Steganography using the Fourier Transform and Zero-Padding Aliasing Properties (May 2006)

Robert T. McKeon, *Member, IEEE*

Abstract— **Privacy is an important freedom, and in the modern world of technology, the probability of our private data being comprised is more likely than in the days before computers since all of our data can be stored electronically and possibly accessed from an outside location.**

This paper presents a methodology for steganography based on hiding images in the Fourier domain of an image by using the properties of zero-padding an image with itself causing aliasing to occur. Seventy-five percent of the resulting spectrum before any data is embedded in the image, has the value of zero. These zeros can be changed slightly where the change in the image is not noticeable. The data changes due to formatting an image in a 24-bit color format. This change can be accounted for using a transfer function which would also be needed to decrypt the information.

The reason for this approach comes from the flexibility in the amount of data which can be hidden in an image and the location of the hidden data. The proposed method would also be advantageous in the manner that a person would need four images and a transfer function or a key so to speak, to recover the data. They would also have to know what order to put the four images before decryption can occur. This adds a level of complexity to steganography which can help to insure the data being sent is kept private.

***Key Terms*-- Steganography, Fourier Transform, Image Processing, Zero-padding**

I. INTRODUCTION

Privacy is a matter of trust between us and society at large, but some people in society abuse that trust to violate our privacy. To protect ourselves and our personal data along with proprietary information, cryptology was developed to encrypt data, and steganography was developed to hide our data.

Steganography involves hiding data in different forms of media such as music, movies, and pictures so that when these different forms of media are perceived, there is no noticeable difference to the observer. In the modern age of the internet and multi-media functions galore, steganography is an efficient method of hiding a needle in a hay stack.

Steganography in images is done by slightly modifying the each color plane to hide the desired data. Many images are coded using 24-bits, and there are 8-bits (or one byte) used for each of the color planes. By modifying the two least significant bits of each data byte for each color plane, the change in color to the human eye goes unnoticed. The amount of data that can be stored in an image is 6 bits per color pixel. These data bits are also undetectable unless someone has the original image or knows the image has data hidden in it. Then to find this hidden data, they would AND the image with the number 3 (or 00000011 in binary) [1].

Other methods of steganography are carried out by hiding data in the Fourier domain of an image. Also, data is hidden using the Discrete Cosine Transform (DCT) which allows the data to be more resistant to becoming corrupted [2]. The difference between the proposed method and previous methods using the Fourier domain are specifically seen in the way the Fourier transform is taken and the image outputs.

The current amount of counter measures used to detect steganography are limited due to the encoding done with the data to make the image look as it originally was. Usually, many of these techniques can only provide a probability of an image having something hidden in it. Usually, these look at the spectrum of an image to determine if it has been noticeably modified [2].

One method of steganography is carried out by hiding data in the Fourier domain of an image using the Discrete Cosine Transform (DCT) which allows the data to be more resistant to becoming corrupted [2]. The DCT is similar to the Discrete Fourier Transform (DFT), but it does not have an imaginary component. Instead, it is about twice the length of the DFT, and it operates only on real data with even symmetry. This coincides with the DFT since "the Fourier

transform of a real and even function is also real and even". [2]

The equation for the DCT is as follows for 2-dimensions (2D) [3]:

$$B(k_1,k_2) = \sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1} 4\cdot A(i,j)\cdot\cos\left[\frac{\pi\cdot k_1}{2\cdot N_1}(2i+1)\right]\cdot\cos\left[\frac{\pi\cdot k_2}{2\cdot N_2}(2j+1)\right] \quad (1)$$

where (i,j) are pixel points of an image of size N1 x N2. The form above is the DCT-III, and it is commonly used for images.

The 2D Discrete Fourier Transform is given by the equation [4]:

$$F(u,v) = \frac{1}{NM}\sum_{x=0}^{N}\sum_{y=0}^{M} f(x,y)\ \ e^{-j2\pi(\frac{ux}{N}+\frac{vy}{M})} \quad (2)$$

where (u,v) are the new positions in the DFT image, and (x,y) were the positions of pixels in the original image.

Steganography is done using the DCT primarily because hiding data in the Least Significant Bit (LSB) is more easily detectable. Data can also be lost more easily. However, with the DCT, data can be hidden in such a manner that it is robust enough to still be there even if some of the image data is lost or modified [3].

Steganography using the DCT is usually done by modulating the size of two of more DCT coefficients within a block in an image. This is commonly done by splitting an image into 8 x 8 blocks on which the DCT is performed. Then, two frequencies near the middle are chosen to hide the data because this will insure the hidden data is not changed by image quantization/compression used in turning the image into the JPEG format [4]. A similar practice can be done using the DFT. The DFT method is similar to the DCT by taking frequencies in the mid-band region, and modifying their coefficients [5].

The proposed method involved padding an image with itself instead of zeros as normally done. Usually, an image is padded with zeros to make sure that the sampling frequency is high enough so that aliasing will not occur. However, it was found that by imposing the same signal right after itself, and sampling at the minimum rate one normally would for that signal, the resulting spectrum contains mostly zeros. It is in these places in the image that information is hidden with this method.

The information is hidden in the higher frequency places of the spectrum to insure that the data is not noticeably different to the eye or to a computer unless one would have the original. The amount of data hidden in this area of the spectrum can differ based on the user's preference, but there is a limit. For an image of 64 by 64, the image padded with itself is 128 by 128, and an area centered in the middle of the spectrum of this padded image can be at most 85 by 85 before the data can not be recovered within 0.1% error. In this 85 x 85 section of the spectrum, only 75% of the values would be modified since only 75% of these values would be zeros before the information is embedded.

The image has to be converted back to the time domain, and then it has to be put in the unsigned integers eight format so that it can be saved as a JPEG or a BMP. This process changes the data values in the spectrum domain, but it was found that using the same data values being put in the same place, the change which occurs to those values due to formatting the image in the time domain properly, is negligible. This means, there is a direct relation between the data before it is embedded and after it is embedded. This relation is embodied in a transfer function which works within a given error percentage to reform the original information.

II. BACKGROUND

The Nyquist rate is an important theorem in signal transmission. The proposed methodology of segmentation will be implemented through using some of the implications of this theory as later discussed in this section and the next.

In data transmission, digital samples are taken of an analog signal which is transmitted as binary numbers in some transmission medium. Upon receiving the signal, the receiving end of the transmission line decodes the signal and uses the samples to reconstruct the original signal [6].

Nyquist and Shannon found that if the signal is under sampled, it will be corrupted by itself when reconstructed. This distortion is caused by parts of the signal's spectrum, or the frequency domain of the signal, being overlaid on itself. In particular, if the sampling rate or sampling frequency is not high enough, the higher frequencies in a signal are cut off resulting in not being able to reproduce the original signal [6, 7, 8].

Nyquist and later Shannon proved that the sampling frequency must be twice the highest frequency or the bandwidth for the signal to be able to be reproduced without aliasing [7, 8].

$$f_s \geq 2*B \quad (3)$$

where the B is the bandwidth, and f_s is the sampling frequency [6, 7, 8].

$f_s = 1/T_s$ (4)
where T_s is the sampling time period [6, 7, 8].

The frequency domain of a signal and a sampled signal can be seen in Figure 1. The main distinction between the frequency domains of the signal and the sampled signal is that the sampled signal has the signal's spectrum repeated at every f_s. In Figure 1 (b), it also shows that a low-pass filter is used to keep only the original signal. This is very important when examining the aliasing effects as seen in Figure 2. The signal to be sampled is also low passed before it is sampled, and the cut off frequency for this low pass filter is greater than the bandwidth (B). [6]

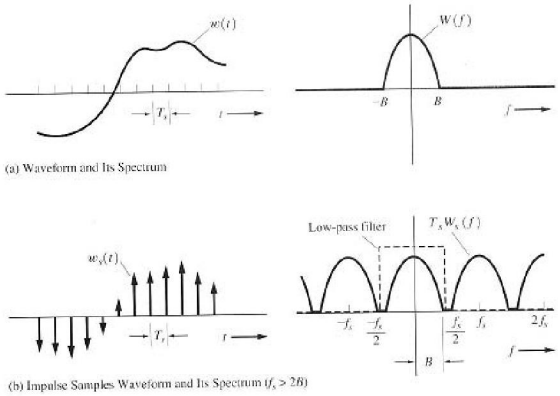

Fig. 1 (a, b): (a) A waveform and its Fourier transform. (b) The waveform in (a) sampled and its Fourier transform [1, page 91, Figure 2-18]

Figure 2 shows visually why aliasing occurs when the sampling rate is not greater than twice the bandwidth. This is due to overlapping the reflections of the original signal's frequency domain.

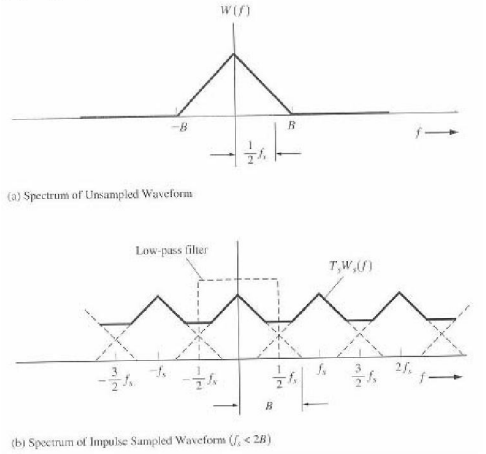

Fig. 2 (a, b): (a) Spectrum of unsampled waveform. (b) Spectrum of a sampled waveform with $f_s \leq 2*B$ [1, page 92, Figure 2-19]

The bandwidth of a signal is the highest frequency required in a signal for it to be meaningful. The trade off with increasing the bandwidth is that the sampling rate increases which is equal to an increase in cost because devices have to record signals faster. At least for audio, the human ear has a frequency range from 20 Hz to 20,000 Hz [9]. So it is unnecessary to have a sampling rate much higher than 40,000 Hz since any extra gain from the added bandwidth would go unnoticed.

Nyquist rate is important for the proposed method of steganography because the proposed method will use some of the implications of aliasing to hide data in a signal. The signal used will be a two dimensional discrete signal since images are stored as a matrix of samples taken by a camera.

III. METHOD

The Nyquist rate is used in image processing by padding an image with zeros before the Discrete Fourier Transform (DFT) is taken. This is particularly important when convolving an image with another image such as a low pass or high pass filter. Otherwise, aliasing will occur, and in images, the effects are seen as rippling coming off of off of details of the image. This ripple effect looks like a drop of water causing a ripple in a pond. However, if the aliasing is very severe, the image will appear to have a Moiré pattern, but for the proposed method of steganography, the Moiré pattern is not extremely important.

If one of the images to be convoluted has dimensions A x B, and the other image dimensions C x D, then the dimensions of the zero padded images for both must be P x Q where P = A + C – 1 and Q = B + D – 1. If the dimensions of the zero-padded image (Figure 3) are less than P x Q, the image will have aliasing. [10]

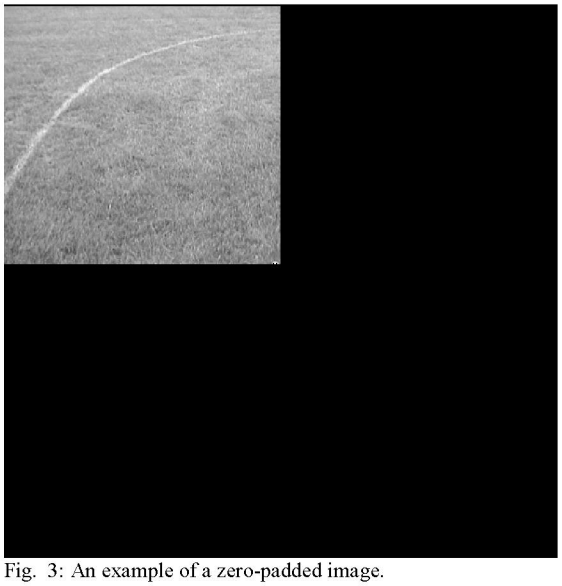

Fig. 3: An example of a zero-padded image.

The DFT is taken on each color plane separately, so the focus will be on one arbitrary plane whether it is the blue, red, or green, it does not matter. The reason the DFT is used in the convolution is that the convolution in the spatial domain is multiplication in the frequency domain as seen in the equations below [10]:

$$f(x,y) = h(x,y) * r(x,y) \Leftrightarrow F(u,v) = H(u,v)R(u,v) \quad (5)$$

While doing experiments to see how different types of padding might affect a convolution, an interesting result was found. When the same image was placed in the three areas of the zero padded part of the image (Figure 2), which are each large enough to contain the image itself, the DFT came out to be mostly zeros as seen in (Figure 3). There is a color bar for the DFT image showing that most of the image is composed of zeros.

Where these zeros are will be where data will be placed, but before starting that, the next part of this paper will show the mathematical proof proving that 75% of the DFT as show in (Figure 3) is zero. This will go to show that this result is not unique to this image or any other image.

The equation below for f(x, y) is a small example representing a 2 x 2 image which is padded with itself. It can be shown that the two-dimensional DFT of f(x, y) (below in Equation 2) is equal to F(u, v) (below in Equation 3). This mathematical result is to show that the result in question was not some random happening.

$$f(x,y) = \begin{bmatrix} a1 & a2 & a1 & a2 \\ a3 & a4 & a3 & a4 \\ a1 & a2 & a1 & a2 \\ a3 & a4 & a3 & a4 \end{bmatrix} \quad (6)$$

$$F(u,v) = \begin{bmatrix} \frac{1}{4}a1 + \frac{1}{4}a2 + \frac{1}{4}a3 + \frac{1}{4}a4 & 0 & \frac{1}{4}a1 - \frac{1}{4}a2 + \frac{1}{4}a3 - \frac{1}{4}a4 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{4}a1 + \frac{1}{4}a2 - \frac{1}{4}a3 - \frac{1}{4}a4 & 0 & \frac{1}{4}a1 - \frac{1}{4}a2 - \frac{1}{4}a3 + \frac{1}{4}a4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (7)$$

IV. RESULTS

The first test done changed the first row of the Discrete Fourier Transform (DFT) of the image padded with itself (Figure 5 (a)). The original image used was 64 x 64 pixels, and only the red plane was used (Figure 4).

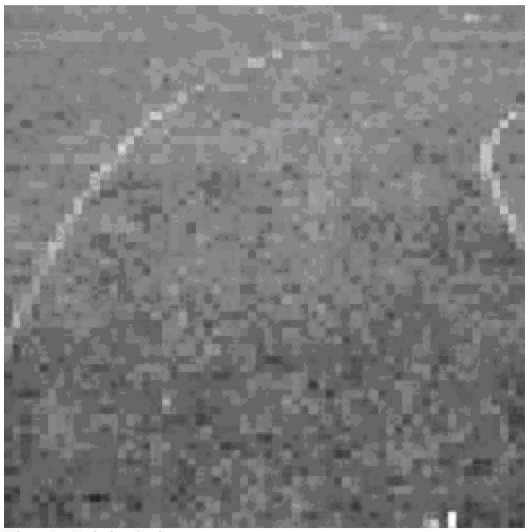

Fig. 4: Original picture.

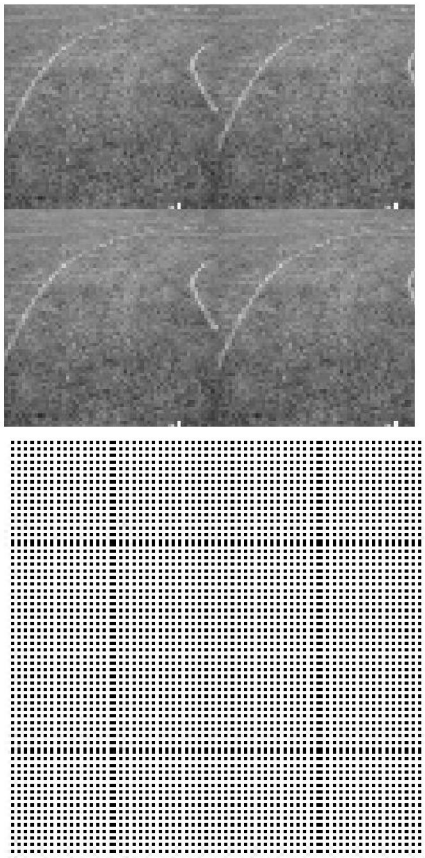

Fig. 5 (a, b): (a) Original picture padded with itself. (b) The zero components of the DFT of the image.

The parts of the DFT that equaled zero are shown as the white pixels in Figure 5 (b). The first row was changed of the DFT to a single value ranging from 1 until 100 to examine the effects of converting the image back using the inverse DFT, and then changing the image in to a format that could be saved like BMP or JPEG. Then the DFT was again applied, and those places where the values were changed, were examined to see how much the values changed.

Until the value changes to 80, the other values can be rounded down to recover the original data. Figure 6 (a) shows the image after change the zeros in the first row of the DFT of the image to 1 while Figure 4 (b) shows the image after changing the zeros in the first row of the DFT of the image to 100. This just shows there is some noticeable change in the image, but not that much.

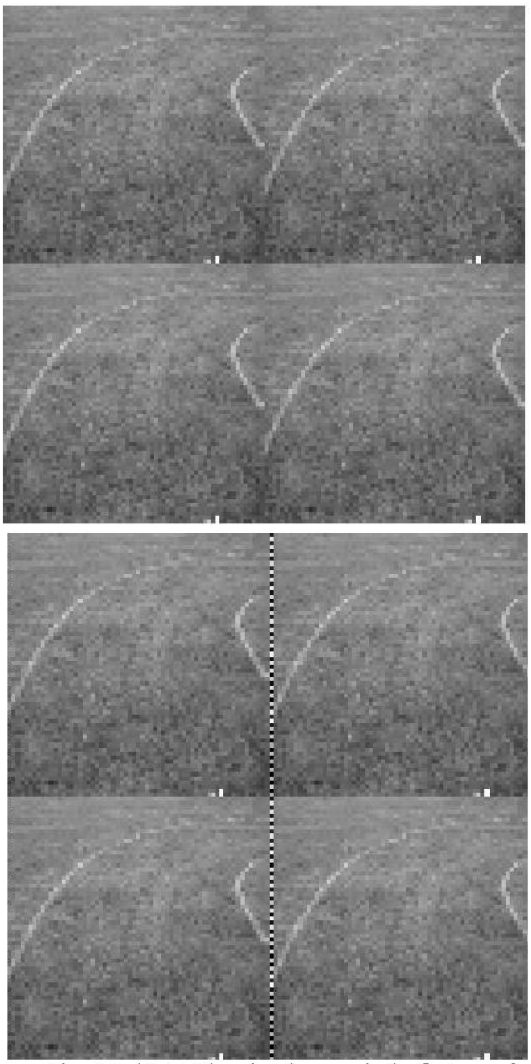

Fig. 6 (a, b): (a) Changing the zeros in the first row of the DFT to 1. (b) Changing the zeros in the first row of the DFT to 100.

The frequencies in the 64 x 64 center area of a spectrum of the image in Figure 4 were changed to include some hidden data resulting in a new image (Figure 7) without a noticeable change to the human eye when compared to the original. There is nothing really noticeably different about the image. Random data was selected which was greater than 0 but less than 2. With the transfer function, the data can be extracted without error.

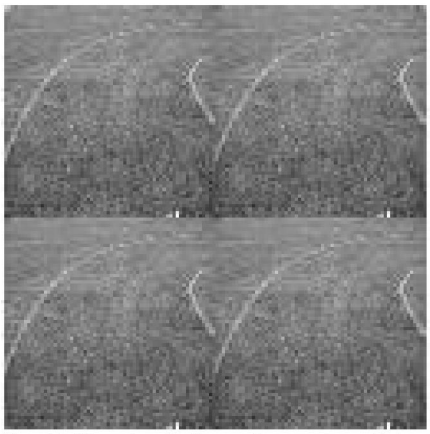

Fig. 7: Changing the lower frequencies which equal zero to certain data points.

This same data could be embedded in the same place in another image, and the percent error for the difference between the resulting data was around 5.149 x 10^-13 %. This shows that the data will change nearly independent of the image it is embedded in. Using a transfer function made from embedding the data into a different image, all the data could be recovered within an error percentage of 3.5%. However, it is not necessary to use a different transfer function, and transfer functions can be made specific for the data and the images used. This can insure not losing any data.

V. FUTURE WORK

Future work should include more testing on making sure the transfer function is valid, and on what range it is valid. A study will also be done on the amount of data that can be hidden in the image.

REFERENCES

[1] Johnson, Neil F; Jajodia, Sushil. Exploring steganography: Seeing the unseen, COMPUTER. Vol. 31, no. 2, pp. 26-34. Feb. 1998

[2] M. Kharrazi, H. T. Sencar, N. Memon, Image Steganography: Concepts and Practice, Lecture Note Series, Institute for Mathematical Sciences, National University of Singapore, 2004.

[3] "Issues in Information Hiding Transform Techniques," NRL/MR/5540-02-8621 http://chacs.nrl.navy.mil/publications/CHACS/2002/2002chang-NRL-MR-5540-02-8621.pdf#search='2D%20DFT%20in%20steganography

[4] Pennebaker, W. & Mitchell, J. "JPEG STILL IMAGE DATA COMPRESSION STANDARD". van Nostrand Reinhold, 1993.

[5] Marvel, L. \Image Steganography for Hidden Communication". Ph.D. Dissertation, Univ. of Delaware, Dept of EE, 1999.

[6] Couch, Leon W. Digital and Analog Communication Systems. Sixth Edition, Prentice-Hall, Upper Saddle River, NJ. © 2001. Pages 86-93.

[7] H. Nyquist, "Certain topics in telegraph transmission theory," Trans. AIEE, vol. 47, pp. 617-644, Apr. 1928.

[8] C. E. Shannon, "Communication in the presence of noise", Proc. Institute of Radio Engineers, vol. 37, no.1, pp. 10-21, Jan. 1949.

[9] Cutnell, John D. and Kenneth W. Johnson. *Physics*. 4th ed. New York: Wiley, 1998: 466.

[10] Gonzalez, Rafael C. and Woods, Richard E. Digital Image Processing, Second Edition. © 2002, Prentice Hall, Upper Saddle River, New Jersey. Pages 199-205.