Passive Warden Using Statistical Steganalysis

David Stacey
School of Applied Technology
Illinois Institute of Technology
201 East Loop Road
Wheaton, IL 60189
dstacey@hawk.iit.edu

ABSTRACT

This paper examines the statistical techniques used in blind steganalysis of JPEG images. Blind steganalysis attempts to detect the presence of covert data without knowing the particular steganographic algorithm used. This paper begins by discussing steganography and steganalysis in general with a focus on common techniques. JPEG images are then introduced with a detailed discussion of their format and how they are created. Once the details of JPEG images are understood, common JPEG steganographic algorithms are explained. These algorithms are available in programs like JSteg, Outguess, and F5. This paper focuses on the Calibration Technique and associated features developed by Fridrich [2].

The goal is to develop an application that detects covert data hidden in JPEG images by common steganographic algorithms. The output from this application is an indicator of whether or not the image contains covert data.

Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: Miscellaneous

Keywords

JPEG; Steganography; Steganalysis

1. INTRODUCTION

This project implements a Passive Warden application, with the goal of detecting the presence of hidden messages in JPEG images. The roots of the Passive Warden comes from an article by G. Simmons entitled "Prisoners' problem and the subliminal channel" [8], which is an illustrated story of Alice, Bob, and Wendy. Alice and Bob are prisoners and Wendy is their warden. Like all prisoners, Alice and Bob are planning to escape. Unfortunately, the warden must approve all communications between prisoners. In order to plan their escape, they need to communicate details of their plan to each other, without Warden Wendy finding out. To do this, Alice hides her message in a document, which she passes to Bob, who extracts the message. Warden Wendy acts passively and only examines the document to determine if it contains a hidden message. Technically, Alice and Bob are using steganography to hide and extract the message while Wendy uses steganalysis to determine if a message has been hidden.

RIIT'14, October 15–18, 2014, Atlanta, Georgia, USA.

ACM 978-1-4503-2711-4/14/10 ...$15.00.
http://dx.doi.org/10.1145/2656434.2659756.

2. STEGANOGRAPHY

Steganography is defined as the art and science of writing hidden messages. It comes from the Greek words *steganos* meaning "covered" and *graphei* meaning "writing." It has been in use, in one form or another, for thousands of years, particularly during wartime. Recently, the suspected use of the Internet by drug dealers and terrorists has sparked a renewed interest in steganography, which some believe is used for covert communications.

In steganography, there are two types of objects: carrier and hidden. Carrier or overt is the host or where the data is hidden. Hidden or covert is the source or the data that is hidden. The goal is embedding the covert data into the carrier in such a way that the embedding is not obvious to visual observation of the carrier or to the application that processes the carrier.

There are three common techniques used in steganography: insertion, substitution, and generation.

2.1 Insertion

The insertion technique hides the message in a location that the application ignores. Two locations used by the insertion technique are comment fields and the area after the end of file (EOF) marker. The key point is the application ignores any data in these two locations. While the rendering application may ignore the embedded data, this technique is susceptible to discovery by visual inspection. The most obvious is the file size increases based on the amount of data embedded. A HTML file that is 5MB in size would be suspicious, to say the least.

2.2 Substitution

The substitution technique replaces insignificant bits of the carrier file content with the message. The goal is to do this with minimal distortion of the carrier. Unlike the insertion technique, visual inspection doesn't work, since the file size normally doesn't change. The unwillingness to increase the file size does limit the amount of data that can be embedded.

The most common substitution technique is Least Significant Bit (LSB) substitution. JSteg, StegHide, and EzStego are JPEG steganographic tools that use LSB substitution. These tools select a sequence of data from the covert file either by some predetermined sequence or by using a pseudo-random number generator to determine the sequence. The LSB of the selected data is substituted by a bit from the overt data.

Gary Kessler [3] gives an example of LSB substitution, where the letter 'G' is "hidden" across the following eight bytes of a carrier file (the least significant bits are underlined):

```
10010101 00001101 11001001 10010110
00001111 11001011 10011111 00010000
```

In ASCII, the letter 'G' is represented in binary as 01000111. These eight bits are "written" to the least significant bit of each of the eight carrier bytes as follows:

```
10010100 00001101 11001000 10010110
00001110 11001011 10011111 00010001
```

In the example above, only half of the least significant bits were actually changed (shown in bold).

2.3 Generation

The generation technique uses an algorithmic scheme to create an overt file from a covert file. An example of this technique is creating a fractal image from the covert file. Both the insertion and substitution techniques require an overt file, while this technique does not. This means the generation technique is immune to comparison tests, which compare the modified file with an unmodified version of the same file.

3. STEGANALYSIS

Steganalysis is defined as the art and science of detecting hidden messages. This section examines the techniques used for steganalysis. Similar to steganography, where different techniques are used to hide messages, steganalysis uses different techniques to detect those messages. These techniques can be broken down into three broad categories; knowledge of the steganography tool or algorithm used, presence of structural changes to the file, and analysis of statistical properties of the image.

3.1 Steganographic Tool Knowledge

Steganalysis based on knowledge of the steganography tool or embedding algorithm used is called specific or targeted steganalysis. Kumar [4] states if the steganalyst knows the embedding algorithm and its statistical signature then this type of steganalysis is very effective. This approach is similar to how anti-virus software works by looking for a specific signature, in this case a statistical signature. As with anti-virus software, targeted steganalysis doesn't perform well with new or unknown embedding algorithms.

3.2 Structural Changes

Some steganographic algorithms alter the structure of the image. These algorithms primarily use the insertion steganographic file technique where covert data is hidden in areas ignored by the application, such as comment fields and after the end of file. By examining the structure of an image and comparing it to the structure of clean images, steganalysis can determine if covert data is present.

3.3 Statistical Analysis

Current techniques in steganalysis focus on statistical analysis and form the basis of universal or blind steganalysis. Blind steganalysis is agnostic with regard to the steganographic tool or algorithm used to hide data within an image. This approach allows it to detect new steganographic algorithms. Most statistical analysis focuses on how images with hidden data deviates statistically from clean images. As an example, consider two steganographic tools, JSteg and OutGuess.

JSteg is an older steganographic tool and one of the first to do JPEG steganography. It uses Least Significant Bit (LSB) substitution using a simple algorithm to select the bits to modify. This makes it susceptible to detection by statistical analysis, particularly by using the Chi-square test. Westfeld and Pfitzmann [9] developed the use of the Chi-square test for detecting embedded data. They discovered when data is embedded using LSB substitution, it causes adjacent values to change. The Chi-square test measures "goodness of fit" or how an observed distribution differs from a theoretical distribution. Westfeld and Pfitzmann measured the value of adjacent pixels. They calculated a theoretical value by averaging the adjacent pixels and comparing that to the observed value.

OutGuess (www.outguess.org) was developed by Niels Provos in 2001. Like JSteg, OutGuess identifies redundant bits in the carrier and replaces them with bits from the hidden message. Provos [7] wanted to develop a steganographic algorithm for JPEG images that would not be susceptible to statistical analysis, like the Chi-square test. To do this, OutGuess applies additional transforms in order to correct statistical deviations as a result of hiding the message. The resulting carrier image is statistically the same as the clean image.

4. JPEG FORMAT

This project focuses on JPEG images, which are the most common image format found on the Internet. The JPEG format was developed by the Joint Photographic Experts Group and became an approved standard in 1992. The JPEG standard specifies how an image is converted into a stream of compressed bytes and how it is decompressed back into an image. However, JPEG is not a file format. There are two standard file formats for JPEG-compressed images; Exit and JFIF.

In order to understand the approach taken by this project, it is necessary to understand some of the technical details of JPEG images. Figure 1 is a Block Diagram of the JPEG Conversion Process.

This project is interested in the quantized DCT (Discrete Cosine Transform) coefficients that result from the conversion process. The DCT converts an 8x8 pixel block from the spatial to the frequency domain. This means the RGB color model is converted to the YC_bC_r color space where Y represents luminance or brightness and C_bC_r represents chrominance or color. The DCT equation is shown in Figure 2.

To clarify the JPEG conversion process, Figure 3 shows the intermediate results from each step of the conversion

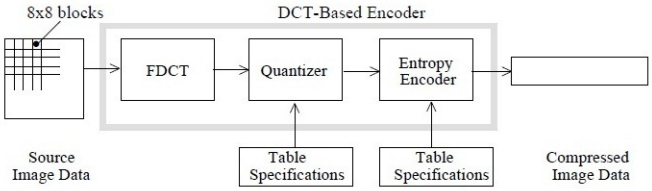

Figure 1: JPEG Conversion Block Diagram [5]

$$F(u,v) = \frac{c(u)c(v)}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} \cos\left(\frac{(2i+1)u\pi}{16}\right) \cos\left(\frac{(2j+1)v\pi}{16}\right) f(i,j),$$

$$c(e) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } e = 0, \\ 1, & \text{if } e \neq 0. \end{cases}$$

Figure 2: Discrete Cosine Transform (DCT) [1]

process. The process begins with an 8x8 block of source data (a), which is then converted from the spatial to the frequency domain using the DCT (Figure 2). The result of this transform is the forward DCT coefficients (b). Next, a quantization table (c) is used to normalize the DCT coefficients (d). The element at index [1,1] of d is called the DC coefficient; the others are called AC coefficients. It is these normalized quantized AC DCT coefficients this project is interested in. Steps e and f are the results of the reverse process and are not used by this project. The quantization table is specified by the JPEG standard and determines the level of compression applied to the image.

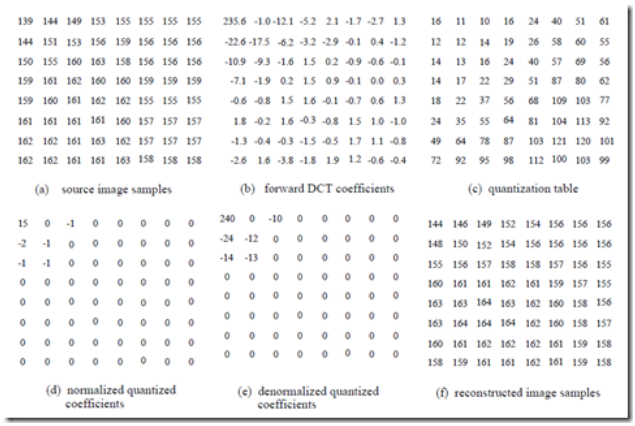

Figure 3: Example JPEG Conversion [5]

5. APPROACH

The approach taken by this project uses the Calibration Technique and associated features developed by Fridrich[2].

The Calibration Technique attempts to create the statistical equivalent of the clean image from the stego image. It does this by decompressing the stego image to the spatial domain, cropping by 4 pixels in each direction, and then recompressing using the same quantization table as the original stego image. Fridrich says this approach should "produce a 'calibrated' image with most macroscopic features similar to the original cover image." [2] The reasoning behind this approach is the cropped stego image is "similar to cover image and thus its DCT coefficients should have approximately the same statistical properties as the cover image." The choice of 4 pixels is important because of its relationship to the 8x8 grid. Fridrich explains that during recompression new DCT coefficients are calculated that have not been influenced by the previous compression and potential embedding in the DCT domain. This results in two images, the original stego image (J_1) and the calibrated image (J_2). Fridrich calculates a calibrated form of feature f which is the difference using $f(J_1) - f(J_2)$.

5.1 First Order Statistics or Features

The histogram of DCT coefficients is the primary first order statistic or feature. Fridrich's assumption is that DCT coefficients are independent and identically distributed (iid) random variables. This means each DCT coefficient has the same probability distribution as the others and all are mutually independent. Fridrich concludes, "their complete statistical description can be captured using their probability mass function." [2]

The following features are derived from the sample probability mass function (pmf) computed from the DCT coefficients.

5.1.1 Kronecker Delta

The Kronecker Delta is used in many steganalysis formulas. It returns a 1 when the value of x is 0 and a 0 when the value of x is not equal to 0. Typically, x is the difference of two values, so the Kronecker Delta is used to determine if the two values are equal and to count how many times the value occurs. The formula for the Kronecker Delta is shown below:

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

5.1.2 Global Histogram

The Global Histogram is considered a normalized histogram (i.e., displays relative frequencies) of all luminance (Y) DCT coefficients. It returns a D-dimensional vector as shown by its formula (1). The range is $-5 \leq r \leq 5$, which produces 11 features. Most of the features described use a limited range. Fridrich explains that without a limited range the feature space dimensionality would be too large and larger values of r exhibit fluctuations that are of little value. [2]

$$H[r] = \frac{1}{64xN_B} \sum_{k,l=0}^{7} \sum_{b=1}^{N_B} \delta(r - D[k,l,b]) \qquad (1)$$

This project focused primarily on the Global Histogram. Table 1 shows the calculations using Fridrich's Calibration Technique. The first row (r) represents the range used in the calculations. J_1 is the original image. J_2 is the cropped image. The third row is the difference between the original and cropped images. Fridrich specified the use of the difference between the two images [2]. This project extended Fridrich's technique by using the absolute value of the difference as shown in the fourth row. A single value per image was calculated by adding the absolute values of the differ-

ence. For the sample calculation in Table 1, that value was 0.02174.

r	-5	-4	-3
Image(J_1)	0.01096	0.01555	0.02458
Cropped(J_2)	0.01154	0.01559	0.02534
$J_1 - J_2$	-0.00058	-0.00004	-0.00076
$\|J_1 - J_2\|$	0.00058	0.00004	0.00076
r	-2	-1	0
Image(J_1)	0.04727	0.12242	0.39056
Cropped(J_2)	0.04930	0.12700	0.38188
$J_1 - J_2$	-0.00203	-0.00458	0.00868
$\|J_1 - J_2\|$	0.00203	0.00458	0.00868
r	1	2	3
Image(J_1)	0.12587	0.04642	0.02332
Cropped(J_2)	0.12749	0.04628	0.02512
$J_1 - J_2$	-0.00162	0.00014	-0.00180
$\|J_1 - J_2\|$	0.00162	0.00014	0.00180
r	4	5	$\sum \|J_1 - J_2\|$
Image(J_1)	0.01524	0.01131	
Cropped(J_2)	0.01591	0.01047	
$J_1 - J_2$	-0.00067	0.00084	
$\|J_1 - J_2\|$	0.00067	0.00084	*0.02174*

Table 1: Global Histogram Calculation using Fridrich's Calibration Technique

Using the Global Histogram calculation described above, this project calculated values for clean images, JSteg embedded images, and F5 embedded images. A sample of the results is shown in Table 2. The sample shows a very distinct range for each type of image. However, the sample is somewhat misleading and when all images were compared, there existed some overlap that contributes to the results obtained.

Image #	Clean	JSteg	F5
1	0.02174	0.10527	0.06304
2	0.02437	0.13497	0.08288
3	0.02196	0.13561	0.06497
4	0.02093	0.10148	0.07639
5	0.02702	0.11471	0.07781

Table 2: Global Histogram Results Sample

5.1.3 AC Histogram

The AC Histogram is also considered a normalized histogram. It is a histogram of individual DCT modes. Fridrich says, "it is possible to consider the coefficients as 64 parallel iid channels, each corresponding to one DCT mode." [2] While some steganographic tools preserve the Global Histogram, not all preserve the histogram of individual DCT modes. Analysis of the AC Histogram may allow detection of embedding from the use of these tools.

The range is $-5 \leq r \leq 5$ and $0 < k + 1 \leq 2$, which produces 11 x 5 (55) features. The AC Histogram limits the indices examined to the first five AC DCT coefficients in "zigzag" order. The formula for AC Histogram is shown in 2.

$$h^{(kl)}[r] = \frac{1}{N_B} \sum_{b=1}^{N_B} \delta(r - D[k, l, b]) \tag{2}$$

5.1.4 Dual Histogram

The Dual Histogram returns an 8x8 matrix and determines "how many times the value r occurs as the (k, l)th DCT coefficient." [2] Fridrich states that "it captures the distribution of a given coefficient value r among different DCT modes." [2]

The range is $-5 \leq r \leq 5$ and $0 < k + 1 \leq 3$, which produces 11 x 9 (99) features. The Dual Histogram limits indices examined to the first nine AC DCT coefficients in "zigzag" order. The formula for Dual Histogram is shown in 3.

$$g^{(r)}[k, l] = \frac{1}{N_B(r)} \sum_{b=1}^{N_B} \delta(r - D[k, l, b]) \tag{3}$$

5.2 Inter-Block or Second Order Features

Fridrich [2] explains an Inter-block feature by saying "that natural images exhibit dependencies over distances larger than the block size."

5.2.1 Variation

The Variation feature is based on Ueli Maurer's "Universal Statistical Test for Random Bit Generators" [6] and Niels Provos [7] application of Maurer's research to JPEG steganalysis.

Provos [7] discovered images with hidden data have higher entropy than those without. He used Maurer's test to measure entropy and found the expected result from a truly random source is 7.184. Fridrich adds, "most steganographic techniques in some sense add entropy to the array of quantized DCT coefficients and thus increase the difference between dependent coefficients across blocks." [2] The dependencies are measured mathematically using a quantity called variation. Fridrich developed the formula shown in 4 for measuring variation. An interesting aside, the Variation Technique produces a single result, unlike the other techniques examined by this project.

$$V = \frac{\sum_{i=1}^{8\lceil \frac{M}{8} \rceil - 8} \sum_{j=1}^{8\lceil \frac{N}{8} \rceil} \left| \mathbf{D}[i, j] - \mathbf{D}[i + 8, j] \right|}{64(\lceil M/8 \rceil - 1)\lceil N/8 \rceil} + \frac{\sum_{i=1}^{8\lceil \frac{M}{8} \rceil} \sum_{j=1}^{8\lceil \frac{N}{8} \rceil - 8} \left| \mathbf{D}[i, j] - \mathbf{D}[i, j + 8] \right|}{64\lceil M/8 \rceil(\lceil N/8 \rceil - 1)} \tag{4}$$

5.2.2 Blockiness

Fridrich [2] defines Blockiness as "the sum of discontinuities along the 8x8 block boundaries in the *spatial domain*." Notice this feature is the only feature that works in the spatial domain. There are two blockiness measures for $\gamma = 1$ and $\gamma = 2$. The formula for Blockiness is shown in 5.

$$B_\gamma = \frac{\sum_{i=1}^{\lfloor \frac{M-1}{8} \rfloor} \sum_{j=1}^{N} \left| x[8i,j] - x[8i+1,j] \right|^\gamma}{N\lfloor (M-1)/8 \rfloor + M\lfloor (N-1)/8 \rfloor} + \frac{\sum_{i=1}^{M} \sum_{j=1}^{\lfloor \frac{N-1}{8} \rfloor} \left| x[i,8j] - x[i,8j+1] \right|^\gamma}{N\lfloor (M-1)/8 \rfloor + M\lfloor (N-1)/8 \rfloor} \quad (5)$$

5.2.3 Co-occurrence Matrix

Fridrich [2] defines the Co-occurrence Matrix as the "distribution of pairs of neighboring DCT coefficients." It is actually the average of two matrices, one in the horizontal direction and one in the vertical direction. The range is $-2 \leq s,t \leq 2$, which produces 25 features. The Co-occurrence Matrix formula is shown in 6.

$$C[s,t] = \frac{\sum_{i=1}^{8\lceil \frac{M}{8} \rceil - 8} \sum_{j=1}^{8\lceil \frac{N}{8} \rceil} \delta(s - D[i,j]) \delta(t - D[i+8,j])}{64(\lceil M/8 \rceil - 1)\lceil N/8 \rceil} + \frac{\sum_{i=1}^{8\lceil \frac{M}{8} \rceil} \sum_{j=1}^{8\lceil \frac{N}{8} \rceil - 8} \delta(s - D[i,j]) \delta(t - D[i,j+8])}{64\lceil M/8 \rceil (\lceil N/8 \rceil - 1)} \quad (6)$$

5.3 Intra-Block Features

Intra-block features such as the Average Markov matrix measure dependencies within one 8x8 block. This project did not examine Intra-block features, but is including this section for completeness so the reader is aware they exist and how they are used. Fridrich [2] goes into great detail explaining the concepts and mathematics of Average Markov matrices and their applicability to JPEG steganalysis.

6. RESULTS

The results of using Global Histogram for Blind Steganalysis are presented below.

This project examined 600 images, 300 were used for training and 300 for testing Blind Steganalysis. Each group of 300 was divided into three sets of 100 each. The first set was clean, the second embedded using JSteg, and the third embedded using F5. For the 100 clean images, an average plus 2σ (standard deviation) was calculated. The average represents the sum of the absolute values of the difference in the Global Histogram of the original and cropped images. The 300 training images were then evaluated against this value. The results of the training session are shown in Table 3.

Image Type	Correct
Clean	94%
JSteg	91%
F5	64%

Table 3: Training Results

A second group of 300 images was used for testing Blind Steganalysis. Like the ones used for training, the 300 were divided into three sets of 100 each. The first set was clean, the second embedded using JSteg, and the third embedded using F5. These images were evaluated against the same value (average + 2σ) as the training session. The results of the Blind Steganalysis session are shown in Table 4.

Image Type	Correct
Clean	98%
JSteg	99%
F5	74%

Table 4: Blind Results

7. CONCLUSION

Based on this project, three conclusions can be drawn. First, Global Histogram is a better differentiator than the other features evaluated. Considering a single value was calculated from the 11 Global Histogram features, the Blind Steganalysis results are very encouraging. Admittedly, the sample set is small; it does indicate this approach is viable. Future work would be to analyze thousands of images to determine if consistent results are obtained.

Second, the Variation Technique is not as good a differentiator as originally hypothesized. This project's initial premise was that Variation would be a very good differentiator. This was based on the fact that embedding causes an increase in the entropy or randomness in an image. Since Variation is a measure of entropy, the assumption was any embedding would cause a noticeable change in the Variation feature. Future work is planned in this area because the theory behind Variation still has merit and deserves further investigation.

Third, the combination of features should increase predictive accuracy. Most statistical steganalysis techniques use multiple features, sometimes hundreds of features, to increase accuracy. An extension of this project would be to combine multiple features to determine if the predictive accuracy increased and by how much relative to the additional computational complexity.

8. ACKNOWLEDGMENTS

I would like to acknowledge the contributions to this project made by Stephen Felix, Abel Zerazion, and Phil Shriner.

9. REFERENCES

[1] C.-C. Chang, C.-C. Lin, C.-S. Tseng, and W.-L. Tai. Reversible hiding in dct-based compressed images. *Information Sciences*, 177(13):2768–2786, 2007.

[2] J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, 2010.

[3] G. C. Kessler. An overview of steganography for the computer forensics examiner. *Forensic Science Communications*, 6(3):1–27, 2004.

[4] M. Kumar. *Steganography and Steganalysis of JPEG Images: A Statistical Approach to Information Hiding and Detection*. LAP LAMBERT Academic Publishing, 2011.

[5] F. Liu. Jpeg standard - a tutorial based on analysis of sample picture - part 1. coding of a 8x8 block, August 2011.

[6] U. M. Maurer. A universal statistical test for random bit generators. *Journal of cryptology*, 5(2):89–105, 1992.

[7] N. Provos. Defending against statistical steganalysis. In *Usenix Security Symposium*, volume 10, pages 323–336, 2001.

[8] G. J. Simmons. The prisoners' problem and the subliminal channel. In *Advances in Cryptology*, pages 51–67. Springer, 1984.

[9] A. Westfeld and A. Pfitzmann. Attacks on steganographic systems. In *Information Hiding*, pages 61–76. Springer, 2000.