Measures for Classification and Detection in Steganalysis

A Project Report

Submitted in partial fulfilment of the

requirements for the Degree of

Master of Engineering

in

Faculty of Engineering

by

GUJAR SUJIT PRAKASH

Computer Science and Automation
Indian Institute of Science, Bangalore
Bangalore – 560 012 (INDIA)

June 2006

Dedicated

To

My parents, sister
and Friends

Acknowledgments

I am extremely thankful my guide, Prof. C E Veni Madhavan for his guidance and support. My association with him over the past two years has been a greatly enriching experience. All discussions with him and his lectures on cryptography were very insightful. I would also like to thank my faculty advisor Dr Shirish Shevade his support during my stay at IISc. I wish to express my gratitude to my friends Shailesh Patil and Deepak Hinge for their valuable comments and feedback throughout my project. I also thank my fellow labmate and friend Sophia Rodrigues who has been always very helpful and cooperative.

I would also be very thankful to Mr Sameer Pawar, Muralidhar and all my friends in B-mess who were extremely nice company and when I had fracture in my hand, were of great help which cannot be expressed in words. Finally I would like to say thanks to all my friends in department and in the institute for their support cooperation.

Abstract

Still and multi-media images are subject to transformations for compression, steganographic embedding and digital watermarking. We propose new measures and techniques for detection and analysis of steganographic embedded content. We show that both statistical and pattern classification techniques using our proposed measures provide reasonable discrimination schemes for detecting embeddings of different levels. Our measures are based on a few statistical properties of bit strings and wavelet coefficients of image pixels.

Keywords : Steganography, Steganalysis, SVM, Wavelets, Radon Transform, Ridglets

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Steganography is a Greek word meaning *covered or hidden writing*. It is the art and science of secret communication, aiming to conceal the existence of the communication. This is a different from Cryptography, where the existence of the communication is not disguised but the message is obscured by scrambling it. Use of cryptography would not stop a third party knowing that some secret communication is going on. In steganography, the message to be sent is concealed in such a way that an intruder would not know whether any secret communication is going on or not. Hiding information inside digital carriers is becoming popular([1, 13]). A rapid growth in demand and consumption of multimedia has resulted in data hiding techniques for files like audio (*.wav*), images (*.bmp, .pnm, .jpg*). Digital images are most common sources for hiding message. The process of hiding information is called an ***embedding***. Least Significant Bit (**LSB**) embedding is the most widely used steganographic technique. In LSB embedding, the LSBs of uncompressed images are replaced with the message bits. We will be seeing this in detail in 2.3.2. The amount of embedding (the number of bits embedded) referred to as *level*, is given as the percentage of the total number of pixels.

Steganalysis is the art of seeing the unseen. Steganalysis will analyze whether a given content, contains any secret message camoflagued into it. We will be concentrating on steganalysis of images. Natural images carry some statistical properties. These get disturbed due to steganographic operations. A steganalyst explores this fact by analyzing the images. Some of the powerful methods for the analysis of steganographic images are [5, 6, 7, 11]. We propose new measures and techniques for detection and analysis of steganographic embedded content. Our approach is to blend different techniques together viz. statistical, pattern classification techniques, run length, transform coding techniques, and also encryption techniques. Using our proposed measures, we show that with statistical and pattern classification techniques, we obtain discrimination schemes for detecting embeddings of different levels. Our measures are based on a few statistical properties of bit strings and wavelet coefficients of image pixels.

In Chapter 4, we explain our approach towards classification of given data based on a feature vector consisting of statistical measures and using **S**upport **V**ector **M**achine (SVM) tools. In Chapter 5, we propose the use of wavelet transforms for steganalysis. Our results, presented in Chapter 4 and 5 show the efficacy of our measures in discriminating different levels of embedding. We then explore the power of *Ridgelets* and describe experimental setup for that in Chapter 5.8. We conclude with our plans for improved and finer steganalysis in Chapter 6.

Chapter 2

Background

2.1 History

This art of covert communication is very ancient [1]. Till date, multitude of methods and variations have been developed, for hiding information. Hiding the secret message under a wax coating of a wax coated tablets is one of the oldest methods. The message can be camouflaged in text message. e.g.

> Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by products, ejecting suets and vegetable oils.

Taking the second letter in each word the following message emerges:

> *Pershing sails from NY June 1. I*

These types of techniques are called "Null Ciphers". Invisible inks and microdots were used in World War II. Invisible ink is, the one with which if we write on a plain paper, nobody will be able to read with naked eyes. If we heat the paper, the message is visible. In Microdot technology, the photograph of document to send is taken and the photograph is miniaturized to the size of period of printed document. This 'dot' size photograph is put on any period of a document. The tools like 'Genomic Steganography' (hiding message in human DNA) are recent advances in this science.

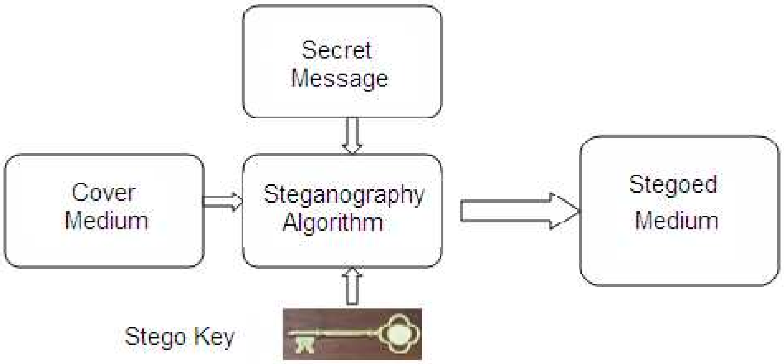


Figure 2.1: *Typical Steganography Model*

2.2 Steganography Model

The commencement of computer era has given a new dimension to the art of secret communication. Computer based techniques hide data in digital carriers by changing the carriers in such a manner that even after altering them, they appear to be innocuous. These carriers are called as “Cover Media”. Cover media can be audio files, images etc. The process of hiding information on the cover is called as “embedding”. Different steganographic schemes will have their own ways of embedding the message. These are called ‘steganographic algorithms’. Fig. 2.1 shows the typical model of steganographic techniques. A stegokey is used to provide additional security. Even if an object is suspected to contain steganographic embedding by a third person, the stegokey will preclude the detection of the secret message. So ideally for a third person, without knowledge of stegokey and stego algorithm, it is not possible to read the hidden message from stegoed object.

Capacity of Cover Object : The maximum length of message that can be embedded into a cover without affecting perceptual quality or signal strength, is referred as capacity of the particular object.

Level of embedding: The actual amount of embedding as percent of capacity of cover object is level of embedding. So it can be anywhere between 0% to 100%.

2.3 Image Steganography

We will see how information can be camouflaged in images. First we will see what are digital images.

2.3.1 Digital Images

To store an image on computer, it is divided into small parts called pixels. The value of intensity of these pixels is stored for three basic colors as an image on computer. '.bmp', '.pnm' are some file formats to store such images, which are uncompressed file formats for images. These images have a lot of redundancy. Also the loss of small information in pixel intensity is not captured by the human eye. So there exists compression techniques like jpeg for images. The compression techniques will try to de-correlate the redundancy and may also introduce some loss of information.

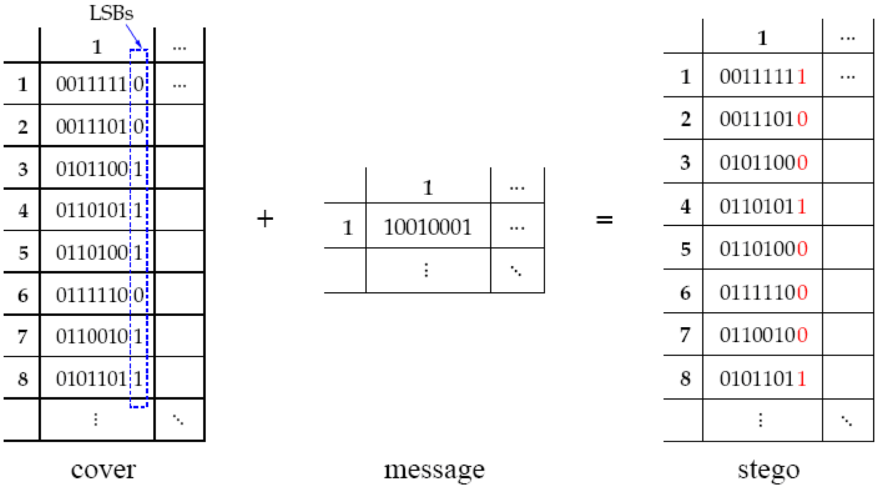
2.3.2 LSB Steganography for Uncompressed images

One simple and yet effective method of steganography is LSB replacement. As mentioned, small perturbation in pixel intensity is not detected by an eye; these techniques take advantage of it by changing the LSB of a few pixels. The algorithm used will decide which pixels in an image to be modified. Some algorithm will pick the pixels in image at regular interval depending upon image size and message size. Sophisticated steganographic software viz. S-Tools [13], CSA-Tool [18] can add further layers of complexities, such as distributing messages in a pseudo-random way and encrypting messages. The disadvantage of such schemes is that lossy compression techniques can not be applied on such images after the process of embedding as information is hidden in LSBs which are highly susceptible to change. This technique is explained in Fig. 2.2. One byte of the message, the middle one in the Fig. 2.2, is embedded into the LSBs of eight consecutive pixels in the cover image by modifying the eight LSBs of the eight pixels in the cover image to the same as those in the message, right in figure.

2.3.3 LSB Steganography for JPEG images

For compressed image formats (e.g., JPEG), LSB insertion is performed on the compressed data streams, for instance, the quantized DCT coefficients in a JPEG image. Similar to embedding in raw pixels, LSB insertion on the compressed data stream introduce

Figure 2.2: *LSB Steganography*



negligible perceptual difference between the cover and stego images. There are schemes for jpeg images, to conceal data in transform domain i.e. frequency domain, after quantization of DCT coefficients (F5 [3]). Disadvantage of such schemes is that the message length that can be hidden in such images is small. Also image quality degrades very fast, as concealed message size increases. Such techniques are relatively easier to crack. So LSB hiding and detection are of most interest.

2.3.4 Quantization Index Modulation (QIM) Steganography

A message can be embedded in the host medium through the choice of a scalar quantizer. For example, consider a uniform quantizer of step size δ, used on the host's coefficients in some transform domain. Let odd reconstruction points represent a signature data bit '1'. Likewise, even multiples of 'δ' is used to embed '0'. Thus, depending on the bit value to be embedded, one of the two uniform quantizers of step size $2 * \delta$ is chosen. Moreover, the quantizers can be pseudo randomly dithered, where the chosen quantizers are shifted by a pseudo-random sequence available only to encoder and decoder. As such, the embedding scheme is not readily decipherable to a third party observer, without explicit knowledge of the dither sequence. Decoding is performed by quantizing the received coefficient to the nearest reconstruction point of all quantizers. An even reconstruction point indicates that a ‘0’ has been hidden. Likewise, if a reconstruction point lies on an odd quantizer, a ‘1’ has been hidden [2].

Chapter 3

Steganalysis of images

All natural images have a lot of correlation among neighboring pixels. Image pixel data has statistical properties. All these are disturbed by the process of embedding. These are exploited in steganalysis of images. The various kinds of problems handled by steganalysis are,

- Identification of embedding algorithm
- Detection of presence of hidden message in cover signal
- Estimation of embedded message length
- Prediction of location of hidden message bits
- Estimation of secret key used in the embedding algorithm
- Estimation of parameter of embedding algorithm
- Extraction of hidden message (!!!)

Various techniques for steganalysis are described below.

3.1 Visual attacks

Most steganographic programs embed the message bits either sequentially or in some pseudo-random fashion. In most programs, the message bits are chosen non adaptively independently of the image content. If the image contains homogeneous areas or areas with the color saturated at either 0 or 255, we can look for suspicious artifacts using simple visual inspection. Even though the artifacts cannot be readily seen, we can plot one bit-plane (for example, the LSB plane) and inspect just this bit-plane. This attack is especially applicable to palette images for LSB embedding in indices to the palette. If, at the same time, the message is embedded sequentially, one can have a convincing argument for the presence of steganographic messages in an image. Although visual attacks are simple, they are hard to automate and their reliability is highly questionable [5].

3.2 Chi Square / Pair of Values (PoVs) method

In LSB replacement, while embedding the message, fixed set of Pairs of Values (PoVs) are flipped into each other. e.g. 0 - 1, 2 - 3, ... 254 - 255. 2 will never become 1 or vice versa. Pfitzman and Westfield [5] introduced a powerful statistical attack that can be applied to any steganographic technique in which fixed set of Pairs of Values (PoVs) are flipped into each other to embed the message bits. This method is based on statistical analysis of PoVs exchanged during message embedding. As the number of pixels for which LSB has been replaced increases, the frequencies of both values of each PoV tend to become equal. The idea of the statistical attack is to compare the theoretically expected frequency distribution in stego image with some sample distribution observed in the possibly changed carrier medium.

3.3 RS Steganalysis

Just statistical measures on LSBs for detecting level of embedding is unreliable as the LSB bit plane does not contain any easily recognizable structure. But even though it appears random, it has some relation with other bit planes. RS Steganalysis exploits this property. Fridrich et al. [6] developed a steganalytic technique based on this for detection of LSB embedding in color and grayscale images. They analyze the capacity for embedding lossless data in LSBs. Randomizing the LSBs decreases this capacity. To examine an image, they define Regular groups (R) and Singular groups (S) of pixels depending upon some properties. Then with the help of relative frequencies of these groups in the given image, in the image obtained from the original image with LSBs flipped and an image obtained by randomizing LSBs of the original image, they try to predict the levels of embedding.

3.4 DCT domain Steganalysis

Since we can not compress image using frequency domain techniques,after LSB hiding, specific DCT domain steganography has been developed [3, 4]. The algorithm F5 stores the information in DCT coefficients leading to change in DCT histogram. Fridrich et al [7, 8] have shown that this change is proportional to the level of embedding. They also showed that, if we crop an image by 4 rows and 4 columns, we can get the original DCT histogram. The basic assumption here is that the quantized DCT coefficients are robust to small distortions and after cropping the newly calculated DCT coefficients will not exhibit clusters due to quantization. Also, because the cropped stego image is visually similar to the cover image, many macroscopic characteristics of cover image will be approximately preserved. After predicting DCT coefficient's histogram in the original image and comparing with that of a stegoed image, the hidden message length can be calculated.

Tools like Outguess [4] are developed to counter this attack. Fridrich et al [9] have developed techniques using *Blockiness* introduced in images due to histogram equalization to attack Outguess.

We described above a few of steganalysis techniques. There are many more methods to discover the presence of embedding. But detecting the complete message, or predicting the message length is not possible in most of these analysis tools. RS Steganalysis [6] is the most powerful among this for predicting message length. But RS Steganalysis does not perform well on all images. Shree Lekshmi and Veni Madhavan [19] a measure "adjacent probable transition" (APT) and developed a method which predicts the message length based on this measure. This has a coarse resolution of predicting the message length but performs well on a wider range of images than that of RS Steganalysis.

The disadvantage of frequency domain (DCT) stego algorithms is that the hidden message length small. Also image quality degrades very fast, as concealed message size increases. Such techniques are relatively easier to crack. Thus LSB hiding in an uncompressed images is of most interest. hence, we concentrate only on LSB data hiding and detection for uncompressed images. There are few schemes which are slight variants of LSB replacement techniques. Instead of replacing LSB of pixel value, the pixel value is incremented or decremented depending upon data bit and pixel value.

3.5 Assumptions

We develop schemes for LSB Steganalysis under following assumptions,

1. The hidden message contains approximately equal number of 0's and 1's.
2. The embedded bits are uniformly distributed in an image.
3. LSB replacement technique is used by the steganographic algorithm.

Our analysis holds true for any stego scheme which satisfies the above assumptions.

Chapter 4

Classification based on statistical measures and SVM

Image steganography is a kind of transformation of a *cover image* and embedded data. The embedding operation will perturb the statistical properties. We try to capture the perturbation. We use statistics defined below as feature of non-random strings. As a first step we establish the power of our feature vector of measures based on statistical properties of bit strings in discriminating a variety of standard file types (Section 4.2). Then we explore the possibility of discriminating images with different levels of embeddings. (Section 4.3). Once the level of embedding is determined to reasonable accuracy, we can proceed to the next step of *location* of embedded bits by other statistical and combinatorial techniques. For classification we use Support Vector Machines.

4.1 Support Vector Machines

Support vector machine (SVM) is a supervised learning technique for classification. For classification there are many techniques like neural networks, perceptron, Fisher Discriminant, SVMs. Out of this, SVM is widely used and most popular in Machine learning community. The key to the success of SVM is the kernel function which maps the data from the original space into a high dimensional (possibly infinite dimensional) feature space. By constructing a linear boundary in the feature space, the SVM produces non-linear boundaries in the original space. When the kernel function is linear, the resulting SVM is a maximum-margin hyperplane. Given a training sample, a maximum-margin hyperplane splits a given training sample in such a way that the distance from the closest cases (support vectors) to the hyperplane is maximized. Typically, the number of support vectors is much less than the number of the training sample. Nonlinear kernel functions such as the polynomial kernel and the Gaussian (radial basis function) kernel are also commonly used in SVM. The computational complexity of the SVM depends on the training sample, thus it avoids the traditional problem of "Curse of dimensionality". One of the most important advantage for the SVM is that it guarantees generalization to some extent. The decision rules reflect the regularities of the training data rather than the incapabilities of the learning machine. Because of the many nice properties of SVM, it has been widely applied to virtually every research field. More detailed discussion of SVM and kernel methods can be found in [17].

4.2 Classification of different types of files

We use a statistical feature space. We propose a vector of statistical measures [12] for this purpose. Our feature vector $\mu \in \mathbb{R}^9$ consists of nine statistical measures. We consider a bit string S of size $32 * n$ bits as concatenation of 'n' 32 bit words, S_i, $i = 1, \ldots, n$. We define the measures $\mu(S_i) = < \mu_1(S_i), \ldots, \mu_9(S_i) >$ for the words S_i and define the measure for entire string S, namely $\mu(S)$ as a weighted sum of the measures $\mu(S_i)$. The measures are as follows.

μ_1 : *Weighted sum of the of k-gram frequencies.* Let $f(k, j)$ denote the overlapping frequency of the k-gram binary pattern of the integer j in S_i. For example $f(4, 3) =$ number of occurrences of the pattern $< 0011 >$ in S_i. For a 32 bit word W, we define

$$\mu_1(W) = \sum_{k=1}^{4} (\max_j(f(k,j)) - \min_j(f(k,j)))2^{4(k+1)}$$

We expect the measure μ_1 to be smaller for random strings as compared to non-random strings.

μ_2 : *Weighted sum of run lengths.* Let the vector $< l_1, l_2, \ldots >$ denote the sequence of run lengths of 0's and 1's in a 32 bit word W. Then we define,

$$\mu_2(W) = \sum 2^{c_i l_i}$$

where c_i are specifically chosen weights. We set $c_i = 1 \;\forall\; i$, without loss of generality. For random strings, we expect the measure μ_2 to be smaller as compared to non-random strings, since one expects very few long runs.

μ_3 : *Weighted sum of byte-wise hamming weight transition.* Let $W = < b_0, b_1, b_2, b_3 >$, where b_i's are the bytes of the 32 bit word. Let $\#1(b)$ denote the number of 1's in a byte b. Then we define,

$$\mu_3(W) = 2^{\#1(b_0)} + 2^{\#1(b_0 \oplus b_1)} + 2^{\#1(b_1 \oplus b_2)} + 2^{\#1(b_2 \oplus b_3)}$$

For random strings, we expect μ_3 to be higher than for non-random strings. It is also possible to define the measure μ_3 with respect to overlapping bytes in a word, to measure the smoothness/suddenness of transitions.

μ_4 : *Fourier transform of the autocorrelation function of the sequence bits* in W. Let $W = < a_0, ..., a_{31} >$ be a 32 bit word. The autocorrelation function $A(W)$ is the sequence of communication itself is secret,so $A(W) = < c_0, .., c_{31} >$ where $c_i = \sum_{j=0}^{31} a_j.a_{j+1}$ (mod 32), $i = 0, .., 31$. The discrete Fourier transform $F(A(W))$ is given by the sequence $F(A(W)) =< f_0, ..., f_{31} >$; where $f_k = \sum_{j=0}^{31} c_j\, \omega^{jk\, mod 32}$ $k = 0, ..., 31$. Here ω is a 32^{nd} root of unity. Finally, the measure $\mu_4(W)$ is a root mean square average of F and is given by,

$$\mu_4(W) = (\sum_{j=0}^{31} |f_j|^2)^{1/2}$$

For random strings, we expect μ_4 to be smaller than for non-random strings.

μ_5 : *Weighted Hadamard transform.* Using an 8x8 Hadamard matrix (H) and the operation $y = Hx$, where x is 8x1 bit vector, we get measure μ_5. x is single data byte. When the Hadamard transform is applied on image data, x is taken as the bit string corresponding to a pixel value.

μ_6, μ_7, μ_8, μ_9 : These measures are based on the weighted entropy measures $-\sum p_i \log p_i$ where p_i's are probabilities of non-overlapping occurrences of 1,2,3,4 grams in string S.

Thus given a file S of some data, we compute the feature vector $\mu(S)$. This vector captures the statistical characteristics of the bit string corresponding to S. We note that the statistical properties such as k-gram frequencies, run lengths, auto-correlation and entropy together are powerful features that discriminate a wide variety of non-random data. In the following we demonstrate this by classification based on our feature vector.

We use the feature vector μ defined above as follows. For training of SVM, we measure statistics on 3 different chunks of 2000 words (8000bytes) from 30 different files to get 90 different μ vectors for each class. For testing, we measure statistics, in similar fashion, on 20 different files from each i.e. 60 μ vectors for each class. Though we have used measures calculated on 2000 words, our experiments shows that even 400 words are sufficient for testing a data for classification.

The SVM tool is obtained from *http://www.csie.ntu.edu.tw/˜cjlin/libsvm/*. This tool provides scripts to find the best values of hyper parameters required for SVM, based on the train data set. We used the most widely used 'Gaussian kernel' for SVM. For avoiding some features dominating the classification, we scale each measure to zero mean, unit variance. We studied the following eight different classes:

1. jpeg 2. bmp/pnm 3. zip files 4. gz files 5. text files 6. ps files 7. pdf files and 8. c files.

We present our classification results in confusion matrix Table 4.2. The ij^{th} entry is the probability of a test data belonging to class i and being classified as class j. We see from the table that in all but two of the eight cases, the classification accuracy is near 1. We used a total of 540 μ vectors for testing and achieved overall accuracy of 80%.

	jpeg	bmp/pnm	zip	gz	txt	ps	pdf	c
jpeg	0.9	0.05	0.05	0.0	0.0	0.0	0.0	0.0
bmp/pnm	0.0	0.9	0.0	0.0	0.0	0.0	0.05	0.05
zip	0.0	0.0	0.6	0.35	0.0	0.0	0.05	0.0
gz	0.0	0.0	0.1	0.9	0.0	0.0	0.0	0.0
txt	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
ps	0.0	0.0	0.0	0.0	0.05	0.95	0.0	0.0
pdf	0.0	0.0	0.6	0.05	0.0	0.05	0.3	0.0
c	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.95

Table 4.1: Confusion Matrix For Data Classification

4.3 Analysis of LSB planes from Stegoed and non-Stegoed Images

In the above experiments, we measured statistics on the whole sequence of bits of the given data. An embedding operation is performed on LSB of an image. So to detect the perturbation due to steganographic operation, we measure statistics only of LSB of images. In this direction, we first consider only two classes : one is LSB obtained from *non-stegoed image* and the other is LSB obtained from *images with 50% embedding.* For our experiments described in this section, we use a random embedding instead of using any particular steganographic tool. This random embedding satisfies the assumptions stated in Section 3.5. We are conducting separate studies on different types of tools. The feature vector μ defined above is computed on LSB of 30 colour images (3 colors/image) from both classes. We take different sample each color. So we obtain a total 180 μ vectors on LSB bit planes. (i.e. 30Images*3bitplanes/image*2classes). Out of these, 150 μ vectors are used for training SVM and 30 for testing. Thus we have two classes,

1. LSB plane of non-Stegoed image. 2. LSB plane of stegoed image.

We present the results of classification using SVM in a confusion matrix in Table 4.3.

	non-Stego	Stegoed Image
non-Stego	0.67	0.33
Stegoed Image	0.0	1.0

Table 4.2: Confusion Matrix For 2 Category LSB Classification

The overall accuracy of classification is 85%.

This clearly indicates that the feature vector μ defined above is powerful enough to detect the presence of steganographic operations. To explore its power in discriminating the level of embedding, we consider the four category classification problem with four classes:
Class 1. LSB plane of non-Stegoed image. Class 2. LSB plane of 25% stegoed image.
Class 3. LSB plane of 50% Stegoed image. Class 4. LSB plane of 75% stegoed image.

	0%	25%	50%	75%
0%	0.6	0.0	0.33	0.07
25%	0.0	0.6	0.27	0.13
50%	0.0	0.0	0.4	0.6
75%	0.0	0.0	0.0	1.0

Table 4.3: Confusion Matrix For 4 Category LSB Classification

The confusion matrix for this experiment is in Table 4.3. The overall classification accuracy is 65%. Thus, this indicates that the statistical measures alone are not sufficient for detection of levels of embedding. The reason is that image is a 2D signal and our statistics are based on sequential traversal of image bytes. We take alternative approach in the next chapter.

Chapter 5

Steganalysis : Wavelet Transforms

Our feature vector μ considers a linear sequence of bits as input. However, image properties are in general captured more accurately by two dimensional transforms. Our goal is to classify images accurately under different levels of embedding. The approaches in Section 4.2 and 4.3 serve as good handles in this direction.

To further enhance our understanding of the effects of embedding, we study the behavior of wavelet coefficients. Farid et al [10, 11] have shown that wavelet domain can capture image characteristics, such as whether an image is a natural image or a computer generated one or is a scanned one. They have shown that the feature vector given by them can be used for universal steganalysis. Their aim was only to find whether an image contains any kind of hidden information or not. We further explore the detection of the level of embedding.

5.1 Hypothesis

Our motivation to study the wavelet domain, rather than pixels directly, is that the averages in wavelet coefficients smoothen the pixel values and hence *it is expected* that even minor anomalies in neighboring pixels introduced by stego operation would lead to *amplified* changes in the wavelet domain. We intend to capture and attempt to calibrate these changes w.r.t graded embedding. We consider second level wavelet sub-bands of images. The *Haar wavelet* is used as the mother wavelet.

5.2 Notations

For our experiments, we use 15 images that do not contain any hidden information. These are images taken with a Nikon Coolpix camera at full 8M resolution with most of images stored in RAW format. These images are then cropped to get 800x600 images without doing any image processing operations. Let,

I = $\{I_j : j = 0, 1, 2, \ldots, 14\}$ be the set of natural unstegoed images.

k : The initial LSB embedding present in an given image. i.e. $k\%$ LSB's
of an image have been modified by steganographic operations.

S_k : The Start Image, that is an image $\in \mathrm{I}$ with $k\%$ embedding.
(k is the unknown to be detected.)

i : The forced embedding level.
(will be defined in Section 5.3)

S_{ki} : An image $\in \mathrm{I}$ with $k\%$ original embedding and $i\%$ forced embedding.

5.3 Our Approach

Let S_k be the given image. We call this as the start image. We do additional embedding on it to get S_{ki} and refer this kind of embedding as '*forced embedding*'. Our approach is to compute some transforms on both S_k and S_{ki}, and study a measure of the difference between the transform coefficients for finding k. This procedure is explained with the help of Fig. 5.1. In Fig. 5.1 the transform used is the wavelet transform.

5.4 Definitions

We consider second level wavelet sub-bands. So, each $4 * 4$ block in images will contribute to exactly one wavelet coefficient in each sub-band viz. LL, LH, HL, HH. Let, We consider the 2^{nd} level LL sub-band coefficients, since most of the energy gets concentrated in this sub-band.

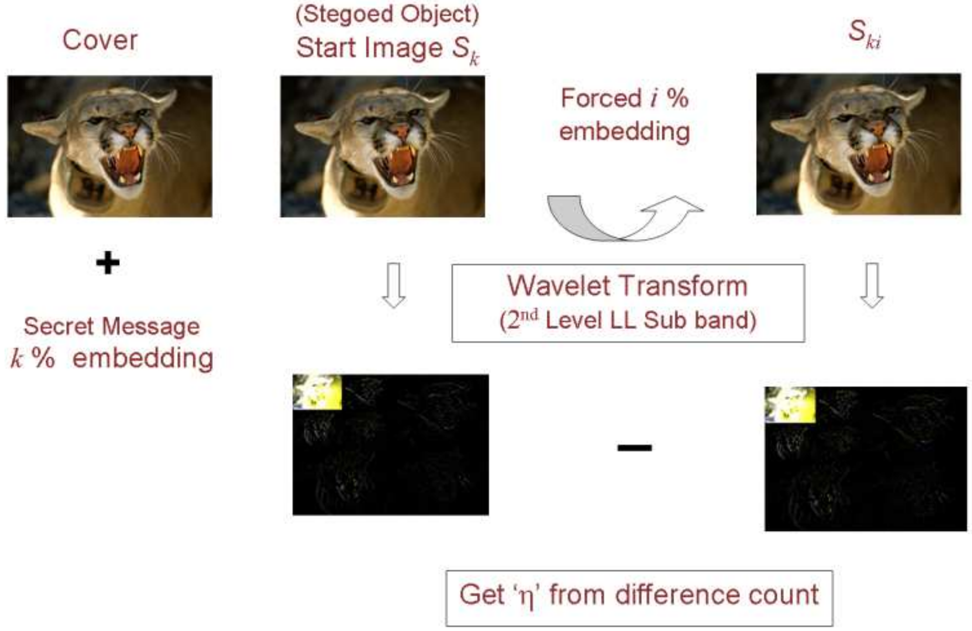


Figure 5.1: *The process to get η*

Let a $4 * 4$ subblock of an image be denoted by :

$$\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}$$

The 2^{nd} level LL wavelet coefficient is given by

$$\tfrac{1}{4} * (a + b + c + d + e + f + g + h \\ +i + j + k + l + m + n + o + p)$$

(Note : The 2^{nd} Level LL sub-band size is $\frac{1}{4}^{th}$ of the original image size in both directions.) The LH coefficient is given by

$$\frac{1}{4} * \{(a + b + e + f + +i + j + m + n) - (c + d + g + h + k + l + o + p)\}$$

The HL coefficient is given by

$$\begin{gathered}\tfrac{1}{4} * \{(a+b+c+d+e+f+g+h) \\ -(i+j+k+l+m+n+o+p)\}\end{gathered}$$

The HH coefficient is given by

$$\begin{gathered}\tfrac{1}{4} * \{(a+b+e+f+k+l+o+p) - \\ (c+d+g+h+i+j+m+n)\}\end{gathered}$$

Let the image S_k be considered as made up of $4*4$ blocks.

Let P denote a $4*4$ block in S_k. $P = (u_{ij})$
and P' denote corrosponding $4*4$ block in S_{ki}. $P' = \left(u'_{ij}\right)$

We define the following random variables,

$$\begin{aligned}
X_0 &= \#\{\mid \sum \left(u_{ij} - u'_{ij}\right) \mid \neq\ 0 : \text{for all non-overlapping blocks P in } S_k\} \\
X_1 &= \mid \sum \left(u_{ij} - u'_{ij}\right) \mid \ \text{ over non-overlapping blocks P in } S_k \\
X_2 &= \sum \left(u_{ij} - u'_{ij}\right) \ \text{over non-overlapping blocks P in } S_k \\
\eta &= \frac{X_0 * 500}{\text{image size in pixels}} \\
\Gamma_W^{ki} &= \text{SNR between } 2^{nd} \text{ level LL sub-band of } S_{ki} \text{ and } S_k
\end{aligned}$$

We have chosen the factor 500 to normalize the quantity η to be near 100 for the size of images being considered $(800 * 600)$.

5.5 Analysis

Let,

$$\begin{aligned}
p &= \text{probability of LSB of pixel in a } 4*4\text{block be even i.e. `0' in Cover } S \\
p' &= \text{probability of LSB of pixel in the } 4*4\text{block be even i.e. `0' in } S_k \\
&= \frac{k}{2} + (1-k)*p \\
&\quad \text{(under assumptions stated in 3.5)} \\
p'' &= \text{probability of LSB of pixel in the } 4*4\text{block be even i.e. `0' in } S_{ki} \\
&= \frac{i}{2} + (1-i)*p' \\
Pr &= \text{probability of a particular } 2^{nd} \text{ level LL wavelet coefficient in } S_{ki} \text{ is different} \\
&\quad \text{from corresponding wavelet coefficient in } S_k \qquad (5.1) \\
\mathrm{E}[X_0] &= \frac{\text{Image Size in Pixels}}{4*4} * Pr \\
\Rightarrow & \\
\mathrm{E}[X_0] &\propto Pr
\end{aligned}$$

Let,

$$\eta_{ki} = \text{expected value of } \eta \text{ with } k\% \text{ initial embedding and } i\% \text{ forced embedding.}$$

Theorem :

i. η_{ki} increases with i and decreases slightly with k.

ii. Γ_W^{ki} increase with increase in k.

Proof: Observe that,

$$\eta_{ki} \propto \mathrm{E}[X_0] \propto Pr$$

$$
\begin{aligned}
Pr &= 1 - \text{prob}\{\mid \sum \left(u_{ij} - u'_{ij}\right) \mid = \; 0\} \\
&= 1 - \text{prob}\{\text{No pixel in the particular } 4 * 4 \\
&\quad \text{block has been replaced with data bits} \\
&\quad\; \text{OR} \\
&\quad \text{2 pixels have been replaced with data bits} \\
&\quad \text{in such way that one pixel value} \\
&\quad \text{increases by 1 and other decreases by 1} \\
&\quad\; \text{OR} \\
&\quad \text{4 pixels have been replaced with data bits} \\
&\quad \text{in such way that two pixel value} \\
&\quad \text{increases by 1 and other decreases by 1} \\
&\quad\;\; \vdots \\
&\quad\; \text{OR} \\
&\quad \text{16 pixels have been replaced by data bits} \\
&\quad \text{in such way that for 8 pixels the value} \\
&\quad \text{increases by 1 and other decreases by 1}\} \\
&= 1 - \{(1 - i/2)^{16} \\
&\quad + (1 - i/2)^{14} * (i/2)^2 * 16C_2 * p' * (1 - p') * \frac{2!}{1! * 1!} \\
&\quad + (1 - i/2)^{12} * (i/2)^4 * 16C_4 * p'^2 * (1 - p')^2 * \frac{4!}{2!2!} \\
&\quad + (1 - i/2)^{10} * (i/2)^6 * 16C_6 * p'^3 * (1 - p')^3 * \frac{6!}{3!3!} \\
&\quad\;\; \vdots \\
&\quad + (i/2)^{16} * 16C_{16} * p'^8 * (1 - p')^8 * \frac{8!}{4!4!}
\end{aligned}
\tag{5.2}
$$

It is logically correct that η_{ki} increases with i. Also it can be seen from equation 5.2 for Pr, that η_{ki} increases with i for $0 \leq i \leq 1$. This can be proved by differentiating equation 5.2 w.r.t. i or can be empirically verified with ease. A close look at the equation reveals that Pr depends upon $(p' * (1 - p'))^{(\text{Some positive integer power})}$. Given a Start image S_k, p' is fixed. But which in turn depends upon p and k. (Refer to Eq. 5.1). As k increases to values of 1, p' goes to $\frac{1}{2}$ irrespective of p. In general, adjacent pixels are very similar in natural images. So, p is biased towards 0.35 or 0.65.

$$\begin{aligned} \Rightarrow \quad p' * (1 - p') \text{ increases} &\quad \text{as } k \text{ increases,} \\ \Rightarrow \quad Pr \text{ decreases} &\quad \text{as } k \text{ increases.} \\ \Rightarrow \quad \eta_{ki} \text{ decreases} &\quad \text{as } k \text{ increases.} \end{aligned}$$

Thus, as Pr decreases with increasing k, the number of wavelet coefficients of S_k and S_{ki} that are equal, increases, i.e. noise in $W(S_{ki})$ w.r.t $W(S_k)$ decreases.
$\Rightarrow \Gamma_W^{ki}$ increases as k increases.

We have verified these experimentally as follows.

5.6 Results

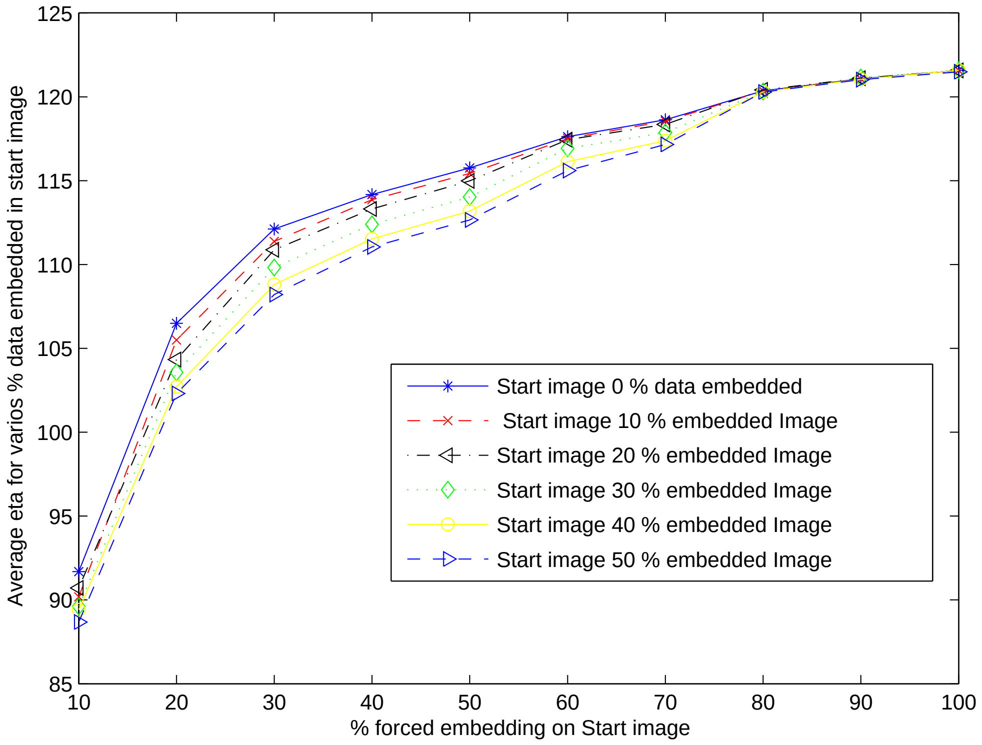


Figure 5.2: *Graph of η_{ki} vs 'i' for various 'k' Hide4PGP*

We use the stego algorithm Hide4PGP in our experiments. In our experiments, we

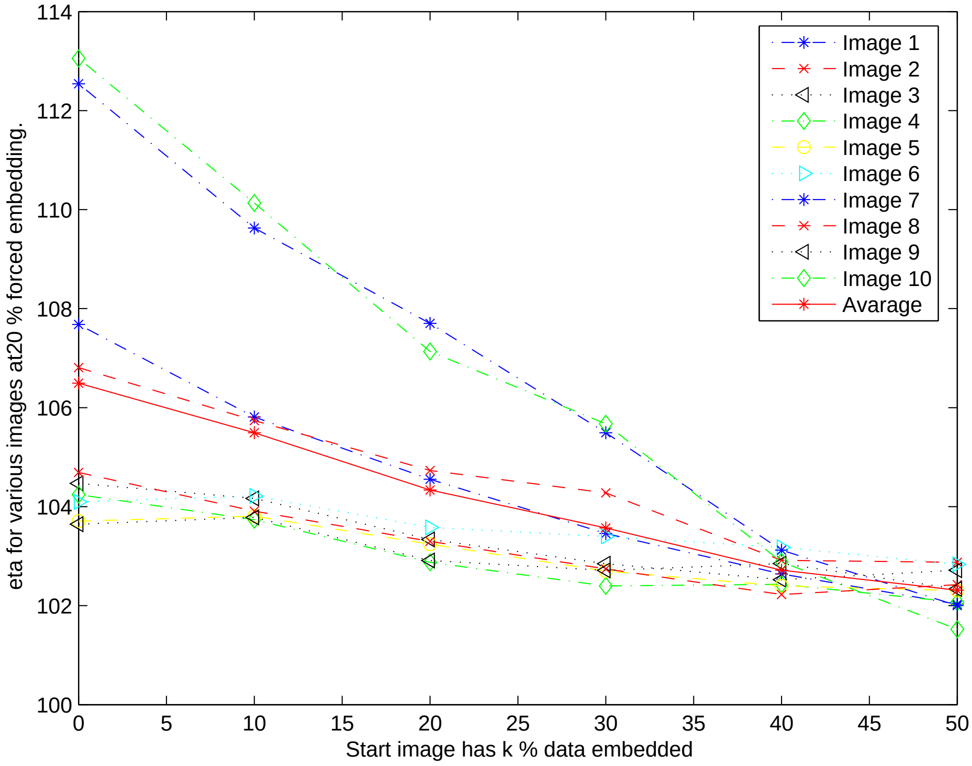


Figure 5.3: *Graph of η vs 'k' for various at fixed forced embedding 20% for various images Hide4PGP*

use $i = 10, \ldots, 100$. $k = 0, 10, 20, 30, 40, 50$. The plots of η_{ki} vs. i for various k is as shown in Fig. 5.2.

For a particular forced embedding say i, it can be observed that η_{ki} decreases as k increases. Encouraged by this monotonic trend, we now look closely at the variations in measure η at a fixed forced embedding of $i = 20\%$, with respect to k on different start images. The results are shown in Fig. 5.3.

The continuous line shows the average value, η_{k20} vs. k. The other curves show the η values for the individual images. These also show the monotonic decreasing trend around the average value. We note that such trends are quite significant especially at low levels of 20% embedding. Thus, this serves as a first indicator for detecting approximately the amount of embedding (even at low levels) in any given image.

It is quite difficult to conduct a large number of data generation experiments under various parameter choices using a public domain tool as we do not get appropriate handles into the source code. Hence, in our lab we have built a tool called CSA-Tool for simulating the behavior of S-Tool. We have taken care to incorporate our own functions for encryption, randomized location generation and embedding analogous to the steps performed by S-Tools. The statistical characteristics of our tools would closely resemble those of S-Tools.

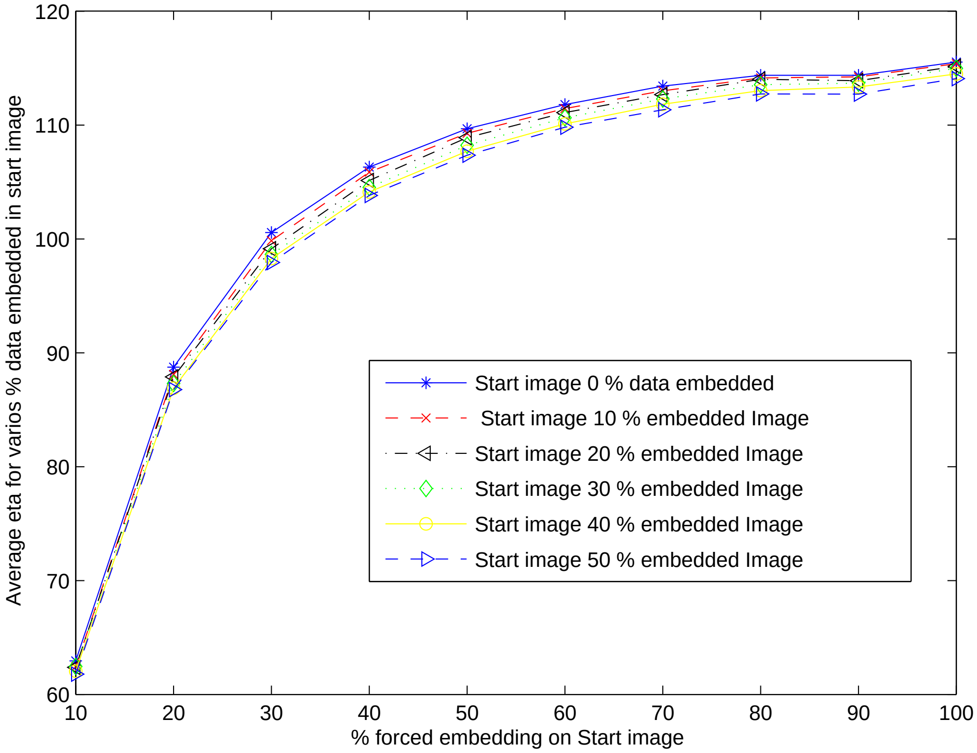


Figure 5.4: *Graph of* η_{ki} *vs. 'i' for various 'k' CSA Tool*

We performed similar experiments as detailed above using the CSA tool. Fig.5.4 and Fig.5.5 show the results. We note that the results are along the same trends as for Hide4PGP. However, the separations in Fig.5.4 are smaller than in Fig. 5.2 and fluctuations in Fig.5.5 are more than in Fig. 5.3. A reason is that the CSA Tool (and S-Tools) employs more strong random generators for choosing the LSB for embedding than the tool Hide4PGP.

The plot, Γ_W^{ki} vs. i for various k is as shown in Fig. 5.6. As per the theorem proved in Section 5.5, it can be observed that for a particular forced embedding say i, Γ_W^{ki} increases

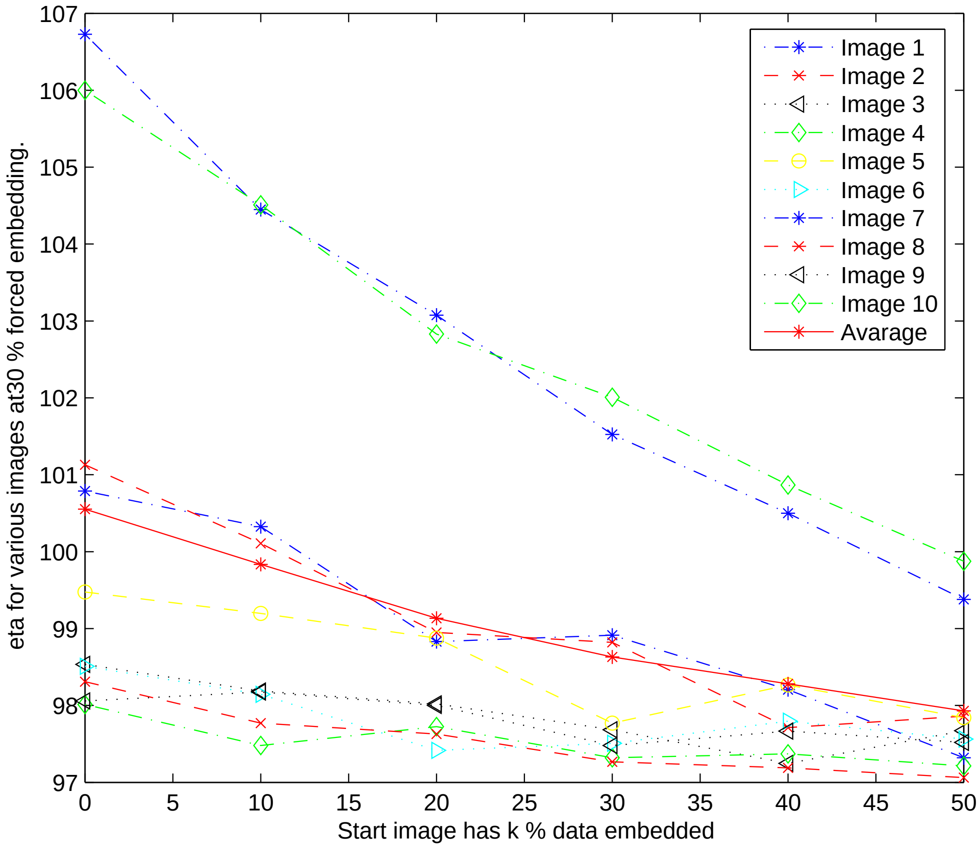


Figure 5.5: *Graph of η vs. 'k' for various at fixed forced embedding 30% for various images CSA Tool*

as k increases. The zoomed version of Fig. 5.6, for $i = 70$ is shown in Fig. 5.7. Encouraged by this monotonic trend, we now look closely at the variations in measure Γ_W^{ki} at a fixed forced embedding of $i = 70\%$, with respect to k on different start images. The results are shown in Fig. 5.6. The continuous line shows the average value, Γ_W^{k70} vs. k. The other curves show the Γ_W^{k70} vs. k values for the individual images. These also show the monotonic decreasing trend around the average value.

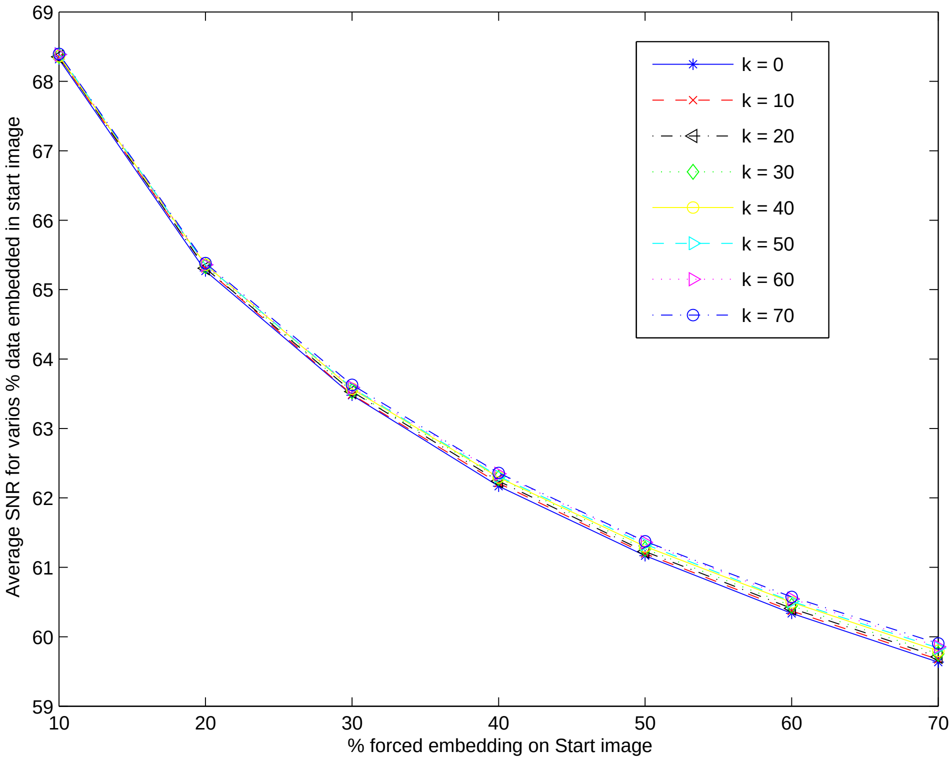


Figure 5.6: *Graph of* Γ_W^{ki} *vs. 'i' for various 'k' CSA Tool*

5.7 Other Experiments based on wavelet transform based statistics

(Note : We will be discussing only the 2^{nd} level wavelet coefficients, so it assumed implicitly). We studied the distribution of wavelet coefficients of LL, HH, HL, LH bands, similar to Farid et al [10] for the purpose of detection of the level of embedding. We note that, Farid [10, 11] had tried just to detect whether given image contains any hidden information or not. Some important observations we have are:

1. The LL subband is the weighted average of neighborhood pixels in $4 * 4$ block, it is expected that the histogram of any color of an image and LL suband of the same color should have similar patterns. The fig. 5.9 is for the green color LL subabnd histogarm of B1.bmp and fig. 5.10 is the green color histogram of the same image. It clearly verifies the fact.

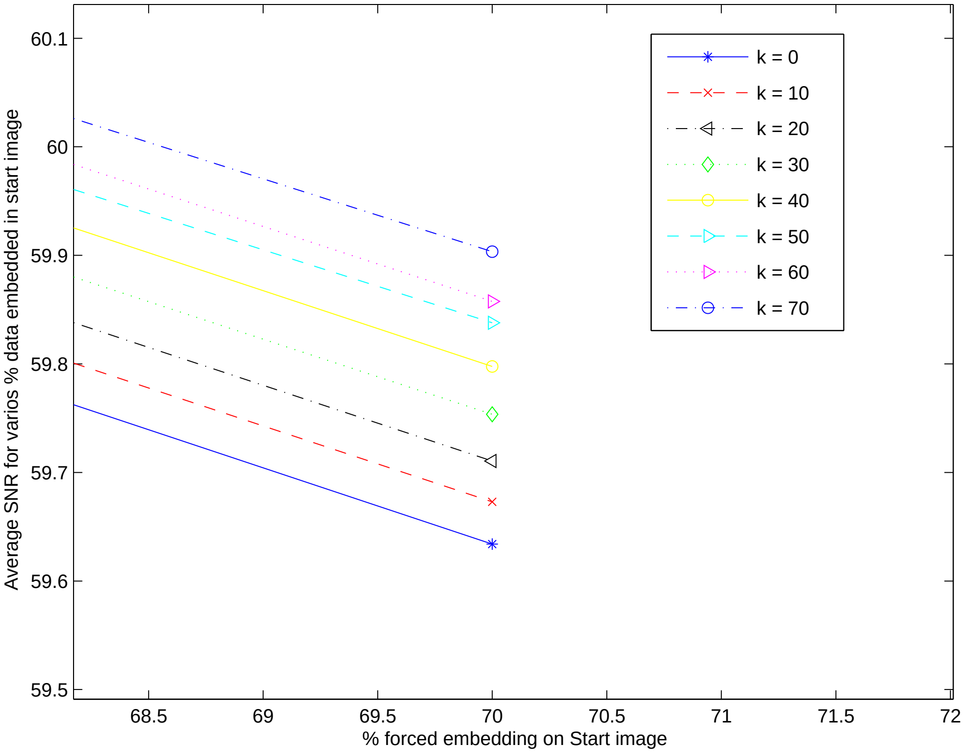


Figure 5.7: *Graph of* Γ_W^{ki} *vs. 'i' for various 'k' CSA Tool - zoomed version*

2. The LH,HL,HH subands of an image follow Gaussian distribution with zero mean. (fig. 5.11-5.13)

Apart from these, we studied the probability density functions of X_1 , X_2 and their characteristic functions w.r.t. i and w.r.t. k at fixed i. These show variations w.r.t. i and w.r.t. k. But the variations in these random variables due to embedding are not sufficient to detect the level of embedding i.e. k. Histogram of X_1 for various i when $k = 0$ is shown in fig. 5.14. And Histogram of X_1 for various k when $i = 50$ is shown in fig. 5.15

We explore the power of the ridgelet transform [14], for the purpose of steganalysis in the next section.

5.8 Ridgelet Transforms

The use of Ridgelet transform [15] for image representation is a recent advancement in image processing. Wavelet transforms are useful in capturing zero dimensional i.e. point

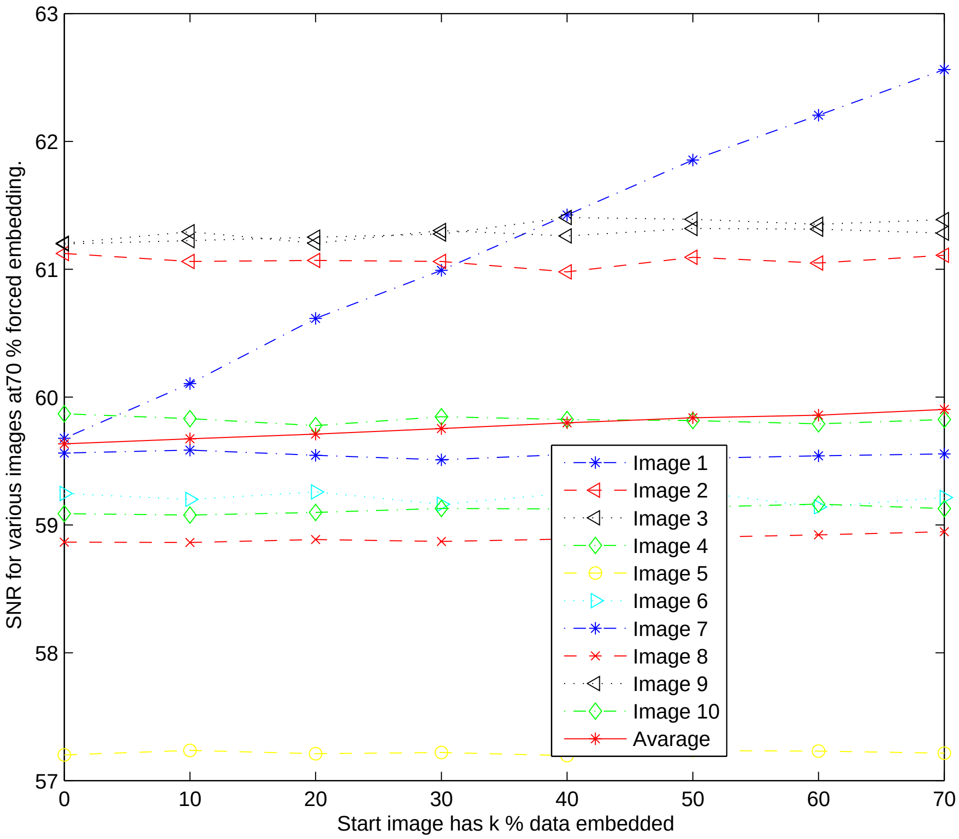


Figure 5.8: *Graph of* Γ_W^{ki} *vs. 'k' for various at fixed forced embedding 70% for various images CSA Tool*

singularities. But an image is a 2-D signal and contains some one dimensional i.e. line singularities. To handle these, first a Radon transform is used to map line singularities to point singularities. Then the wavelet transform is used. This system is called Ridgelet transform and was developed by Candes and Donoho [14, 15]. Minh Do has introduced changes in the ordering of Radon transform coefficients to use it effectively for image representation [16]. The Radon transform works on a $p * p$ size block, where p is prime number.

We studied the use of Ridgelet transform for steganalysis. Our hypothesis is : As Radon transform captures line singularities and any distortions along edges due to steganographic operations, should yield change in Radon coefficients.

Our approach is organized as shown in Fig. 5.1. The transform used is the Finite

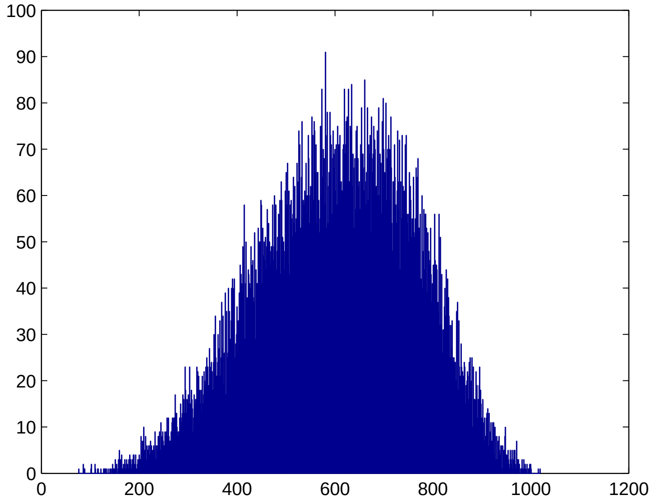


Figure 5.9: *Histogram for LL of green color of B1*

Ridgelet Transform *(FRIT)*. The code for FRIT is obtained from *http://www.ifp.uiuc.edu/˜ minhdo*. (Author of [16].) We used $p = 599$ in our experiments. For Ridglet analysis we defined the following Random Variables.

Let R(I) : denote ridglet transform coefficients of an image I. As before S_k and S_{ki} will denote the start image with unknown $k\%$ embedding and image with forced $i\%$ forced embedding respectively. We consider the coefficients in R(S_k) and R(S_{ki}).

We define, the set X_1 as the collections of p largest coefficients in the set $\mid R(S_k) - R(S_{ki}) \mid$.

We define X_2 as the set containing p numbers from the difference of the p largest coefficients of $R(S_k)$ and p largest coefficients of $R(S_{ki})$.

We define X_3 as the set containing p numbers from the difference of the p largest coefficients of $R(S_k)$ and corresponding p largest coefficients of $R(S_{ki})$.

We define X_4 as coefficients of $R(S_k)$

$\Gamma_R^{ki} =$ SNR between coefficients of $R(S_{ki})$ and $R(S_k)$

We have performed many experiments to study the behavior of the random variables defined above. However so far none of these has led to effective detection of the level of embedding.

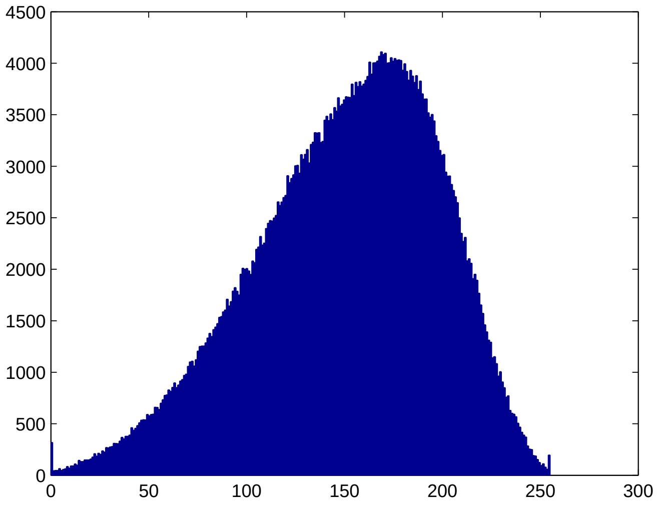


Figure 5.10: *Histogram for green color of B1*

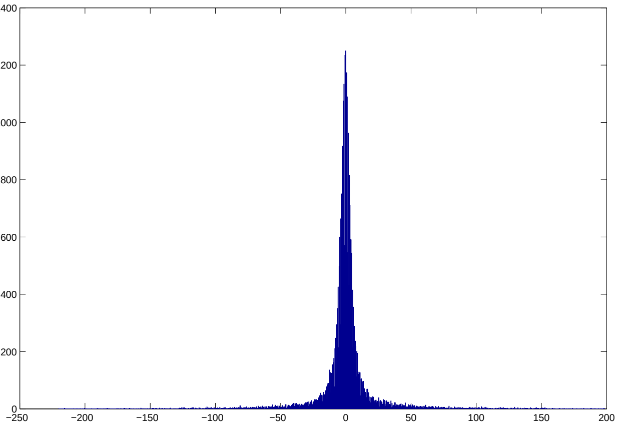


Figure 5.11: *Histogram for LH of red color of B6*

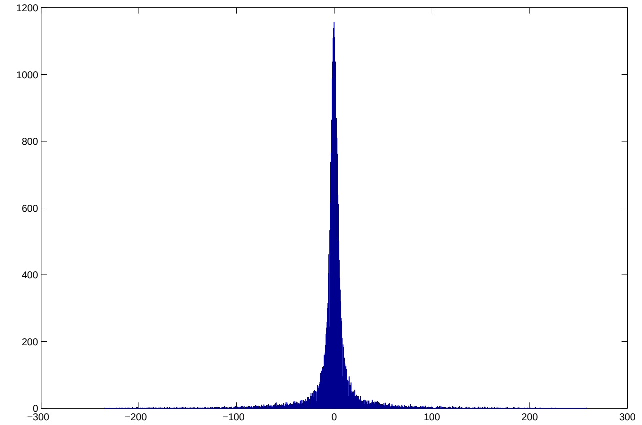


Figure 5.12: *Histogram for HL of red color of B6*

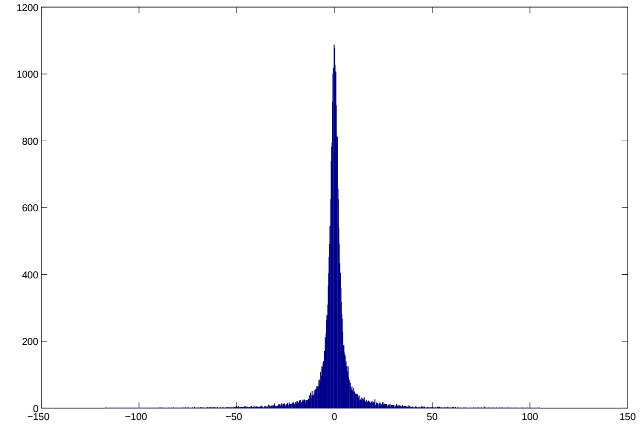


Figure 5.13: *Histogram for HH of red color of B6*

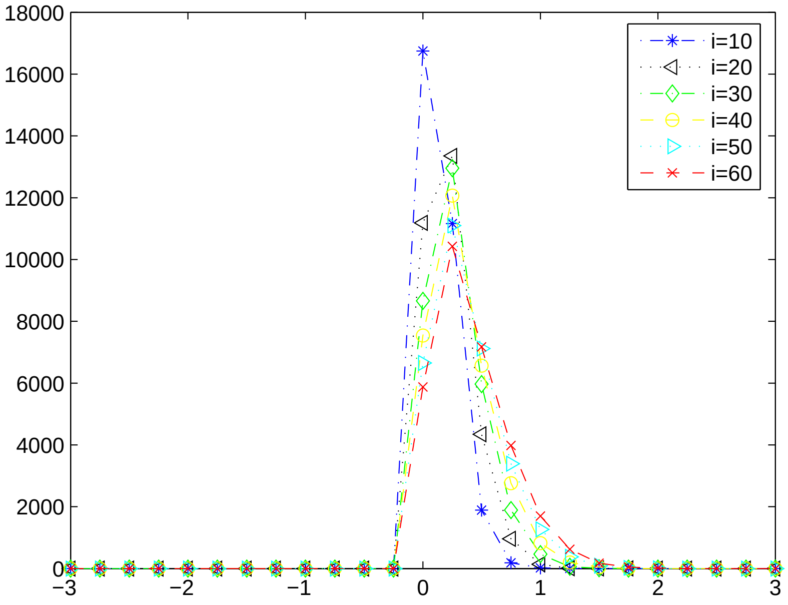


Figure 5.14: *Histogram of X1 of red color of image B4 for various i when $k = 0$*

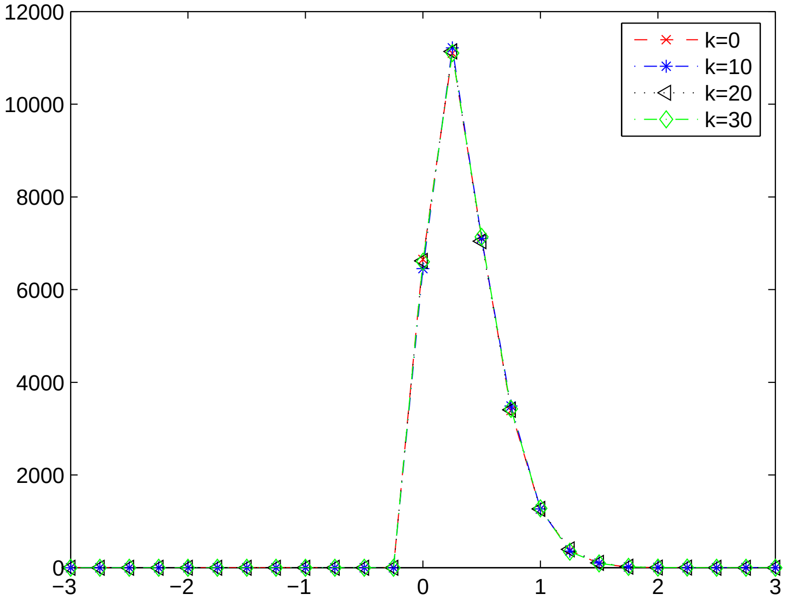


Figure 5.15: *Histogram of X1 of red color of image B4 for various k when $i = 50$*

Chapter 6

Conclusions and Future Work

We discussed two new approaches towards analysis of stego images for detection of levels of embedding. Our approach of using wavelet coefficient perturbations holds promise. We will also consider a modified wavelet coefficient based measure that takes into account the numerical changes in the pixel values introduced by embedding. We plan to use this measure in addition to the statistical measures to arrive at finer detection. We suggest, similar to Expectation Maximization (EM) algorithm used by Machine learning community, for finer detection of level of embedding, the use of two phase approach,

1. First, find some rough estimate of 'p' (defined in Eq. 5.1) using statistical measures.
2. Then using η, Eq. 5.1, p' estimate, k. Use this k to refine p and iterated until not much change.

Once we have a reasonably accurate estimate of embedding, we can use a combinatorial search as well as Bayes formula to estimate the original pixel values. This is at stage hypothetical, and needs to be studied carefully.

The ridgelet transform is very stable with respect to steganographic operations, we may infer that one should used for digital water marking purposes, this transform would lead to robust watermarking schemes

We have presented some of our results of Section 5 in [20].

Bibliography

[1] N Johnson, Sushil Jajodia, “Information Hiding : Steganography and Watermarking - Attacks and Countermeasures,” *Kluwer Academic Publishers*, Third Print, 2003.

[2] N. Jacobsen, K. Solanki, U. Madhow, B. S. Manjunath and S. Chandrasekaran, “Image-adaptive high-volume data hiding based on scalar quantization,” in *Proc. IEEE Military Communications Conference (MILCOM),* Anaheim, CA, USA, Vol. 1, pp. 411-415, Oct. 2002.

[3] A. Westfeld, “High Capacity Despite Better Steganalysis (F5A Steganographic Algorithm)”, in *LNCS* Vol.2137, Springer-Verlag, New York Heidelberg Berlin, pp. 289302, 2001.

[4] N. Provos, “Defending Against Statistical Steganalysis,” in *10th USENIX Security Symposium,* Washington, DC, 2001.

[5] Westfeld and A. Pfitzmann, “Attacks on Steganographic Systems,” *Lecture Notes in Computer Science,* Vol.1768, Springer-Verlag, Berlin, 2000, pp. 61-75.

[6] J. Fridrich, M. Goljan and R. Du, “Reliable Detection of LSB Steganography in Color and Gray-Scale Images,” in *Magazine of IEEE Multimedia Special Issue on Security,* October-November 2001, pp. 22-28.

[7] J. Fridrich, M. Goljan and D. Hogea, “New Methodology for Breaking Steganographic Techniques for JPEGs,” in *Proc. SPIE Electronic Imaging* Santa Clara, CA, Jan 2003, pp. 143-155.

[8] J. Fridrich, M. Goljan and D. Hogea, “Steganalysis of JPEG Images: Breaking the F5 Algorithm,” *5th Information Hiding Workshop,* Noordwijkerhout, The Netherlands, 79 October 2002, pp. 310-323.

[9] J. Fridrich, M. Goljan and D. Hogea, “Attacking the OutGuess,” in *Proc. of the ACM Workshop on Multimedia and Security 2002,* Juan-les-Pins, France, December 6, 2002.

[10] S. Lyu and H. Farid, “Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines,” in *5th international workshop on Information Hiding,* 2002.

[11] Farid H, “Detecting Steganographic Message in Digital Images,” Report TR2001-412, Dartmouth College, Hanover, NH, 2001

[12] C.E. Veni Madhavan, “Statistical Techniques for Cryptanalysis and Steganalysis,” *Workshop on Steganography,* C-DAC, Kolkata, October 2004.

[13] Andy Brown, S-ToolsV4.0,
ftp://ftp.demon.net/pub/mirrors/crypto/idea/
s-tools4.zip

[14] E. J. Cand, “Ridgelets: Theory and applications,” Ph.D. dissertation, Dept. Statistics, Stanford Univ., Stanford, CA, 1998.

[15] E. J. Cand and D. L. Donoho, “Ridgelets: A key to higher-dimensional intermittency,” *Phil. Trans. R. Soc. Lond. A.,* pp. 2495-2509, 1999.

[16] Minh N. Do, “The Finite Ridgelet Transform for Image Representation,” in *IEEE TRANSACTIONS ON IMAGE PROCESSING,* VOL. 12, NO. 1, pp. 16-28, JANUARY 2003

[17] B. Scholkopf and A Smola. *“Learning with kernels”*, MIT Press, Cambridge, MA, USA, 2002.

[18] Shree Lekshmi, CSA Tool Version 4.

[19] Shreelekshmi, *Statistical Techniques for Digital Image Steganalysis*, ME Thesis, under supervision of C E Veni Madhavan, June 2005.

[20] Sujit P Gujar and C E Veni Madhavan, “ Measures for Classification and Detection in Steganalysis”, in *Proceedings of* 3^{RD} *Worshop on Computer Vision, Graphics and Image Processing- WCVGIP 2006,* pp. 210-214, JANUARY 2006.