The Fourier Transform and MRIs

Abby Miller

August 3, 2012

Abstract

The ability to see the internal structure of the body is important, not only for research and the continuation of knowledge, but also for medicinal purposes. Diseases and other injuries that otherwise would not have been noted can be found and treated. The problem arises in transferring the incoming signal into an image that can be seen and data that can be analyzed. My work involved learning the background behind such a process, groundwork that can eventually develop into more in depth research. The most basic way of recovering the image from the signal is using the inverse Fourier Transform. The Fourier Transform expresses a function in its frequency spectrum, a spectrum of sine and cosine functions that are contained within the function. Using the inverse Fourier Transform, the image can be seen and different image processing can occur, such as sharpening or smoothing the image. The Fourier Transform is just an approximation, however. In the future, more accurate approximation, as well as different techniques to image processing, could be found and may even replace the Fourier Transform technique.

1 Introduction

In 1977, the first Magnetic Imaging resonance (MRI) scan was performed. MRI scanners are a tool used to see inside the human body and look at the different tissues without cutting the body open. Diagnostics conclusions can be made without having to cut the body open, and a number of different diseases can be examined earlier, potentially saving lives. The way the MRI scanner works is by creating a magnetic field using electricity passing through wire. Radio waves are sent which react with the protons, which in turn release an energy signal that is sent to the computer. [3] Once the signal reaches the computer, it must be transformed into an image. This past summer, my work was to learn the background behind the process of creating an image out of the signal received from the MRI scanner. The information I learned could be developed into a more in depth research project. However, with only ten weeks to learn an entirely new concept, there is hardly time for expanding upon that. The actual process is used by the Fourier Transform, developed by Joseph Fourier, a French mathematician, in the early 1800s. [1] This process is the most common way of recreating the image of an MRI scanner. Once I had taught myself the Fourier Transform and had simulated some data using MatLab, I also learned several image processing techniques, different filters that are common when using the MRI scanner. [4]

2 The Fourier Series

Before a person can begin to learn about the Fourier Transform, the Fourier Series must be first examined. Also developed by Joseph Fourier, the Fourier Series is a way of rewriting a function in a series of sine and cosine functions. In this way, it is very similar to the Taylor Series which using a series polynomials to rewrite a function. Like the Taylor Series, the Fourier Series is an infinite series and can be approximated to designated *N*. The Fourier Series only works with periodic functions with period *T*. The way the Fourier Series is written is

$$f(x) = a_0 + \sum_{n=0}^{\infty} \left[a_n cos\left(\frac{2\pi nx}{T}\right) + b_n sin\left(\frac{2\pi nx}{T}\right) \right]$$

The coefficients can be found by

$$a_0 = \frac{1}{T}\int_{T/2}^{-T/2} f(x)dx$$

$$a_n = \frac{2}{T}\int_{T/2}^{-T/2} cos\left(\frac{2\pi nx}{T}\right) f(x)\, dx$$

$$b_n = \frac{2}{T}\int_{T/2}^{-T/2} sin\left(\frac{2\pi nx}{T}\right) f(x)\, dx$$

To approximate the Series, the summation then, instead of continuing to infinity, can be stopped at the *Nth* value. [2] For a triangle wave function

$$f(x) = \begin{cases} x + L/2 & -L \leq x \leq 0 \\ x - L/2 & 0 \leq x \leq L \end{cases}$$

where L is equal to half the period, the coefficients would be

$$a_0 = 0$$

$$a_n = \begin{cases} \frac{4L}{n^2\pi^2} & \text{n is even} \\ 0 & \text{n is odd} \end{cases}$$

$$b_n = 0$$

The Fourier Series for this particular triangle wave would then be

$$f(x) = \sum_{\text{odd n=1}}^{\infty} \left(\frac{4L}{n^2\pi^2}\right) cos\left(\frac{\pi nx}{L}\right)$$

The Figure 1 shows the original function and Figure 2 the Fourier Series approximated to *N=1*, *N=3*, *N=5*.

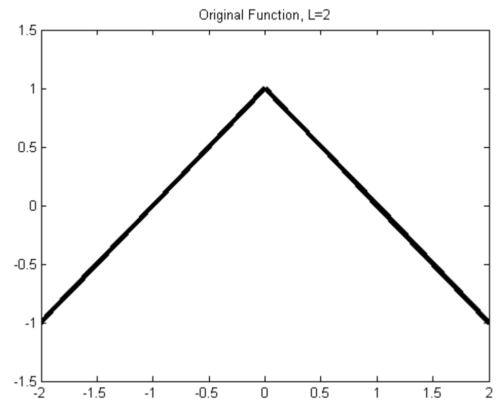

Figure 1: Original Triangle Function

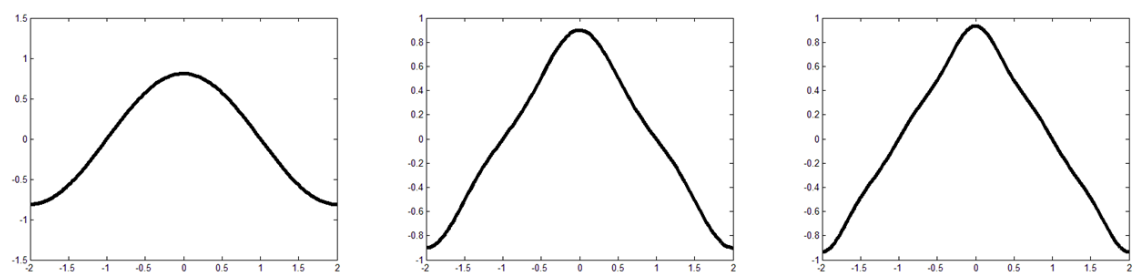

Figure 2: Fourier Series N=1, N=3, N=5

3 The Fourier Transform

The Fourier Series is only used in periodic functions. Therefore, the Fourier Transform is needed for aperiodic functions. Because all the incoming signals from the MRI scanners will be aperiodic, the Fourier Transform will be used.

The Fourier Transform creates a frequency spectrum of *f(x)*.Where the Fourier Series was a way of rewriting the original function, the Fourier Transform displays all the different sine and cosine functions that exist inside the original function. This would be similar to taking the Taylor Series and instead of writing it as a sum of polynomials keeping track of all the different pieces. For every x^2 variable that is found, a mark is placed at two, and for every x^5 variable, another mark is placed at five. The Fourier Transform, in this regard, is like a prism. When a light is passed through a prism, it does not look the same once it has gone through. Instead, the prism splits apart the light and shows all the different colors that create the original lights color. The Fourier Transform takes the original function and breaks it into all the different sine and cosine functions that make it up. If the Fourier Transform where performed on only one frequency, the amplitude of that frequency would be the result. The way that this is done is by multiplying $f(x)$ by $e^{-i\pi kx}$ and then integrating over the set of all real numbers. In formula, this looks like

$$F(k) = \int_{-\infty}^{+\infty} f(x)e^{-i2\pi kx}\,dx$$

also denoted

$$\mathscr{F}\{f(x)\}$$

The inverse Fourier Transform would then be

$$f(x) = \int_{-\infty}^{+\infty} F(k)e^{+i2\pi kx}\,dk$$

or

$$\mathscr{F}^{-1}\{F(k)\}$$

If a function has a Fourier Transform, the function and its Fourier Transform are known as a Fourier Transform pair. This is because the two are distinct pairs. If the Fourier Transform is known, then the inverse can be taken to see the original function. Likewise, if the function is known, the Fourier Transform can be computed. [2] To show an example, take the equation (Figure 3)

$$f(x) = cos(2\pi x) + cos(4\pi x) + cos(6\pi x)$$

To break it into its different frequencies would look like this (Figure 4)

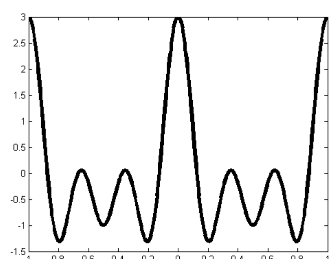

Figure 3: Original function

$$f_1(x) = cos(2\pi x)$$

$$f_2(x) = cos(4\pi x)$$

$$f_3(x) = cos(6\pi x)$$

In the image below, the original function can be seen, along with the different frequencies that make it up. Taking the Fourier Transform of the individual frequencies looks like this (Figure 5)

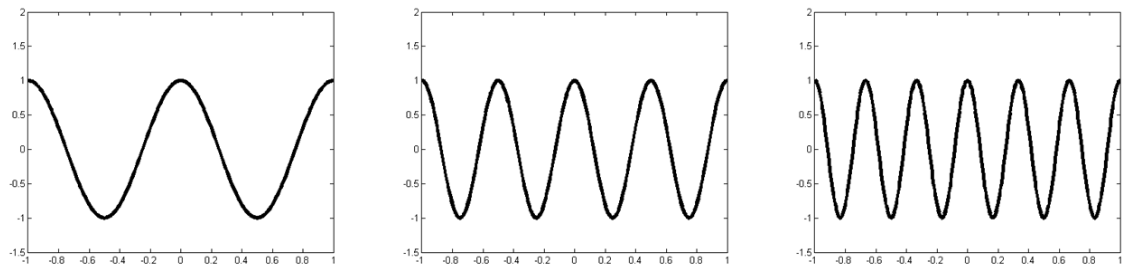

Figure 4: Frequencies Inside the Function

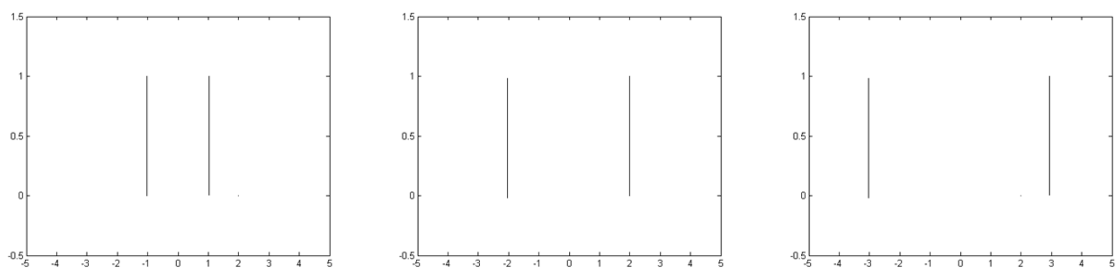

Figure 5: Fourier Transform of Separate Frequencies

$$F_1(k) = \frac{1}{2}\delta(k-1) + \frac{1}{2}\delta(k+1)$$

$$F_2(k) = \frac{1}{2}\delta(k-2) + \frac{1}{2}\delta(k+2)$$

$$F_3(k) = \frac{1}{2}\delta(k-3) + \frac{1}{2}\delta(k+3)$$

Adding them all together to form the Fourier Transform of *f(x)* would become

$$F(k) = \frac{1}{2}\delta(k-1) + \frac{1}{2}\delta(k+1) + \frac{1}{2}\delta(k-2) + \frac{1}{2}\delta(k+2) + \frac{1}{2}\delta(k-2) + \frac{1}{2}\delta(k+2)$$

Figure 6 shows these in images. This example only contains real numbers. In general, the Fourier Transform and Inverse Fourier Transform create complex numbers, even if the original function is made of only real numbers. The reason that the function in the example does not have any imaginary numbers is because it only has cosine terms inside of it. Cosine creates the real parts of the Fourier Transform and Inverse. Sine creates the imaginary parts. The real terms are called "inphase" while the imaginary terms are called "quadrature". If there had been any sine functions in the above example, there would have been delta dirac functions multiplied by an imaginary coefficient. This shows the one dimensional Fourier Transform, but the two and three dimensions are very similar and perform the same procedure. [2]

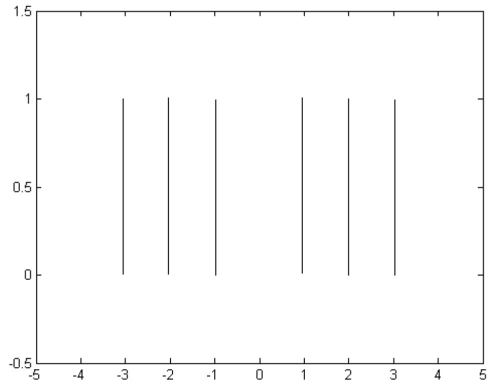

Figure 6: Frequencies Inside the Function

4 The Fourier Transform Properties

The Fourier Transform Properties The Fourier Transform not only exists in distinct, individual pairs, but it has many useful properties, such as linearity, similarity, the shift theorem, the derivative theorem, magnitude and phase, the convolution theorem and the correlation theorem. Of all these, the convolution theorem will be used the most in later image processing. The linearity property is as follows $\mathscr{F}\{(af(x) + bg(x)\} = aF(k) + bG(k)$. This can be helpful in breaking up complicated functions that would be nearly impossible to compute together into smaller, more easily manageable pieces. The similarity property $\mathscr{F}\{f(ax)\} = \frac{1}{|a|}F(k/a)$. By multiplying the x term by a constant, it is just stretching or squeezing the original function. In the Fourier Transform, this constant will either increase or decrease the frequencies at a similar rate. The shift theorem $\mathscr{F}\{f(x-a)\} = e^{-i2\pi ak}F(k)$ is useful in manipulating and correcting data before the image is reconstructed. If the data needs to be shifted, it can easily be done so. Also, the converse is also true, a shift in frequency space by k_0 results in multiplication of the x space by $e^{-2\pi k_0 x}$. The derivative theorem

$$\mathscr{F}\{f'(x)\} = i2\pi kF(k)$$

and its inverse

$$\mathscr{F}^{-1}\left\{\frac{d^l F(k)}{dx^l}\right\} = (i2\pi k)^l f(x)$$

can aid in edge finding. Because edge detection algorithms can become very complicated due to noise, these theorems can make those complicated functions easier to calculate. The magnitude and phase are useful for finding more information about the human body. Many times, the image of the magnitude is looked at and computed as often as the original image, though the phase is seldom examined. If the function comes in as $F(k) = R(k) + iI(k)$ then its magnitude and phase would be

$$|F(k)| = \left\{ [R(k)]^2 + [I(k)]^2 \right\}^{\frac{1}{2}}$$

$$\phi(k) = tan^{-1} \left[\frac{I(k)}{R(k)} \right]$$

The magnitude is useful because, while the incoming function is often complex, its magnitude is always real. The convolution of a pair $f(x)$ and $g(x)$ is

$$f(x) * g(x) = \int_{-\infty}^{+\infty} f(\alpha) g(x - \alpha) d\alpha$$

This is often calculated to determine the effect of the systems on each other. When two functions have convolution performed on them, their Fourier Transform is multiplied together $\mathscr{F}\{f(x) * g(x)\} = F(k)G(k)$. Likewise, if two Fourier Transform functions are convolved, the two original functions are multiplied together $\mathscr{F}\{f(x)g(x)\} = F(k) * G(k)$. This property is particularly useful for performing operations on an image, such as windowing, sub-sampling or, as I used it the most often, smoothing and sharpening. The correlation between $f(x)$ and $g(x)$ is

$$f(x) \circ g(x) = \int_{-\infty}^{+\infty} f^*(\alpha) g(x + \alpha) d\alpha$$

In image processing, correlation is used for template matching. These properties are all described in one dimension. When looked at in two or three dimensions, they are very similar and all apply. [2]

5 Discrete Sampling

The continuous Fourier Transform is useful in theory, but signals cannot be continuously measure for an infinite amount of time. Instead, to sample a continuous function, it would be the same as multiplying that function by the rect and comb functions (see Figure 7 for example). This is the same as measuring the function at a set distance (Δx)for a set period of time. As the intervals between the measuring getting closer and closer, it gives the illusion of being a continuous sampling. Because this sampling is different than a continuous function, the Discrete Fourier Transform was developed. [2]

6 The Discrete Fourier Transform

The Discrete Fourier Transform is written

$$F(p\Delta k) = \sum_{q=-n}^{n-1} f(q\Delta x) e^{-\frac{i2np}{2n}}$$

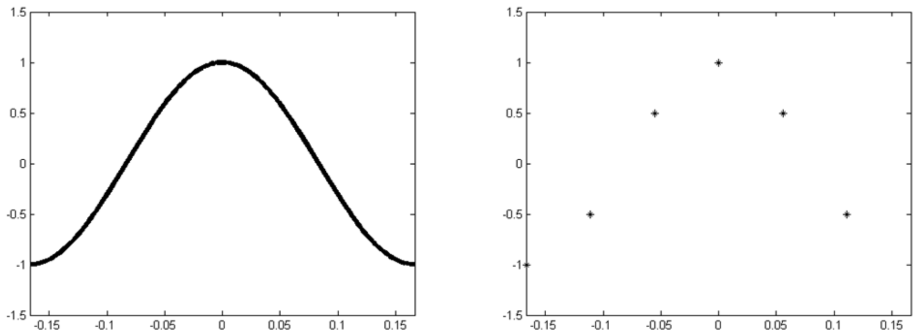

Figure 7: Discretely Sampled Continuous Function

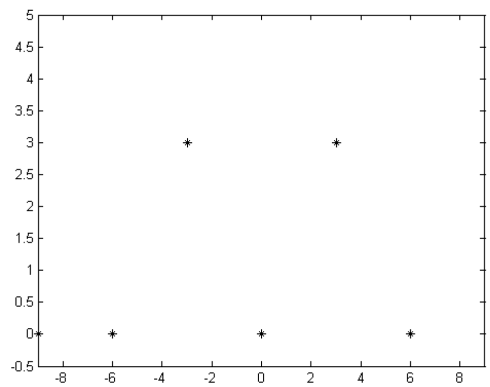

Figure 8: Fourier Transform of Discrete Sampling

where $p = -n, ..., n-1$. The same properties that apply to its continuous counterpart also apply to it. Figure 8 shows an image of the Discrete Fourier Transform of the sample data from above (Figure 7). MRI signals do not come in as one dimensional data, though. Due to the way the MRI reads data, the information most often comes in as a two dimensional matrix or vector. The Fourier Transform for such information is

$$F(p_x \Delta k_x, p_y \Delta k_y) = \sum_{q_y=-m}^{m-1} \sum_{q_x=-n}^{n-1} f(q_x \Delta x, q_y \Delta y) e^{-i2\pi(\frac{p_x q_x}{2n} + \frac{p_y q_y}{2m})}$$

and its inverse

$$f(q_x \Delta x, q_y \Delta y) = \frac{1}{4mn} \sum_{q_y=-m}^{m-1} \sum_{q_x=-n}^{n-1} F(p_x \Delta k_x, p_y \Delta k_y) e^{i2\pi(\frac{p_x q_x}{2n} + \frac{p_y q_y}{2m})}$$

where $p_x = -n, ..., n-1$ and $p_y = -m, ..., m-1$. The inverse is the same as the Fourier Transform, only scaled. When programming in MatLab, I found it easier to use matrix representations of this to compute the two dimensional Fourier Transform. The way to do this is

$$F[p_x \Delta k, p_y \Delta k] = \left(W_y^{p_y(-m)}, ..., W_y^{p_y(m-1)} \right) f(x, y) \begin{pmatrix} W_x^{p_x(-n)} \\ . \\ . \\ . \\ W_x^{p_x(n-1)} \end{pmatrix}$$

where $W = e^{\frac{-i2\pi}{2n}}$. The inverse is the same thing, except the sign switched on $e^{+\frac{i2\pi}{2n}}$ and scaled to $\frac{1}{4mn}$. [2] When I created my own function on Matlab, I first created the W_x and W_Y matrices. Then I multiplied together with the $f(x)$ given to the Matlab function. Below is a picture of the outcome of this process. The original image(Figure 9) is of MatLabs phantom function, a function created to simulate the MRI of a brain for testing purposes. On the other side is the Fourier Transform, performed by my Matlab function (Figure 10).

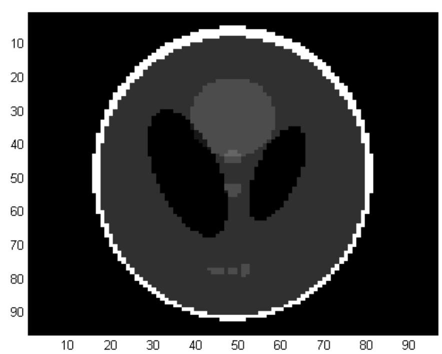

Figure 9: Simulated Image

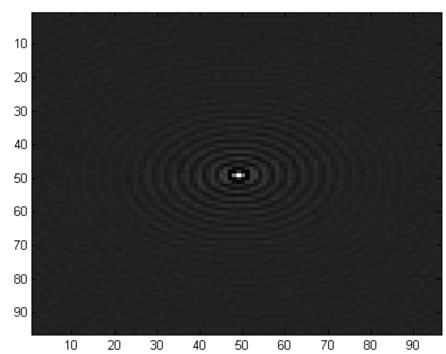

Figure 10: Simulated Incoming Data

7 Magnetic Resonance Imaging

The MRI machine is never turned off. It continually runs, but signal is centered in the middle (see Figure 11). When the machine is in use, the data is then read following the path below, in a zigzag pattern (Figure 12). Rather than reading the entire brain in one data set, the information is read in slices. This is due to the fact that it takes 30 to 40 milliseconds for the information to be read. If ten slices were taken, that would be 300 to 400 milliseconds. In that time, a person could move, corrupting the data, which means that the information needs to be taken another time, and using the machine is expensive. The incoming data from the machine is in the Fourier Transform of the image. Therefore the inverse Fourier Transform must be performed in order to see the original image. Because the data is read as slices, the third dimension can be disregarded, and the two dimensional inverse Fourier Transform may be used. The incoming data is in either vector or matrix form. If the entire path is sent at one time, the data is in a vector form. If it is sent in by rows and columns, then the data is a matrix. The function that I created in MatLab accepts matrices. If the information comes in as a vector, it could easily be transferred to a matrix, as long as the number of rows and columns is known. It is at this time that the image can be processed and any kind of filter, be it smoothing or noise reduction, can be performed. [2]

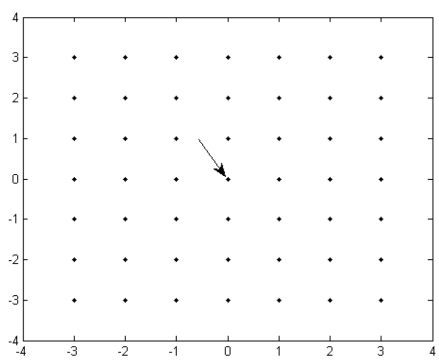

Figure 11: Machine at Rest

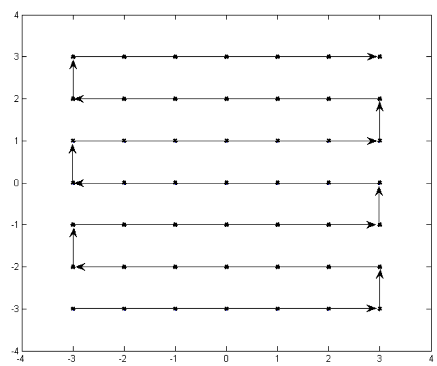

Figure 12: Path machine reads data

8 Image Processing

Once the data has been recorded from the machine, different kind of filters can be used on it. Because of limited time, I was only able to use a few of them. It is during this that convolution is so important. A significantly smaller matrix, called a kernel, is chosen and the kernel and original image are convoluted. Depending on the kernel that is chosen, the image will be filtered accordingly. Several of the methods I will discuss below involve kernels. [2]

8.1 Smoothing Filters

8.1.1 Mean Filter

The first filter I looked at is called a mean filter. The way that this filter works is by first selecting one cell in the matrix. From that one cell, a 3x3 matrix is created, with the chosen cell at its center. All the values in that matrix are then added together and average, the new value replacing the center one. In the figure below, the 3x3 has been chosen from a larger set of data. The center cell is equal to 186. All the other cells are added together and the total divided by nine. The new number to be place is 199.99 which is rounded to 200. The new 3x3 matrix then looks like the second figure.

Original Matrix

$$f(x) = \begin{matrix} 202 & 198 & 207 \\ 195 & 186 & 201 \\ 211 & 189 & 208 \end{matrix}$$

After Mean Filter

$$f(x) = \begin{matrix} 202 & 198 & 207 \\ 195 & 200 & 201 \\ 211 & 189 & 208 \end{matrix}$$

When this process is applied to over the entire image the differences between the cells is less, so the image is blurred. When the edges are reached, the data is wrapped around so that all the different cells of the matrix have the operation performed on it. Figure 13 shows the original image and Figure 14 the new image once the mean filter has been applied. [2]

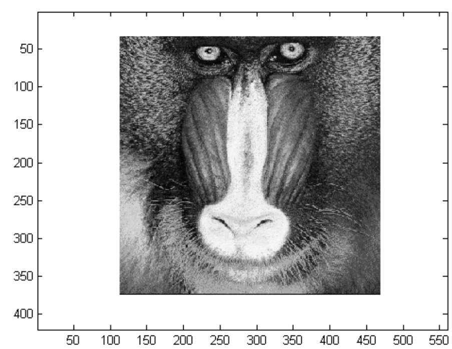

Figure 13: Original Image

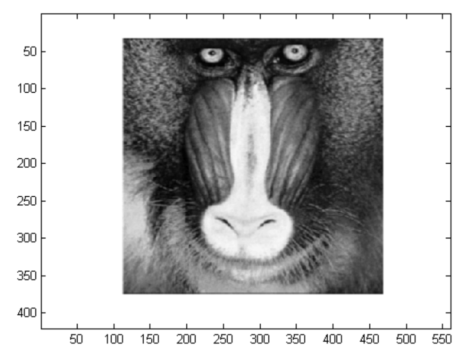

Figure 14: Mean Filtered Image

8.1.2 Median Filter

The median filter works very similarly to the mean filter. Instead of finding the mean of the 3x3 matrix, the median is found instead. The center value is then replaced with the median. Figure 15 shows the original image, followed by the new image after the median filter has been applied (Figure 16). [2]

8.1.3 Gaussian filter

The Gaussian filter works by creating a Gaussian kernel. The way a Gaussian kernel is created is through this formula $g(r,s) = ce^{-\frac{r^2+x^2}{2\sigma^2}}$. r and s are the dimensions of the Gaussian filter. The coefficient c will be chosen later to set the first value in the matrix equal to one. A 5x5 Gaussian filter with variance

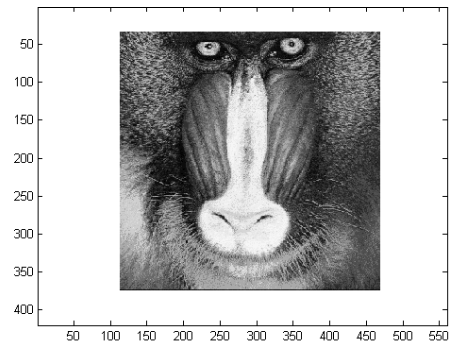

Figure 15: Original Image

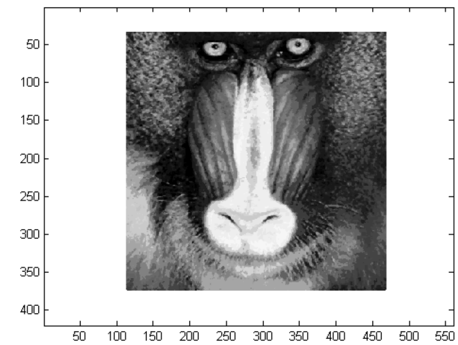

Figure 16: Median Filtered Image

$\sigma^2 = 0.5$ looks like

(r,s)	-2	-1	0	1	2
2	0.0003	0.0067	0.0183	0.0067	0.0003
1	0.0067	0.1353	0.3679	0.1353	0.0067
0	0.0183	0.3679	1.0000	0.3679	0.0183
-1	0.0067	0.1353	0.3679	0.1353	0.0067
-2	0.0003	0.0067	0.0183	0.0067	0.0003

Once that has been found, the c that sets g(2,2) = 1 can be found to be 2981 so the entire filter is multiplied by 2981 and will look like

(r,s)	-2	-1	0	1	2
2	1	20	55	20	1
1	20	403	1097	403	20
0	55	1097	2981	1097	55
-1	20	403	1097	403	20
-2	1	20	55	20	1

However, the values must now be normalized, so all the values added up, adding to 9365. Each number can then be divided by 9365 and when all of them are added up again, the total will be equal to 1. To use this filter, the kernel is convolved with the image to be smoothed, or the Fourier Transforms are multiplied together, smoothing the image. Figures 17 and 18 show an example of the outcome of this filter.

8.2 Sharpening Filters

There are many examples of sharpening filters. Many times they are 3x3 and contain an assortment of negative and positive numbers. An example of a

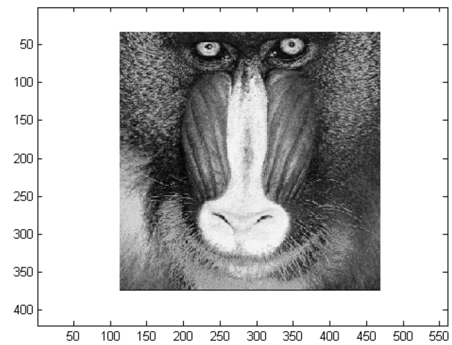

Figure 17: Original Image

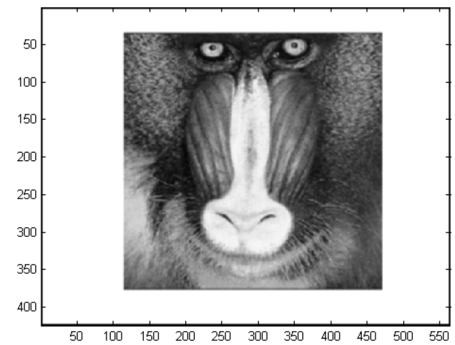

Figure 18: Gaussian Filtered Image

sharpening filter would be

$$g(x,y) = \frac{1}{9} \begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$

This kernel would be convolved with the original function and then the con-

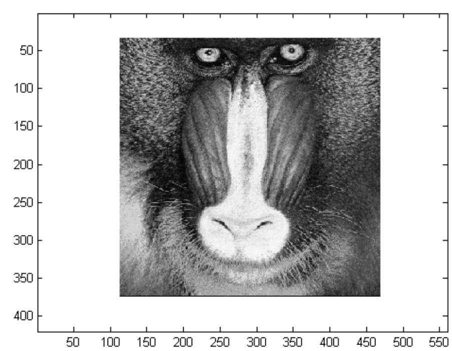

Figure 19: Original Image

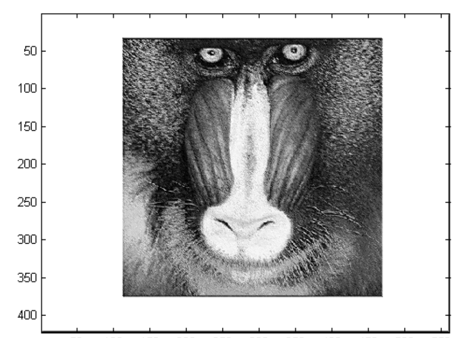

Figure 20: Gaussian Filtered Image

voluted image would be added to the original image to produce the sharpened image. Figures 19 and 20 show an example of this. An important thing to note is that no matter what the numbers in the kernel are, they must sum to zero. Other famous sharpening kernels are: first derivative, Robert Cross gradient operator, Prewitt operator, Sobel operator and 3x3 Laplacian filter mask. Rather than convolution being performed on many of these operators, they are multiplied with a piece of the larger matrix that is the same size with them and added or subtracted together to find the value to go in the middle, similar to the mean smoothing filter. [2]

9 Future Work

While the Fourier Transform is a powerful tool, it is only an approximation. Inside the f(x) used in the equations, there are many assumptions made about that incoming signal that I have not even touched on. A new technique may be developed that could replace the Fourier Transform, giving a more accurate image and information. Even with the current system, new filters could be developed or old ones perfected. I barely began to learn about these filters, but I can see there is much room for expansion. The Fourier Transform is a powerful tool for the MRI, but its applications do not rest simply there. It is currently used in other medical and non-medical processes and perhaps its applications will continue even further.

References

[1]

[2] R. D. B., *The Fourier Transform: a technical understanding with applications to MRI.* Wisconsin, USA: Daniel B. Rowe, 2012.

[3] M. Kalapurayil. (2009, Apr.) What is mri? how does mri work? [Online]. Available: http://www.medicalnewstoday.com/articles/146309.php

[4] H. E. Mark, *Magnetic Resonance Imaging: PHysical Principles and sequence Design.* New York: Wisely-Liss, 1999.