibility for improvement in image reconstruction using curvelets. By efficiently encoding the edge information, curvelets can create a sparser image representation which might allow for more efficient image reconstruction.

The coefficients can be calculated using the Radon data. Therefore, the next step would be to extract the curvelet coefficient from partial or truncated data. Future research on curvelets may show that these functions are an accurate representation for more medical uses. Localizing image reconstruction would be useful, requiring less radiation for the patient. Finding an effective way to invert the radon data, localized in the problem area, will lower the amount of harmful X-rays. To reduce the negative effects of the damaging radiation, the ability to focus on a specific part of the body is critical. Previous studies have shown the ability to localize the inversion of radon data using wavelets [19]. Those results could be extended into using curvelets as a basis in reconstruction.

It is important to note that each curvelet's corresponding coefficient was calculated using an approximation. Therefore, these experiments indicate a "rough" estimation for the coefficients. The true values can be calculated more accurately, but will be more costly. Using a higher processing machine will help improve efficiency. For more accurate results in calculating the curvelet coefficients, the issue of aliasing also needs to be addressed.

In this paper's experiment, each of these coefficients were computed one at a time, so reconstructing an entire image may be too computationally expensive. With current processes, the computational cost of the curvelet transform is higher than that of some of the competing methods, such as FBP. Fast curvelet based algorithms need to be used to make this approach practically relevant. The development of the Fast Discrete Curvelet Transform made the curvelet transform more efficient, which will lead to more practical applications for curvelets [12]. An option might be to create a more efficient "pseudo-curvelet" transform [17]. This transform would have most of the characteristics of curvelets, but be more efficient.

APPENDIX: CODE

```

% reconstruction.m does a partial reconstruction of an image using functions
% in Curvelab.

% Set the percentage of coefficients used in the partial reconstruction
pctg = 0.3;

% Load image
%x= phantom('Modified Shepp-Logan'); %load Lena; X = Lena; clear Lena;
%x=imread('C:\Users\Jill\Desktop\MATH programs\AroundDeHeadCartoon','jpg');

x=imread('C:\Users\Jill\Desktop\MATH programs\thesis\Jill Set\Stuff to try
for doing magic on\grandcyanon2','jpg');
x=rgb2gray(x);
X=imresize(x,[256 256]);

% Forward curvelet transform
disp('Take curvelet transform: fdct_usfft');
tic; C = fdct_usfft(double(X),0); toc;

% Get threshold value
cfs = [];
for s=1:length(C)
    for w=1:length(C{s})
        cfs = [cfs; abs(C{s}{w}(:))];
    end
end
cfs = sort(cfs); cfs = cfs(end:-1:1);
nb = round(pctg*length(cfs));
cutoff = cfs(nb);

% Set small coefficients to zero
for s=1:length(C)
    for w=1:length(C{s})
        C{s}{w} = C{s}{w} .* (abs(C{s}{w})>cutoff);
    end
end

disp('Take inverse curvelet transform: ifdct_usfft');
tic; Y = ifdct_usfft(C,0); toc;
Z = afdct_usfft(C,0);
F = ifftshift(fft2(fftshift(Z)));

figure
colormap gray; imagesc(real(X)); axis('image'); title('original image');
figure
colormap gray; imagesc(real(Y)); axis('image'); title('partial
reconstruction');

```

```

function curvelet=one
tic
[P_hat P_hat]=curvelet_MATRIX(5,-45,-45,10);

%%% To make a sparser matrix, set small coefficients to zero by uncommenting
%%% the following code.
%for i=1:512
%   for j=1:512
%       if abs(P_hat(i,j))<=.1
%           P_hat(i,j)=0;
%       end
%   end
%end

curvelet=real(P_hat);
colormap gray; imagesc(curvelet); axis('image');
A=zeros(512,512);

pic=imread('C:\Users\Jill\Desktop\MATH programs\thesis\asdfasdf','jpg');
pic=rgb2gray(pic);
X=imresize(pic,[512 512]);

for x=1:512
    for y=1:512
        A(x,y)=real(X(x,y))*real(curvelet(x,y));
    end
end
Approx=sum(A,2);
approx=sum(Approx)
toc

```

```

function U=U_m_MATRIX(j,l,k1,k2,m,n)

% this function creates a matrix of Um values to use with the Radon data

tic
alpha=2^j;
Kposition= [k1/alpha k2/sqrt(alpha)];
bkjl=Kposition*inv(r(thetaJL(j,l)));
%+2*pi
integral=0;
U=zeros(m,n);
for s=1:m
    s0=ceil(m/2)-s;
    for w0=1:n
        w1=(w0-1)*pi/180-2*pi;
        beta=-bkjl*theta0(w1+thetaJL(j,l));

        for count=1:alpha
            t=2^(j+1)/3+(count-1);
            integral(1,count)=abs(t)*exp(1i*s0*(t-beta))*w(t/alpha);
        end
        approx=sum(integral)/alpha;
        vwindow=v((2^(ceil(j/2)+1)/pi)*(w1+thetaJL(j,l)));
        c=2^(-7*j/4)*sqrt(2*pi)*vwindow*approx;
        U(s,w0)=c;
    end

    end
toc
% figure
% imagesc(real(U))

end

```

```

function [P P_hat]=curvelet_MATRIX(j,k1,k2,l)

P=zeros(129,129);
alpha=2^j;
Kposition= [k1/alpha k2/sqrt(alpha)];
bkjl=inv(r(thetaJL(j,l)))*Kposition';
for x=1:512
    x0=x-255;
    for y=1:512

        y0=256-y;
        s=sqrt(y0^2+x0^2);
        if x0>0
            w0=atan(y0/x0)-2*pi;
        elseif x0<0
            if y0<=0
                w0=atan(y0/x0)-pi;
            else
                w0=atan(y0/x0)-pi;
            end
        else
            if y0<=0
                w0=pi/2-2*pi;
            else
                w0=3*pi/2-2*pi;
            end
        end

        % for j of even value, insert w0= w0+2*pi, otherwise output is
        % automatic zero.

        beta=[y0 x0];
        e=beta*bkjl;

        b=exp(-1i*e)*alpha^(-3/4)*w(s*alpha^-1)*v((w0+thetaJL(j,l))/thetaJL(j,l));
        P(y,x)=b;
    end
end

P_hat=ifftshift(fft2(fftshift(P)));

```

```
function theta=theta0(omeg)
theta=[cos(omeg) sin(omeg)]';
end
```

```
function thetajl= thetaJL(j,1)
thetajl=1*pi*2^-(ceil(j/2)+1);
end
```

```
function vwindow=v(x)
if abs(x)<=1/3
vwindow=1;
elseif abs(x)>2/3
vwindow=0;
else
vwindow=cos(pi/2*Nu(3*abs(x)-1));
end
```

```
function Nu_value=Nu(x)
if x<=0
Nu_value=0;
elseif x>=1
Nu_value=1;
else
Nu_value=exp(-(1/x^2+1/(2-x)^2))/(exp(-(1/x^2+1/(2-x)^2))+exp(-
(1/(1+x)^2+1/(1-x)^2)));
end
```

```
function Rotation= r(x)
Rotation=[cos(x) sin(x); -sin(x) cos(x)];
end
```

```
function wwindow=w(x)
if x<=5/3
if x<=4/3
if x<=5/6
if x>=2/3
wwindow=cos(pi/2*Nu(5-6*x));
end
wwindow=0;
end
wwindow=1;
end
wwindow=cos(pi/2*Nu(3*x-4));
else
wwindow=0;
end
```



```

function [R, xp]=radar

iptsetpref('ImshowAxesVisible','on')
% I = zeros(100,100);
% I(25:75, 25:75) = 1;

% Load Picture

%I=phantom('Modified Shepp-Logan');
% I=imresize(I,[256 256]);
x=imread('C:\Users\Jill\Desktop\MATH programs\thesis\Jill Set\Stuff to try
for doing magic on\grandcyanon2','jpg');
x=rgb2gray(x);
X=imresize(x,[512 512]);

theta = 0:360;
[R,xp] = radon(X,theta);
figure
imshow(R,[],'Xdata',theta,'Ydata',xp,...
        'InitialMagnification','fit')
xlabel('\theta (degrees)')
ylabel('x')
colormap(hot), colorbar
iptsetpref('ImshowAxesVisible','off')

```

LIST OF REFERENCES

- [1] G. T. Herman, *Fundamentals of Computerized Tomography: Image reconstruction from projection*, 2nd ed., Springer, 2009.
- [2] E. Candes and D. Donoho, "Curvelets – a surprisingly effective nonadaptive representation for objects with edges," in *Vanderbilt University Press*, Nashville, 2000.
- [3] K. Lange and R. Carson, "EM reconstruction algorithms for Emission and transmission tomography," *Journal of Computer Assisted Tomography*, no. 8, pp. 306-316, 1984.
- [4] B. Grignon, L. Mainard, M. Delion, C. Hodez and G. Oldrini, "Recent advances in medical imaging," *Surgical and Radiologic Anatomy*, vol. 34, pp. 675-686, 2012.
- [5] G. S. Hurlock, H. Higashino and T. Mochizuki, "History of cardiac computed tomography," *International Journal of Cardiovascular Imaging*, pp. 31-42, April 2009.
- [6] Siemens Medical, "Computed Tomography: Its History and Technology," Forchheim, Germany.
- [7] M. Bentourkia, "Understanding by seeing before treating," *Computerised Imaging and Graphics*, vol. 36, pp. 515-518, 2012.
- [8] W. R. Hendee and M. K. O'Connor, "Radiation Risks of Medical Imaging," *Radiology*, vol. 264, no. 2, pp. 312-321, August 2012.
- [9] K. L. Winger, "Foundations of CT: The Radon Problem," *Radiologic Technology*, vol. 84, no. 2, pp. 206-211, November/December 2012.
- [10] S. Helgason, *The Radon Transform*, Second ed., Boston: Birkhäuser, 1999.
- [11] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 1998.
- [12] E. Candes, L. Demanet, D. Donoho and L. Ying, "Fast discrete curvelet transforms," *Multiscale Model Simulations*, vol. 5, no. 3, pp. 861-899, 2006.
- [13] J. Ma and G. Plonka, "The Curvelet Transform," *IEEE Signal Processing Magazine*, pp. 118-133, March 2010.
- [14] B. F. Hutton, "An Introduction to Iterative Reconstruction," *Alasbimn Journal*, vol. 5, no. 18, pp. Article N° AJ18-6, October 2002.

- [15] J. Friel, "Sparse regularization in limited angle tomography," *Applied and Computational Harmonic Analysis*, 2012.
- [16] E. J. Candes and D. L. Donoho, "Recovering Edges in Ill-Posed Inverse Problems: Optimality of Curvelet Frames," *Annals of Statistics*, vol. 30, no. 3, pp. 784-842, 2002.
- [17] J. Ma and G. Plonka, "A Review of Curvelets and Recent Applications," *IEEE Signal Processing Magazine*, 2008.
- [18] L. Demanet, E. Candes, D. Donoho and L. Ying, "Curvelab," [Online]. Available: <http://www.curvelet.org/>.
- [19] D. Walnut and C. Berenstein, "Local Inversion of the Radon Transform in Even Dimensions Using Wavelets," Institute for Systems Research, 1993.
- [20] L. Shepp and Y. Vardi, "Maximum likelihood reconstruction for emission tomography," *IEEE Trans Med Imaging*, no. 1, pp. 113-122, 1982.