3-2005

An Analysis of Perturbed Quantization Steganography in the Spatial Domain

Matthew D. Spisak

AN ANALYSIS OF PERTURBED QUANTIZATION STEGANOGRAPHY IN THE SPATIAL DOMAIN

THESIS

Matthew D. Spisak

AFIT/GIA/ENG/05-04

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GIA/ENG/05-04

AN ANALYSIS OF PERTURBED QUANTIZATION STEGANOGRAPHY IN THE SPATIAL DOMAIN

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science

Matthew D. Spisak, BS

March 2005

AFIT/GIA/ENG/05-04

AN ANALYSIS OF PERTURBED QUANTIZATION STEGANOGRAPHY IN THE SPATIAL DOMAIN

Matthew D. Spisak, BS

Approved:

/signed/

_____ _____

Richard A. Raines, Ph.D. (Chairman) Date

/signed/

_____ _____

Gilbert L. Peterson, Ph.D. (Member) Date

/signed/

_____ _____

Rusty O. Baldwin, Ph.D. (Member) Date

Table of Contents

List of Figures

Page

List of Tables

Page

Abstract

Steganography is a form of secret communication in which a message is hidden into a harmless cover object, concealing the actual existence of the message. Due to the potential abuse by criminals and terrorists, much research has also gone into the field of steganalysis – the art of detecting and deciphering a hidden message. As many novel steganographic hiding algorithms become publicly known, researchers exploit these methods by finding statistical irregularities between clean digital images and images containing hidden data. This creates an on-going race between the two fields and requires constant countermeasures on the part of steganographers in order to maintain truly covert communication.

This research effort extends upon previous work in perturbed quantization (PQ) steganography [FrG04] by examining its applicability to the spatial domain. Several different information-reducing transformations are implemented along with the PQ system to study their effect on the security of the system as well as their effect on the steganographic capacity of the system. Additionally, a new statistical attack is formulated for detecting +/- 1 embedding techniques in color images. Results from performing state-of-the-art steganalysis reveal that the system is less detectable than comparable hiding methods. Grayscale images embedded with message payloads of 0.4bpp are detected only 9% more accurately than by random guessing, and color images embedded with payloads of 0.2bpp are successfully detected only 6% more reliably than by random guessing.

AN ANALYSIS OF PERTURBED QUANTIZATION STEGANOGRAPHY IN THE SPATIAL DOMAIN

I. Introduction

1.1 Background

Steganography, a discipline of information hiding, is a form of secret communication in which a message is hidden into a harmless cover object while concealing the actual existence of the message. In the midst of a digital world, steganographers have found many applicable carrier signals including digital images, audio, and video. Over the past decade, researchers have revealed many different approaches by which to hide data into digital media, especially digital images. These methods vary from simple bit substitution with pixels or Discrete Cosine Transform (DCT) coefficients to more advanced techniques which attempt to minimize the added noise introduced from a hidden message. Many of these hiding methods have been developed into freeware, allowing anyone with access to the Internet the ability to communicate messages covertly.

1.2 Motivation

Due to the potential abuse of this form of communication by criminals and terrorists, much research has also gone into the field of steganalysis – the art of detecting and deciphering a hidden message. As novel steganographic hiding algorithms become publicly known, researchers exploit these methods by finding statistical irregularities between clean images and images containing hidden data. This creates an ongoing race

between the two fields and requires constant countermeasures on the part of steganographers to maintain truly covert communication. An ideal steganographic system is undetectable in a sense that images containing hidden data are detected with accuracy no better than random guessing. Further, to prevent future steganalytic attacks, an ideal steganographic system produces stego-images which are statistically identical to clean images. All of these requirements for a secure steganographic system must be accomplished despite the fact the hiding algorithm is publicly known.

Perhaps the most advanced hiding technique to date was recently introduced by Dr. Jessica Fridrich of SUNY-Binghamton [FrG04, FrG05]. This hiding method, entitled perturbed quantization steganography, minimizes the amount of noise added from a secret message by embedding the data while the image is being subject to an information reducing transformation. Elements of the image are chosen to carry hidden data based on their values prior to a rounding step that occurs after a lossy transformation. The system was implemented in the frequency domain by embedding secret messages into DCT coefficients during double JPEG compression. Results from this preliminary work showed that the perturbed quantization hiding method outperformed all of the publicly known steganographic systems which hide messages in the frequency domain.

While this groundbreaking research did apply the system to the process of double JPEG compression, the algorithm itself can be applied more generally to any image transformation resulting in a loss of information. Moreover, the frequency domain limits the number of applicable information reducing operations since the data must be hidden into DCT coefficients.

1.3 Research Objectives

Thus, the primary objective of this research is to apply the concept of perturbed quantization steganography into the spatial domain by surveying lossy image processing operations which involve the rounding of pixel values. In doing so, the overall security of the algorithm in the spatial domain is studied by performing state-of-the-art steganalytic techniques which target spatial hiding methods. Additionally, the lossy image transformations are compared in terms of their security and steganographic capacities in order to determine which operations are best for hiding data in the spatial domain using the perturbed quantization algorithm.

1.4 Preview

This thesis is organized as follows. The next chapter gives a thorough review of previously researched hiding methods as well as state-of-the-art steganalytic techniques which look to detect the presence of hidden data in digital images. A perturbed quantization steganographic system using several different lossy image transformations is explained in Chapter III along with a methodology for testing the system. Chapter IV presents the results from performing state-of-the-art steganalysis on the system, and compares this performance to that of other publicly known steganographic systems. Finally, recommendations are made in Chapter V for future areas of exploration in the field.

II. Related Work

2.1 Chapter Overview

This chapter provides an overview of state-of-the-art research in various areas of steganography and steganalysis. The following section describes the fundamentals behind steganography, and a discussion on digital images in the context of steganography is presented in Section 2.3. Section 2.4 surveys numerous hiding techniques, while Section 2.5 reviews security models. Finally, Section 2.6 outlines various steganographic detection techniques.

2.2 Fundamentals of Steganography

In Katzenbeisser's book [KaP00], three fundamental steganographic systems are defined: a *pure* steganographic system, a *secret key* steganographic system, and a *public key* steganographic system. In pure steganography, no additional information such as keys is needed for the communicating parties other than the embedding and extracting algorithms. However, the security of such a system exclusively relies on the secrecy of the embedding and extracting algorithms. While pure steganography does not involve the use or exchange of any secret information such as stego-keys, both secret key steganography and public key steganography rely on the sharing of such keys. In a secret key steganographic system, a sender hides a secret message into a cover object using a secret key. The key used in the embedding process can also be used to reverse the process in order to extract the hidden data. In this steganographic system, it is assumed that the communicating parties are able to transmit secret keys over a secure channel. Finally, public key steganography models after public key cryptography. In this system, two keys

are needed to transmit secret messages: a public and a private key.

2.3 Steganography and Digital Images

The process of creating multimedia in a digital format introduces a considerable amount of noise [Way02]. Photos, audio clips, and video clips in their digital form are simply a set of numbers which represent an intensity in space and time. For example, a digital photograph is merely a large matrix of numbers signifying an intensity of light at a given place and time. Further, devices such as digital cameras are subject to the randomness of the world which can affect the camera's conversion of photons to bits. Therefore, the creation of digital media is far from perfect, and noise is extremely common. From a steganographers point of view, noise is a good thing. The noise associated with digital media offers an excellent hiding place for secret messages and secret data. In fact, hiding in the noise is the most common approach to steganographic techniques.

Digital images are frequently used as cover objects for steganography. However, there exist both positives and negatives to using digital images for the carrying of secret messages. On the one hand, images are small in data size in comparison to other forms of digital media. As a result, there exists a limited amount of space in which to encode hidden data. For example, an 8-bit grayscale image of size 200 x 200 pixels offers at most 40 kilobytes of data to embed. This is equivalent to a 5 second voice audio clip or 1 frame of video from a NTSC TV [BeG96]. Therefore, digital images do not make good cover objects for hiding large amounts of data (video and audio). Additionally, digital images transmitted over the Internet and otherwise are liable to information reduction transformations such as cropping and lossy compression, and thus pose a threat to

altering the hidden data.

In spite of these limitations, digital images make good stego-objects because changes in them are imperceptible to the human eye. In particular, the human eye has very little sensitivity to changes in brightness across an image [BeG96]. Finally, the omnipresence of images on the Internet makes them an excellent choice for cover objects in covert communication.

2.4 Steganography Techniques

In this section, an overview of steganographic embedding techniques for digital images is provided.

2.4.1 Hiding in the Spatial Domain

Perhaps the simplest form of steganography involves the substitution of message bits for cover data. In such techniques, pixels of the cover image are chosen and substituted with message bits in such a way that it is imperceptible to the human visual system. Typically, the least significant bits (LSB) of pixels of the cover image are chosen for substitution as they alter the value of the pixels by the least amount. This form of steganography is also common with audio [KaP00]. Figure 1 depicts this form of steganography.

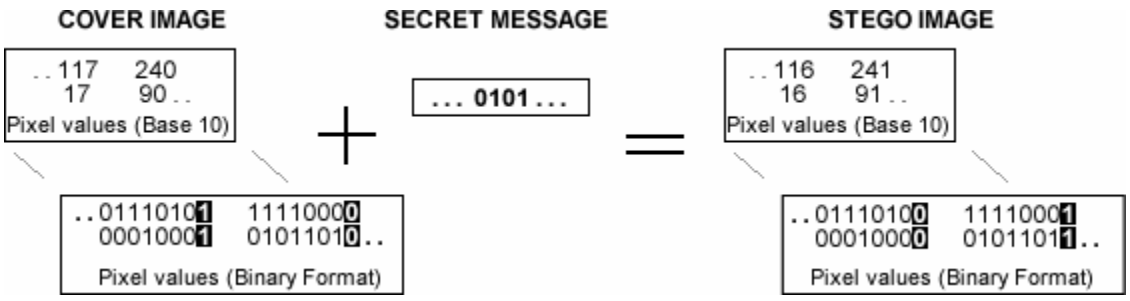

Figure 1. Least Significant Bit Substitution

There exist numerous ways to select a subset of pixels from the cover image for substitution. The most elementary approach involves starting at the very first pixel and embedding the stream of message bits by substituting them one by one into the LSB of each and every cover pixel until the message stream has ended. This method is a bad choice for two reasons: 1) In most cases, the number of bits in the secret message will be less than the number of pixels in the cover image. As a result, the part of the cover image where the message exists will be statistically different than the remainder of the cover image. 2) An attacker knows which elements contain the secret message and can easily extract the message.

A somewhat better selection method uses a pseudorandom number generator to distribute the message bits throughout the cover image. In this technique, the sender and receiver agree on a secret stego-key which is used to seed a pseudorandom number generator (PRNG). The sequence generated by the PRNG represents a sequence of cover pixel indices, and the stream of message bits is then substituted into the least significant bits of those cover pixels with indices in the sequence. The receiver can then regenerate the same sequence of indices using the shared stego-key and extract the message bits accordingly. Steganos [Ste04] is a freeware utility which utilizes a random number generator to choose a more dispersed subset of pixels for LSB substitution.

Another technique for LSB steganography incorporates the idea of hiding bits within a set of cover pixels rather than hiding a bit into one element [AnP98]. The secret message bit is embed into the parity of a group of pixels. For example, a parity function is calculated for a given group of elements, and if the message bit differs from the parity of the group only the LSB of one element within the group needs to be modified.

Kurak [KuM92] demonstrated the ability to embed classified images into unclassified images using a simple bitplane substitution of each pixel in a cover image. In the cautionary paper, Kurak warned that downgrading images – the process of declassifying an image – should be done carefully to ensure that a downgraded image has not been "contaminated" with a secret image. Two equally sized 8-bit grayscale images: a cover-image and a secret image, were combined using only the four most significant bits of each pixel in both images. The four most significant bits of every pixel in the secret image were substituted into the four least significant bits of every pixel in the cover image. Extracting the hidden image simply consisted of shifting the four LSB's of the cover image over to the four most significant bits. Results showed that keeping only the four most significant bits per pixel produced a surprisingly small degradation in image quality, and provided sufficient data to view both the cover image and the secret image.

2.4.2 Hiding in the Transform Domain

While LSB steganography hides data in the least significant areas of image pixels, this data is not robust against lossy digital processing operations such as compression, and cropping. Therefore, other information hiding techniques embed data in more significant parts of the cover image.

A common steganography technique involves embedding data in the transform domain. In particular, the Discrete Cosine Transformation (DCT) makes it possible to hide data in images of the JPEG format. This is because the JPEG lossy compression algorithm is centered around the DCT. The mathematical formulas for the Discrete Cosine Transform and inverse transform are presented in Figure 2.

$$T(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{\pi u(2x+1)}{2N}\right)\cos\left(\frac{\pi v(2y+1)}{2N}\right)$$

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u)\alpha(v)T(u,v)\cos\left(\frac{\pi u(2x+1)}{2N}\right)\cos\left(\frac{\pi v(2y+1)}{2N}\right)$$

$$\text{where} \quad \alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for} \quad u = 0 \\ \sqrt{\frac{2}{N}} & \text{for} \quad u = 1,2...,N-1 \end{cases}$$

Figure 2. The Two-Dimensional Discrete Cosine Transform [GoW02]

JPEG (Joint Photographic Experts Group) is generally accepted as a standard for compression of digital images [AnM00]. In JPEG compression, an image is first divided into blocks of 8x8 pixels. Next, the two-dimensional DCT is applied to each pixel within each block resulting in blocks of 8x8 DCT coefficients. The DCT coefficients are then divided by quantization steps (quantum's) from the quantization matrix and rounded off to the nearest integer. Finally, the quantized DCT coefficients are encoded into a binary stream, and the stream is written to an output file with the extension .jpg or .jpeg. The quantization of DCT coefficients [Wol04] is where information from the original image is lost, and as a result the data is compressed to a smaller size. Figure 3 shows a diagram of the JPEG compression process.

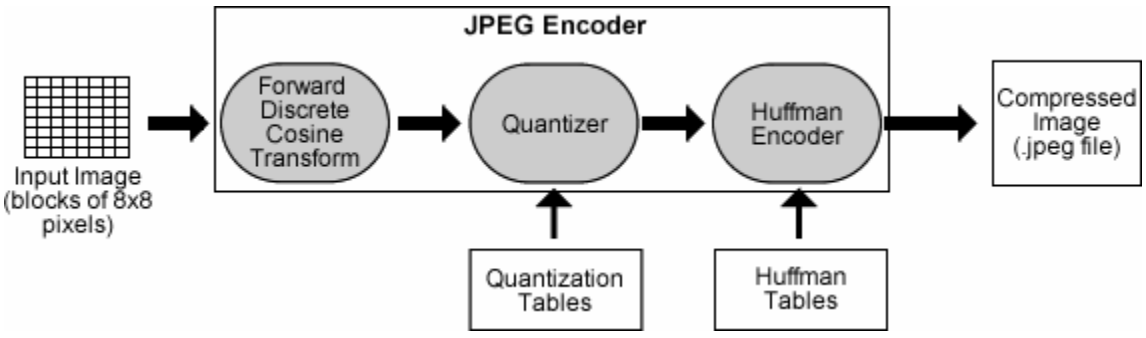

Figure 3. Diagram of JPEG Compression Algorithm

Katzenbeisser [KaP00] introduced a steganographic system which uses the relative size of a pair of DCT coefficients to hide message bits. The encoding algorithm chooses blocks b_i from the cover image in a pseudorandom order, and the i^{th} message bit is encoded in the block b_i. The communicating parties agree on a pair of indices such that their corresponding quantization values from the quantization matrix are equal. Within each block, when the first DCT coefficient of the known pair of indices is greater than the second coefficient the block represents the message bit '1'. Likewise, when the second coefficient is greater than or equal to the first, the block signifies a '0'. Further, the encoder simply swaps DCT coefficients when the message bit is not already encoded.

As was described in the JPEG lossy compression system, there is no information lost after the quantization of DCT coefficients has occurred. Once DCT coefficients have been divided by quantums from the matrix, the coefficients are usually rounded to the nearest integer. Derik Upham [Uph97] proposed a technique for hiding data in the DCT coefficients of a JPEG image by tweaking the rounding that occurs. The DCT coefficient rounding process is modified in such a way that the coefficient is rounded up or down to match the message bit. The secret message bit stream is thus embedded into the least significant bit of consecutive non-zero DCT coefficients from the cover image [Uph97].

2.4.3 The Selection Channel: Non-adaptive vs. Adaptive Steganography

The previously described information hiding techniques all have one thing in common: each one is non-adaptive. Non-adaptive steganography does not take into consideration the unique characteristics of a given cover image. Instead, hidden data is always embedded into predefined locations or embedded using PRNG seeds independent of the cover image. While such techniques are relatively straightforward and easily

decoded by the receiver, an adversary can use their knowledge of the fixed selection rules to mount an educated attack. Additionally, non-adaptive techniques do not take advantage of the possibility of hiding in the random noise which occurs in digital images. As a result, many researchers suggest using techniques which adaptively select areas of a cover image in which to hide information. It is believed that adaptively choosing good hiding spots can increase the security of steganographic systems. However, Fridrich [FrG04] noted that a publicly known selection rule or a rule which is "weakly dependent on a key" provides an adversary with a starting point to mount an attack.

Neils Provos [Pro01] introduced a more statistically conscious approach to Upham's JSTEG. In his program Outguess, the hidden message is embedded into the least significant bit of DCT coefficients of a cover image similar to JSTEG. However, the Outguess algorithm attempts to perform statistical error correction by offsetting all data bits that are changed. Each time a bit is flipped to hide information, a second bit is changed in the opposite way in order to maintain a statistical balance in the image.

In an attempt to preserve statistical properties of an image, Franz [Fra03] described an adaptive steganographic approach which selects pixels for embedding such that after the message has been embedded the characteristics of the histogram of the stego-image match that of the original cover image. A histogram, a first order statistic, is often used to characterize the frequencies of the colors or shades of gray in an image. Simple LSB substitution modifies this distribution of colors (shades) which makes stegdetection easier. In Franz's method, after a histogram is computed for the cover image, pixel values are separated into usable groups. A usable group is defined as a group of consecutive pixel values (color or shades of gray) in which every element of the group

occurs at least once in the cover image. Message bits are then embed only in those pixels which belong to usable groups. Therefore, the selection rule for this adaptive technique involves hiding message bits only in pixels that belong to usable groups. As a result, both the cover image and the stego-image have matching distributions of frequency of pixel values.

Topkara [ToT04] introduced a protocol for adaptive steganography by partitioning the cover object into a hierarchical tree-like structure. In the hierarchical structure, the cover object is partitioned into finer and finer regions as you traverse down levels. Therefore, the lowest level (leaf nodes in a tree structure) is a representation of the cover object partitioned into the smallest blocks. These blocks are referred to as elementary blocks $R(N_i)$. Additionally, a metric is chosen to measure the detectability $d()$ or presence of suspicious data in each elementary block. For example, a suggested detectability metric is the deviation of certain statistics of an elementary block from statistics from a similar region in a known database. The protocol thus selectively chooses blocks of the cover object such that the addition of a message bit will not increase the detectability $d(R(N_i))$ above some threshold τ. Further, a suitability function $S(N_i)$ is defined to determine whether or not the addition of any message bit in node Ni will increase the detectability metric above the threshold τ ($d(R(N_i)) > \tau$). The secret message is then only embedded into those elementary blocks $R(N_i)$ in which the detectability of steganography in that block does not surpass a given threshold.

In 1996, Toby Sharp [Sha01] used his own steganographic software to communicate covertly with a colleague living in a country which monitored e-mail. In his software utility entitled Hide v2.1, secret messages are encrypted and embed into the

least significant bits of pixels. Whereas simple substitution systems replace the LSB of pixels with the message bit, Sharp's software randomly increments or decrements the entire pixel value in order to match the LSB with the message bit. This embedding method is frequently called +/- 1 embedding. Table 1 shows the difference between simple substitution and +/- 1 LSB embedding. Further, the algorithm is adaptive in that the pixels selected to contain hidden data are dependent on the content of the image as well as a secure stego-key. More specifically, once a pixel has been embedded with secret data, the most significant bit and least significant bit from the pixel value are concatenated with randomly generated bits in order to determine the next pixel in the image to hide data into.

Table 1. A Comparison of Embedding Techniques [FrG04b]

	LSB SUBSTITUTION		LSB +/- 1 EMBEDDING	
Pixel Value	$MessageBit = 0$	$MessageBit = 1$	$MessageBit = 0$	$MessageBit = 1$
$2k$	$2k$	$2k+1$	$2k$	$2k+1$ *or* $2k-1$
$2k+1$	$2k$	$2k+1$	$2k$ *or* $2k+2$	$2k+1$

The concept of information hiding using added side information only known to the sender was proposed by [FrG05] in an attempt to alleviate the problem of adaptive steganographic selection rules being publicly known. In this model, it is proposed that the sender could utilize some side information (such as a raw, unmodified digital image) which is unavailable to both the receiver and an adversary. It is shown that the sender can embed their message in a modified/compressed form of the original cover object, while using the side information from the original object to select where data will be hidden. For the selection channel, Fridrich introduces a concept called perturbed quantization steganography. Whereas Upham [Uph97] introduced the idea of tweaking the rounding

process of the JPEG compression by rounding DCT coefficients up or down to represent the message bit, Fridrich expands on this concept to enhance security. Instead of simply rounding up or down all DCT coefficients as Upham introduced, it is proposed that prior to the rounding of the coefficients, only those coefficients whose fractional part is in a close interval about the value 0.5 will be selected as candidate elements for hidden information. This interval is defined with some tolerance ϵ (0.5- ϵ, 0.5+ ϵ), and the coefficients which lie in this interval are called changeable coefficients. Further, Fridrich defines that the perturbed quantization model can be applied to any information reducing operation such as image downsampling, and analog to digital conversion. Additionally, the model applies not only to DCT coefficients in the frequency domain, but also to pixel values in the spatial domain. The algorithm was implemented using the reduction operation of JPEG double compression and results showed that the stego-images which were embedded using this added side information were very rarely detected in comparison to the Outguess algorithm. The perturbed quantization steganographic system is described in detail in Chapter III.

2.5 The Security of a Steganographic System

It is difficult to define and classify what constitutes a secure steganographic system; however, some of the properties of secure cryptographic systems do apply. Auguste Kerckhoffs published a document in 1883 entitled *La Cryptographie Militaire* [Ker83] which outlined six principles that a cryptographic system should possess. In his second principle, Kerckhoffs states that the technique used to encipher data should not require secrecy, and can be stolen by the enemy without causing trouble. Therefore, the

security of a crypto-system relies solely on the secrecy and choice of key. Following Kerckhoffs principles, Katzenbeisser [KaP00] defines a secure steganographic system as follows: *If an enemy knows the stego-system and it's technique for hiding data, but has no information about the stego-key, then it follows that the system is secure if and only if the enemy cannot obtain any evidence or suspicion that a covert communication occurred.*

Zöllner [ZöF98] characterized the breaking of a steganographic system in two phases: 1) The attacker can detect the presence of steganography in a cover object and 2) The attacker is able to extract and read the hidden message. However, Zollner states that only the first phase needs to be achieved for a steganographic system to be declared insecure.

Additionally, theoretical information security models have been described which attempt to define security in the context of steganographic systems. For example, Cachin's model [Cac04] parallels Shannon's security model for cryptographic systems [Sha49]. However, many of these theoretical security models have rather impractical assumptions. First, in the case of unconditional security it is assumed that a warden, someone monitoring traffic for hidden messages, has unlimited computational power in order to exhaust all possible stego-keys [AnP98]. Other security models such as Cachin's [Cac04], assume that the communicating parties have knowledge of the probability distributions of a finite set of cover objects, and stego objects. These assumptions will seldom hold true in a real world scenario. Katzenbeisser [KaP02] attempts to define secure steganography in a more realistic manner. Additionally, Fridrich [FrG02] states that a steganographic technique is considered secure if its stego-objects have nearly

identical statistical properties to the corresponding cover objects. That is, if the process of embedding a secret message in the cover object does not introduce any detectable feature into the resulting stego-object, the system is secure. Furthermore, a system is broken if a detection algorithm can differentiate between cover objects and stego-objects with a greater likelihood than guessing at random (better than 50% detection).

2.6 Steganalysis Techniques

It is evident that there exist numerous ways to hide data into cover objects such as digital images. For each newly invented technique to hide data, researchers attempt to formulate novel counterattacks. Thus, the invention of novel ways to embed secret data fuels the field of steganalysis – the art of detecting the existence of secret communication. Therefore, the fields of steganography and steganalysis incorporate a spy vs. spy mentality.

The goal in steganalysis is to detect the existence of a hidden message, to extract the hidden message, and to disable or corrupt the secret message. The latter two are extremely difficult tasks, especially for novel forms of steganography. Therefore, current research in steganalysis is focused primarily at detecting the very existence of steganography. Subsequent efforts have been made at estimating the secret message length once stego-images are accurately detected. However, in order to differentiate clean objects from stego-objects, a set of discriminating features are needed. As a result, many researchers have investigated discriminating features between clean images and stego-images. The following section provides an overview of some state-of-the-art steganalytic techniques and algorithms.

2.6.1 RS Steganalysis

Fridrich [FrG01] introduced a technique for detecting least significant bit (LSB) steganography in digital images called RS Steganalysis. RS Steganalysis produces a threshold-free statistic which provides an estimate of the secret message length hidden in a cover image. For those images which don't contain a hidden message, the outputted RS-statistic is approximately normally distributed about 0. The method was tested and found to be much more reliable in detecting LSB steganography in non-sequential pixel embedding. As the RS-statistic will be used in the experiments within this investigation, the method is discussed in detail.

The technique starts by splitting the image into separate groups of n neighboring pixels $(x_1, x_2, \ldots, x_n)$. Each pixel group $G = (x_1, x_2, \ldots, x_n)$ is assigned a real number value based on f, a discriminating function $f(x_1, x_2, \ldots, x_n) \in \Re$. The discriminating function f classifies the noisiness of each group G. As the noisiness of an image increases, so does the value of f. A commonly used function f is one which measures the variation of the group G:

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n-1} \left| x_{i+1} - x_i \right| \qquad (1)$$

If P is defined as the set of all possible pixel values for an image, then for an 8-bit grayscale image $P = \{0,1,2,\ldots,255\}$. Next, three flipping operations are defined on the set P: F_1, F_{-1}, F_0 such that they are invertible $\left(F_i(F_i(x)) = x, \forall\, x \in P\right)$. Further, the three flipping operations are defined as follows:

$$F_1 : 0 \leftrightarrow 1,\ 2 \leftrightarrow 3, \ldots,\ 254 \leftrightarrow 255$$

$$F_{-1} : -1 \leftrightarrow 0, 1 \leftrightarrow 2, \ldots, \ 255 \leftrightarrow 256$$

$$F_0 : 0 \leftrightarrow 0, 1 \leftrightarrow 1, \ldots, \ 255 \leftrightarrow 255$$

Therefore, the flipping operation F_1 has the effect of flipping the LSB of a given pixel, while F_{-1} performs a shifted flipping of the LSB of a pixel. Finally, each pixel group G is classified using f and F into one of three categories: (R) Regular, (S) Singular, or (U) Unusable groups.

$$G \in R \ \Leftrightarrow \ f(F(G)) > f(G)$$

$$G \in S \ \Leftrightarrow \ f(F(G)) < f(G)$$

$$G \in U \ \Leftrightarrow \ f(F(G)) = f(G)$$

However, it is possible for different flipping operations to be applied to various pixels within each group G. Thus, a mask M is defined to encapsulate which flipping operation is applied to which pixel for each group G. M is a vector of n elements such that $\forall i \in M, i \in \{-1, 0, 1\}$. Flipping an entire group G can thus be defined as $F(G) = \left(F_{M(1)}(x_1), F_{M(2)}(x_2), \ldots, F_{M(n)}(x_n)\right)$. By flipping only certain pixels within each group, the function $F(G)$ has the effect of additive noise similar to that introduced by pixel based LSB steganography. In clean cover images, the flipping function results in an increase in variation as measured by the discriminating function. As a result, clean images contain a greater amount of Regular groups (R) than Singular (S).

2.6.1.1 The RS Hypothesis

The relative number of Regular groups is represented by R_M, and S_M signifies the relative number of Singular groups using the mask M. Similarly R_{-M} and S_{-M}

denote the relative number of respective groups using the mask $-(M)$. Fridrich, introduced and verified the RS-hypothesis that for a clean cover image: $R_M \cong R_{-M}$ *and* $S_M \cong S_{-M}$. Additionally, it is shown that for a clean cover image: $R_M > S_M$ *and* $R_{-M} > S_{-M}$. Once a message has been embedded into the LSB of certain pixels in an image, the difference between R_M and S_M decreases. Moreover, as the message payload approaches 100% (1 secret message bit per pixel), around 50% of the pixels LSB are changed, and as a result the difference between R_M and S_M approaches zero $R_M \cong S_M$.

2.6.1.2 RS Statistic

Experiments showed that the relevant number of regular and singular groups form quadratic curves as the message payload varies (see Figure 4). Accordingly, a statistic p was derived to estimate the secret message length (proportion of an image which contains a hidden message). Therefore, the statistic itself determines first the estimated secret message length, and in turn a decision can be made as to whether a given image is clean or contains secret data. The statistic was able to detect message payloads as small as 1%. The RS-Statistic is derived for a given image as follows:

1) RS Steganalysis is applied to the provided image using masks $M, -(M)$ in order to obtain the points $R_M(p/2), R_{-M}(p/2), S_M(p/2)$, and $S_{-M}(p/2)$.
2) Every LSB of every pixel in the image is flipped, and RS-Steganalysis is applied to the "flipped" image using masks $M, -(M)$ in order to obtain the points $R_M(1-p/2), R_{-M}(1-p/2), S_M(1-p/2)$, and $S_{-M}(1-p/2)$.

3) The quadratic equation $2(d_1 + d_0)x^2 + (d_{-0} - d_{-1} - d_1 - 3d_0)x + d_0 - d_{-0} = 0$ is solved to obtain the roots x_0, x_1 with $d_0 = R_M(p/2) - S_M(p/2)$, $d_1 = R_M(1-p/2) - S_M(1-p/2)$, $d_{-0} = R_{-M}(p/2) - S_{-M}(p/2)$, and $d_{-1} = R_{-M}(1-p/2) - S_{-M}(1-p/2)$

4) Lastly, the RS-statistic p is computed by

$$p = x \div (x - 1/2) \qquad (2)$$

where $x = \min(|x_0|, |x_1|)$.

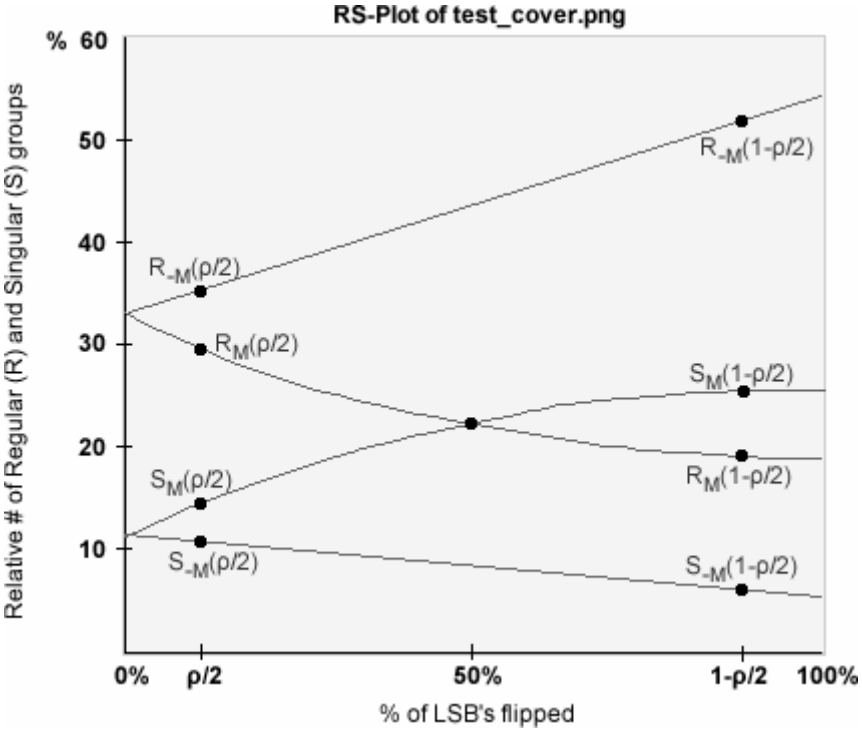

Figure 4. RS Statistics of a Test Image [FrG01]

Andrew Ker [Ker04] further researched the concept of RS-steganalysis by focusing on analyzing the distribution of the RS-statistic, as well as studying the effect of varying the mask M. Whereas, Fridrich [FrG01] claimed that the distribution of the RS-

statistic for clean images was normally distributed, Ker provided evidence that this claim was not entirely accurate. It was found that the distribution of the RS-statistic had a much more heavily tailed curve than that of a normal curve. That is, the kurtosis values for the RS-statistic were significantly greater than that of the normal distribution's curve. Additionally, Ker varied the mask M from various flat masks (1-Dimensional row vectors) and various square masks ($n \times n$ matrices). Results showed that masks are a factor which affects detection performance. In particular, it was found that square masks performed better than flat masks, with the masks $M = [0,1,0]$ and $M = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ performing the best out of all masks tried. Finally, Ker concluded that reliable detection could not be made for payloads lower than 1%-2%. Nonetheless, RS-steganalysis is still the best choice for detecting LSB steganography.

2.6.1.3 Other applications of RS-Steganalysis

There exist several additional distinguishable features from the RS-hypothesis which can be used to differentiate clean images from stego-images. McBride [McB03] computed the value $S_M \div R_M$ for each image in order to express the relationship between the relative number of singular groups versus the relative number of regular groups. The closer the ratio gets to 1, the higher the probability that a given image contains a hidden message in the LSB of pixels (If $S_M \div R_M \cong 1$, then the secret message payload $\cong 100\%$). However, clean digital images usually contain a small amount of noise due to the randomness of the world. This initial bias of noise can result in $S_M \div R_M$ ratios as high as 0.5 for clean images. Hence, the initial bias hinders McBride's proposed

statistic from detecting small message payloads. Other possible distinguishable features from RS-Steganalysis which could be used include: $R_M \div R_{-M}$ (The ratio is approximately equal to 1 for clean images, and the ratio decreases towards zero as the message payload increases), and $S_{-M} \div S_M$ (The ratio is approximately equal to 1 for clean images, and the ratio decreases towards zero as the message payload increases).

2.6.2 Chi-Squared Attack

Westfeld [WeP99] presented a statistical attack on stego-images by analyzing the histogram of a given image. This technique centers around the concept of a pair of values (PoVs). A pair of values can be defined as two elements from the histogram (such as pixel values) whose frequency distribution only differ by the least significant bit. Prior to embedding, the PoV's are distributed unevenly; however, after embedding a message using a LSB technique the pairs of values become equally distributed. Westfeld's approach calculates a theoretical expected frequency distribution of stego-images as well as a sample distribution from the image under question, and the values are subsequently tested for equality using the Chi-Squared test. For instance, an 8-bit grayscale image has 256 possible pixel values c_i. Therefore, there exist k PoV's such that $k \leq 128$. For a given pair i, $(i = 1,2,\ldots,k)$, the theoretical expected frequency of i is calculated by $n_i* = (number\ of\ occurrences \in \{c_{2i}, c_{2i+1}\}) \div 2$. Additionally, a measured frequency of distribution $n_i = (number\ of\ occurrences\ of\ c_{2i})$ is computed for each pair i. The X^2 statistic is defined as:

$$X_{k-1}{}^2 = \sum_{i=1}^{k} \frac{(n_i - n_i*)^2}{n_i*} \qquad (3)$$

with k-1 degrees of freedom. The probability that the distributions n_i* and n_i are equal is determined by:

$$p = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma\left(\frac{k-1}{2}\right)} \int_0^{X_{k-1}^2} e^{-\frac{x}{2}} x^{\frac{k-1}{2}-1} dx \qquad (4)$$

Therefore, the p-value is close to zero for clean images and close to 1 for stego images. However, this technique is most effective at detecting hidden messages which are embed in consecutive pixel LSB's rather than LSB hiding techniques which spread out the hidden message.

2.6.3 Raw Quick Pairs

Fridrich [FrD00] created an attack on color images involving the number of unique colors of an image. The motivation behind the attack comes from the idea that LSB substitution in the spatial domain, results in an increased number of close colors – colors whose pixel values are extremely close. Two colors $(R_1, G_1, B_1); (R_2, G_2, B_2)$ are considered a close color pair *iff* $|R_2 - R_1| \le 1, |G_2 - G_1| \le 1, and\ |B_2 - B_2| \le 1$. Further, the number of unique colors present in the image is denoted by *U*, and the total number of close color pairs on an image is symbolized by *P*. Therefore, the relative frequency of the number of close color pairs in an image is represented by

$$R = \frac{P}{\binom{U}{2}} \qquad (5)$$

Next a small secret message is embedded into an image under test by substituting the LSB of random pixels with bits from the secret message. Once the test message has been

embedded into the image, the relative frequency for close color pairs is calculated a second time from the image as R'. The detection hypothesis states that if the image already had a message hidden in it, the ratios R and R' will be extremely close; however, if no message is present in the image the ratio R' will be greater than R. Therefore, the statistic R/R' can potentially be used as a discriminating feature for colored images. A limitation to the attack lies with the number of unique colors present in an image. The statistic becomes unreliable when more than 30% of the pixels in the image are unique colors [FrG01]. Consequently, high-resolution scans and uncompressed digital images are less likely to be detected by this attack because of the large number of unique colors present in such images.

2.6.4 Histogram Characteristic Function

Jeremiah Harmsen [HaP03] formulated an attack based on the premise that spatial hiding methods have a similar effect on the histogram of an image to applying a low-pass filter to the histogram. Further, the attack can be applied to both grayscale images and RGB color images. For color images, a three-dimensional histogram is first computed to determine the total number of occurrences of each (r,g,b) color in the image. Next, the histogram is transformed into the frequency domain by taking the 3-dimensional discrete Fourier transform (DFT) in order to obtain a histogram characteristic function (HCF). A center of mass (COM) of the HCF is computed to measure the distribution of the histogram in the frequency domain. For color images, the HCF center of mass is a vector of 3 elements denoting the metric for each of the three color dimensions (r,g,b). The equation used by Harmsen for computing the HCF COM of an image is explained as follows. Let $h[n]$ denote the histogram of an image, and the histogram transformed into

the frequency domain is denoted as $H[k] = DFT(h[n])$. Then the HCF COM, $C(H[k])$, is computed as follows:

$$C(H[i]) = \frac{\sum_{i \in K} i \times |H[i]|}{\sum_{j \in K} |H[j]|} \qquad (6)$$

with $K = \{0,1,\ldots,\frac{N}{2} - 1\}$, and N is the length of the DFT. It is shown that after an image is embed with a message in the spatial domain, the HCF COM for the resulting stego-image decreases or remains equal to the clean image. Similarly, a HCF COM can be computed for grayscale images using only a one dimensional histogram. Using this statistic as a discriminating feature, Harmsen showed that stego-images embed via simple LSB substitution could be detected with perfect accuracy for message payloads of 100%.

2.6.5 The Neighborhood Attack

Andres Westfeld [Wes02] devised an attack to counter the +/- 1 embedding technique of the Hide v2.1 steganographic software created by Toby Sharp. Similar to the raw quick pairs method, this attack involves detecting an increase in the number of close colors present in a stego-image. Any given (r, g, b) color can have up to 26 neighbor colors which only differ by 1 in any of its three color components (Figure 5).

Therefore, the attack involves computing the number of neighbors present in an image for each unique, non-saturated color from the image. A histogram is then created to chart the count of colors in the image which contain *i* neighbors, $i = 0,1,\ldots,26$. Studies revealed that colors in typical clean images have no more than 9 neighbors, whereas the addition of even a small message using Hide v2.1 creates colors in the image with 10 or

more neighbors. However, it was later discovered that the attack only works for decompressed JPEG images. High-resolution scans, and images not subject to compression contain a much greater amount of unique colors present in an image; therefore, the neighborhood attack does not work for such images.

$(r-1,\ g-1,\ b-1)$	$(r,\ g-1,\ b-1)$	$(r+1,\ g-1,\ b-1)$
$(r-1,\ g-1,\ b)$	$(r,\ g-1,\ b)$	$(r+1,\ g-1,\ b)$
$(r-1,\ g-1,\ b+1)$	$(r,\ g-1,\ b+1)$	$(r+1,\ g-1,\ b+1)$
$(r-1,\ g,\ b-1)$	$(r,\ g,\ b-1)$	$(r+1,\ g,\ b-1)$
$(r-1,\ g,\ b)$		$(r+1,\ g,\ b)$
$(r-1,\ g,\ b+1)$	$(r,\ g,\ b+1)$	$(r+1,\ g,\ b+1)$
$(r-1,\ g+1,\ b-1)$	$(r,\ g+1,\ b-1)$	$(r+1,\ g+1,\ b-1)$
$(r-1,\ g+1,\ b)$	$(r,\ g+1,\ b)$	$(r+1,\ g+1,\ b)$
$(r-1,\ g+1,\ b+1)$	$(r,\ g+1,\ b+1)$	$(r+1,\ g+1,\ b+1)$

Figure 5. The 26 Neighbors of an RGB Pixel [Wes02]

2.6.6 Universal Blind Steganalysis

The steganalytic methods described above are considered model-based algorithms because they target specific steganographic techniques. For example, RS Steganalysis and the Chi-Squared attack are both targeted for embedding techniques which hide data in the spatial domain. In that sense, these analytical techniques are considered non-blind – the embedding and extracting algorithms are known to the warden. On the other hand, universal blind steganalysis attempts to detect numerous different steganographic techniques including novel ones without knowledge of the embedding and extracting algorithms being used. In universal blind steganalysis, a set of features (image statistics) is chosen which accurately discriminates between clean images and stego-images. Features are then extracted from a set of both clean and stego-images, and the data is then trained and classified using such tools as neural networks [HoS04], a fisher linear

discriminant [HoS04, Far01], support vector machines [HoS04,LiF02], or hyper-geometric classifiers [McB03]. Therefore, much of the focus in this field is aimed at finding a discriminating set of features which accurately classify images as being clean or containing hidden data no matter how the data was hidden.

2.7 Summary

This chapter gives a thorough review of the state-of-the-art in steganographic hiding methods as well as detection mechanisms. Further, much of the focus of the survey of techniques is given to those involving the spatial image domain. The following chapter further explains the perturbed quantization steganographic system as well as how it is applied and tested in the spatial image domain.

III. Methodology

3.1 Chapter Overview

This chapter first revists the perturbed quantization system in greater detail and then redefines the system in the context of the spatial image domain. Section 3.3 describes two information-reducing processes to be used with the system. The goals of the study are outlined in Section 3.4, and a discussion on digital image formats is presented in Section 3.5. A small study investigates the steganographic capacities of the various lossy image transformations in Section 3.6, and another pilot study in Section 3.7 introduces a new steganalytic attack. The remainder of the chapter is devoted to describing a test plan which accomplishes the goals of this research.

3.2 The Perturbed Quantization Steganographic System

Perturbed quantization steganography applies to many different types of digital signals; however, this research focuses on digital images and thus the system is described using image terminology. A more thorough description of perturbed quantization steganography and the mathematical theory involved can be found in previously published works [FrG04, FrG05]. The system terminology and variable names used herein to explain the hiding technique, especially within the mathematical equations, are the same terms used in previous studies.

3.2.1 Basic Terminology

The principle concept behind the perturbed quantization steganographic system is that two parties are able to communicate hidden data embedded within a digital image.

In this scenario, the sender uses additional side information not known to the receiver or warden in order to hide the data. More specifically, the sender applies lossy processing to a digital image, and in the course of this image processing a secret message is hidden within the image. Thus, the technique works in conjunction with a lossy image processing operation as depicted in Figure 7, whereas most steganographic techniques simply embed data after such an operation as shown in Figure 6.

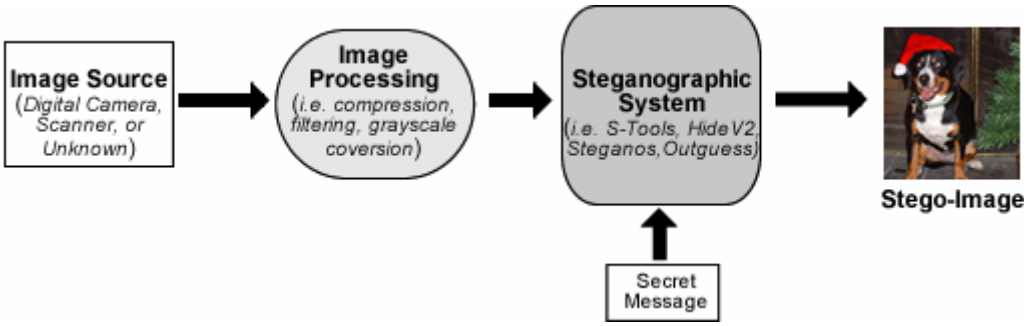

Figure 6. Diagram of a Typical Steganographic System

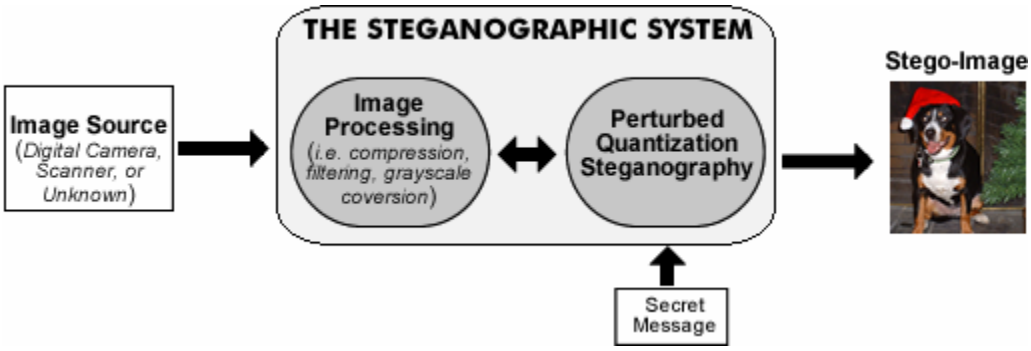

Figure 7. Diagram of the Perturbed Quantization Steganographic System

The sender starts with a raw digital image taken directly from its original source. Next, the sender applies a transformation to the image which results in some information loss. Typically, lossy transformations in the spatial domain alter pixel values, and/or reduce the number of pixels in an image. In many cases, the rounding of pixel values to the nearest integer is required after an image has been transformed, but before it has been encoded into an image format. It is the process of rounding pixel values that is "perturbed" by the sender in order to embed a secret message as shown in Figure 8.

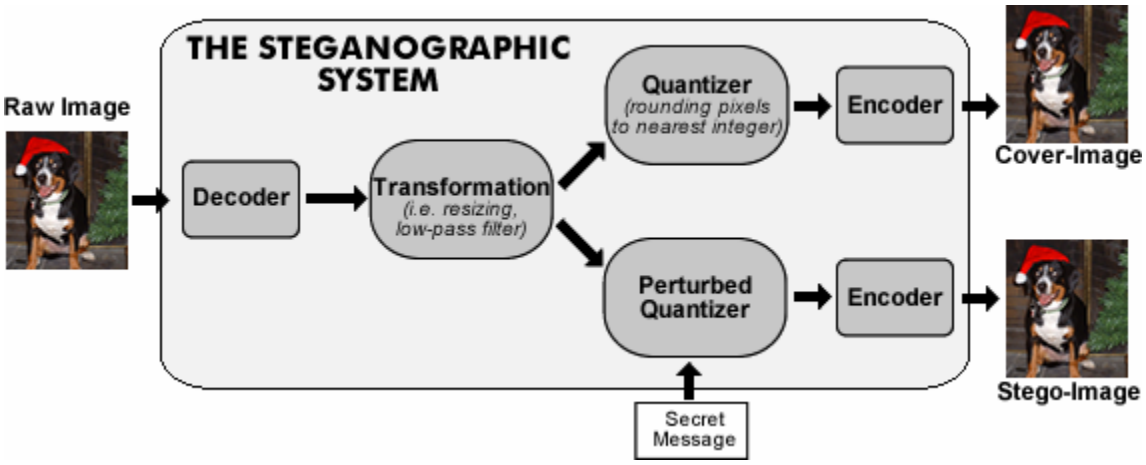

Figure 8. A look Under the Hood of the Perturbed Quantization System

Assume that once the sender has applied a lossy transformation to their digital image, the image contains n pixels. Let P_i represent pixel values prior to rounding, where $i \in \{1,\ldots,n\}$. The sender then utilizes a selection rule to choose only those pixels P_i, that can be rounded either up or down while minimizing any additional rounding error. The basic selection rule suggested by Fridrich chooses a pixel

$P_i \leftrightarrow P_i - \lfloor P_i \rfloor \in (0.5 \pm \varepsilon), where\ \varepsilon \leq 0.1$ [FrG04]. The index i for each pixel meeting the selection rule is stored in the set $C = \{i_1, \ldots i_k\}$ with a total of k changeable pixels. Therefore, the sender can round k pixels P_i where $i \in C$ in either direction in order to encode a secret message, and the pixels P_i where $i \notin C$ are rounded to their nearest integer. The maximum steganographic capacity for a given image using this method is given by $\frac{k}{n}$. The cover image can be defined as the image which results from rounding all pixels P_i, $i = 1, \ldots, n$ using the normal rounding function. Thus, for purposes of steganalysis, the system has two outputs: a cover image which was rounded normally, and the stego-image which uses a perturbed quantizer. This is conceptually shown in Figure 8.

Fridrich also showed that the amount of additional rounding error which occurs during perturbed quantization steganography is ε^2. For LSB substitution systems which choose pixels values in sequence or at random, the epsilon threshold ε is 0.5 because the selection interval covers the entire interval (0,1), and because it doesn't matter what value a pixel had before any rounding that took place. Therefore, the amount of additional rounding error introduced in such a system is $0.5^2 = 0.25$. With perturbed quantization steganography, setting the threshold $\varepsilon \leq 0.1$ ensures that the maximum amount of additional rounding error introduced to an image is $\leq 0.1^2 = 1/100$.

3.2.2 Embedding a Secret Message

After the set of changeable pixels *C* is defined by the sender's chosen lossy transformation, the sender is now prepared to encode a secret message *m*. However, first

the sender rounds all n pixels in the image in order to obtain a cover image Y. In doing so, another set $A = \{t_1, \ldots, t_k\}$ *where* $t_i \in \{0,1\}$ is defined to keep track of whether a changeable pixel P_i was rounded up or down in order to obtain Y. This set ensures that the sender knows whether to increment or decrement a changeable pixel upon encoding the stego-image. Rather than substituting message bits into the least significant bit of a changeable pixel, the system encodes message bits by changing the parity of a pixel so that the set C does not have to be known by the receiver. Let $b_j = Parity(P_j)$ represent the parity bit from pixel P_j, where the parity function is defined as $Parity(P_j) = LSB(P_j)$. Thus, to encode a message bit into one of the changeable pixels, if the message bit does not already match b_j, then the sender flips the parity bit to match the message bit by incrementing or decrementing the pixel P_i according to set A. Therefore, on average only 50% of the pixel values which are encoded with a secret bit need to be altered to match the message bit.

If the sender wants to hide a secret message of q bits where the message $m = \{m_1, \ldots, m_q\}$, they first compute the binary parity vector b, where $b_i = Parity(Y_i)$ for each $i = 1, \ldots, n$. Thus, b is a binary vector of dimension $n \times 1$. In addition, it is assumed that the sender and receiver have agreed on a secret stego-key. This key is used by the sender in order to seed a pseudo random binary sequence generator (PRBSG) which generates a matrix D, containing q rows and n columns ($q \times n$). It is then the senders job to calculate a modified parity vector b' such that

$$Db' = m \tag{7}$$

Therefore, the sender has to solve a system of equations using Gaussian elimination in a Galois field of 2. Once the modified parity vector b' is solved such that the indices for modified parities exists in the set *C*, the sender encodes message bits by either leaving the pixel unchanged (message bit matches the parity bit), incrementing or decrementing the pixel based on the corresponding set *A*.

3.2.3 Extracting a Secret Message

Because of the way in which the secret message is encoded in Equation 7, the job of the decoder is extremely easy. Recall that it is assumed the sender and receiver have agreed on a secret-stego key. The receiver uses this shared secret key to seed the same pseudo random binary sequence generator (PRBSG) as used by the sender in order to construct the matrix *D* ($q \times n$ elements). Additionally, the receiver constructs the parity vector b' by simply taking the LSB of each pixel in the received image. Finally, the receiver multiplies the matrix *D* by the vector b' in order to obtain the message *m*.

3.2.4 The Implementation

Digital images contain thousands upon thousands of pixel values. For example, a 512x512 color image contains 262,144 pixels in 3-dimensions. As a result, an implementation of the system described above would require huge computational power in order to solve a system of q equations (q = message length in bits) and 786,432 unknowns. Therefore, Fridrich implemented the perturbed quantization system by performing structured Gaussian elimination in which the cover image *Y* was broken into β blocks. Then Equation 1 was solved for each block β_i, where *i* is the total number of blocks. However, this requires that the receiver must know *q*, the length of the secret

message, or must be able to compute q as well as the size of the blocks β_i. Thus, Fridrich's implementation involved embedding a header stream within the secret message for the receiver to know the length of the secret message q_i in each block β_i. It is then assumed that the receiver knows the length of the header stream h in order to decode the message length q_i within each block β_i. Fridrich's previously published papers contain a complete outline of the implementation used in this research. The perturbed quantization system in this investigation was implemented in an identical manner as to the previous work done by Fridrich with the following modifications:

First, in Fridrich's implementation, the headers h_i identifying the length of q_i within each block β_i were embedded and concatenated together in the final block β of the image. In contrast, the implementation used in this thesis embeds a header h_i at the beginning of each block β_i to denote the length of q_i. This is merely a design decision which makes the decoding process easier for the receiver.

Secondly, when the secret message length q is less than the steganographic capacity $\frac{k}{n}$, there are then $k-q$ changeable pixels which are not used for message encoding. In this case, Fridrich's theoretical design utilizes the first q_i available pixels within each block β_i. The implementation in this thesis seeks to select changeable pixels uniformly distributed throughout the image. This is accomplished by adding an algorithm which selects changeable pixels within a block β_i by ensuring that the pixels selected to carry secret message bits are spread uniformly throughout each block and in turn throughout the image. This is important because it spreads out any artifacts introduced by

the secret message throughout the image so that subsets of the image do not contain distinguishable differences which could be exploited by steganalysis. An even more advanced selection rule could be explored in future work which further selects only those changeable pixels located in noisy areas of an image.

Finally, the perturbed quantization algorithm used for this study is implemented using Matlab 7 Release 14 [Mat04]. In a real world situation, an optimized version of the algorithm would be important for both the sender and receiver; however, it is not a focus of this investigation.

3.3 Lossy Image Transformations

Whereas previous work focused on applying perturbed quantization steganography to double JPEG compression [FrG04, FrG05], this research looks to investigate using perturbed quantization steganography in conjunction with the following lossy image transformations: color to grayscale conversion, and image downsampling.

3.3.1 Color to Grayscale Conversion

Converting an image from color to black and white is extremely common amongst graphic designers, photographers, and steganographers alike. Image processing applications such as Adobe Photoshop, Macromedia Fireworks, ImageMagick, and the GIMP toolkit offer users several different variations of color to grayscale functions.

A 24-bit RGB color image actually contains three separate 8-bit channels corresponding to a red channel, green channel, and a blue channel. The color of a pixel is determined from the corresponding values from each of the three color channels. This results in $256^3 = 16{,}777{,}216$ possible colors for each RGB triplet. On the other hand, a grayscale

image contains just one single channel. Hence, a color to grayscale conversion simply involves reducing an image from three channels to one. Let *X* be an RGB color image, *X* can be represented as a three-dimensional matrix of pixel values (integers) sized $l \times w$, where *l* and *w* are the dimensions of the image. A color to grayscale operation converts the $l \times w \times 3$ matrix into a one-dimensional $l \times w$ matrix of real numbers. Finally, the values are rounded to the nearest integer in order to produce a grayscale image *Z* from image *X*.

There are actually several different grayscale conversion functions which map an 8-bit single channel grayscale image from a 24-bit RGB color image. The most popular grayscale conversion functions for image processing are the standard weighted sum, and the desaturate function [Bun00].

3.3.1.1 Standard Color to Grayscale Weighted Sum

Perhaps the most straightforward grayscale operation uses a weighted sum function. In this function, a 24-bit RGB color image is transformed into an 8-bit grayscale image by multiplying a weight to each of the three color components: red, green, and blue. The specific weights are defined in the following color to grayscale function:

$$GrayPixel = .299R + .587G + .114B \tag{8}$$

where R,G,B represent the value for the Red, Green, and Blue channels from each pixel. The weights were constructed to create a grayscale image that is perceptually identical to its originating color image in brightness and luminance. In addition, it is known that the human visual system is most sensitive to green. This formula is recognized by most as the standard color to grayscale operation. The exact weights and formula are also used in GIMP [Bun00], Matlab [Mat04], and ImageMagick [Ima04].

3.3.1.2 The Desaturate Function

To most people, desaturating an image is a foreign concept. However, to most photographers desaturating an image is the method of choice for converting a color digital image into a grayscale image [Bai04]. Whereas the previous method produces a weighted sum of the RGB components, the desaturate function attempts to find an "average" value from the three RGB channels. However, rather than simply taking a weighted sum where each component is multiplied by 1/3, desaturating an image involves taking the average of the maximum and minimum components of an RGB pixel. More specifically,

$$GrayPixel = \left\lfloor \frac{Max(R,G,B) + Min(R,G,B)}{2} \right\rfloor \tag{9}$$

where (R,G,B) represents the red pixel value, green pixel value, and blue pixel value of a given pixel from the color image [Bun00].

By examining color images converted to grayscale using the two conversion functions, it can be seen what kind of effect each operation has. Notice in Figure 9 that the standard grayscale conversion of the color wheel closely resembles the brightness and luminosity from the original color image. In comparison, the desaturate function has a much different effect. The brightness and luminosity values are consistent throughout the wheel regardless of the original color.

However, looking at Figure 10, one can see that with normal digital photographs there is not a huge difference between the conversion functions other than the fact the desaturated image is a shade darker. The desaturate function has become an extremely popular grayscale function throughout the digital imaging community, especially for

those printing black and white images [Bai04]. It is currently implemented in Adobe Photoshop, GIMP, and is the default grayscale conversion function in Macromedia Fireworks; therefore, a desaturated image by itself would not raise any suspicion from a steganographic warden.

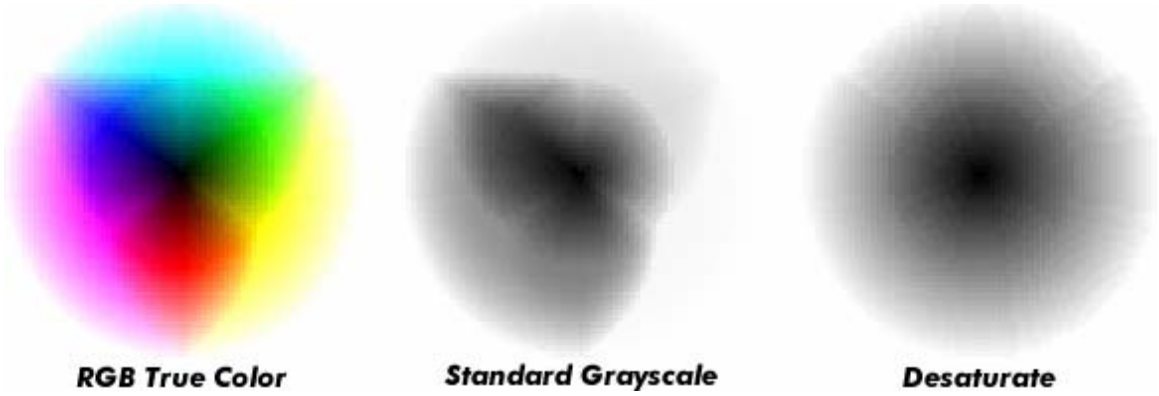

Figure 9. A Comparison of the Grayscale Functions on a Color Wheel

Figure 10. A Comparison of the Grayscale Functions on a True Color Image

3.3.2 Image Downsampling

The process of changing the number of pixels in an image is called resampling,

and downsampling refers to the process of reducing the number of pixels in an image. Downsampling an image is actually a very complicated process with several parameters which affect the outcome of the smaller image. A typical sequence of operations which occur while downsampling an image is:

1: The reduced image's pixels are interpolated from the original image.

2: The reduced image's pixel values are rounded to the nearest integer.

In some cases, an anti-aliasing filter may be applied before interpolation occurs. This anti-aliasing filter reduces the amount of aliasing which might occur from pixel interpolation. However, images which have been properly sampled such as images taken directly from a digital camera do not require the anti-aliasing filter. This filter can be overlooked as a parameter which affects the downsampled image. Some image processing applications apply this filter before interpolating the pixels. For example, Matlab uses a Hamming filter in its resize function as an anti-aliasing filter. Thus, the anti-aliasing filter is recognized as an important parameter in image downsampling, but it is not a focus of this study. This study looks to examine the interpolation methods, and as a result the default filter in Matlab is used for downsampling throughout this study.

The second parameter which affects the outcome of a downsampled image is the interpolation method. When an image is reduced in size, an entirely new image is created with the smaller dimensions. The pixel values in the smaller image are interpolated from the original image, and it is the method of interpolation which is of interest to this study. Most image processing applications such as Photoshop, Matlab, and Fireworks implement three primary interpolation methods: nearest neighbor, bilinear, and bicubic

interpolation. Many other methods do exist for downsampling an image [Hof02], but they are not explored in this investigation.

3.3.2.1 Nearest Neighbor Interpolation

Perhaps the most basic form of interpolation is the nearest neighbor approach. In this approach, the pixel values in the smaller image are taken directly from the nearest neighbor in the original image. More specifically, this method applies a direct mapping of the pixel coordinate in the smaller image to the closest pixel coordinate in the original. As a result, no mathematical operations are applied to the pixels, which results in no new colors introduced in the image. Additionally, the nearest neighbor approach is the only interpolation method in which an anti-aliasing filter is not applied beforehand. This method does not make sense to use with perturbed quantization steganography as no pixel values are ever rounded, thus the steganographic capacity for an image downsampled and interpolated using the nearest neighbor approach is always zero.

3.3.2.2 Bilinear Interpolation

In contrast, the bicubic and bilinear methods are much more applicable to the steganographic system under study. Bilinear interpolation maps target pixels from the original image's nearest four neighbor pixels. The average value from these four closest neighbors is calculated and weighted according to their relative distances. As a result, each pixel in the smaller image is actually a weighted average of four pixels from the original image. This has the effect of both introducing new colors into the smaller image, as well as requiring a quantizer to round the pixel values prior to the image's encoding into its appropriate format.

3.3.2.3 Bicubic Interpolation

The bicubic method is identical to the bilinear interpolation method except for the fact that rather than average the four closest pixel values in the original, the bicubic method takes an average value from the nearest sixteen pixels in the original image. Again, this weighted average introduces new colors into the image and requires a rounding function. Therefore, both the bilinear and bicubic interpolation methods make good candidates for lossy image transformations which can be applied to perturbed quantization steganography.

Thus far it has been suggested that the anti-aliasing filter as well as the interpolation method used can affect the outcome of downsampling an image. The final parameter which can affect the pixel values of a smaller image is the factor of which the image is reduced. However, in the context of the steganographic system under study, it is assumed that the only way in which this parameter will have any affect is on the steganographic capacity. This factor will be studied within the steganographic capacity pilot study, but for the actual implementation of the system the size factor for downsampling an image remains a constant.

3.4 Goals and Expectations

While the primary objective of this research is to apply the proposed steganographic system into the spatial image domain by hiding data in the least significant bits of pixels, there exist more specific tasks. Namely, the goals of this research are as follows:

1) To determine which lossy image transformation can accommodate the largest amount of hidden data when used with the perturbed quantization algorithm.

2) To revise the neighborhood attack originally introduced by Andreas Westfeld [Wes02] in search of a better discriminating feature for +/- 1 embedding.

3) To measure the performance of the perturbed quantization steganographic system in the spatial domain as the payload (secret message length) increases.

4) To vary the threshold ε within the selection rule in order to investigate the relationship between smaller epsilon values and the security of the system.

5) Finally, to compare the performance of the system against other publicly known steganographic systems which hide in the spatial domain.

It is expected that decreasing the epsilon value ε in the interval $(0.5 \pm \varepsilon)$ will reduce the secret message capacity of a given image, but also improve security of the system by making stego-images less distinguishable from clean images. This is because the closer an un-rounded pixel value is to 0.5, less noise will be added to an image. For example, when $\varepsilon = 0.1$ the selection rule calls for all pixels whose fractional part is between (0.4, 0.6) to be considered changeable pixels – pixel can be rounded either way. In such a system, the largest rounding error that would occur would be when a pixel whose fractional part is 0.4 gets rounded up, and when a pixel with fractional part of 0.6

gets rounded down. However, when $\varepsilon = 0.05$, only those pixels in the interval (0.45, 0.55) will be selected to carry secret message bits. In such a scenario, the maximum error introduced during rounding is also smaller as the worst case scenario calls for a pixel with fraction of 0.45 to get rounded up or a pixel with fraction 0.55 to get rounded down. Clearly, the smaller the epsilon value, the smaller the rounding error, and in turn less noise added to an image. Therefore, it is hypothesized that lowering the epsilon value in the selection rule increases security of the steganographic system. However, a tradeoff exists in that lowering the epsilon value also reduces the possible set of changeable pixels.

It is also expected that perturbed quantization steganography in the spatial domain will outperform other spatial steganographic techniques. First, research done by Fridrich showed that in the frequency domain, perturbed quantization steganography outperformed every comparable method, and that should hold true for its application into spatial image formats. Secondly, most other spatial hiding techniques involve selecting pixels at random or pixels in noisy regions to hide data. The pixel selection rule for perturbed quantization steganography is more sophisticated than that of other hiding techniques.

3.5 Steganography and Digital Images

Steganography and steganalysis of digital images is an increasingly popular research area over the past few years. There exist numerous published works on various topics and techniques of steganography and steganalysis. However, an area of inconsistency amongst researchers in the field lies with the choice of images on which to

perform experiments. Many researchers utilize digital images taken by their own cameras and equipment [Ker04], some choose to download images from large online databases [FrG04, LiF02] such as Philip Greenspun's server (philip.greenspun.com), and others download images from random online locations such as EBay listings [PrH01]. Ideally, all researchers would utilize the same set of images in order to maintain a baseline and consistency across studies. Nevertheless, researchers are forced to select an image workload to use for their own studies.

3.5.1 Choosing an Image Format

The PNG format, is a good choice for lossless images for many reasons. First, whereas the GIF format can only store 256 unique colors in any given image, the PNG format can support a full 16,777,216 colors in one image. Secondly, the PNG format performs a small amount of compression by looking for patterns in the image data. Any compression that occurs during encoding into the PNG format is fully reversible thus maintaining its lossless status. As a result, the PNG format creates smaller file sizes than that produced by the BMP format. Finally, PNG is the best choice for lossless images posted on the web. Modern web browsers support the PNG format, and for some web browsers, the PNG format is the only available choice for lossless images.

In summation, the PNG format offers a full spectrum of colors similar to Bitmaps (BMP) making them excellent choices for digital photographs. The PNG format performs some compression of redundant data in order to reduce file sizes much less than the Bitmap (BMP) format, and the PNG format is becoming increasingly popular on the web as most modern web browsers support the format. Still, the JPEG image format is the most popular image format due to its balance of quality and file size compression.

Nevertheless, for steganographic systems requiring the use of a lossless image format, the PNG (Portable Network Graphics) image format is the best choice. Accordingly, the PNG format is used exclusively throughout this investigation.

3.5.2 The Workload

In order to avoid using one image database which could be biased in some manner, the experiments within this investigation use two separate image databases of different origin. These image databases are characterized as follows:

3.5.2.1 Image Set A

Image Set A consists of 50 JPEG images sized 2048×1360 pixels, and are all 24-bit RGB color images. The images are courtesy of philip.greenspun.com, a large server hosting over 10,000 digital photos taken by Philip Greenspun. The source of the images is consistent with previous work [FrG04, LiF02]; however, it is not known what specific images were used in the previous studies. The images are all original files taken directly from a digital camera; therefore, the images have not been exposed to any information loss other than any compression built-in to the digital camera. The images were selected in order to create an un-biased set. Many of the photographs are taken outdoors in daylight; others are taken indoors or at nighttime. A wide variety of images contain people and animals, while other photographs are scenic displaying the natural world. Some images contain drab, dreary skies, yet others contain vivid and beautiful colors. The only limitation with Image Set A is that the images are all assumed to be unedited; nonetheless, the website does explicitly state that the images are clean.

3.5.2.2 Image Set B

Image Set B contains 1,000 JPEG images all of which are sized 512×512 pixels, and are 24-bit RGB color images. This database was used in previous steganalysis research at the Air Force Institute of Technology [McB03, Jac03]. The images originated from an Air Force website, and contain a variety of scenes. However, the majority of the images are shots of Air Force planes, personnel, labs, and other equipment. It seems as though the set of images might be slightly biased in that there is little variation in colors between the images. Another limitation of Image Set B is that the images have all been JPEG compressed, cropped, and/or downsampled to their current size. Because of the unknown editing that has taken place, it can be said that the images are not "natural" as are the images from Image Set A. Nonetheless, Image Set B offers images of a different size and origin.

3.6 Pilot Study #1: Determining the Steganographic Capacity

Recall that a secret message *payload* characterizes the length of a secret message in terms of the percentage of elements of the stego-image which contain data from the secret message. For example, a 50% *payload* means that exactly one half of all elements of the stego-image (DCT coefficients in the frequency domain, pixel values in the spatial domain) contain one bit of the hidden message. Whereas an 8-bit grayscale image contains only one value for each pixel, a 50% payload message simply means that half of the pixel values contain hidden data. However, a 24-bit RGB color image contains three separate values for each pixel (red, green, and blue values). Thus, in a color image a 50% payload means that one half of all red pixel values contain hidden data, one half of all

green pixels values contain hidden data, and one half of all blue pixel values contain hidden data.

On the contrary, a steganographic *capacity* of a cover image refers to the percentage of elements in the image that can be used to hide data. In the spatial domain many software tools such as S-Tools, Steganos, and Hide4PGP allow a full 100% *capacity* for hiding data. This means that there is no selection rule as to which pixels can be chosen for hiding data in their least significant bit. Adaptive steganographic systems such as the perturbed quantization method, select only those pixel values whose fractional part falls in a predefined interval. Therefore, it is assumed that a full 100% capacity is unachievable.

As goal four above indicates, this research effort not only looks to compare perturbed quantization steganography with other spatial hiding methods, but to analyze the security of the system as the secret message payload increases. Accordingly, the secret message *payload* is one factor in the experiments. However, in order to determine the possible secret message payloads embedded with perturbed quantization steganography it is necessary to first investigate the steganographic capacity for each of the information-reducing operations under study. Once the capacities for each lossy transformation are determined, the payload values can be set appropriately.

This pilot study utilizes both image database A and image database B in order to validate the results. The four lossy image processing operations:

- *Weighted color to grayscale conversion*
- *The desaturate function*
- *Downsampling using bilinear interpolation*

- *Downsampling using bicubic interpolation*

are applied to each image. Prior to rounding the pixel values, the capacity of each image is calculated by dividing the number of pixels whose fractional part falls within the range $0.5 \pm \varepsilon$ by the total number of pixels in the image.

$$capacity = \frac{\#of\ unrounded\ \ pixels \in (0.5-\varepsilon, 0.5+\varepsilon)}{Total\ \#of\ \ pixels\ in\ the\ image} \qquad (10)$$

The factor ε is tested using values 0.1, and 0.05 as these were the values previously used by Fridrich [FrG04]. Making the epsilon value larger defeats the purpose of perturbed quantization steganography. Since the downsampling operations are using colored images, the capacities for all three pixel domains (Red, Green, and Blue) are calculated separately and averaged. A probability density estimate is then generated for each operation, and for each image database. The densities plot the distribution of capacity sizes expressed in percentage values (0.0, 1.0). In addition the mean, median, standard deviation, maximum, and minimum capacities are calculated for each operation using each image database. Furthermore, this study downsamples images to several different sizes in order to study the effect that the downsampling scalar has on the steganographic capacity of a given image. Thus, both of the downsampling interpolation techniques are used to reduce images to 25%, 50% and 75% of the original image resolution size.

3.6.1 Weighted Color to Grayscale Conversion

Recall that the standard color to grayscale conversion takes the three RGB pixel values and uses the formula 0.299R + 0.587G + 0.114B to create one grayscale pixel. First using Image Set A, each image is converted to an un-quantized grayscale image.

The capacity for each image is then calculated by selecting only those pixels whose fractional part lies between $0.5 \pm \varepsilon$, where $\varepsilon = 0.1$.

Figure 11 shows a probability density graph for the distribution of images in Image Set A that contains the capacity specified on the x-axis. The immediate results are quite surprising. Nearly every image appears to have less than 5% of its pixels fall in the interval 0.5 ± 0.1. Using such a simple formula one would expect the fractional part of pixel values to be uniformly distributed between 0 and 1. In such a scenario, 20% of the pixel values would fall in the interval 0.5 ± 0.1, and 10% of the pixel values would fall in the interval 0.5 ± 0.05. However, it is clear that the images in Image Set A when used with the weighted grayscale conversion function, provide a smaller than expected steganographic capacity.

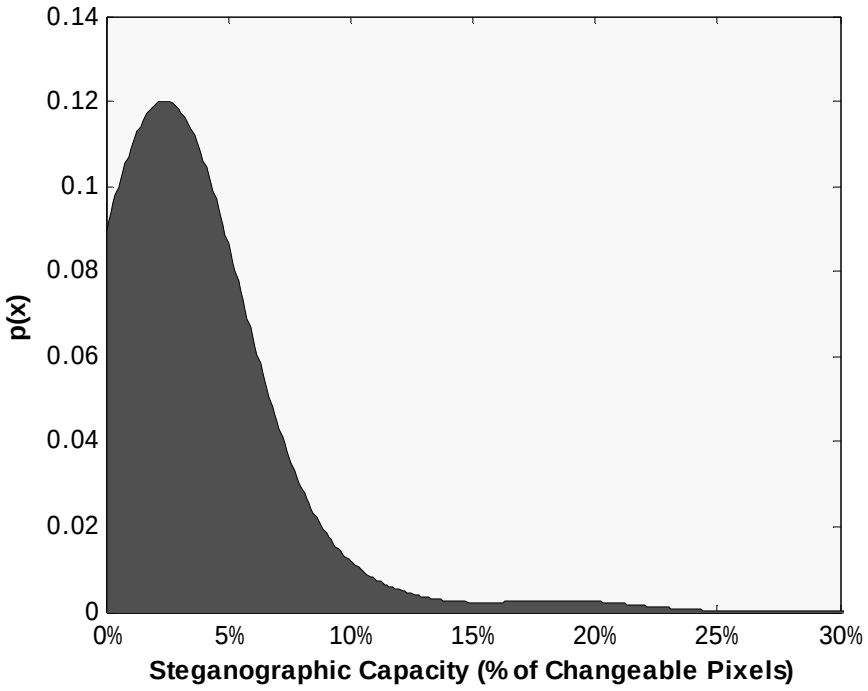

Figure 11. Capacity of Image Set A Using Weighted Grayscale Function

Next, the experiment is repeated using Image Set B in order to validate the results from Image Set A. The results from Image Set B are similar to those from Image Set A. Table 2 summarizes the capacity statistics from each image set. The results show that the average capacity for an image in Image Set A using $\varepsilon = 0.1$ is close to 3% whereas for Image Set B the average capacity for an image using $\varepsilon = 0.1$ is around 2.1%. These results are extremely unexpected. Moreover, because these capacity amounts are already extremely low when using $\varepsilon = 0.1$, it is known that using $\varepsilon = 0.05$ only produces capacity levels even smaller as the interval 0.5 ± 0.05 is included in the interval 0.5 ± 0.1. Therefore, no further capacity studies are needed using the color to grayscale formula.

Table 2. Steganographic Capacity using the Weighted Color to Grayscale Function

Image Set	Mean	Median	Standard Deviation	Maximum	Minimum
A	3.0333 %	2.3813 %	2.8047 %	18.7180 %	0.9978 %
B	2.1017 %	1.8519 %	1.6865 %	28.2520 %	0 %

3.6.2 The Desaturate Function

The desaturate function is of particular interest for its application to perturbed quantization steganography. In the desaturate function, pixel values in the resulting grayscale image are derived by taking the average of the maximum and minimum values from each corresponding RGB 3-tuple. As a result, dividing by two always gives a remainder of zero or a remainder of one. Therefore, every pixel in the un-rounded desaturated image will either have a fractional remainder of zero or exactly 0.5. Moreover, it does not matter what value of ε is used since the same set of pixels will be selected for 0.5 ± 0.1, and 0.5 ± 0.05. Each image from Image Set A is first desaturated

without rounding any values, and the steganographic capacity is calculated by selecting only those pixels whose remainder is 0.5 and dividing by the total number of pixels in the image.

The results from Image Set A are displayed in the density graph in Figure 12. As expected, most images have a capacity around 50%. Table 3 notes that the average capacity is around 47% and the maximum capacity of all the images from Set A is 55%. Next the experiment is repeated for Image Set B. The capacity is calculated for each image in Image Set B and the results are plotted in the density graph in Figure 13. Again, the results are pretty similar between the two image sets. The average capacity for images from Image Set B is 47% (see Table 3). However, there is a bit more variation amongst capacity values in Image Set B as noted by the standard deviation. This is likely due to the smaller image sizes of set B (512x512) as well as the fact that these images have been subject to lossy compression at some point in time. Another interesting finding is that the maximum capacity amongst all of the images in Set B is slightly above 80%, and a couple of images from the set have a steganographic capacity of 0%. Upon further review, most of the images with capacity outliers in Image Set B are pictures dominated by a solid sky color. In general, smooth regions of an image are not the greatest choice for the hiding of data as such areas are vulnerable to visual attacks [WeP99].

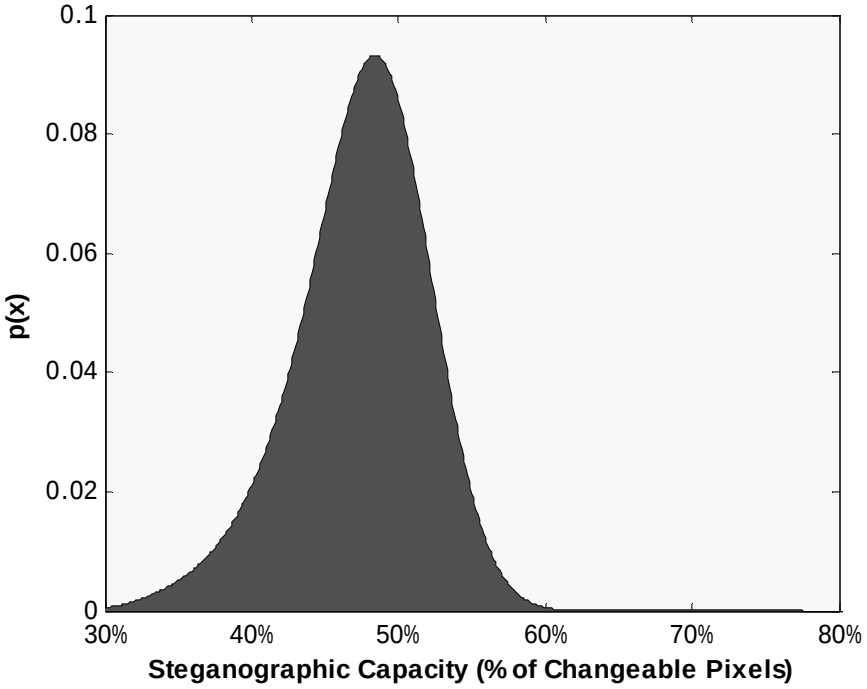

Figure 12. Capacity of Image Set A Using the Desaturate Function

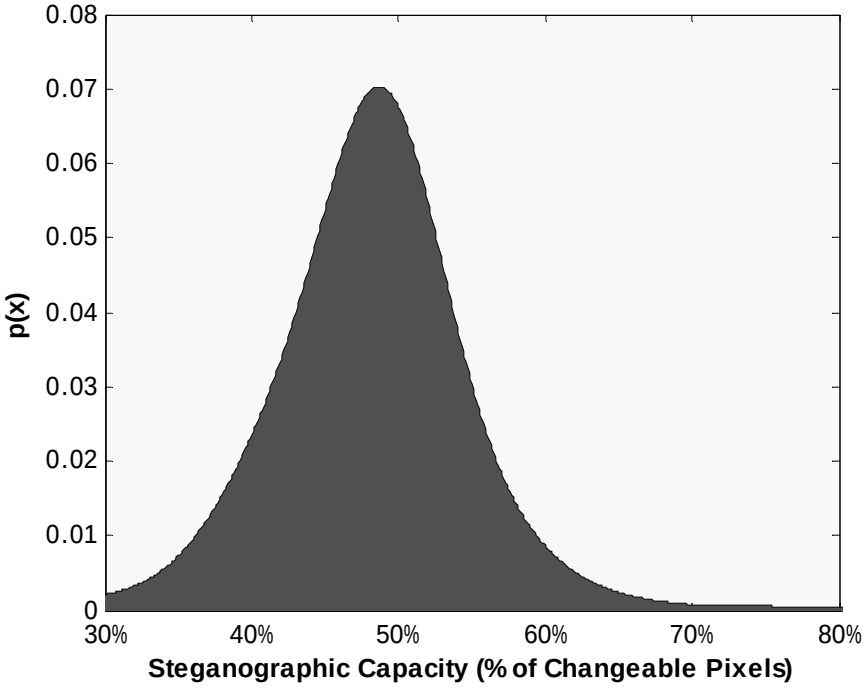

Figure 13. Capacity of Image Set B Using the Desaturate Function

Table 3. Steganographic Capacity Statistics Using the Desaturate function

Image Set	Mean	Median	Standard Deviation	Maximum	Minimum
A	47.3542 %	48.1000 %	3.5690%	53.1170 %	35.5760%
B	47.8655 %	48.2975 %	6.9590 %	80.1450 %	0 %

Overall, the desaturate function offers relatively large capacities when used as a lossy transformation for perturbed quantization steganography. The capacities are especially large when the desaturate function is compared to the capacities of the color to grayscale weighted conversion. Moreover, the pixels which are selected for data hiding with the desaturate function ensure that the absolute minimum amount of rounding error is added to the cover object ($\varepsilon = 0$). This is because when an un-rounded pixel value is chosen after an image has been desaturated, a rounding function could actually go either way since the fractional part is at exactly 0.5.

3.6.3 Downsampling using Bilinear Interpolation

The steganographic capacity is first calculated for each set of images using a downsampling factor of four, meaning that images are reduced to 0.25 their original size. The results from using bilinear interpolation during downsampling are shown in Figures 14 and 15, which depict the capacity for image sets B and A respectively. Additionally, Table 4 charts the statistics from both image sets. The results show that the capacity statistics are very similar for both image sets. The one difference being the slight variation in standard deviation and minimum capacities in image set B. Again, this is likely the result of smaller images having more variation.

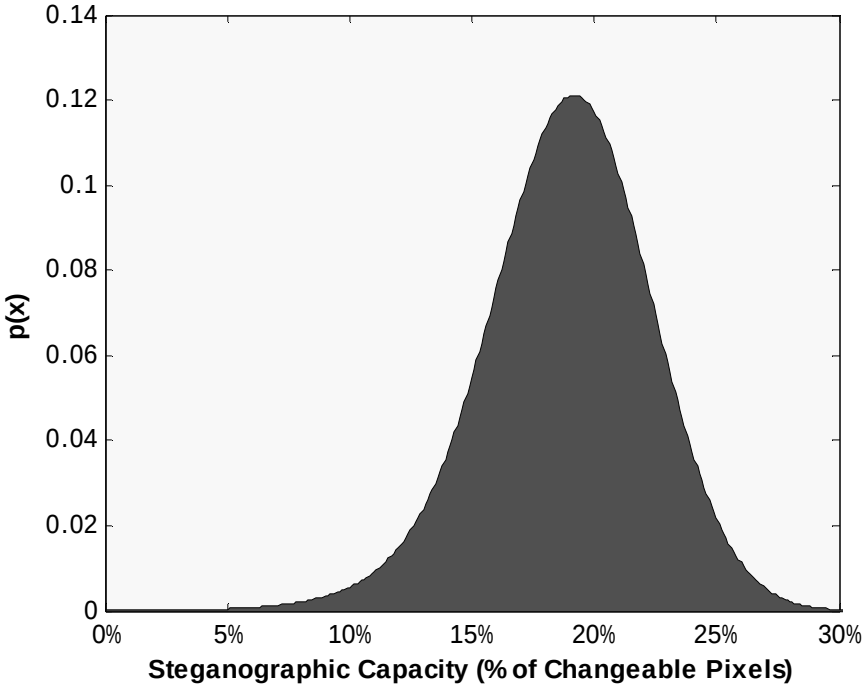

Figure 14. Capacity of Image Set B for Bilinear interpolation and a Scaling Factor of ¼

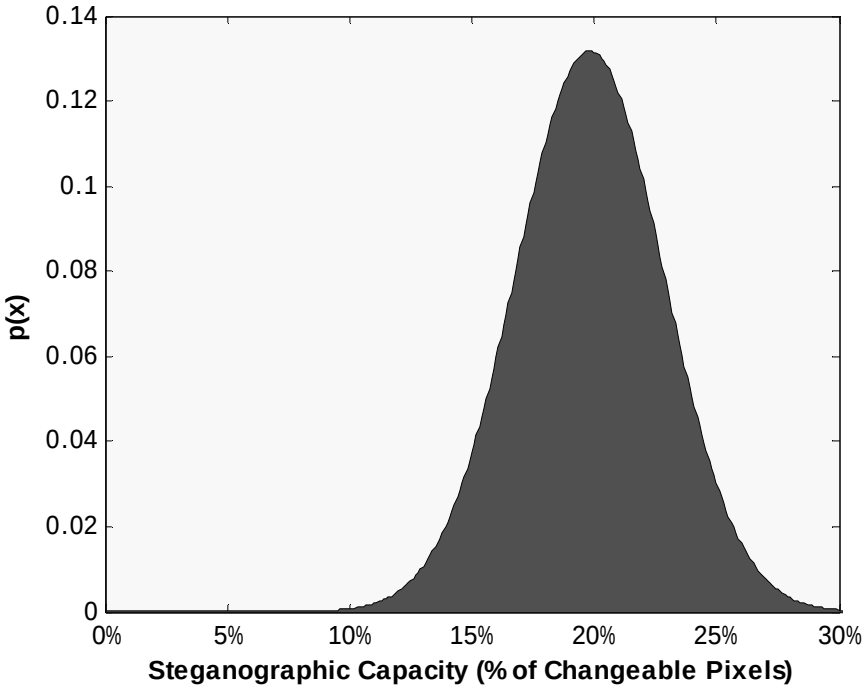

Figure 15. Capacity of Image Set A for Bilinear Interpolation and a Scaling Factor of ¼

Table 4. Steganographic Capacity Statistics Using Bilinear interpolation

Image Set	Mean	Median	Standard Deviation	Maximum	Minimum
A	19.8038 %	19.9295 %	0.4352%	20.1620 %	17.7900%
B	18.8645 %	19.4090%	1.6955 %	20.7520 %	6.9824 %

Next, the effect of the downsampling factor is studied by looking at how changing the downsampling factor affects the steganographic capacity of bilinear interpolation. Image Set A is used, and the capacities for each image are calculated after downsampling to 0.25, 0.5, and 0.75 the original size. The resulting statistics are charted in Table 5. The results show that regardless of the downsampling factor used, the steganographic capacity remains constant. This means that the downsampling size factor does not have any impact on the steganographic capacity of the system.

Table 5. Steganographic Capacity Statistics using Bilinear Interpolation

Factor	Mean	Median	Standard Deviation	Maximum	Minimum
0.25	19.8038 %	19.9295 %	0.4352%	20.1620 %	17.7900%
0.5	19.6879 %	19.8480%	0.4395 %	20.0660 %	17.9510%
0.75	19.4879 %	19.6470 %	0.5468%	20.0440 %	17.7890%

3.6.4 Downsampling using Bicubic Interpolation

The same experiment is repeated using bicubic interpolation. First the steganographic capacities are calculated from both image sets using a downsampling factor of 0.25. The capacity statistics for both image databases are charted in Table 6. The results show that the statistics are similar between the two image sets, with slight variation in minimums.

A second study examines the affect of the downsampling size factor used with bicubic interpolation by studying steganographic capacities from image set A using a size reduction of 0.25, 0.5, and 0.75. The resulting statistics from this investigation are shown

in Table 7. Similar to the bilinear interpolation, there seems to be no affect on capacity when the size factor is varied.

Table 6. Steganographic Capacity Statistics using Bicubic Interpolation

Image Set	Mean	Median	Standard Deviation	Maximum	Minimum
A	19.7813 %	19.8865 %	0.4072%	20.1150 %	17.9460%
B	18.8856 %	19.4305%	1.6875 %	20.935 %	7.2260 %

Table 7. Steganographic Capacity Statistics using Bicubic Interpolation

Factor	Mean	Median	Standard Deviation	Maximum	Minimum
¼	19.7813 %	19.8865 %	0.4072%	20.1150 %	17.9460%
½	19.8786 %	19.8110%	0.4596 %	20.0430 %	17.9200%
¾	19.3843 %	19.5850 %	0.6178%	20.0160 %	17.5100%

3.6.5 Downsampling Using a Lower Epsilon Threshold

Finally, the affect of the epsilon value on steganographic capacities is explored with bicubic and bilinear interpolation. The capacities for all images in set B are computed using an epsilon value of 0.05. Whereas previous capacities were all calculated using an epsilon value of 0.1, there were on average 20% changeable pixels in an image. Thus, for an epsilon value of 0.05, one would expect approximately 10% of the pixels to fall in the range 0.5 ± 0.05. The results from performing bilinear interpolation and bicubic interpolation using a size reduction factor of 0.25 are shown in Table 8. As expected, for both interpolation methods the average steganographic capacity is around 10%.

Table 8. Steganographic Capacity Statistics Using $\varepsilon = 0.05$

Interpolation	Mean	Median	Standard Deviation	Maximum	Minimum
Bilinear	9.4096 %	9.6558 %	0.8769%	10.6690 %	3.4912%
Bicubic	9.4224 %	9.6802%	0.8643 %	10.6320 %	3.5400%

3.6.6 Conclusions of Pilot Study #1

In summation of the steganographic capacity pilot study, the only surprising results occurred with the color to grayscale weighted function. The weighted grayscale function provided extremely small capacity sizes even with an epsilon value of $\varepsilon = 0.1$. As expected, the desaturate function provided capacities around 50%, and the interpolation methods from downsampling images provided capacities around 20% for $\varepsilon = 0.1$ and 10% with $\varepsilon = 0.05$. Additionally, within the downsampling study, the size reduction factor did not appear to influence a given images capacity. Now that the steganographic capacity for each of the image processing operations under study is known, the methodology for this research can be thoroughly explained.

3.7 Pilot Study #2: Revising the Neighborhood Attack

Andreas Westfeld's original attack on the Hide v2.1 steganographic technique involved calculating the number of neighbors each unique color has in a given image [Wes02]. This attack only takes into consideration the neighborhood of a color once. More useful information could possibly be found by calculating the number of neighbors present for every pixel in an image including duplicate colors. This results in a study of the frequency of occurrence of a neighborhood rather than the existence of a neighborhood. Therefore, this method is explored in a brief pilot study in hopes of finding a discriminating feature for cover and stego-images.

In this pilot study, Westfeld's algorithm is modified to the following:

1) Extract all pixels in a color image which are not saturated (do not contain a value 0 or 255 in any of the red, green, or blue color components).

2) For each pixel obtained in 1), calculate the number of neighbors by searching the image for all 26 possible neighbors.
3) Depending on the number of neighbors for a given pixel, increment the appropriate neighbor counter.
4) Once all pixels are searched, compute the probability density for each # of neighbors by dividing the number of pixels who contain x neighbors by the total number of non-saturated pixels extracted in 1).

The algorithm is run on a test image, and its results are plotted in Figure 16. The test image is then embedded with a secret message of 50% payload, and the algorithm is run again to chart the probability densities for both a clean and stego-image. Notice that for most categories of neighbors there is not much change; however, the probability density for the number of pixels who have all 26 neighbors increases a moderate amount from the clean image to the stego-image.

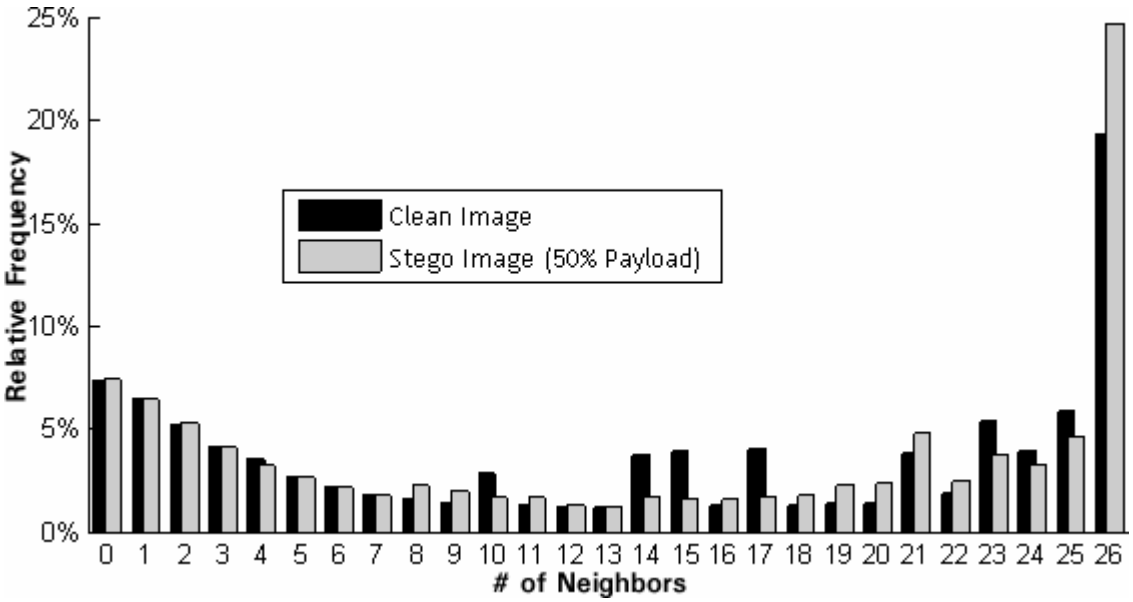

Figure 16. Probability Densities of the Number of Neighbors Present for all Pixels

Next, the pilot study shifts focus to determine whether the probability density of a full neighborhood (pixel containing all 26 neighbors) is a good discriminating feature. Therefore, five more test images are pooled from Image Set A. For each image, the probability density of pixels containing all 26 neighbors is computed. A random secret message with 50% payload is then embed into each image, and the probability density for the number of pixels containing all 26 neighbors is again computed. The results are shown in Table 9. Notice that for each image, there is a slight increase in probability densities from the cover images to the stego-images. Finally, an average of the five images shows that there is approximately a 3% increase in pixels containing 26 neighbors when an image contains a hidden message with a 50% payload.

Table 9. Probability Density for the Number of Pixels Containing all 26 Neighbors

	Image #1	Image #2	Image #3	Image #4	Image #5	Avg.
Clean	12.436 %	9.196 %	5.854 %	22.237 %	19.375 %	13.8196 %
Stego (50%)	14.825 %	14.487 %	9.004 %	25.145 %	22.289 %	17.1500 %

To summarize the findings of this pilot study, it is shown that by calculating the probability density for the number of pixels that contain all 26 neighbors, there is a slight difference between clean and stego images. Therefore, the steganalysis of images embed with hidden data using the perturbed quantization algorithm can attempt to be detected using this feature.

3.8 Performance Metrics

For the analysis of system performance, comparison of systems at varying payloads, and the comparison of the PQ system using different image processing operations, the Receiver Operating Characteristics Curve (ROC Curve) is the primary

tool used in this study. A ROC curve is extremely common in analyzing the discriminability of a set of features used to perform steganalysis. The ROC curve plots the probability of correct detection (true positive) along the *y*-axis versus the probability of a false alarm (false positive) along the *x*-axis. The resulting curve shows a systems performance as the percentage of false positive increases. A straight line at 45 degrees along the y = x axis means that a system could not correctly detect true positives any better than random guessing.

Further, the performance metric used throughout this study is derived from the ROC curve. The area under the curve of a ROC curve is an important metric. For example, a system which performs no better than random guessing will have an area under the curve (AUC) equal to 0.5. At the opposite end of the spectrum, a perfect system which always detects true positives and never creates false alarms will have an area under the curve $A = 1$. Further, Fridrich [FrG04] utilized the performance metric ρ where $\rho = 2A - 1$, and A = the area under the ROC curve. This measures the area between the ROC curve and the line Y=X (random guessing line). The equation $\rho = 2A - 1$ has the effect of normalizing the areas such that a system with performance no better than random guessing (A=0.5) is normalized to $\rho = 0$, and a system with perfect performance (A=1) is normalized to $\rho = 1$. Both the use of a ROC curve as well as the normalized area under the ROC curve (ρ) are extremely common performance analysis tools for steganographic systems and steganalysis detection systems. Accordingly, this metric is used throughout the performance analysis of the perturbed quantization system.

3.9 Comparison of Systems

Many of the downloadable steganographic systems which hide data in the spatial domain require the use of a specific image format. For example, S-Tools, Hide4PGP, WBStego, Steganos, and Steghide require the use of Bitmap files (BMP) for data hiding. Other tools use various image formats as well; for instance, White Noise Storm (WNStorm) requires PCX files. However, all of these tools share one thing in common, and that is that each of them hides the secret message in the least significant bits of pixel values. The only variation amongst these tools is in how pixels are chosen for carrying hidden data. None of these techniques is adaptive – hides data depending on the cover image; rather, pixels are chosen either at random or sequentially in order to carry the hidden data. Hiding data in sequential pixels such as WBStego is extremely elementary and can be easily detected and decoded by an attacker. Thus, when comparing the perturbed quantization system, it makes sense to compare it to a tool which hides data by spreading the message throughout the cover image randomly. S-Tools, Steganos, and Hide4PGP all accomplish hiding data in this manner. However, each of these tools requires the use of Bitmap (BMP) files. This is not a problem with color images, as PNG files and BMP files can be converted back and forth to both formats without losing any data. Conversely, when working with grayscale images, converting from PNG files to BMP files does pose a challenge. An 8-bit grayscale image in the PNG format is structured identically to a 24-bit RGB true color image except for the fact that only a single channel is present. An 8-bit grayscale image in the BMP format actually requires the use of a color map, similar to a color palette used in GIF formats. While no data would be lost in a conversion of an 8-bit grayscale from PNG to BMP, the BMP format

would be stored in a palette based style. Consequently, data hidden in the two formats would be done in a completely different manner as hiding data in palette based images requires a different hiding technique. As a result, using grayscale images interchangeably between PNG and BMP does not make practical sense.

3.9.1 A Generic LSB Hiding Approach

In order to compare performance of the PQ system to common non-adaptive steganographic tools which require the BMP format, a generic Least Significant Bit (LSB) hiding method is implemented to "simulate" the effect of those tools previously mentioned. Actually, it is not uncommon for researchers to utilize a generic LSB hiding method. Hany Farid [LiF02] used a generic LSB hiding technique in some of his work studying wavelet statistics. Similarly, Jeremiah Harmsen [Har03] implemented a generic LSB hiding method to study other image features for steganalysis. The generic LSB hiding method in this study is implemented to randomly choose pixels in the cover image and substitute the next bit from the secret message into the least significant bit of the chosen pixel. Some tools try to maintain first order statistics such as the number of unique colors in an image; however, all of the tools have the same statistical effect of flipping LSB's of pixels randomly scattered throughout the cover image. In summation, one of the systems for comparison to the PQ system is simply a generic LSB hiding method. This method actually simulates the effect of the following tools: S-Tools, Hide4PGP, Steganos, and White Noise Storm. The LSB algorithm is implemented in Matlab 7 Release 14 [Mat04], and all of the resulting stego images created from this method are stored in the PNG format. This maintains the consistency of image formats

for 8-bit grayscale images, while simulating the effect of flipping pixel LSB's as done in public steganographic software.

3.9.2 Hide v2.1

In addition to comparing the PQ system against a non-adaptive steganographic system, this study seeks to also compare the performance of the system against the adaptive steganographic software Hide v2.1 created by Toby Sharp [Sha01]. This steganographic tool hides a hidden message adaptively based on the pixels within a cover image, and utilizes the PNG format as well. Hide v2.1 provides an excellent comparative system to PQ steganography because of its adaptive algorithm.

3.10 The Testing Plan

The perturbed quantization steganographic system is tested and analyzed in two different phases. First, the system is tested using grayscale images as carrier files. The second phase of the research focuses on color images.

3.10.1 Perturbed Quantization Steganography & Grayscale images

The first phase of this research explores using color to grayscale conversion as the lossy image transformation in the perturbed quantization steganographic system. The standard weighted color to grayscale conversion is omitted from the study because of the results from the steganographic capacity pilot study in Section 3.6.1. As a result, the desaturate function is the only operation understudy throughout this testing phase. Recall from pilot study #1, that the average steganographic capacity for an image converted to grayscale via the desaturate function is just below 50%. As a result, secret messages are

embedded into the 1000 images from Image Set B using payloads of 5%, 10%, 20%, and 40%.

Each of the 1000 images from Image Set B are sent into the perturbed quantization system in which both a clean image and stego-image is output. The clean image is obtained by rounding the pixel values using a normal rounding function, whereas the stego-image is produced via the perturbed quantization method (shown in Figure 8). Furthermore, in the process of desaturating the image, a secret message is embedded into the image. The image is embedded with a message payload of 5%, 10%, 20%, and 40% up to the maximum allowable capacity for the image. In order to maintain consistency with the images being embedded, the same secret message is embedded into each and every image. The secret message is simply a random sequence of binary numbers generated from a pseudo random binary sequence generator (PRBSG). Using a random sequence of binary numbers simulates the effect of encrypting a secret message. Most steganographic systems encrypt a hidden message using a passphrase prior to embedding. Therefore, a hidden message is generated, and provided as input to the system for use with all of the images embedded using the desaturate function. Besides the image and the secret message, the only other input supplied to the system is a random integer $\leq 2^{31}$ to be used as the shared stego-key in order to generate the matrix D used within the PQ algorithm.

To compare the system against other steganographic techniques, the set of images from Image Set B are also embedded with data using the generic LSB hiding technique as introduced in Section 3.9.1, and the Hide v2.1 steganographic software. In this embedding process, the desaturated "cover images" output from the PQ system are used

to maintain consistency of cover images. Additionally, the exact same secret message used by the PQ embedding process is hidden in the cover images via the generic LSB hiding method and the Hide v2.1 software. Again, the secret message payload embedded into the set of images is 5%, 10%, 20%, and 40%.

Once the process is completed, the exact same set of images is embedded using the Perturbed Quantization method, the generic LSB hiding method, and the Hide v2.1 steganographic software. Therefore, regardless of the hiding technique used, all of the images are embed using the same desaturated "cover images" and secret messages.

3.10.2 Steganalysis of Grayscale Images

Despite the abundance of statistical attacks that don't apply to grayscale images, two of the more reliable features for detecting steganography in the spatial domain are the RS statistic [FrG01], and the HCF COM statistic [HaP03]. Therefore, both of these statistics are used in the steganalysis of desaturated images embedded via the PQ algorithm, generic LSB hiding method, and the Hide v2.1 software.

The two features are extracted from each image in order to perform pattern classification. The features are extracted from both the clean images output from the PQ steganographic system as well as all three sets of stego-images generated from the three different hiding methods. Pattern classification is done in a "known-classifier" manner meaning that clean and stego-images from each hiding method as well as at each message payload are trained and classified separately. For each hiding method and for each payload, there are approximately 1000 clean cover images, and 1000 stego-images. Exactly one half of the stego-images are chosen at random, and their corresponding cover images are used to form the Fisher's linear discriminant. The remaining half of stego-

images combined with their corresponding cover image are then projected onto the fisher line obtained from the training set. Next, the probability densities for both classes are estimated using Gaussian parzen windows of equal width, and a ROC curve is generated plotting the classification results. Finally, the normalized area under the ROC curve ρ is computed. This entire classification process is repeated 15 times for each hiding method, and secret message.

3.10.3 Perturbed Quantization & Color Images

The second phase of this research examines the perturbed quantization steganographic system used in combination with various downsampling methods. Because the nearest neighbor interpolation method does not require any rounding function, only the bicubic and bilinear interpolation methods are at the focus of this study. Additionally, the anti-aliasing pre-filter applied prior to down-sampling which is a system parameter, remains a constant throughout this study. Namely, the default filter in Matlab, the Hamming filter, is used. The other system parameter not at the focus of this study is the downsampling scaling factor – the amount at which an image is reduced in size. Results from the steganographic capacity study showed that varying this factor does not have any effect on the steganographic capacity of the system. Therefore, this investigation looks to compare the performance of the two interpolation methods when used as the lossy image processing operation within the perturbed quantization system.

A second factor at the center of this study is the epsilon value ε used to define a selection rule for the choosing of changeable pixels. The steganographic capacity pilot study (Section 3.6) revealed that as the epsilon value is decreased within the downsampling interpolation methods, the capacity is also decreased. Steganalysis of

extremely low embedding rates is not very reliable. Thus, the only values used for the epsilon factor are 0.1 and 0.05. Any epsilon value less than 0.05 will allow for an extremely small steganographic capacity which will be difficult to detect regardless of the hiding method, and any epsilon value greater than 0.1 will defeat the purpose of the perturbed quantization selection rules.

The final performance analysis of this study is to examine three different information hiding techniques: the perturbed quantization system, the generic LSB hiding method, and the Hide v2.1 hiding technique introduced by Toby Sharp. Image Set A is used as the workload for this portion of the experiments, and all images are downsampled from the original 2048x1360 to 0.25 of its original size (512x340) which is a common size for images posted on the web. First, the entire set of images from Image Set A are downsampled to 512x340 and in the process of downsampling, a secret message is hidden into each image using an epsilon value of 0.1 and 0.05. With the epsilon value at 0.1, a message payload of 5%, 10% and 20% is embedded into every image using both the bicubic and bilinear interpolation methods. In the process of downsampling, a second set of images is created by rounding the pixel values normally. This results in one set of 50 cover images created from bicubic interpolation, and one set of 50 cover images created from bilinear interpolation. Next, the same original images (2048x1360) are downsampled using an epsilon value of 0.05 for the selection rule within the PQ system. The same message is embedded at a payload of 10% using Image Set A, but using only the bicubic interpolation method. Again, the secret message is generated as a pseudo random binary sequence in order to simulate the effect of an encrypted file.

For system comparison, the two sets of cover images (bicubic cover images, bilinear cover images) are then used along with the same secret message, and hidden using the generic LSB hiding method. Using the LSB hiding method, message payloads of 5%, 10%, 20%, and 40%, are embedded for both sets of cover images. Finally, the embedding process is repeated with the Hide v2.1 software. Message payloads of 5%, 10%, 20%, and 40% are again used to hide with this third hiding technique. Once all stego-images have been created, there are a total of 12 sets of 50 stego images generated from bicubic interpolation, and 12 sets of 50 stego-images generated from bilinear interpolation.

3.10.4 Steganalysis of Color Images

Again, steganalysis using a pattern classifier is used in order to measure the performance of the three hiding techniques. Similar to working with grayscale images, a "known-classifier" is generated meaning that all systems, interpolation techniques, and message payloads are trained and tested separately.

3.10.4.1 Feature Extraction

As mentioned in the previous phase of this research, the RS-statistic and histogram characteristic function center of mass have emerged as the best image features for discriminating clean images from images containing hidden data in the spatial domain. Therefore, both of these statistics are again used in the steganalysis of color images embedded using the various downsampling techniques. However, color images contain three color channels, and thus each of the features provides a statistic for each color component. For a final feature, the probability density of colors which contain all 26 neighbors is used in an effort to help distinguish stego images from clean images. The

pilot study in Section 3.7 showed that stego-images contain a slightly higher probability density of colors that contain all 26 possible neighbors than do clean images. In summation, seven features are extracted and used to classify stego-images and clean images. The seven features are displayed in Figure 17.

RS-Statistic Red Channel	RS-Statistic Green Channel	RS-Statistic Blue Channel	HCF COM Red Channel	HCF COM Green Channel	HCF COM Blue Channel	% of Pixels with all 26 Neighbors

Figure 17. The Feature Set for Color Image Steganalysis

3.10.4.2 Pattern Classifier

Once all features are extracted from the 600 images generated, pattern classification is performed using the Fisher's linear discriminant. Each hiding technique and payload size is trained and tested separately. For each technique, 40 out of the 50 clean images and corresponding stego images are trained on the seven features in order to create a fisher line. The remaining 10 cover images and corresponding stego-images are then projected onto this fisher line. Density estimation of the two classes is done using Gaussian parzen windows of equal width, and the resulting data is used to generate a ROC curve. Classification performance is then computed as the normalized area under the ROC curve. This process is repeated 150 times due to the small amount of testing data. Further, each hiding technique is plotted on the same ROC curve in order to compare performance amongst the systems at a given payload. Finally, a three-way ANOVA is computed for the data in order to verify any visual conclusions drawn from the ROC curves.

3.10.5 Repeating Experiments

The entire set of experiments, both the first and second phase, are repeated a total of three times in order to validate the results. Each time the entire experiment is repeated, a new secret message is used; therefore, a total of three secret messages are used in order to verify that the results hold true with different secret messages.

3.11 Summary

This chapter first introduces the perturbed quantization steganographic system and its application to the spatial domain by introducing two different lossy image transformations: color to grayscale conversion, and image downsampling. A pilot study explores the steganographic capacities of the various lossy image processing operations used with the perturbed quantization system. A second pilot study briefly revisits the neighborhood attack originally formulated by Andreas Westfeld, in order to derive a more discriminating feature for stego-images. Finally, a test plan is described in which the performance of the system under study is compared to other information hiding techniques. The results and analysis are provided in the next two chapters.

IV. Results and Analysis

4.1 Chapter Overview

The results from the testing methodology described in Chapter III are presented in this chapter. First, the steganographic detection results from the grayscale image study are shown in Section 4.2, and then the detection results from the downsampling study are displayed in Section 4.3. Lastly, an explanation of the results in Section 4.4 offers insight into the meaning and significance of this entire investigation.

4.2 Steganalysis of Grayscale Images

In this section, the performance of the perturbed quantization system using grayscale images is presented. This system is also compared to the generic LSB hiding method as well as the Hide v2.1 steganographic software.

4.2.1 Perturbed Quantization Steganography & the Desaturate Function.

In the first experiment, steganalysis is done using features extracted from the desaturated cover images and stego-images in which data was hidden using the perturbed quantization hiding method. The classification results, measured by the normalized area under the ROC curve, ρ, are displayed in Table 10. Each ρ value shown in Table 10 represents an average of the 15 classification trials at the given payload and using the corresponding secret message.

Without any comparison to other hiding methods, the results in Table 10 clearly illustrate that desaturated stego-images embedded via the perturbed quantization hiding method are difficult to decipher from clean images using the features described in

Chapter III. Even for a secret message payload of 40% (0.4bpp), the greatest classification results only provided an area under the ROC curve of 0.5476 ($\rho = 0.0952$). This is only a slight advantage over guessing at random.

Table 10. Mean Detection Rates of PQ Steganography Using Desaturated Stego-Images

Message Payload *(Bits per Pixel)*	Message A	Message B	Message C
5%	0.0131	0.0206	0.0229
10%	0.0201	0.0257	0.0287
20%	0.0429	0.0397	0.0427
40%	0.0952	0.0945	0.0937

4.2.2 Simple LSB Substitution & the Desaturate Function.

The experiment is repeated for a generic LSB hiding system. Results from steganalysis classification of the generic LSB hiding method are presented in Table 11. Again, the normalized area under the ROC curve, ρ, is displayed as an average of 15 classification trials for each payload using each secret message

Table 11. Mean Detection Rates of LSB Steganography Using Desaturated Stego-Images

Message Payload *(Bits per Pixel)*	Message A	Message B	Message C
5%	0.3935	0.4211	0.4246
10%	0.7169	0.7130	0.7132
20%	0.9420	0.9393	0.9399
40%	0.9965	0.9961	0.9955

A visual analysis of the results in Table 11 show that images embed using the generic LSB hiding method are reliably detected. Recall that perfect classification would result in a ρ value of 1.0, while random guessing results in a normalized area under the ROC of 0. Notice in Table 11 that near perfect detection is achieved at secret message

payloads of 20% and 40%. Even stego-images embed with smaller messages (5%, 10% payloads) are classified correctly much more frequently than random guessing. Additionally, the ROC curves in Figure 18 display the increase in classification accuracy as message payloads increase, as well as the improvement in classification compared to the random guessing line.

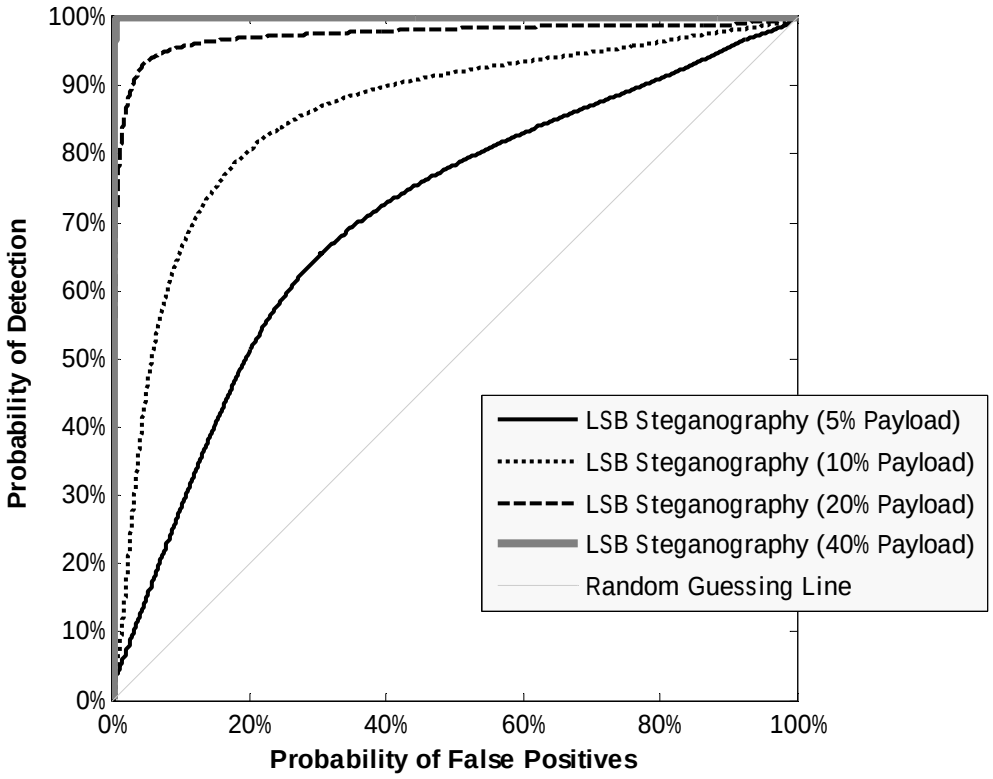

Figure 18. ROC Curves from Classification of Desaturated Stego-Images Embed via a Generic LSB Hiding Method

4.2.3 Hide v2.1 Steganographic Software & the Desaturate Function.

Finally, the experiments using desaturated images are completed by performing steganalysis of the Hide v2.1 software created by Toby Sharp. Table 12 displays the

classification results from this hiding technique, measured by taking the average normalized area under the curve from 15 trials of classification at each payload value and secret message.

Table 12. Mean Detection Rates of Hide v2.1 Using Desaturated Stego-Images

Message Payload *(Bits per Pixel)*	Message A	Message B	Message C
5%	0.0230	0.0294	0.0271
10%	0.0345	0.0302	0.0355
20%	0.0382	0.0365	0.0375
40%	0.1103	0.0991	0.0944

A quick analysis of the detection results in Table 12 shows that deciphering clean grayscale images from stego-images embed using Hide v2.1 is difficult. Even at a secret message payload of 40%, classification accuracy is only a little bit better than guessing at random. In addition, as with the two previous systems, there appears to be an increase in detectability as the message payload increases. The detectability of secret message A increases from 0.0230 for a 5% payload to 0.1103 for a 40% payload.

4.2.4 Studying the Effect of the Secret Message Payload.

One of the goals of this research effort as described in Chapter III is to study the effect that the secret message length has on the performance of the perturbed quantization system. Before doing a numerical analysis of the classification data, some conclusions about the payload's effect can be drawn from a visual analysis of the data. Notice in Table 10 that as the message payload increases, ρ values also increase a small amount. For example, the ρ value for secret message A at 5% payload is 0.0131, and the ρ value for secret message A at 40% payload is 0.0952. The increase in ρ as the payload

increases is consistent across all three secret messages. Further, plotting the ROC curve at each payload of secret message A displays this minor variation in detection performance.

In order to verify that the factor of secret message length does have a main effect on detection performance of the PQ hiding technique, a two-way ANOVA is computed for the gathered data. The resulting ANOVA Table is shown in Table 13.

Table 13. ANOVA Table for the Factors of Secret Message Payload and Message Content

Source	Sum of Squares	Degrees of Freedom	Mean Square	F-Value	P-Value
Secret Message	0.00052	2	0.00026	2.31	**0.1024**
Message Payload	0.15960	3	0.05320	470.97	**0**
Interaction	0.00096	6	0.00016	1.41	**0.2134**
Errors	0.01898	168	0.00011		
Total	0.18005	179			

In Table 13, the P-value for the factor of secret message content is 0.1024. Thus, it can be stated that all samples drawn from this factor are not statistically different; hence, the content of a secret message does not have a main effect on the performance of the system. There also doesn't appear to be a main effect from the interaction of the secret message content and secret message payload. However, Table 13 notes that the p-value for the factor of secret message payload is 0. This means that there is strong statistical evidence that the payload does have a main effect on system performance. More specifically, as the payload increases the classification performance as measured by the normalized area under the ROC also increases. However, this increase in detectability is relatively small.

4.2.5 Performance Comparison of the Three Steganographic Systems.

Another goal of this research is to compare the performance of the perturbed quantization technique to the other two spatial hiding methods. A visual test clearly depicts much better classification results for the generic LSB algorithm than the two adaptive hiding techniques: the PQ method, and the Hide v2.1 software. The ROC curves presented in Figure 20 show that for message payloads of 10%, the generic LSB hiding method is detected with significantly greater accuracy than either of the adaptive algorithms. Similar conclusions can be drawn from the ROC curves for other payloads which are presented in Appendix A. This means that the two features used to discriminate clean images from stego-images are much more effective with the generic LSB method. As a result, it can be concluded that stego-images embedded by either adaptive hiding technique are considerably more difficult to detect compared to those stego-images embed with a generic LSB substitution system. However, from the ROC curves in Figures 19 and Appendix A it cannot be determined whether the PQ method or Hide v2.1 software is less detectable. Therefore, a statistical analysis of data is performed in order to determine which of the two adaptive hiding techniques is less detectable using desaturated images.

The data from Tables 10 and 12 are summarized in Table 14 in order to take a closer look at the comparison in performance of the perturbed quantization system and Hide v2.1 software.

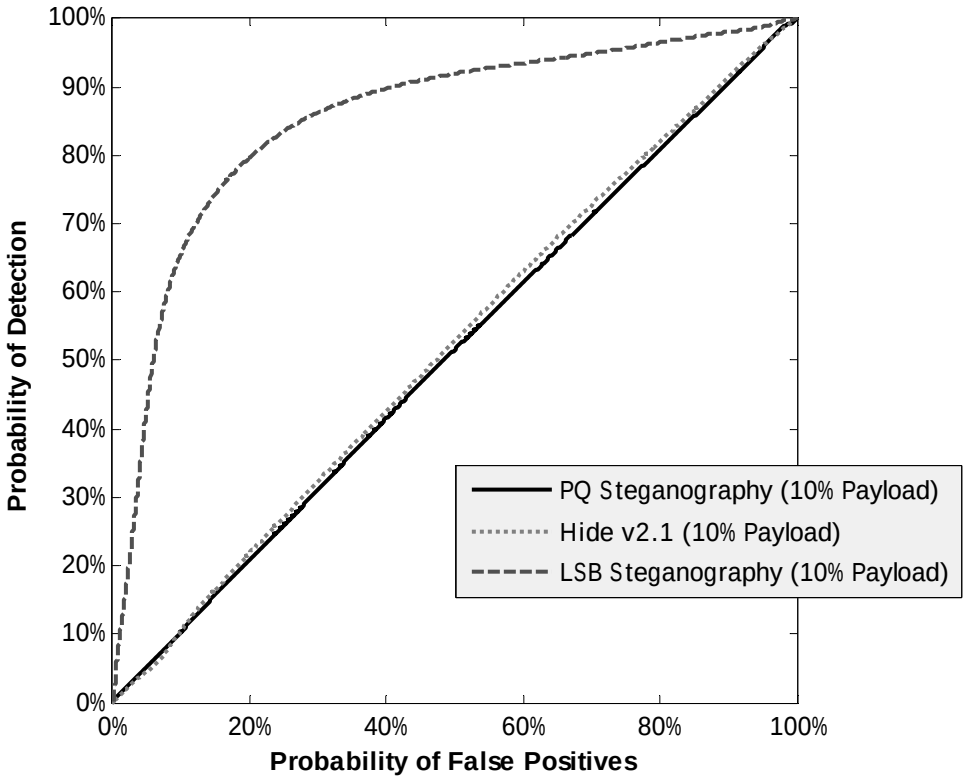

Figure 19. ROC Curves from the Classification of Stego-Images for All Three Systems Using Desaturated Grayscale Images

Table 14. Performance Comparison of Detection Rates for PQ System and Hide V2.1

	PQ System			Hide V2.1		
Payload	**Message A**	**Message B**	**Message C**	**Message A**	**Message B**	**Message C**
5%	0.0131	0.0206	0.0229	0.0230	0.0294	0.0271
10%	0.0201	0.0257	0.0287	0.0345	0.0302	0.0355
20%	0.0429	0.0397	0.0427	0.0382	0.0365	0.0375
40%	0.0952	0.0945	0.0937	0.1103	0.0991	0.0944

After looking at Table 14 it is not immediately clear which system is less detectable; however, it appears that the PQ system has slightly smaller ρ values than does the Hide software. For example, at 5% payload, the detection performance of the PQ

system is around 0.01-0.02 whereas the Hide v2.1 software has detection performance between 0.02-0.03. Similar results hold true for message payloads of 10% and 40%, where detection appears to be slightly smaller for the PQ system. In order to verify this visual analysis, a three-way analysis of variation is computed using the factors of secret message payload, secret message content, and the two steganographic systems being compared. An ANOVA table from this test is displayed in Table 15 showing the p-values for the three factors as well as the three interactions.

Table 15. ANOVA Table for System Comparison Study (Hide v2.1 vs. PQ)

Source	Sum of Squares	Degrees of Freedom	Mean Square	F-Value	P-Value
System	0.00196	1	0.00196	11.62	**0.0007**
Message Payload	0.31979	3	0.10660	632.8	**0**
Message Content	0.00004	2	0.00002	0.13	**0.8807**
System x Payload	0.00248	3	0.00083	4.9	**0.0024**
System x Content	0.00079	2	0.00040	2.35	**0.0971**
Payload x Content	0.00263	6	0.00044	2.6	**0.0177**
Errors	0.05761	342	0.00017		
Total	0.38530	359			

From Table 15 it can be seen that similar to the previous ANOVA, the secret message content does not have a main effect on the detectability of either system. This is denoted by a very large p-value, 0.8807. Additionally, the factor of message payload has a p-value of zero meaning that the payload does have a main effect on the outcome of detectability. However, the focus of this performance comparison lies with the factor of "System" as written in the ANOVA Table. The p-value for the system factor is also very

close to zero, and thus it can be concluded that there is statistical evidence that the performance of the two systems as measured by ρ is significantly different. Therefore, the original conclusions drawn from the visual test are verified by the three-way ANOVA. It can be said that when using desaturated images the perturbed quantization steganographic hiding technique is somewhat less accurately detected than is the Hide v2.1 software. Although, the difference in performance between the two hiding techniques is very small, especially when compared to the generic LSB hiding method. A much safer conclusion from these results is that the two adaptive hiding algorithms (PQ, Hide v2.1) are much less detectable than the generic LSB hiding method. An interpretation of these conclusions is explained in broader context in Section 4.4.

4.3 Steganalysis of Color Images

The results from the second phase of testing the steganographic systems are presented in this section. First the results from each individual system are discussed, and then the performance of the various steganographic systems are compared.

4.3.1 Downsampling with Bicubic Interpolation

This subsection presents the results from experiments involving downsampled images via bicubic interpolation.

4.3.1.1 Perturbed Quantization Steganography & Bicubic Interpolation

Seven features are extracted from each of the stego-images from image set B that contain data hidden via the perturbed quantization technique. In this particular experiment, the set of stego-images which were downsampled using bicubic interpolation are trained and classified using a fisher linear discriminant. The results from the

classification of this data are presented in Table 16, where the detection performance is charted by calculating the normalized area under the ROC curve ρ. Each of the ρ values displayed in Table 16 are averages from 15 trials of pattern classification.

Table 16. Mean Detection Rates of PQ Steganography Using Bicubic Interpolation

Message Payload *(Bits per Pixel)*	**Message A**	**Message B**	**Message C**
5%	0.0058	0.0090	0.0052
10%	0.0409	0.0356	0.0392
20%	0.0633	0.0566	0.0602

The detection results in Table 16 show that distinguishing between clean images and stego-images that are embed with secret messages using the perturbed quantization method and downsampling with bicubic interpolation is extremely difficult. For instance, detection rates for message payloads of 5% are on par to random guessing. A ρ value of 0.0058 means that the area under the ROC is only 0.5029, whereas the area under the ROC for a random guessing system is 0.5. Even at the maximum allowable payload of 20% for the PQ system under study, detection results are still extremely low.

4.3.1.2 Simple LSB Substitution & Bicubic Interpolation

Next, the same features are extracted from the set of stego-images generated from the generic LSB hiding technique. The detection results, denoted by the value ρ, are displayed in Table 17. The results are displayed for message payloads of 5%, 10%, 20%, and 40%, and three secret messages.

The ρ values in Table 17 are averages from 15 trials of classification of clean and stego images, and note the extremely high detection rates of the generic LSB hiding

method. With secret message payloads of 20% and 40%, the derived fisher linear discriminants are able to decipher clean from stego-images almost perfectly. Even message payloads of 10% are detected very accurately. At 5% payloads, classification was not done perfectly; however, there still exists a substantial advantage over random guessing.

Table 17. Mean Detection Rates of LSB Steganography Using Bicubic Downsampled Images

Message Payload *(Bits per Pixel)*	**Message A**	**Message B**	**Message C**
5%	0.5083	0.5259	0.5165
10%	0.7929	0.8154	0.8141
20%	0.9435	0.9606	0.9481
40%	0.9827	0.9836	0.9851

4.3.1.3 Hide v2.1 Steganographic Software & Bicubic Interpolation

The bicubic interpolation experiments conclude with the performance analysis of the Hide v2.1 software. Again, features are extracted from clean images which are downsampled using bicubic interpolation, as well as the corresponding stego-images embedded with secret message payloads of 5%, 10%, 20%, and 40%. Training and classification of the images is done a total of 15 times for each message and message length, and the results are shown in Table 18.

Table 18. Mean Detection Rates of Hide v2.1 Using Bicubic Downsampled Images

Message Payload *(Bits per Pixel)*	**Message A**	**Message B**	**Message C**
5%	0.0971	0.0965	0.1043
10%	0.1589	0.1508	0.1518
20%	0.2458	0.2492	0.2310
40%	0.3929	0.3798	0.3787

It appears that images embedded with hidden data via the Hide v2.1 software are moderately difficult to detect. However, as the message payload increases, the classification accuracy also increases. For example, detection accuracy as measured by the normalized area under the ROC for a message payload of 5% is only 0.10, but for a message payload of 40% the ρ value is 0.3929. This is a reasonable increase over a system which performs no better than chance.

4.3.2 Downsampling with Bilinear Interpolation

This subsection presents the results from experiments involving downsampled images via bilinear interpolation.

4.3.2.1 Perturbed Quantization Steganography & Bilinear Interpolation

Whereas the previous sections examined the results from experiments involving downsampling with bicubic interpolation, this section reveals the outcome of steganalysis on stego-images that were downsampled using bilinear interpolation. The perturbed quantization steganographic system, which hides data during the downsampling process, is found to be extremely difficult to detect. The overall performance of this system is charted in Table 19, where the normalized area under the ROC curve, ρ, is calculated and averaged over 15 trials in order to obtain each of the values in Table 19.

Table 19. Mean Detection Rates of PQ Steganography Using Bilinear Interpolation

Message Payload *(Bits per Pixel)*	**Message A**	**Message B**	**Message C**
5%	0.0062	0.0039	0.0079
10%	0.0393	0.0327	0.0321
20%	0.0533	0.0618	0.0599

The detection of images embed with messages as small as 0.05 bits per pixel is not much better than a random guessing system. Detection accuracy increases slightly as the message payload increases; however, this increase is extremely small. For larger payloads of 10% and 20%, the detection capabilities only increase to ρ values of 0.05 - 0.06. These ρ values mean that the actual area under the ROC curve, A, is between 0.5266 and 0.53. Again, these areas are not significantly greater than the area under the random guessing line.

4.3.2.2 Simple LSB Substitution & Bilinear Interpolation

Next, the results from the steganalysis of images embed with data using the generic LSB hiding method are displayed in Table 20. Steganalysis is performed using the same seven features on images containing secret message payloads of 5%, 10%, 20%, and 40%.

Table 20. Mean Detection Rates of LSB Steganography Using Bilinear Downsampled Images

Message Payload *(Bits per Pixel)*	**Message A**	**Message B**	**Message C**
5%	0.5099	0.5358	0.5131
10%	0.8098	0.8119	0.8117
20%	0.9627	0.9697	0.9580
40%	1.0000	1.0000	1.0000

A quick look at the results in Table 20 reveal that the feature set used to decipher clean from stego-images is very accurate. In fact, the detection of images with 40% payload is a perfect 1.0, meaning that all images were classified at 100% accuracy with 0% false positives. Even detecting stego-images containing 20% payloads are done so

almost perfectly. Finally, detection performance of images with smaller payloads is still much better than random guessing.

4.3.2.2 Hide v2.1 Steganographic Software & Bilinear Interpolation

The results from the last system under study, the Hide v2.1 software, are shown in Table 21. Again, the normalized area under the ROC is averaged and charted for each of the given payloads and secret messages.

Table 21. Mean Detection Rates of Hide v2.1 Using Bilinear Downsampled Images

Message Payload *(Bits per Pixel)*	**Message A**	**Message B**	**Message C**
5%	0.0788	0.0810	0.0850
10%	0.1313	0.1214	0.1201
20%	0.2091	0.2180	0.2129
40%	0.3507	0.3684	0.3749

The early conclusion from the results in Table 21 is that detection of stego-images with low payloads is difficult; however, as the payload size increases, detectability increases a moderate amount. The correct classification of stego-images with moderate message payloads (> 20%) are far from perfect; nevertheless, the feature set used provides a good advantage over randomly guessing which images contain hidden data.

4.3.3 Studying the Effect of the Secret Message Payload

Similar to the experiments involving desaturated grayscale images, the downsampling study also looks to study the effect that the message payload has on detection capabilities of the perturbed quantization system under study.

Just by looking at the data in Table 16, which corresponds to detection performance of the PQ system with bicubic interpolation, one can see that the normalized

area under the ROC curve, ρ, increases a small amount as the secret message payload increases. For instance, the ρ value increases from 0.0058, to 0.0409, to 0.0633 for message A at payloads of 5%, 10% and 20% respectively. Therefore, a visual conclusion would be that as the message payload increases the ability to detect stego-images increases with the PQ system. In order to verify this visual analysis, a two-way ANOVA is performed using the factors of payload and message content. The corresponding ANOVA table is shown in Table 22.

Table 22.ANOVA Table for the Factors of Secret Message Payload and Message Content

Source	Sum of Squares	Degrees of Freedom	Mean Square	F-Value	P-Value
Message Content	0.00020	2	0.00010	0.62	**0.5384**
Message Payload	0.06486	2	0.03243	199.6	**0**
Content * Payload	0.00049	4	0.00012	0.76	**0.5555**
Errors	0.02047	126	0.00016		
Total	0.08603	134			

The ANOVA Table in Table 22 does indeed verify this visual conclusion. The P-value for the effect of message payload is 0; therefore, there is significant statistical evidence that the detection distributions do vary as the payload increases. Thus, it can be concluded that embedding images with the PQ algorithm which involve downsampling images with bicubic interpolation will be detected slightly more reliably as the length of the secret message increases. However, the actual increase in detection is extremely small. Similar results are found from the bilinear interpolation experiments. The ANOVA Table verifying this conclusion is found in Appendix B.

4.3.4 Performance Comparison of the Three Steganographic Systems

Conclusions can be drawn from the classification data presented in previous sections; however, it is one of the primary objectives of this research to compare results from the three steganographic systems in order to get an idea of how the perturbed quantization hiding technique fares against the other hiding methods present in this investigation.

To compare the three hiding techniques, a closer look is needed at the detection data for bicubic interpolation. In addition, ROC curves portraying the three systems performance with message payloads of 10% are displayed in Figure 20.

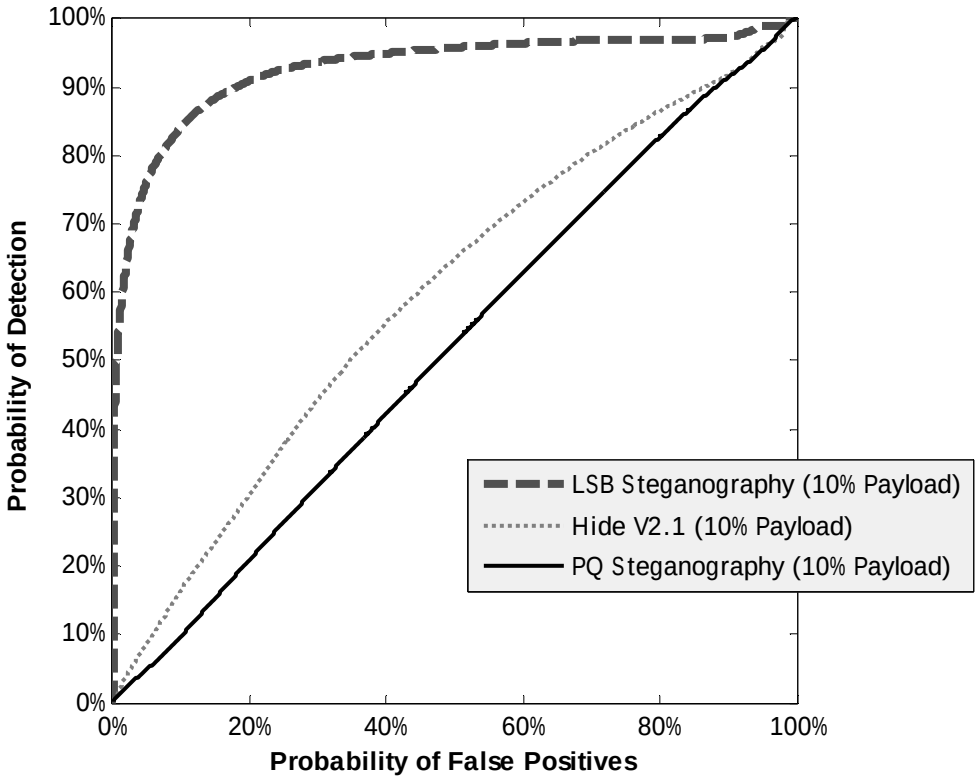

Figure 20. ROC Curves from the Classification of Downsampled Stego-Images via Bicubic Interpolation for All Three Systems

Perhaps the most obvious difference in performance from the three systems as shown in the ROC curves in Figure 21 is the fact that the generic LSB hiding technique is detected with much greater accuracy than either of the other two hiding algorithms. Additional ROC Curves for other message payloads are presented in Appendix A. Therefore, based on a visual analysis of the data and ROC curves, it can be concluded that the feature sets used in this experiment detected stego-images generated from the generic LSB method the most reliably of the three hiding systems.

The remaining two hiding systems: the PQ algorithm, and the Hide v2.1 software are detected at rates that are much more comparable. Even so, the data from Tables 16 and 18, as well as the ROC curves in Figure 21 do show that stego-images embed via the Hide v2.1 software system are detected more often than those images derived from the PQ system. To verify this claim, a three-way ANOVA is performed using the factors of message length, message content, as well as the steganographic algorithm used. The resulting ANOVA Table is presented in Table 23.

Table 23. ANOVA Table for System Comparison Study (Hide v2.1 vs. PQ)

Source	Sum of Squares	Degrees of Freedom	Mean Square	F-Value	P-Value
System	1.13994	1	1.13940	2619.71	**0**
Message Payload	0.43443	2	0.21722	499.18	**0**
Message Content	0.00055	2	0.00027	0.63	**0.5336**
System * Payload	0.09710	2	0.04855	111.57	**0**
System * Content	0.00022	2	0.00011	0.25	**0.7781**
Payload * Content	0.00168	4	0.00042	0.97	**0.4262**
Errors	0.11140	256	0.00044		
Total	1.78531	269			

The ANOVA table verifies this claim that there is a difference between the two systems performance, this is noted by the p-value for the system factor in Table 23. Therefore, it can be concluded both visually and numerically that the Hide v2.1 steganographic software is detected more accurately with the provided features than is the perturbed quantization hiding algorithm.

Similar results are found when using bilinear interpolation. Namely, the generic LSB steganographic system is detected with high reliability, and the two adaptive algorithms are detected with much more comparable results. ROC Curves pertaining to these experiments are presented in Appendix A. Additionally, there is again a noticeable difference between detection rates of the Hide v2.1 software and the PQ system. Specifically, the PQ system is detected with lower accuracy than the Hide software. An ANOVA table verifying this claim is displayed in Appendix B.

4.3.5 Performance Comparison of the Two Interpolation Techniques

Another objective of this research is to determine if any particular lossy image transformation is better when used within the perturbed quantization system in terms of being less detectable. Hence, a comparison of the detection results from both downsampling processes is used in order to determine if stego-images generated from one interpolation technique are detected more or less frequently than the other interpolation technique. Table 24 summarizes the detection results from both interpolation techniques.

An initial examination of the data in Table 24 does not demonstrate a large difference in detection performance. There appears to be a very small difference between the two sets of data. More specifically, the images embed with bicubic interpolation seem

to be detected with slightly higher accuracy. However, regardless of whether a statistical difference does exist between the two interpolation techniques, such a difference is so small that it cannot be concluded that one algorithm is better than the other. There does not exist enough data or difference in data from the two techniques in order to conclude anything meaningful. Thus, a general conclusion can be made from this research effort, that there is comparable detection performance from stego-images generated from both the bicubic and bilinear interpolation techniques.

Table 24. Performance Comparison of Detection Rates for PQ System and Hide V2.1

	PQ Steg w/ Bicubic Interpolation			PQ Steg w/ Bilinear Interpolation		
Payload	**Message A**	**Message B**	**Message C**	**Message A**	**Message B**	**Message C**
5%	0.0058	0.0090	0.0052	0.0062	0.0039	0.0079
10%	0.0409	0.0356	0.0392	0.0393	0.0327	0.0321
20%	0.0633	0.0566	0.0602	0.0533	0.0618	0.0599

4.3.6 Studying the Effect of Epsilon

Finally, the effect of varying the epsilon value used in the PQ selection rule is examined in this pilot study. Recall that during the embedding process, only those pixels whose fractional remainder lies between the interval (0.5- ϵ, 0.5+ ϵ) are selected as potential pixels to carry secret data. In this study only the detection performance from the bicubic downsampling algorithm is used and compared to the detection performance of the bicubic algorithm using a smaller epsilon value in the selection rule. The detection results from using the two selection rules 0.5+/- 0.1, and 0.5+/-0.05 are plotted in Table 25.

The PQ system with a lower epsilon value appears to be detected with even lower accuracy than the PQ algorithm using the higher epsilon value. To verify this analysis, a

two way ANOVA is computed and plotted in Table 26.

Table 25. Performance Comparison of Detection Rates for Various Epsilon Values

	PQ System (ε = 0.1)			PQ System (ε = 0.05)		
Payload	**Message A**	**Message B**	**Message C**	**Message A**	**Message B**	**Message C**
10%	0.0409	0.0356	0.0392	0.0158	0.0215	0.0220

Table 26. ANOVA Table for the Factor of Epsilon

Source	Sum of Squares	Degrees of Freedom	Mean Square	F-Value	P-Value
Epsilon Value	0.00796	1	0.00796	39.24	**0**
Message	0.00009	2	0.00005	0.23	**0.7964**
Message * Epsilon	0.00048	2	0.00024	1.18	**0.3127**
Errors	0.01703	84	0.00020		
Total	0.02555	89			

The ANOVA Table reveals that the epsilon value does in fact have a main effect on the detectability of the system. Additionally, there appears to be no main effect from the interaction of the message content and the epsilon value. Nonetheless, this verifies the visual analysis that lowering the epsilon value within the selection rule does decrease the detectability of the system.

4.4 Secrets for the Secret

Despite the fact that most of the experiments turned out as expected, a great deal of meaningful conclusions can be drawn from this study. Not only do these results reflect the performance of the perturbed quantization system under study, but the results can

benefit the steganography and steganalysis communities as a whole. This section summarizes the findings of the investigation by providing steganographers tips for minimizing their risk of detection as well as a list of reasons why the PQ method should be chosen as their method for covert communication.

4.4.1 The Perturbed Quantization Algorithm outperforms the others

Perhaps, the most obvious inference that can be made from this investigation is the fact that the perturbed quantization algorithm outperforms other state-of-the-art spatial domain hiding techniques in the sense that it is tremendously difficult to detect the system. In comparison, both the generic LSB hiding method and the Hide v2.1 software are detected with much higher reliability. Additionally, this study proved that the perturbed quantization algorithm can be applied to the spatial image domain. Previous work proved the algorithm's application into the transform domain and its subsequent difficulty in detection. Similarly, this examination shows that even using the state of the art in steganalytic image features from the spatial domain, that the hiding technique is detected with not much better accuracy than random guessing. Therefore, it could be argued that hiding secret messages using the perturbed quantization algorithm is the most secure hiding technique presently known. Future steganographers wishing to communicate covertly through a secure channel will be able to do so by hiding messages in the LSB's of pixels of both color and grayscale images using the PQ hiding technique, and the likelihood of the clandestine message being detected is extremely low.

On the contrary, the art of detecting hidden data in images or any other digital carrier signal is especially problematic. One could argue that the field of steganography is further along than is its antithesis, steganalysis. After all, there exist far more ways to

hide data into digital images than there are ways to detect hidden content. Nonetheless, this investigation attempts to detect stego-images using state of the art discriminating features, and the PQ system is considered secure against the attacks tested. It is not out of the question for a future attack to be more applicable to this hiding technique, but given the way the PQ algorithm minimizes any added noise in a stego-image, the system can be expected to be as secure if not more secure than other hiding techniques.

4.4.2 Lossy image transformations provide varying steganographic capacities

In the process of applying the PQ algorithm to the spatial domain, several different lossy image processing operations were introduced and considered. The lossy image transformation used within the PQ system is one of the components which can be varied by a steganographer, and is one of the areas of interest in this study. Results from this research effort proved that all of the transformations considered had similar detection rates. Probably the most important difference in the transformations used within the system is the steganographic capacity allowed by the transformation. For example, in the color to grayscale conversion study it was found that the widely used weighted grayscale conversion function offers an extremely small steganographic capacity for digital images. On the contrary, the desaturate function, another grayscale conversion function, makes an excellent choice for the lossy transformation as it offers a large steganographic capacity while minimizing rounding error. Moreover, detection rates were quite small for such a large message payload of 40% with the desaturate function. However, containing only a single channel, grayscale images do not make good carrier images for large messages such as audio or video clips. For color images, downsampling using various interpolation techniques appears to have similar detection performance for both bilinear and bicubic

interpolation. The steganographic capacity is also not affected by the interpolation technique used while downsampling an image. In conclusion, based on this study, it can be stated that the lossy transformation used within a PQ hiding system will not have a significant effect on the detectability of the generated stego-image, but will effect the steganographic capacity of a given cover image.

4.4.3 Epsilon in the selection rule is a factor which effects detectability

The final component of the PQ system analyzed in this study is the epsilon value ϵ used during the selection rule of choosing pixels which can be embed with hidden data. The pilot study with bicubic interpolation revealed that lowering this threshold value does improve the security of the system. This makes the desaturate function even more appealing to steganographers as it maintains a minimum epsilon value of 0 for all changeable pixels. These results are again not surprising, as one would expect the detectability of a stego-image to decrease as the amount of rounding error introduced by the secret message decreases. Thus, steganographers should take into consideration using a small epsilon value within the selection rule while hiding messages with the PQ algorithm.

4.4.4 Adaptive algorithms are more secure than non-adaptive hiding algorithms

At the time of this research, the author is not aware of many adaptive information hiding algorithms which hide data in the spatial domain. Other than the PQ system under study, the Hide v2.1 software is one such example of an adaptive algorithm. The results throughout all types of experiments revealed that the two adaptive hiding algorithms were detected much less reliably than the generic non-adaptive LSB hiding technique. Intuitively this makes sense, as all of the image features used in this study are computed

uniformly over an image. For example, the RS-statistic is best suited for detecting messages uniformly spread over an image. Adaptive algorithms tend to hide data in varying regions of an image, and thus statistics such as the RS-statistic tend to underestimate the amount of hidden content in an image. In spite of the fact the PQ system outperformed the Hide v2.1 software in terms of being less detectable, both the adaptive algorithms (Hide v2.1 and the PQ system) were shown to be much less detectable than the non-adaptive LSB hiding method. For that reason, steganographers can expect less probability of detection when embedding messages using algorithms that selectively choose unique areas of cover images such as the two adaptive techniques discussed here.

4.4.5 Avoid simple non-adaptive LSB substitution systems

Another interesting finding of this research is the high vulnerability of detection for many of the commonly available and downloadable spatial image hiding techniques. The generic LSB hiding method explained in Chapter III is created to encompass the statistical effects of hiding messages using many of the widely used non-adaptive hiding algorithms. For example, S-Tools, WNStorm, WbStego, Hide 4PGP, and many others all hide data in the least significant bit of pixels either randomly or using sequential pixels. While some attempt to maintain first order image statistics, all of them are vulnerable to the features used in this study. The RS-Statistic, derived from RS-Analysis, can detect many of these hiding tools by itself. Further, given the detection results and accuracy from performing steganalysis of the generic LSB hiding technique, it would be wise for future steganographers to avoid using such hiding systems. Even for extremely low

message payloads of 5% or 10%, detection rates are still very accurate when compared to a random guess.

4.4.6 A tradeoff exists between secret message length and security

The secret message payload, or the length of a secret message, not surprisingly is found to have a main effect on the detection accuracy of the PQ system. However, the apparent increase in detection of the system as the message payload increases is much smaller than is the increase in detection reliability of the other two hiding techniques tested as the message payload increases. In conclusion to the topic of secret message payload, it is a generally known and now a proven trend that as the secret message length increases so does the probability of the message being detected. Therefore, steganographers must deal with this tradeoff of secret message size and probability of detection. For moderate to large payloads, a steganographer will lessen their risk of detection by hiding their message via the perturbed quantization algorithm.

4.5 Summary

This chapter presents the results from the experiments in this study as well as a numerical analysis of the resulting data. Finally, meaningful information is drawn from the results of the study and presented as a list of advice for steganographers wishing to communicate more covertly. The entire study is summarized and future work in the field is suggested in the next and final chapter.

V. Conclusions

5.1 Summary

The art and science behind steganography carries with it a spy versus spy mentality. For every innovative hiding technique introduced by researchers into the public domain, it seems some other work counters with an attack which can defeat the steganographic system. As a result, there exists a need amongst steganographers for a secure hiding technique which can continue to be undetectable well into the future.

This research effort further explores perturbed quantization steganography by applying its theory into the spatial image domain. One of the advantages to working in the spatial domain is the numerous lossy image transformations available to be used in conjunction with the PQ algorithm. The two operations at the heart of this study are the color to grayscale conversion and image downsampling. Interestingly the main effect these operations have on the PQ system is in regards to the steganographic capacity of an image. For example, the standard weighted grayscale function provides extremely low steganographic capacities while the desaturate function allows for nearly 50% secret message payloads. Additionally, the desaturate function is an ideal operation because each of the changeable pixels prior to rounding are at exactly ½. Finally, the steganographic capacities of the two interpolation techniques used with color images are exactly as expected, 20% for ϵ=0.1, and 10% for ϵ=0.05.

A statistical attack is also introduced in Chapter III which attempts to detect whether a color image contains hidden data or not. This attack calculates the probability

density for the number of pixels in an image which contain all of the possible 26 neighbors in the image. A pilot study reveals that there is a slight difference in this density between stego and clean images. Along with this image feature, statistics computed from RS Analysis as well as the histogram characteristic function center of mass form a feature set used to perform pattern classification.

Classification is done on stego-images and clean images using each of three hiding methods: the PQ system, a generic LSB substitution system, and the Hide v2.1 steganographic software. Results of testing these systems reveals that the perturbed quantization system, regardless of the information reducing process, performs much stealthier than either of the other systems tested. In fact, detection reliability for message payloads up to 40% with grayscale images is still not much better than guessing at random. Performance analysis also reveals that adaptive algorithms such as the Hide v2.1 software and the PQ system are much harder to detect reliably than are algorithms which select pixels independently of the cover image. Finally, performance analysis reveals that the epsilon value used within the selection rule of the PQ system does have an effect on the security of the system. It is proven in a small pilot study that lowering the epsilon value, which decreases the amount of rounding error that occurs in the system, results in a lower detectability score. Therefore, it can be concluded that the perturbed quantization system is not accurately detected by the state-of-the-art in steganalytic techniques, and the system offers steganographers the ability to vary the information reducing process used within the system.

5.2 Future Research

With concerns about terrorists using steganography, research will continue in the field for some time to come. Additionally, much of this research will be focused on digital images.

5.2.1 A Large Image Database

In order to maintain a consistency amongst various research efforts, there is a need for a large database of images which can be used by all researchers as their workload for testing. Presently, research efforts use different images originating from different locations, and this makes it difficult to accurately compare performance between different studies involving differing sets of images. This image database needs to contain a wide variety of images coming from a variety of digital sources. The database should contain images originating from several different types of digital cameras as well as scanned images. Some photographs should contain people, animals, and objects, while others should be taken of nature. The database needs to contain as much variety as possible in order to represent all types of images which may be encountered on the Internet. Once a large database is created, future research in steganography can maintain a consistency across the workloads used for testing.

5.2.2 The Neighborhood Attack

This research effort introduces a statistical attack which calculates the probability density of pixels in an image where all of its 26 neighbors are also present in the image. A very small study shows that there is a discrimination between clean and stego-images with this statistic. However, wide scale testing is not done with this steganalytic attack. Therefore, future research can test this statistic on a much bigger scale in order to study

the discriminability of this feature as well as looking into which spatial hiding techniques this attack can detect.

5.2.3 Enhancing the PQ System

Since perturbed quantization steganography has been shown to be stealthy in the spatial domain, future areas of research in PQ steganography should focus on maximizing its security and steganographic capacity. Therefore, the systems adjustable parameter, the information-reducing operation, can continue to be explored in order to implement an optimal system.

One of the limitations with the neighborhood attack described in Chapter III is its likely decrease in discriminability as the number of unique colors in an image increases. Similarly, the raw quick pairs method [FrD00] also suffers in performance as the relative number of unique colors in an image increases. This can be made into a more general statement that the more colors that exist in an image, the more difficult steganalysis will be in the spatial domain. Therefore, a steganographer will have already avoided several attacks simply by using images which contain large amounts of color.

Applying this concept to the PQ system, one can attempt to increase the stealthiness of the system by creating a lossy image transformation which maximizes the number of colors present in an image. The downsampling methods studied in this investigation do increase the number of colors in the image from the interpolation which occurs. Thus, future work can explore other downsampling methods which involve more advanced interpolation methods such as those implemented in ImageMagick [Ima04]. Further, a custom convolution filter poses an intriguing option for the PQ system. In such a system, an optimized convolution kernel would be constructed in which the filtered

image contains a maximum amount of color, while minimizing perceptible distortion of the image. Finally, an information-reducing process can be defined such that it maximizes the steganographic capacity of the PQ system. More specifically, after applying a lossy transformation to an image, the fractional part of *every* un-rounded pixel should be close to 1/2.

Appendix A. ROC Curves

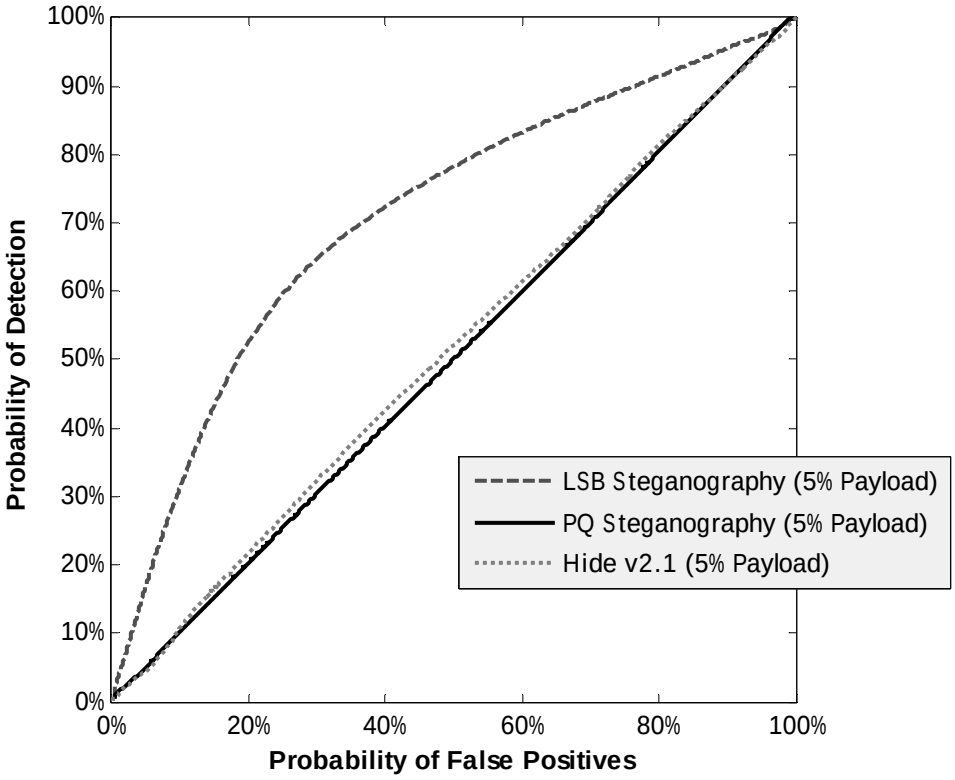

Figure 21. ROC Curves from the Classification of Stego-Images for All Three Systems Using Desaturated Grayscale Images

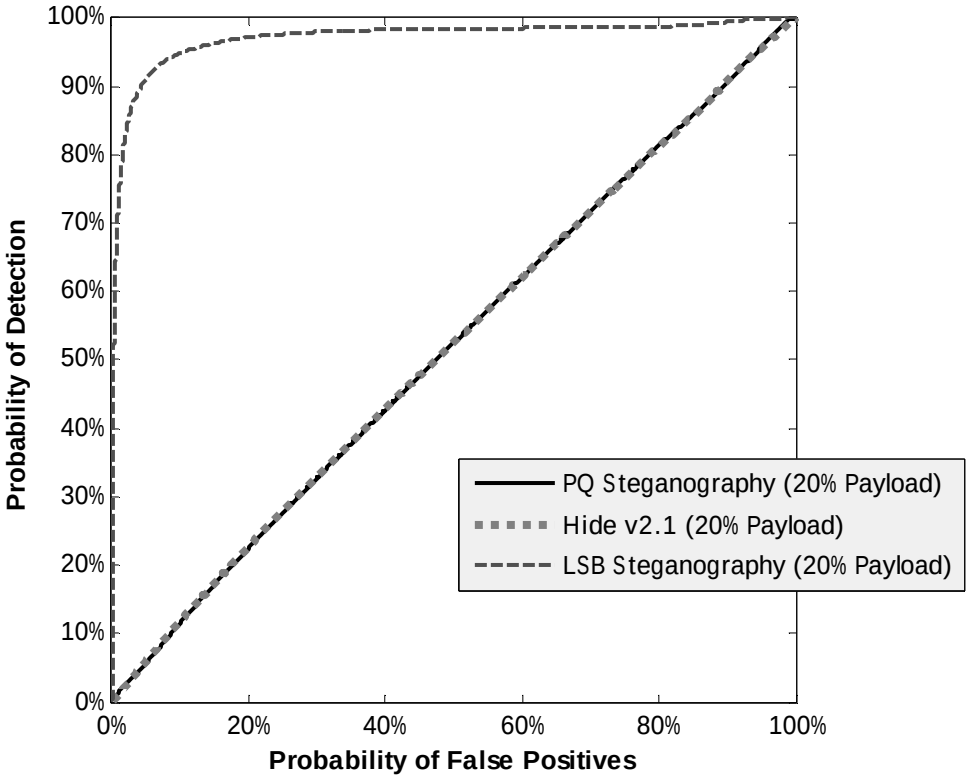

Figure 22. ROC Curves from the Classification of Stego-Images for All Three Systems Using Desaturated Grayscale Images

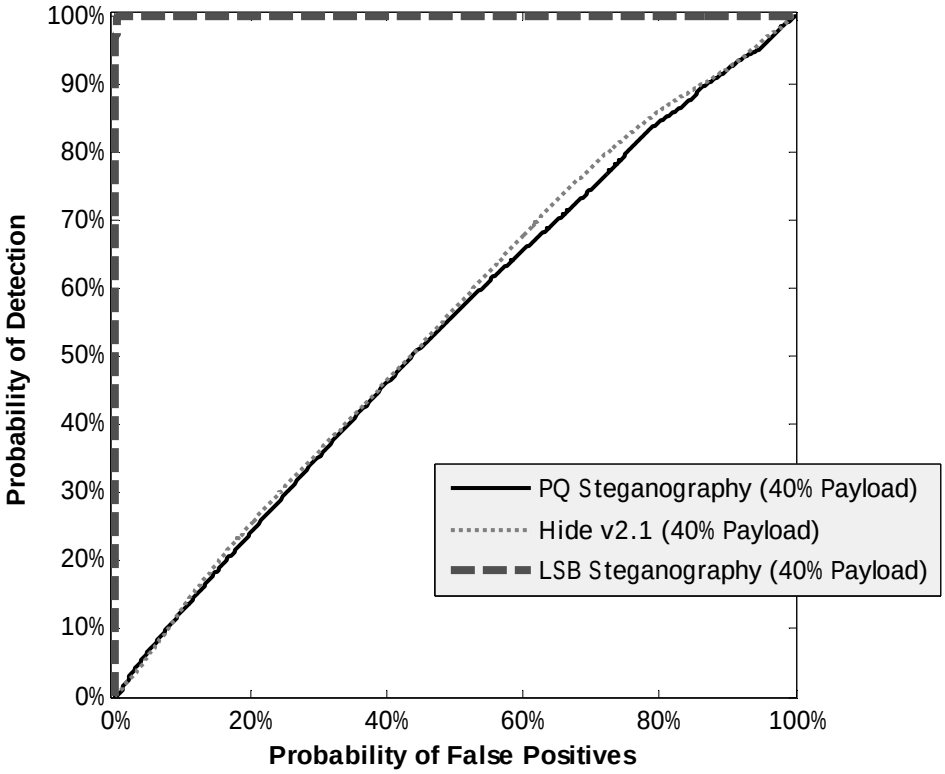

Figure 23. ROC Curves from the Classification of Stego-Images for All Three Systems Using Desaturated Grayscale Images

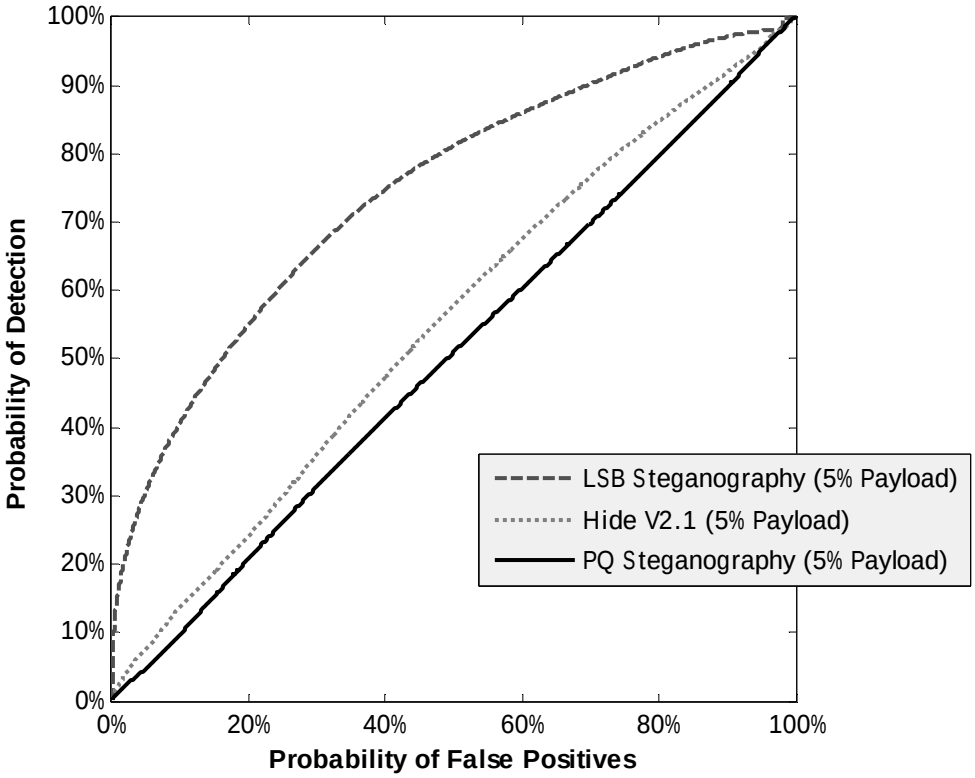

Figure 24. ROC Curves from the Classification of Downsampled Stego-Images via Bicubic Interpolation for All Three Systems

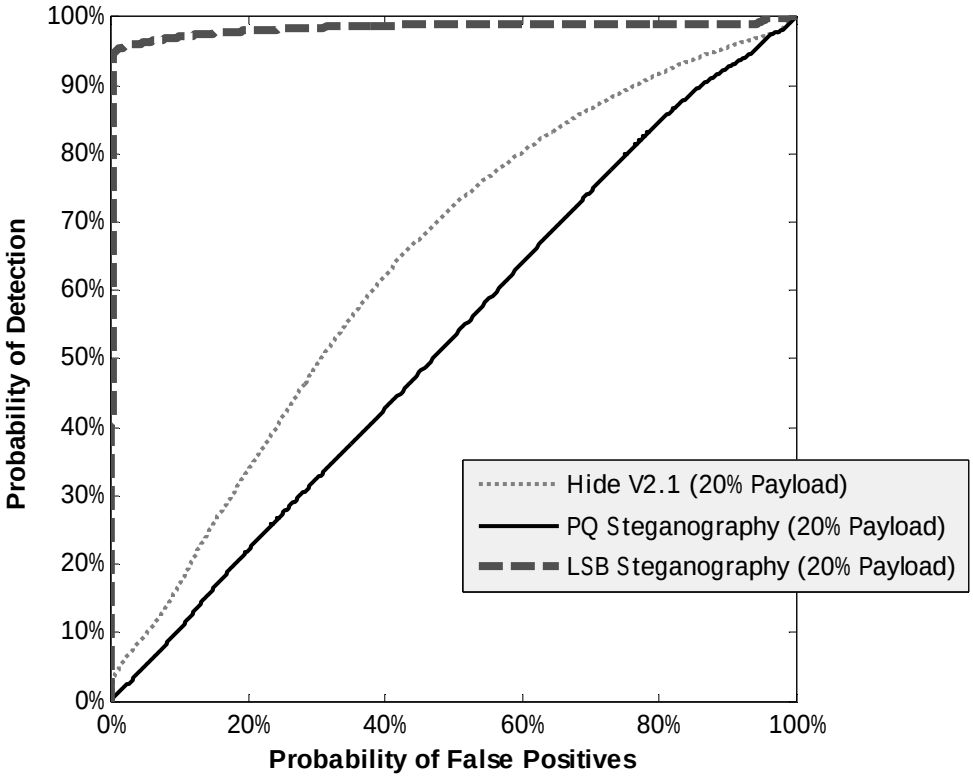

Figure 25. ROC Curves from the Classification of Downsampled Stego-Images via Bicubic Interpolation for All Three Systems

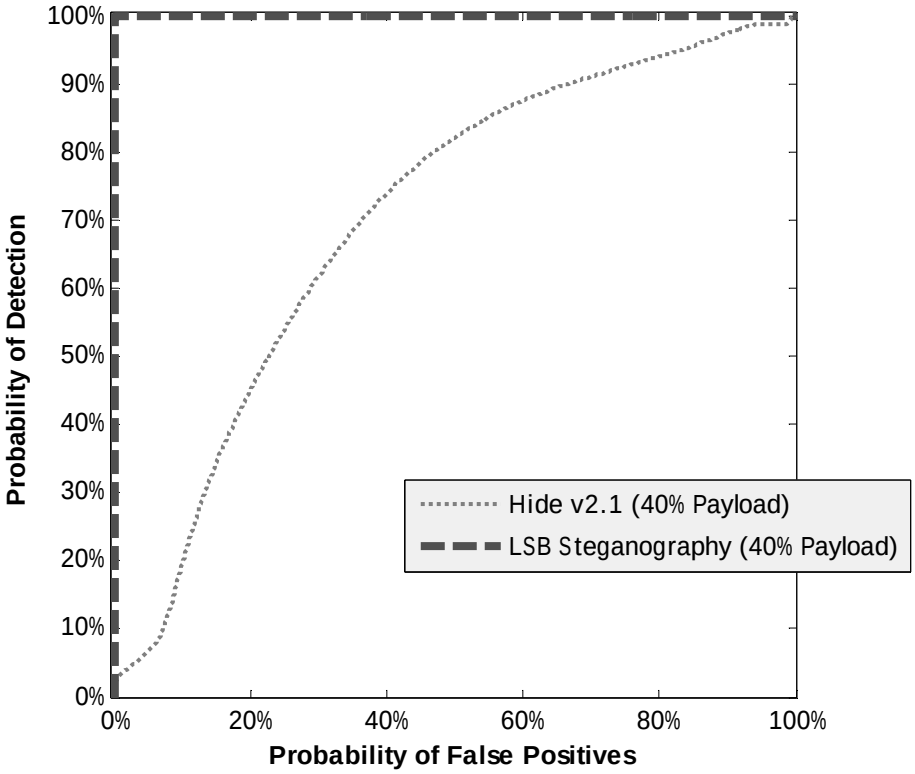

Figure 26. ROC Curves from the Classification of Downsampled Stego-Images via Bicubic Interpolation for Two Systems

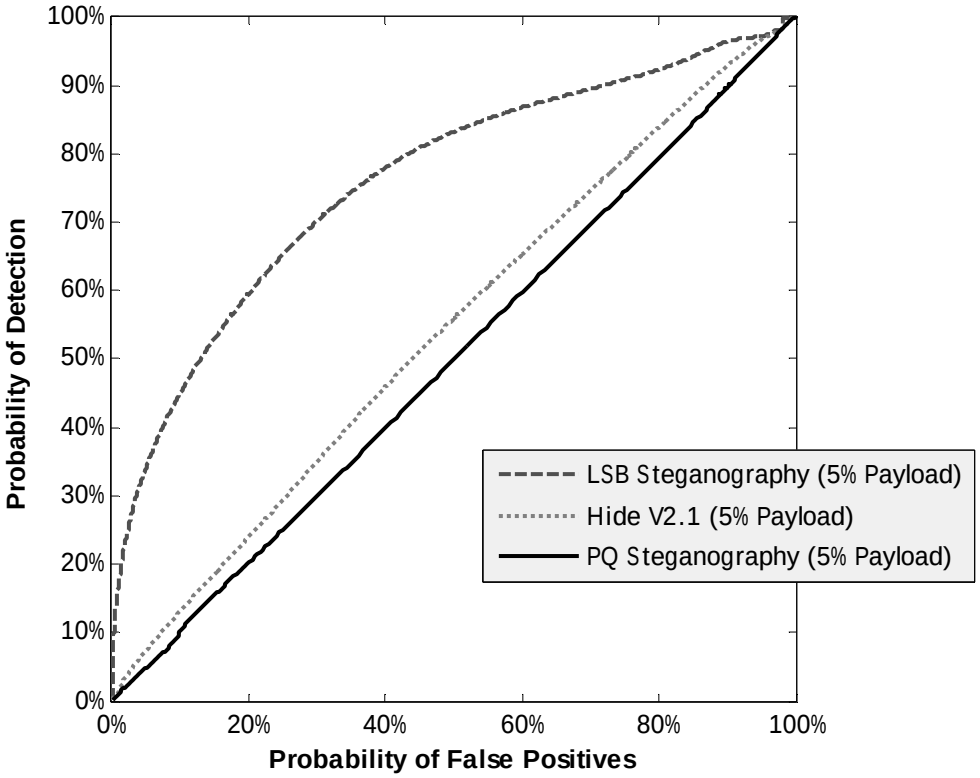

Figure 27. ROC Curves from the Classification of Downsampled Stego-Images via Bilinear Interpolation for All Three Systems

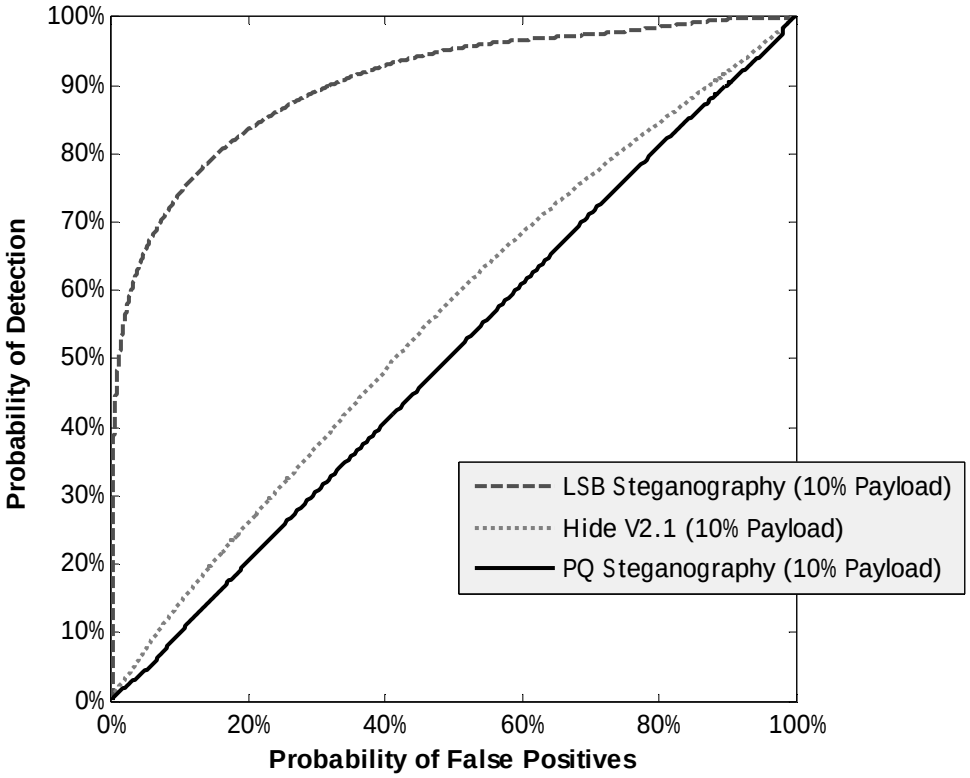

Figure 28. ROC Curves from the Classification of Downsampled Stego-Images via Bilinear Interpolation for All Three Systems

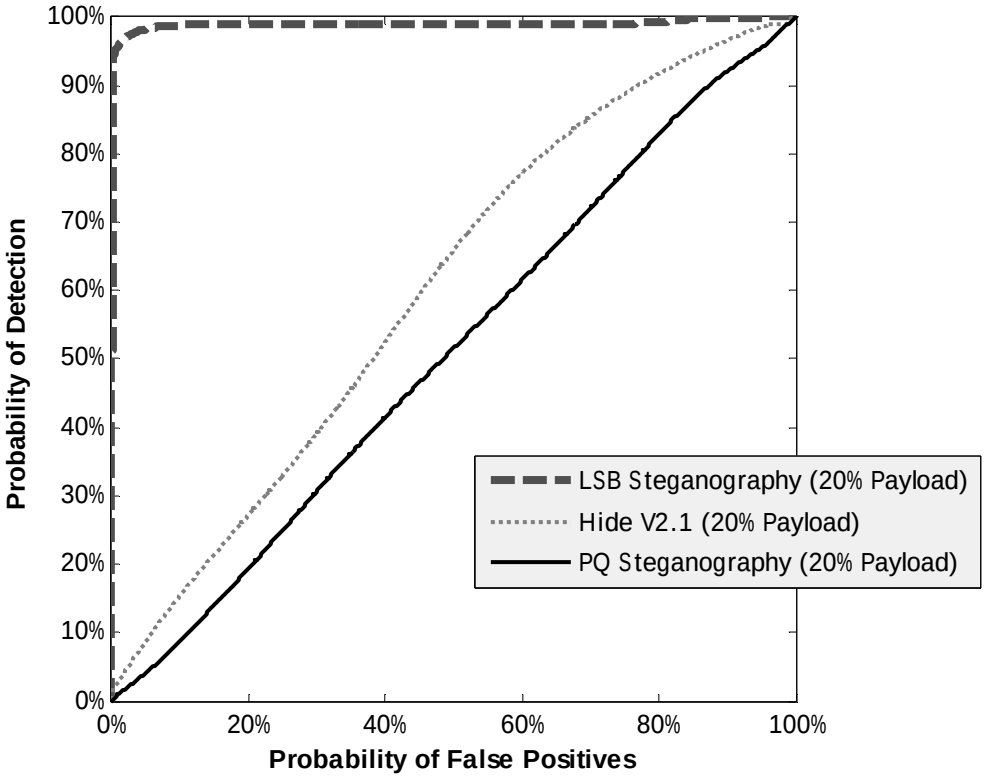

Figure 29. ROC Curves from the Classification of Downsampled Stego-Images via Bilinear Interpolation for All Three Systems

Appendix B. ANOVA Tables

Table 27. The Factors of Message Payload and Message Content within the PQ System – Downsampling with Bilinear Interpolation

Source	Sum of Squares	Degrees of Freedom	Mean Square	F-Value	P-Value
Message Content	0.00001	2	0.00000	0.04	**0.9605**
Message Payload	0.06180	2	0.03090	406.22	**0**
Content * Payload	0.00119	4	0.00030	3.92	**0.0049**
Errors	0.00959	126	0.00008		
Total	0.07259	134			

Table 28. A System Comparison Study (Hide v2.1 vs. PQ) for Downsampled Stego-Images via Bilinear Interpolation

Source	Sum of Squares	Degrees of Freedom	Mean Square	F-Value	P-Value
System	0.76869	1	0.76869	3368.58	**0**
Message Payload	0.38750	2	0.19375	849.06	**0**
Message Content	0.00000	2	0.00000	0.0000	**0.9971**
System * Payload	0.08083	2	0.04041	177.71	**0**
System * Content	0.00002	2	0.00001	0.04	**0.9577**
Payload * Content	0.00301	4	0.00075	3.3	**0.0117**
Errors	0.05842	256	0.00023		
Total	1.29847	269			

References

[AnM00] R. Ansari, N. Memon. "The JPEG Standard," *Handbook of Image and Video Processing*, Academic Press, 2000.

[AnP98] R. Anderson, F. Peticolas. "On the Limits of Steganography," *IEEE Journal of Selected Areas of Communications*, 16(4), pp. 474-481, May 1998.

[Bai04] C. Bair. "Grayscale vs. Desaturate for Black and White Printing," <http://www.inkjetart.com/tips/grayscale/>, 2004.

[BeG96] W. Bender, D. Gruhl, N. Morimoto, A. Lu. "Techniques for Data Hiding," *IBM Systems Journal*, Vol. 35, No. 3 & 4, pp. 331-336, 1996.

[Bis95] C. Bishop. "Neural Networks for Pattern Recognition," Oxford University Press, 1995.

[Bro96] A. Brown. S-Tools, Version 4.0. Computer Software. 1996.

[Cac04] C. Cachin. "An Information-Theoretic Model for Steganography," *Information and Computation*, 192(1), pp. 41-56, July 2004.

[Far01] H. Farid. "Detecting Steganographic Messages in Digital Images," *Technical Report TR2001-412*, Dartmouth College, Computer Science, 2001.

[FaL03] H. Farid, S. Lyu. "Higher-Order Wavelet Statistics and Their Application to Digital Forensics," *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, Wisconsin, June 2003.

[Fra03] E. Franz. "Steganography Preserving Statistical Properties," *In Proceedings of 5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, October 2002.

[FrD00] J. Fridrich, R. Du, M. Long. "Steganalysis of LSB Encoding in Color Images," *In Proceedings of ICME 2000*, New York City, New York, 2000.

[FrG01] J. Fridrich, M. Goljan, R. Du. "Reliable Detection of LSB Steganography in Grayscale and Color Images," In *Proceedings of the 2001 ACM Workshop on Multimedia and Security*, pp. 27-30, Ottawa, Canada, October 5, 2001.

[FrG02] J. Fridrich, M. Goljan. "Practical Steganalysis – State of the Art," *In Proceedings of SPIE Photonics West, Vol. 4675, Electronic Imaging 2002, Security and Watermarking of Multimedia Contents*, San Jose, CA, January, 2002, pp. 1-13.

[FrG04] J. Fridrich, M. Goljan, D. Soukal. “Perturbed Quantization Steganography with Wet Paper Codes,” *Proceedings of the 2004 ACM Multimedia & Security Workshop*, Magdeburg, Germany, September 20-21, 2004.

[FrG04b] J. Fridrich, M. Goljan, D. Soukal. “Searching for the Stego-Key,” *In Proceedings of SPIE Electronic Imaging*, San Jose, January 2004.

[FrG05] J. Fridrich, M. Goljan, P. Lisonek, D. Soukal. “Writing on Wet Paper,” *Submitted to IEEE Transactions on Signal Processing Supplement on Secure Media*, 2004.

[GoW02] R. Gonzalez, R. Woods. “Digital Image Processing [Second Edition],” Prentice Hall, 2002.

[Bun00] C. Bunks. “Grokking the Gimp,” Pearson Educations, February 2000.

[HaP03] J. Harmsen, W. Pearlman. “Steganalysis of Additive Noise Modelable Information Hiding,” *In Proceedings of SPIE Electronic Imaging*, Santa Clara, January 2003.

[Hof02] G. Hoffman. “Windowed Sinc Interpolation,” <http://www.fho-emden.de/~hoffmann/lanczos07112002.pdf>,2002.

[HoS04] M. Hogan, G. Silvestre, N. Hurley. “Performance Evaluation of Blind Steganalysis Classifiers,” *In Proceedings of SPIE-IS&T Electronic Imaging: Security, Steganography, & Watermarking of Multimedia Contents VI*, Vol. 5306, pp. 58-69, San Jose, 2004.

[Ima04] ImageMagick 6.1.9. Open Source Computer Software. 2004.

[Jac03] J. Jackson. “Targeting Covert Messages: A Unique Approach For Detecting Novel Steganography”, MS Thesis, Air Force Institute of Technology, Wright Patterson Air Force Base, Ohio, 2003.

[KaP00] S. Katzenbeisser, F. Peticolas. “Information Hiding Techniques for Steganography and Digital Watermarking,” Artech House, 2000.

[KaP02] S. Katzenbeisser, F. Peticolas. “Defining Security in Steganographic Systems,” *In Proceedings of SPIE Photonics West, Vol. 4675, Electronic Imaging 2002, Security and Watermarking of Multimedia Contents*, San Jose, California, January, 2002, pp. 260-268.

[Ker83] A. Kerckhoffs. “La Cryptographie Militaire,” *Journal des sciences militaires,* vol. IX, pp. 5–38, Janvier 1883, pp. 161–191, Février 1883.

[Ker04] A. Ker. "Quantitative Evaluation of Pairs and RS Steganalysis," *In Proceedings of SPIE-IS&T Electronic Imaging: Security, Steganography, & Watermarking of Multimedia Contents VI*, Vol. 5306, pp. 83-97, San Jose, 2004.

[KuM92] C. Kurak, J. McHugh. "A Cautionary Note on Image Downgrading," *In Proceedings of the 8th Computer Security Applications Conference*, 1992.

[LiF02] S. Lyu, H. Farid. "Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines," *In Proceedings of 5th International Workshop on Information Hiding*, pp. 340-354, Noordwijkerhout, The Netherlands, Springer-Verlang, October 2002.

[Mat04] Mathworks. Matlab 7, Release 14. Computer Software. 2004.

[Mar02] D. Martindale. <davem@cs.ubc.ca> (Oct 2002). S-Spline or Lanczos. <http://www.binbooks.com/books/photo/i/l/57186AE7E6&orig=1> .

[McB03] B. McBride. "A Hyper-Geometric Data Classifier For Blind Detection of Novel Steganography," MS Thesis, Air Force Institute of Technology, Wright Patterson Air Force Base, Ohio, 2003.

[PrH01] N. Provos, P. Honeyman. "Detecting Steganographic Content on the Internet," *CITI Technical Report 01-11*, 2001.

[Pro01] N. Provos. "Defending Against Statistical Steganalysis," *In Proceedings of the 10th USENIX Security Symposium*, Washington DC, 2001.

[Sha49] C. Shannon. "Communication Theory of Secrecy Systems," *The Bell Labs Technical Journal*, pp. 656--715, vol. 28, No 4, May 1949.

[Sha01] T. Sharp. "An Implementation of Key-Based Digital Signal Steganography," *In Proceedings of the 4th International Workshop on Information Hiding*, LNCS 2137, Springer-Verlag, Pittsburgh PA, 2001, pp. 13-26.

[Ste04] Steganos. Steganos Security Suite 7. Computer Software. 2004.

[ToT04] M. Topkara, U. Topkara, M. Atallah, C. Taskiran, E. Lin, E. Delp. **"**A Hierarchical Protocol for Increasing the Stealthiness of Steganographic Methods," *Proceedings of the 2004 ACM Multimedia & Security Workshop*, Magdeburg, Germany, September 20-21, 2004.

[Uph97] D. Upham. "Jpeg-Jsteg (Version 4)" Computer Software, ftp://ftp.funet.fi/pub/crypt/steganography.

[Way02] P. Wayner. “Disappearing Cryptography: Information Hiding: Steganography & Watermarking,” Morgan Kauffman Publishers, 2002.

[WeP99] A. Westfeld, A. Pfitzmann. “Attacks on Steganographic Systems,” *In Proceedings of the 3rd International Workshop on Information Hiding,* Springer-Verlag, London UK, 1999.

[Wes02] A. Westfeld. “Detecting Low Embedding Rates,” *Revised Papers from the 5th International Workshop on Information Hiding*, Springer-Verlag, London UK, pp.334-349,Oct 7-9, 2002.

[Wol04] R. Wolfgang. “JPEG Tutorial,” Society for Imaging Science & Technology, http://www.imaging.org/resources/jpegtutorial/index.cfm.

[ZöF98] J. Zöllner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, G. Wolf. “Modeling the Security of Steganographic Systems,” *In Proceedings of the 2nd International Workshop on Information Hiding*, Portland, LNCS 1525, pp. 345-355, 1998.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
21-03-2005	**Master's Thesis**	June 2004 – March 2005

4. TITLE AND SUBTITLE

AN ANALYSIS OF PERTURBED QUANTIZATION STEGANOGRAPHY IN THE SPATIAL DOMAIN

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Matthew D. Spisak

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 641
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GIA/ENG/05-04

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/IFEC
Attn: Mr. Scott Adams
32 Brooks Rd.
Rome, NY 13441-4114 DSN: 587-1430

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Steganography is a form of secret communication in which a message is hidden into a harmless cover object, concealing the actual existence of the message. Due to the potential abuse by criminals and terrorists, much research has also gone into the field of steganalysis – the art of detecting and deciphering a hidden message. As many novel steganographic hiding algorithms become publicly known, researchers exploit these methods by finding statistical irregularities between clean digital images and images containing hidden data. This creates an on-going race between the two fields and requires constant countermeasures on the part of steganographers in order to maintain truly covert communication.

This research effort extends upon previous work in perturbed quantization (PQ) steganography [FrG04] by examining its applicability to the spatial domain. Several different information-reducing transformations are implemented along with the PQ system to study their effect on the security of the system as well as their effect on the steganographic capacity of the system. Additionally, a new statistical attack is formulated for detecting +/- 1 embedding techniques in color images. Results from performing state-of-the-art steganalysis reveal that the system is less detectable than comparable hiding methods. Grayscale images embedded with message payloads of 0.4bpp are detected only 9% more accurately than by random guessing, and color images embedded with payloads of 0.2bpp are successfully detected only 6% more reliably than by random guessing.

15. SUBJECT TERMS
Feature extraction, image processing, information theory, pattern recognition, pixels, quantization, security, steganography

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
REPORT	**ABSTRACT**	**c. THIS PAGE**			Dr. Richard Raines, ENG
U	**U**	**U**	**UU**	129	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4278; e-mail: Richard.Raines@afit.edu