Locating Secret Messages in Images

Ian Davidson
Computer Science, SUNY Albany
1400 Washington Avenue
Albany, NY 12222, USA
davidson@cs.albany.edu

Goutam Paul
Computer Science, SUNY Albany
1400 Washington Avenue
Albany, NY 12222, USA
goutam@cs.albany.edu

ABSTRACT

Steganography involves hiding messages in innocuous media such as images, while steganalysis is the field of detecting these secret messages. The ultimate goal of steganalysis is two-fold: making a binary classification of a file as stego-bearing or innocent, and secondly, locating the hidden message with an aim to extracting, sterilizing or manipulating it. Almost all steganalysis approaches (known as attacks) focus on the first of these two issues. In this paper, we explore the difficult related problem: given that we know an image file contains steganography, locate which pixels contain the message. We treat the hidden message location problem as outlier detection using probability/energy measures of images motivated by the image restoration community. Pixels contributing the most to the energy calculations of an image are deemed outliers. Typically, of the top third of one percent of most energized pixels (outliers), we find that 87% are stego-bearing in color images and 61% in grayscale images. In all image types only 1% of all pixels are stego-bearing indicating our techniques provides a substantial lift over random guessing.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology – classifier design and evaluation.

General Terms

Algorithms, Experimentation, Security.

Keywords

Steganography, Steganalysis, Outlier Detection.

1. INTRODUCTION AND MOTIVATION

The term *steganography* literally means “covered writing” and involves transmission of secret messages through apparently innocent files without detection of the fact that a message was sent [2]. The innocuous file is known as the *cover* (or *innocent* or *clean*) medium, while the file containing the hidden-message is referred to as the *stego* (or *infected*) medium. There are many tools available [1] that can hide messages in images, audio and video files, and steganography is now in common use [2]. Recent terrorist activity has been tentatively linked to the use of steganography [3] and is seen by various agencies as a growing method of sending covert information [4]. Whereas cryptography was the preferred secret-message-sending tool of the past, relying on complex ciphers to prevent identification of the message, the huge bandwidth of the Internet now offers an alternate and complementary approach. Steganography allows hiding messages innocuously amongst the vast content of Internet sites: for instance, an image containing a hidden message maybe posted to a website (eBay is often cited [5]) where others can download the image and recover the message with the appropriate password.

The process of detecting steganographic messages is known as *steganalysis* and a particular steganalysis technique is called an *attack*. If the image type is carefully chosen (as shown in Figure 1 and Figure 2) then visual detection is difficult. The current state of the art involves identifying a particular signature associated with a particular steganographic technique and devising a statistical test to identify this signature. Such handcrafted approaches are very useful but suffer from a high false positive rate and are vulnerable to steganographic approaches that hide messages in such a way as to reinstate an expected property [6].

Steganalysis can be viewed as a two-stage process: 1) Classification of an image as being stego-bearing or not, and 2) Finding the location of stego-bearing pixels (i.e. the pixels containing the hidden message bits) with an aim to extracting, manipulating or sterilizing the message. There has been considerable work on the first stage using statistical techniques to identify a *particular* steganographic technique [7]. Other work [8], including our own [9] have explored using pattern recognition algorithms to automatically create attacks. However, identifying that an image contains a hidden message written using a specific technique does not enable us to locate the message. This paper explores the second stage: given an image is believed to contain a secret message, identify where the message is hidden. We treat this problem as outlier detection. To our knowledge no approach to steganalysis has tried to locate the actual hidden message or treated it as outlier detection.

2. A QUICK INTRODUCTION TO STEGANOGRAPHY TECHNIQUES

Steganography is most widely used in images, and hence in this paper we focus on detecting hidden messages in this media type. The raster data of an image is generally stored either by using an indexing scheme (as in GIF) or by transforming (smoothing) the raster data to make it more suitable for compression (as in JPEG). The GIF image format stores a palette of colors or grayscales used in the image with each pixel entry being an index to a palette entry. While the message to hide may be text, image, etc., in digital steganography, it is ultimately represented as bits. The hidden message may be compressed or even encrypted before it is hidden, to reduce the amount of information or hide its content. A very common technique is to embed the hidden message by altering the least significant bits (LSBs) of the index entries.

KDD'04, August 22-25, 2004, Seattle, Washington, USA.
Copyright 2004 ACM 1-58113-888-1/04/0008...$5.00.

Consider hiding the letter “A” in a GIF image. The letter is translated into the binary representation of its ASCII value. A pseudo random number generator (PRNG) can be used to determine the location of the eight carrier pixels (one for each message bit). The seed of the PRNG is the password to recover the message. For each chosen location, the existing LSB of the index is examined and flipped if required. For JPEG images, a similar approach is used to flip the LSB of the discrete cosine transformation (DCT) coefficients [2]. Earlier work found identifying GIF files as stego-bearing using automated techniques to be quite challenging [9] due to wide variety of image types.

3. PREVIOUS AND RELATED WORK

Our previous work [9] and the work of others [8] showed the feasibility of applying classic learning and mining techniques to differentiate between stego and clean media. These are some of the earliest work to automatically create blind steganography attacks (i.e. attacks without detailed knowledge of the underlying steganography algorithm). The remaining steganalysis techniques consist of manually crafted measures made in full knowledge of the steganography algorithm’s working.

The first stage of steganalysis (i.e. classification) has received considerable attention and the results are promising and constantly improving. The approaches of blind steganalysis can have as low as 20% false positives [10]. Though hand crafted techniques can break some steganography all the time, since they aim to identify the presence or absence of a signature, they cannot identify the message location. For instance, many steganalysis approaches use a Chi-square test of independence [10] which gives no indication where the message is hidden. Instead, we will treat identifying the message location problem as outlier detection.

4. AN OVERVIEW OF OUTLIER DETECTION

Outlier detection is often cited as a fundamental task in data mining along with classification, associations, clustering and regression [11]. The purpose of outlier detection (also known as anomaly detection) is given a collection of N data points, specify a subset of those points as “not belonging” to the collection and label them as outliers. Outlier detection is used in many data mining applications such as credit card fraud detection and network intrusion detection. Aggarwal et al [12] provides a survey of outlier detection techniques and applications.

There exist two primary forms of outlier detection [13]: distance based and distribution based. With distance based outlier detection, a model of the data is formed (e.g. a clustering model) and an instance’s degree of anomaliness is determined (e.g. its Euclidean distance to the closest cluster center). The model defines normality and the outliers tend to deviate from this normality (e.g. tend to lie between the cluster centers). Distribution based outlier detection involves creating a parametric model of the data and then re-calculating the model parameters with each data point removed one at a time. Essentially N additional sets of parameters are calculated from all possible sets of N-1 points. Note that typically only the model parameters are recalculated, the mining task is not repeated. A data point’s degree of anomaliness is determined by the magnitude of the change in parameters when it is removed.

There are many successful outlier detection techniques available in the field of data mining [14], which require defining normality and some measure of deviation from it. However, the wide variety of image content prevents a ubiquitous definition of normality. Though there exists spatial outlier detection techniques [15], few are readily applicable to image data. Perhaps most importantly, if steganalysis is to be used to scan a large number of images, then a technique whose complexity is linear with respect to the number of pixels is required. We borrow ideas from image restoration to develop our own approach whose complexity is linear in the number of pixels in an image.

5. STEGANALYSIS AS OUTLIER DETECTION

To perform outlier detection in an image, we need to build a model (i.e. a probability distribution over all possible pixel intensities in the image) and measure outliers with respect to that model. Due to the wide variety of image content there is no useful parametric model for even a small subset of images. Therefore, we build a simple non-parametric model for each image. More complicated models could be built if needed.

Examples of such models include the “gradient” of a pixel. Many adjacent high gradient pixels can be identified as an edge [16]. Similarly, a simple vector model of each pixel can identify spatially interesting pixels [17]. Though these models are useful for other applications of image mining, we instead use an Ising model [18] that is popular in image restoration approaches to remove irregularities such as creases and lines in an image [16]. The general idea behind our approach is that if an image has an embedded message then it will be removed when the image is restored. Keeping track of which pixels are restored allows identifying the anomalous stego-bearing pixels.

Consider an image composed of n pixels arranged into a rectangular lattice where each pixel i, i=1,2,…,n, takes one of k possible values from a discrete set C={1,2,…,k}. We define two random variables associated with each pixel i: the observed value X_i and the true value Z_i. We assume that for the most part X_i=Z_i. However, hiding a message changes several X values which visually disturb the image. The goal of image restoration is to maximize $P(Z_1 \ldots Z_n \mid X_1 \ldots X_n)$ which effectively removes the visual disturbances. We can view this in a Bayesian perspective if we consider the Z values to be the model parameters. Algorithms such as the Gibbs sampler can maximize this probability, that is, they find the posterior mode. The Gibbs sampler has various forms [19], the simplest is to sweep through every pixel and conditionally (on the neighbors’ values/states) update its value/state to increase the probabilities. This is repeated until convergence (i.e. no change in probabilities) occurs.

The expression $P(Z_1 \ldots Z_n \mid X_1 \ldots X_n)$ can be simplified by assuming that the Z values are independent of each other and that only the immediate neighbors of the i^{th} pixel effect the corresponding Z value. That is:

$$P(Z_1 \ldots Z_n \mid X_1 \ldots X_n) = \Pi P(Z_i \mid X_1 \ldots X_n) \ = \Pi P(Z_i \mid NBD(X_i)\,)$$

where $NBD(X_i)$ denotes the set of pixels in the immediate neighborhood of X_i.

The form of this probability distribution varies depending on the application and image type. In our experiments, we use a non-parametric distribution as follows:

$$P\left(Z_i \middle| NBD(X_i)\right) = 1 - E_N(X_i) \qquad (1)$$

where $E_N(X_i)$ is the normalized value (between 0 and 1) of the energy $E(X_i)$ defined below. Our results (described later) show that the stego images contain more energy than their cover counterparts and the stego-bearing pixels are the most energized pixels. This can be interpreted as that the stego images are less probable than their cover counterparts and that the stego pixels within an image are less probable. This is to be expected as the stego pixels are artificially embedded in the image. An analogy in physics is that a system of particles tends to be in its lowest energy configuration and if we disturb this configuration, we inherently increase the energy of the system. The energy of a pixel is defined by its state and the states of its immediate neighbors. For color images we use the function in equation (2).

$$E(X_i) = \sum_{X_j \in NBD(X_i)} \left| X_i - X_j \right| \qquad (2)$$

The energy of a color pixel is the average of the energies of its red, green, and blue components. Grayscale images only have an associated intensity value. In our experiments with grayscale images we measure the energy as follows:

$$E(X_i) = Entropy\left(\frac{\left| X_i - X_j \right|}{\sum_{X_k \in NBD(X_i)} \left| X_i - X_k \right|} \right) \qquad (3)$$

where the entropy is calculated over all $X_j \in NBD(X_i)$.

We have two different measures, as the effect of hiding messages in color and grayscale images are quite different. For color images, a small change in a palette index can create a large change in the corresponding color intensity triplet, as the adjacent indices in the palette do not normally map to similar colors. However, since in grayscale images, the palette entries are typically ordered in terms of their intensities, a small change in a palette index only slightly changes the pixel's intensity. Moreover, for some grayscale images, a small change in palette indices results in no change in pixel intensities if adjacent indices have the same intensities. We find that these functions are suitable for commonly used steganography techniques, but acknowledge that more complicated function may be used as new steganography techniques are created.

6. EXPERIMENTAL RESULTS

We use a color database consisting of 150 images of flowers, mountains and trees (50 of each type) and a grayscale database of 30 images that are predominantly old photographs of landscapes and people (downloaded from webshots.com and similar websites). We have purposely chosen natural images as they are conducive to hide messages in. The messages we hide are taken from the verses of Genesis from the Kings James Bible and are typically 400 bytes long each. As approximately half of these message bits require an LSB flip of the indices and our images were all 320x480 pixels, the proportion of stego-bearing pixels is typically 1% of all pixels. We focus on small messages because they are the most challenging to detect. We use the popular steganography tool Hide&Seek [1] that represents the typical LSB-based steganography approach as described earlier. We obtain comparable results with less sophisticated tools such as S-Tools, Third Eye etc [1]. Our work is a blind steganalysis approach using outlier detection since we assume no knowledge of the hiding algorithm. We use a neighborhood of distance 1 for calculating the energy of a pixel. We find in our work that only one iteration of the Gibbs update sufficiently identifies a large number of stego-bearing pixels.

6.1 The Distribution of Outlier Pixels

Figure 1 through Figure 4 show the location of the 500 top outliers out of 320x480 pixels (i.e. 0.33% of all pixels) for an innocent and a stego image in our database. For images with highly contrasting regions, we find that the naturally occurring outliers are located on the transition lines. For other images, the naturally occurring outliers are more uniformly distributed throughout the image. Identifying the pixels to restore can identify between 57 to 96 percent of all stego bearing pixels. Our approach does not assume that the stego tool randomly distributes the message throughout the image. As Figure 3 and Figure 4 show, our approach can detect hidden messages in quite busy images also.

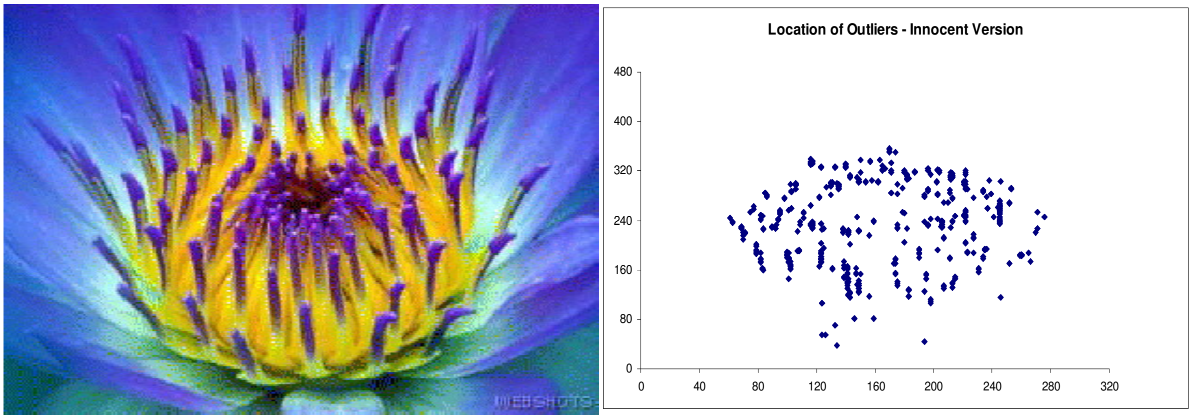

Figure 1 : An innocent image from the color database with the location of the top 500 (0.33%) outliers shown on the right. The average energy per pixel is 166.73.

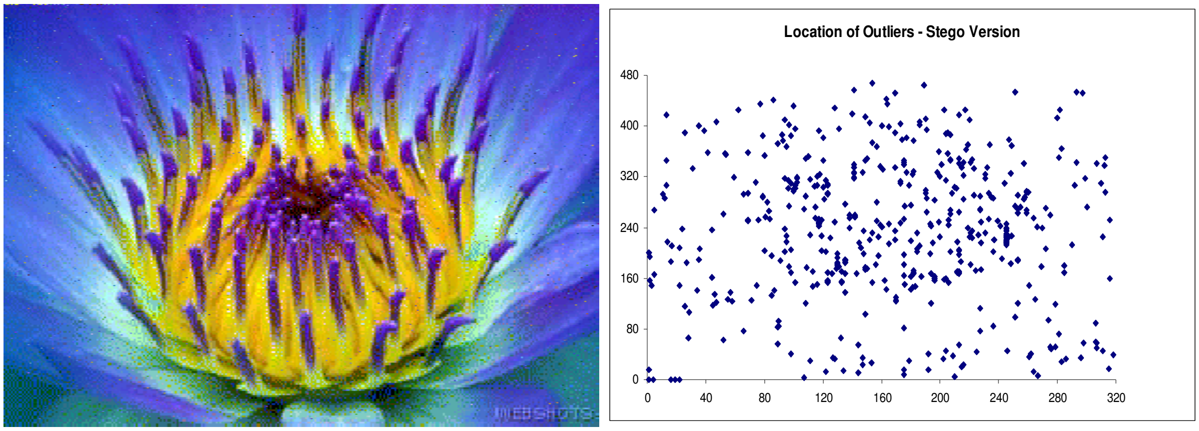

Figure 2 : The stego version of the above image with the location of the top 500 (0.33%) outliers shown on the right. Of these 500 pixels, 71% are stego-bearing pixels. The average energy per pixel is 171.66.

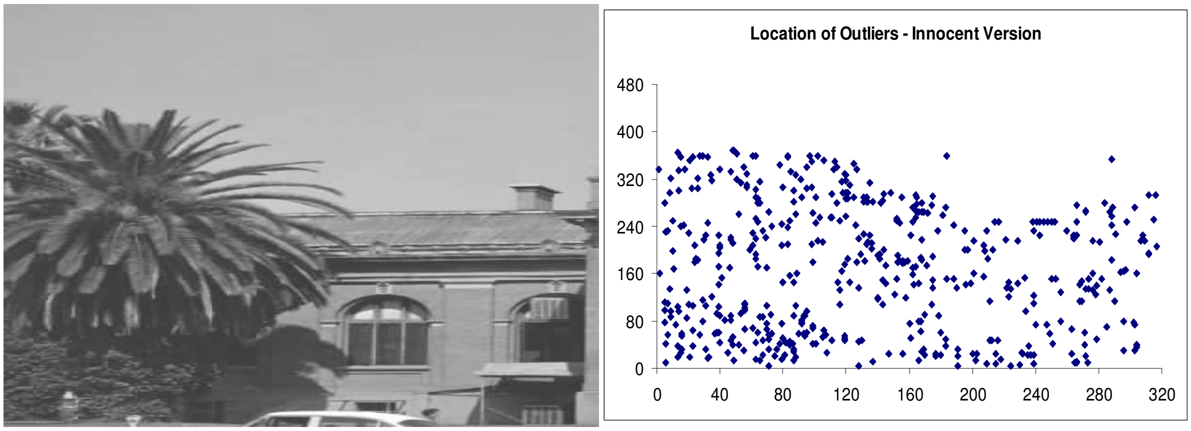

Figure 3: An innocent image from the grayscale database with the location of the top 500 (0.33%) outliers shown on the right. The average energy per pixel is 1.73.

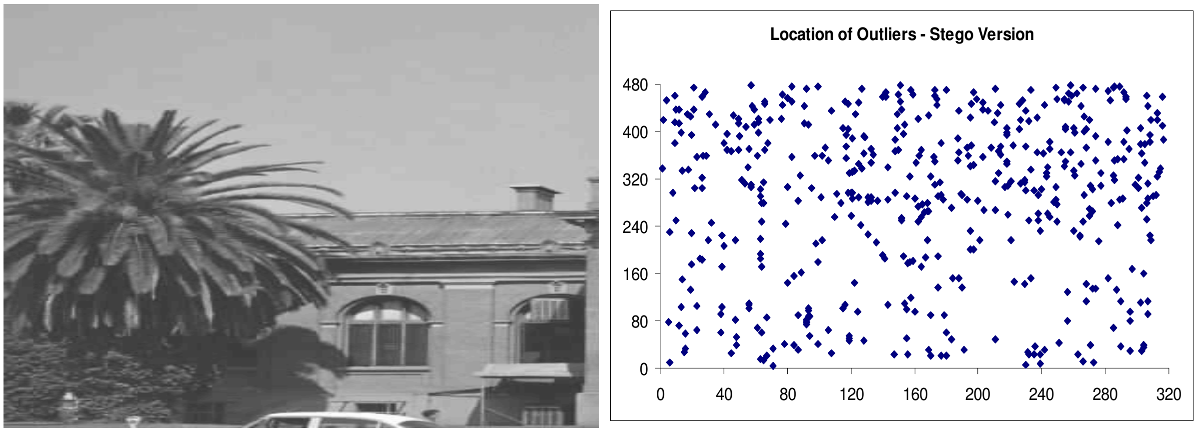

Figure 4: The stego version of the above image with the location of the top 500 (0.33%) outliers shown on the right. Of these 500 pixels, 57% are stego-bearing pixels. The average energy per pixel is 1.74.

6.2 Detecting the Outlier Pixels

We calculate the energy of each pixel and sort the pixels in decreasing order of energy. We take some percentage of pixels at the top of the list and label these as outliers. We determine what pixels are stego-bearing by identifying which pixels differ between the stego and cover image. This is only required to measure the accuracy of our approach. In practice, the cover image is not needed to locate the stego-bearing pixels. The success of our approach is measured by what proportion of the pixels identified as outliers are indeed stego-bearing (precision) and the number of these correctly identified outliers as a proportion of the total number of stego-bearing pixels (recall). As precision increases recall decreases and vice-versa.

6.2.1 *Color Images*

Table 1 shows that the stego images have an average energy increase of 8.5% over their innocent counterparts. Furthermore, we found that every stego image has a higher energy than its innocent counterpart. The standard deviation of the energy of the images is great due to the variation in the image content. Table 2 and Table 3 show the accuracy (precision) and the ability to recover (recall) of the stego-bearing pixels. The pixels are sorted in decreasing order of energy and those 500 (0.33% of the total number of pixels), 1000 (0.65%) and 1500 (0.98%) occurring at the top of the list are examined to determine if they are stego pixels. On average, 87% of all pixels in the top 0.33% are stego-bearing, and these 0.33% pixels represents 28% of all stego-bearing pixels. Looking further down the list until 0.98% of all pixels, we find that the amount of stego pixels recalled is close to 50%. Figure 5 illustrates extended average precision and recall results.

Table 1: Comparing the energy (std. dev.) in innocent and stego color images.

Color Database	Average Energy (std. dev.) Per Pixel	
	Innocent Images	Stego Images
Flowers	113.73 (31.28)	122.60 (30.50)
Mountains	96.79 (21.48)	106.16 (21.73)
Trees	103.72 (25.19)	111.93 (24.45)

Table 2: The average precision (std. dev.) of identifying stego-bearing pixels in color images for different proportions of total pixels occurring at the top of the ordered list.

Color Database	Average Precision (std. dev.)		
	0.33%	0.65%	0.98%
Flowers	86% (0.16)	66% (0.20)	51% (0.18)
Mountains	91% (0.10)	70% (0.15)	54% (0.14)
Trees	85% (0.15)	61% (0.17)	46% (0.13)

Table 3: The average recall (std. dev.) of identifying stego-bearing pixels in color images for different proportions of total pixels occurring at the top of the ordered list.

Color Database	Average Recall (std. dev.)		
	0.33%	0.65%	0.98%
Flowers	28% (0.07)	41% (0.11)	48% (0.12)
Mountains	30% (0.07)	45% (0.08)	51% (0.08)
Trees	27% (0.08)	39% (0.10)	44% (0.10)

6.2.2 *Grayscale Images*

Table 4 shows that the stego images on average have a higher energy. Though the average increase is small (the measure being entropy based and hence on a log scale), every stego image in our study has a higher energy than its innocent counterpart.

Table 4: Comparing the energy (std. dev.) in innocent and stego grayscale images.

Grayscale Database	Average Energy (std. dev.) Per Pixel	
	Innocent Images	Stego Images
	1.73 (0.44)	1.75 (0.44)

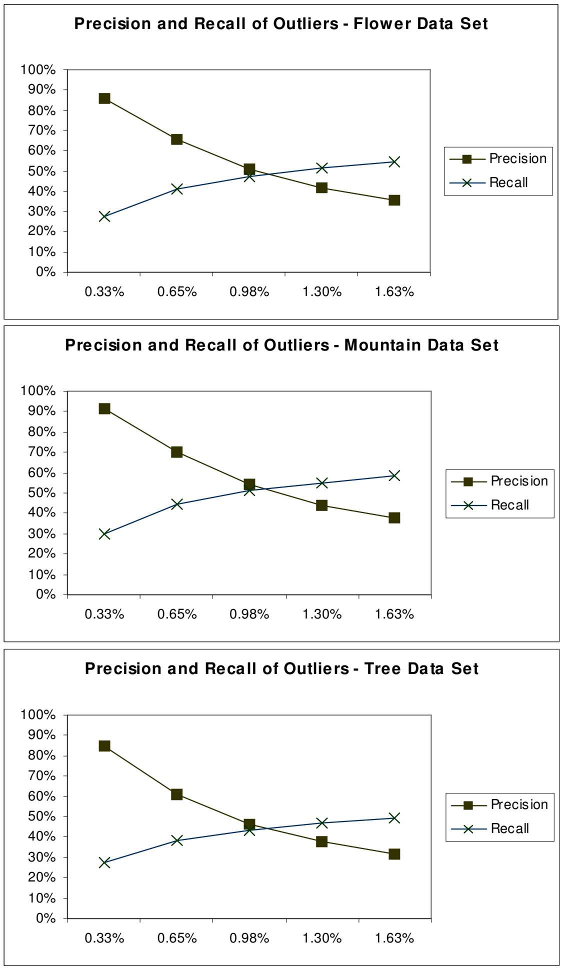

Figure 5: The precision and recall curves for our three color data sets. The x-axis represents the top x% of all pixels ordered in terms of their energy.

Table 5 shows the accuracy (precision) and ability to recover (recall) of the stego-bearing pixels. As in color images, the pixels are sorted in decreasing order of energy and those at the top 0.33%, 0.65% and 0.98% of the list examined to determine if they are stego-bearing. On average, 61% of all pixels in the top 0.33% of pixels are stego-bearing, and these 0.33% of pixels contain on average 20% of all stego-bearing pixels. However, looking further down the list until 0.98% of all pixels we find that the amount of stego pixels recalled is still no more than 28%. Figure 6 shows the plot of average precision and recall against percentages of the total number of pixels occurring on top of the list.

Table 5: The average precision and recall (std. dev.) of identifying stego-bearing pixels in grayscale images for different proportions of total pixels on top of the ordered list.

Grayscale Database	Average Precision & Recall (std. dev.)		
	0.33%	0.65%	0.98%
Precision	61% (0.29)	39% (0.23)	29% (0.16)
Recall	20% (0.11)	25% (0.16)	28% (0.16)

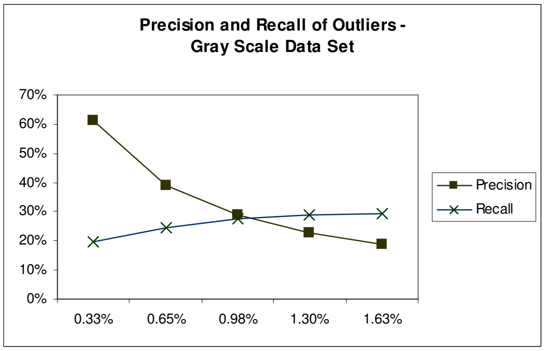

Figure 6: Precision and recall curves for our grayscale data sets. The x-axis represents the top x% of all pixels ordered on energy

7. DISCUSSION

We have shown that:

- Hiding a message increases the energy of an image (Table 1 and Table 4).
- The most energized pixels are typically stego-bearing (Table 2, Table 3 and Table 5).
- As the number of potential outliers taken from the sorted list increases, the number of correctly identified stego-bearing pixels also increases (Figure 5 and Figure 6).
- The most energized pixels for innocent images are concentrated while for stego images they are evenly distributed throughout (Figure 1 through Figure 4).

The monotonicity of the precision and the recall curves implies that our technique is robust and capable of identifying the stego-bearing pixels as outliers. Though our results for grayscale images are quite accurate, they are not as good as for color images. Detecting hidden messages in grayscale images is difficult for two reasons: the variance amongst the palette intensities is very small, and many images are scans of old black and white pictures containing imperfections. Applying image restoration techniques [16] followed by our approach may improve the results.

Steganography and steganalysis techniques are in a continual arms race. Our technique is a developing framework and of course not a panacea. In particular, our approach can be defeated if the steganography algorithm has knowledge of our probability/energy function (i.e. equations 2 and 3), or if the message is carefully embedded in the high energy regions of an image. However, the capacity available to such a steganography algorithm would be greatly reduced. Therefore, to send the same amount of information would require sending more images. An activity monitoring system such as one that monitors Internet traffic or monitors downloads from and uploads to a particular website could then detect the covert transmissions.

8. CONCLUSION

We defined a framework for hidden message location based on image restoration. Hiding a message in an image effectively decreases the probability of an image, or put another way, increases the energy of an image. We defined two energy functions for color and grayscale images. This allows us to measure the probability/energy of each pixel. The outliers in an image are the most energized (i.e. least probable) pixels.

Our results indicate that the stego images contain more energy than their cover counterparts. Our approach can identify stego-bearing pixels with significant accuracy. For the 0.33% most energized pixels, 87% were actually stego-bearing for color images and 61% for grayscale images. Furthermore, we find that the precision and recall of our technique are monotonic functions of the number of predicted outliers. We believe our results could be improved by dividing the image into similar regions using spatial clustering techniques or Kohonen self organized maps [11] and apply our approach to identify outliers in each region.

9. REFERENCES

[1] http://www.jjtc.com/stegoarchive/stego/software.html

[2] Johnson, N., Duric, Z., and Jajodia, S. *Information Hiding: Steganography and Watermarking*, 2001.

[3] Starr, B., and Utley, G. CNN, July 23, 2002: http://www.cnn.com/2002/US/07/23/binladen.internet

[4] http://www.cise.nsf.gov/accomp/index.cfm?div=iis

[5] Kelley, J. USA Today, July 10, 2002: http://www.usatoday.com/news/world/2002/07/10/web-terror-cover.htm

[6] Provos, N. Defending against Statistical Steganalysis. Proc. 10th USENIX Security Symposium, 2001.

[7] Westfeld, A., and Putzmann, A. Attacks on Steganographic Systems. 3rd Information Hiding Workshop, 1999.

[8] Lyu, S., and Farid, H. Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines. Proc. 5th Information Hiding Workshop, 2002, 340-354.

[9] Berg, G., Davidson, I., Duan, M., and Paul, G., Searching for Hidden Messages: Automatic Detection of Steganography. 15th Innovative App. of A.I., 2003, 51-56.

[10] Fridrich, J., and Goljan, M. Practical Steganalysis of Digital Images – State of the Art. Proc. SPIE Photonics West, Vol. 4675, 2002, 1-13

[11] Han, J., and Kamber, M. *Data Mining: Concepts and Techniques*, Morgan Kauffman, 2000.

[12] Aggarwal, C., and Yu, P. Outlier detection for high dimensional data, Proc. ACM SIGMOD 2001, 37-46.

[13] Williams, G., Baxter, R., He, H., Hawkins, S. and Gu, L., A Comparative Study of Replicator Neural Networks for Outlier Detection in Data Mining, Proc. 2nd IEEE ICDM, 2002, 709-712.

[14] Knorr, E. M., and Ng, R. T. A unified notion of outliers: Properties and computation, Proc. KDD, 1997, 219-222.

[15] Shekhar, S. et al. Detecting Graph-based Spatial Outliers: Algorithms and Applications, Proc. KDD, 2001, 371-376.

[16] Jain, A. *Fundamentals of digital image processing*, 1989.

[17] Li, Q., Ye, J., and Kambhamettu, C. Spatially Interest Pixels (SIPs): Useful Low-Level Features of Visual Media Data, 3rd IEEE ICDM, 2003, 63-70.

[18] Cipra, B. An Introduction to the Ising Model, American Mathematical Monthly, Vol 94 No. 10, 937-959.

[19] Neal, R. Probabilistic Inference Using Markov Chain Monte Carlo Method, *TR CRG-TR-93-1*, U. of Toronto.