Higher-order statistical steganalysis of palette images

Jessica Fridrich[*a], Miroslav Goljan[a], David Soukal[b]
[a]Department of Electrical and Computer Engineering, [b]Department of Computer Science,
SUNY Binghamton, Binghamton, NY 13902-6000

ABSTRACT

In this paper, we describe a new higher-order steganalytic method called Pairs Analysis for detection of secret messages embedded in digital images. Although the approach is in principle applicable to many different steganographic methods as well as image formats, it is ideally suited to 8-bit images, such as GIF images, where message bits are embedded in LSBs of indices to an ordered palette. The EzStego algorithm[4] with random message spread and optimized palette order is used as an embedding archetype on which we demonstrate Pairs Analysis and compare its performance with the chi-square attacks[5,7] and our previously proposed RS steganalysis[2]. Pairs Analysis enables more reliable and accurate message detection than previous methods. The method was tested on databases of GIF images of natural scenes, cartoons, and computer-generated images. The experiments indicate that the relative steganographic capacity of the EzStego algorithm with random message spread is less than 10% of the total image capacity (0.1 bits per pixel).

Keywords: Steganalysis, steganography, LSB, EzStego, attacks, detection, steganographic capacity

1. INTRODUCTION

The purpose of steganography is to communicate information in a stealth manner so that anyone who inspects the messages being exchanged cannot collect enough evidence that the messages hide additional secret data. As opposed to cryptography that makes the communication unintelligible to those who do not know the proper cipher keys, steganography should make the communication inconspicuous or invisible. The object that holds the secret information is called the cover object. After a secret message has been hidden in the cover, the cover object becomes the stego object. In this paper, we will deal with steganographic methods for digital images in the palette format GIF.

To mount an attack on a steganographic method, we need to show that it is possible to distinguish cover images from stego images with probability better than random guessing. Obviously, the attacker's ability to do this depends, among other factors, on the choice of cover images, the embedding technique, and the length of the embedded message. In this paper, we propose a new steganalytic technique whose principles can be applied to many different steganographic schemes and image formats. However, it is ideally suited for detection of embedding in palette images, such as the GIF images.

Programs that preprocess the palette by creating clusters of close colors before embedding, such as S-Tools[8], are easily detectable due to obvious artifacts they leave in the palette[3]. A better approach is to leave the palette unchanged and manipulate the image data. A large class of embedding techniques performs this manipulation by internally presorting the palette so that adjacent colors are close to each other and then embedding one bit per pixel as the Least Significant Bit (LSB) of the index to the sorted palette. A number of steganographic programs available on the Internet use this embedding paradigm for data hiding in GIF images. In this paper, we use the EzStego algorithm with random message spread as an example of this embedding archetype (see Section 2).

Previously proposed attacks that are applicable to the above-mentioned steganographic paradigm include the RS steganalysis[2] and the chi-square attack by Westfeld[6] with its generalized versions[5,7]. Because the generalized chi-square attack of Provos[5] was not described in his original paper in sufficient detail and was only discussed in the context of steganography in JPEG files, in Section 3 we have performed our own set of experiments to obtain a fair comparison. In Section 4, we describe a new attack (Pairs Analysis) that uses higher-order statistics and large-scale non-local

[*] fridrich@binghamton.edu; phone: 1 607 777-2577; fax: 1 607 777-4464; http://www.ssie.binghamton.edu/fridrich

correlations among pixels that typically exist in palette images due to their low color-depth. We pay close attention to comparing Pairs Analysis to the existing methods. We show that for palette images, Pairs Analysis outperforms both the chi-square attack including its generalized versions and the RS steganalysis. In Section 5 we present experimental results for 180 images taken with a digital camera and saved in the GIF format and for artificial images, such as cartoons or computer art (fractals). To improve the message length estimation for artificial images, which may often have "singular" palettes, in Section 6 we describe a modification of the proposed detection algorithm and present experimental results. The paper is summarized in Section 7 with outlining possible future research directions.

2. EZSTEGO ALGORITHM

EzStego first orders the palette to minimize color differences between consecutive colors by finding an approximate solution to the traveling salesman problem. Then, the message bits are embedded as the LSBs of color indices to the sorted palette. The original EzStego algorithm embeds bits sequentially but in our version we proceed along a pseudo-random key-dependent walk to make the detection harder. For a more detailed description of the algorithm, including the source code, see the documentation by Machado[4].

EzStego sorts the palette colors c_0, c_1, …, c_{P-1}, $P \le 256$ in a cycle $c_{\pi(0)}$, $c_{\pi(1)}$, …, $c_{\pi(P-1)}$, $\pi(P) = \pi(0)$, so that the sum of distances $\sum_{i=0}^{P-1} \left| c_{\pi(i)} - c_{\pi(i+1)} \right|$ is small. In the last expression, π is the sorting permutation found by EzStego. The set of pairs whose colors will be exchanged for each other during embedding is

$$E = \{(c_{\pi(0)}, c_{\pi(1)}), (c_{\pi(2)}, c_{\pi(3)}), \ldots, (c_{\pi(P-2)}, c_{\pi(P-1)})\}\ . \tag{1}$$

Using the stego-key, generate a pseudo-random walk through image pixels. For each pixel along the walk, replace its color $c_{\pi(k)}$ with the color $c_{\pi(j)}$, where j is the index k with its LSB replaced with b, where b is the message bit: LSB(j) = b. Repeat the embedding steps till all message bits are embedded or the end of image file is reached.

The message extraction algorithm first determines the same palette ordering π and then generates the pseudo-random walk from the stego-key. The message bits are extracted from LSBs of indices to the sorted palette, b = LSB(k), where $c_{\pi(k)}$ is the pixel color visited along the random walk.

3. GENERALIZED CHI-SQUARE ATTACK

The chi-square attack[6] can be applied to any steganographic technique in which a fixed set of Pairs of Values (PoVs), or other fixed groups of values, are flipped into each other to embed message bits. For example, the PoVs can be formed by palette indices that differ in their LSBs. Before embedding, in the cover image the two values from each pair are distributed unevenly. After message embedding, the occurrences of the values in each pair will have a tendency to equalize (this depends on the message length). Since swapping one value into another does not change the sum of occurrences of both indices in the image, we can test for the statistical significance of the fact that the occurrences of both values in each pair are the same. If, in addition to that, the stego-technique embeds message bits sequentially into subsequent pixels/indices/coefficients starting, for example, in the upper left corner, we will observe an abrupt change in the statistical evidence as we encounter the end of the message.

3.1 Chi-square attack

Let us assume that the palette colors c_0, c_1, …, c_{P-1} are already sorted as in expression (1). Since $P \le 256$, we have at most 128 PoVs. For the i-th pair (c_{2i}, c_{2i+1}), i = 1, …, k, we define n_i' = 1/2(number of indices in the set $\{c_{2i}, c_{2i+1}\}$) and n_i = number of indices equal to c_{2i}. The value n_i' is the theoretically expected frequency if a random message has been embedded, and n_i is the actual number of occurrences of color c_{2i}. We can now perform the chi-square test for the equality of n_i' and n_i. The Chi-square statistics is calculated as $\chi_{k-1}{}^2 = \sum_{i=1}^{k} \frac{(n_i - n'_i)^2}{n'_i}$ with k–1 degrees of freedom, the p-value $p = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma\left(\frac{k-1}{2}\right)}\int_0^{\chi_{k-1}^2} e^{-\frac{x}{2}} x^{\frac{k-1}{2}-1} dx$ expressing the probability that the distributions of n_i' and n_i are equal.

For a sequentially embedded message, one can scan the image in the same order in which the message has been embedded and evaluate the p value for the set of all already visited pixels. After a short transient phase, the p value will at first be close to 1 and then it suddenly drops to 0 as we arrive at the end of the message. This test enables us not only to determine with a very high probability that a message has been embedded, but also calculate its length.

If the message-carrying pixels in the image are selected randomly rather than sequentially, this test becomes less effective unless majority of pixels (i.e., more than 97%) have been used for embedding. Westfeld[7] recently described a simple generalization of this idea by grouping colors from one pixel or neighboring pixels and fusing their values using a special hash-like function. The combined values are then analyzed using the same chi-square test as above. Westfeld reports that messages as small as 33% of the maximal image capacity are detectable using this technique. This message length appears to be a fundamental limitation of this approach. Provos[5] describes a different idea to extend the original chi-square attack that he uses for detection of randomly scattered bits in JPEG images. Provos' description of his method is, unfortunately, rather sketchy and no further analysis of his method has been published by the time of writing this paper. Therefore, we included our own analysis of the generalized chi-square attack in the next section so that we can compare it with our Pairs Analysis in a fair manner.

3.2 Generalized chi-square attack

Provos observed that the chi-square attack on stego images with randomly scattered messages produces fluctuating p values in the beginning, and then, as the sample size increases, the p-value eventually drops to zero due to the sensitivity of the test. He proposed to use a sliding window of a *fixed* size that he moves along the image instead of increasing the window size. If the window is not too short, then, because of the stego content, we will register fluctuations of the p value due to uneven distribution of the message bits in the stego image and varying number of classes in the chi-square test (see the upper graph of Figure 2). However, the same sliding window will produce much smaller fluctuations for the cover image (see the upper graph in Figure 1). The difference in the amplitude of the fluctuations can be used to distinguish between stego and cover images.

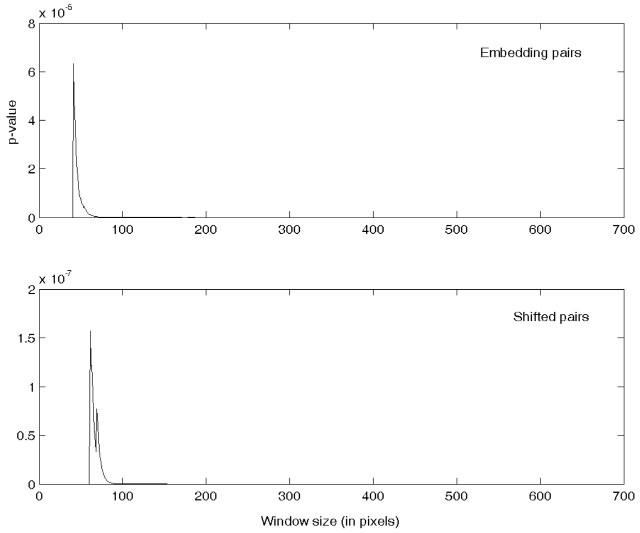


Figure 1: p-values from a window of increasing length for an image without any message. The tested segment started after skipping the first 10% of all image rows. The graphs are very similar to each other.

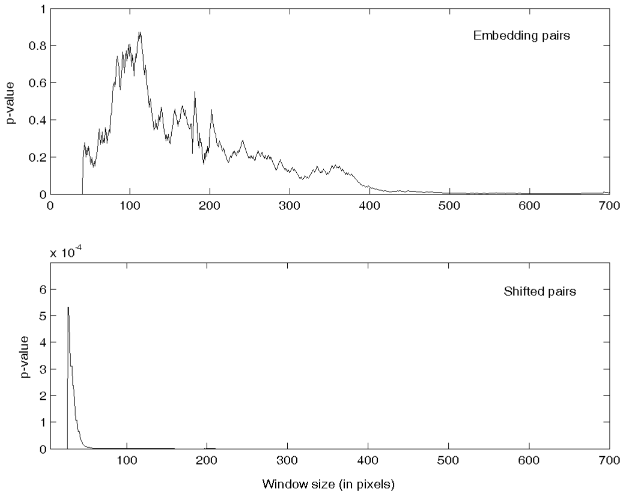


Figure 2: This time, the image contains a message of length of 75% of the image size. The p-value computed from shifted pairs corresponds to the image without a message, but the p-value computed from the embedding pairs strongly exhibits presence of a message until the window length exceeds 400 pixels.

To find the window size that will produce the most sensitive detection, Provos uses the "shifted" pairs of colors E'

$$E' = \{(c_{\pi(1)}, c_{\pi(2)}), (c_{\pi(3)}, c_{\pi(4)}), \ldots, (c_{\pi(P-1)}, c_{\pi(0)})\} , \tag{2}$$

(c.f., Expression (1)) and finds the smallest window size that produces p values during sliding that are all below a certain small threshold. The assumption here is that the fluctuations calculated from the shifted pairs in the stego image are similar to the fluctuations calculated from the pairs (c_{2i}, c_{2i+1}) used for embedding in the cover image. This is indeed confirmed by comparing the bottom graph of Figure 2 with the upper graph of Figure 1.

The generalized chi-square algorithm consists of two steps: calibration and detection.

1. Find a window size s_0, such that $\max_i p'(i, s_0) < T$ and $\max_i p'(i, s_0+\varepsilon) \geq T$. The symbol $p'(i, s)$ denotes the p-value of the chi-square statistics computed from shifted palette pairs (2) and calculated from the segment of image data of the length s taken at position i (this could correspond to sliding the window by one pixel, i.e. $i = 1, 2, \ldots, total\_length-s$ or by some larger amount, such as i = 0%, 10%, …, 100% of the $total\_length-s$). The parameter T is a fixed threshold. Our experiments showed that $T = 10^{-5}$ works well. Finally, ε should be a small integer compared to s_0.
2. For the sliding window of length s_0 (treat the image as a vector, e.g. by concatenation of rows), calculate $p'(i, s_0)$ for the (unrelated) shifted pairs. We define n_i' = 1/2(number of indices in the set $\{c_{2i-1}, c_{2i}\}$) (with the convention $c_0 \equiv c_{256}$).
3. Slide the same window along the image and record the p-values in the window, but this time compute the statistics from the pairs (c_{2i}, c_{2i+1}) used for embedding (1).
4. To determine the presence of a message, threshold the mean p value $|I|^{-1} \sum_{i \in I} p(i, s_0)$ using a threshold determined from a test database of sufficient richness and diversity.

While this algorithm works reasonably well for detection of LSB embedding in the DCT domain (in JPEG files), we observed that its performance is markedly worse for GIF images. The main reason is that, unlike DCT coefficients, the indices forming the palette image exhibit strong spatial correlations. Therefore, it is harder to find an appropriate window size. Due to dithering, areas with a uniform pattern usually are made up of five, four, or fewer colors. In these areas, the number of degrees of freedom was 0 and yet in other areas the p-values computed from shifted pairs were significantly below the specified threshold T. It is not clear how to deal with this situation. One may not want to decrease the threshold and thus enforce enlargement of the window size because too low a threshold might cause loss of sensitivity (in Figure 2, we see that the p-value computed from embedding pairs will eventually fall to zero). Another problem is that even if we find the window size such that all p-values computed from the shifted pairs are well defined (i.e. the number of degrees of freedom is greater than zero), there is no guarantee that the p-value computed from the embedding pairs will be defined as well.

These were the main reasons why we modified the algorithm for palette images so that it finds the appropriate window size for each tested area separately. We also hoped that this would increase sensitivity of the test. It turned out that the overall performance of this adaptive version has improved although it reacts more sensitively to images without any messages, possibly increasing the rate of false positives.

The adaptive generalized chi-square algorithm consists of the following:

1. For all tested areas I, perform these steps
   a. Find an appropriate window size for the current area as in Step 1 of the algorithm above. That means, for an area i, find $s^i{}_0$, such that $\max_i p'(i, s_0) < T$ and $\max_i p'(i, s_0+\varepsilon) \geq T$.
   b. If $p(i, s^i{}_0)$ is not defined because the corresponding χ^2 has zero degrees of freedom, adjust the window size so that it is enlarged as little as needed to define $p(i, s^i{}_0)$.
   c. Record the value $p(i, s^i{}_0)$.
2. Detect the presence of the secret message by computing the mean value of $p(i, s^i{}_0)$, $i \in I$.

The following tables summarize the performance of the two methods on three different image databases. The first database contains 67 images taken by digital cameras. The pictures were converted to the GIF format using Corel Photo Paint 9. The second database had 25 cartoon images collected from different sites on the Internet. The third database consisted of fractal computer-generated images[$].

The embedded messages were random bit-streams and were embedded by EzStego 2.0b4. All images were then subjected to both the chi-square attack and its adaptive version. We represent the results using the number of images in which the message has been successfully detected. The decision threshold was chosen to guarantee less than 1% of false positives. The thresholds are given in Table 4.

Message length (%)		10	20	30	40	50	60	70	80	90	100
Detected files (%)	Extended test	6.5	17.7	51.3	90	100	100	100	100	100	100
	Modified test	6.5	24.2	50.6	96	100	100	100	100	100	100

Table 1. Detection performance of the generalized test and the adaptive generalized test on a database of 67 natural images.

Message length (%)		10	20	30	40	50	60	70	80	90	100
Detected files (%)	Extended test	4.2	8.3	8.3	29.2	75	100	100	100	100	100
	Modified test	12.5	16.6	29.2	50.0	75	100	100	100	100	100

Table 2. Detection performance of the generalized test and the adaptive generalized test on a database of 25 cartoon images.

Message length (%)		10	20	30	40	50	60	70	80	90	100
Detected files (%)	Extended test	5	5	10	10	12	42	77	95	97	100
	Modified test	5	10	10	12.5	37	52	92	97	100	100

Table 3. Detection performance of the generalized test and the adaptive generalized test on a database of 42 fractal images.

Method	Optimal threshold		
	Natural images	Cartoons	Fractals
Extended test	0.0015%	0.049%	5.09%
Modified extended test	0.4327%	0.413%	7.69%

Table 4. The decision thresholds based on detection performance of both methods on images without any messages.

If we postulate that a test is successful if it detects at least 85–90% of all stego images correctly (with false positives less than 1%), then the generalized chi-square test and its adaptive version have very similar performance. Both methods reliably detect a message once it is longer than 40–50% and 50–60% of the capacity for natural images and cartoons, respectively. But the methods perform somewhat worse for fractal images (successful detection for at least 70% of capacity).

We attempted to explain why fractal images produced markedly different results. The palette sorting algorithm of EzStego is sensitive to the initial ordering of the palette. For two identical images with the same palette entries but in a different order, EzStego will very produce two different internal orderings. For some images, the palette sorting by solving the traveling salesman problem produces an ordering with some discontinuities (the approximate solution to the traveling salesman problem is not good). The presence of these discontinuities causes the assumption of the generalized chi-square attack to become invalid (the assumption that fluctuations of the *p* value in the cover image for the shifted pairs is similar to the fluctuations for the embedding pairs). This produces outliers in our experiments and, consequently, leads to a decision threshold that is incomparably greater than for the other two databases (see Figure 3). A larger threshold means that messages must be longer to be detected.

[$] Images provided courtesy of J. C. Sprott, Fractal Gallery, http://sprott.physics.wisc.edu/fractals.htm.

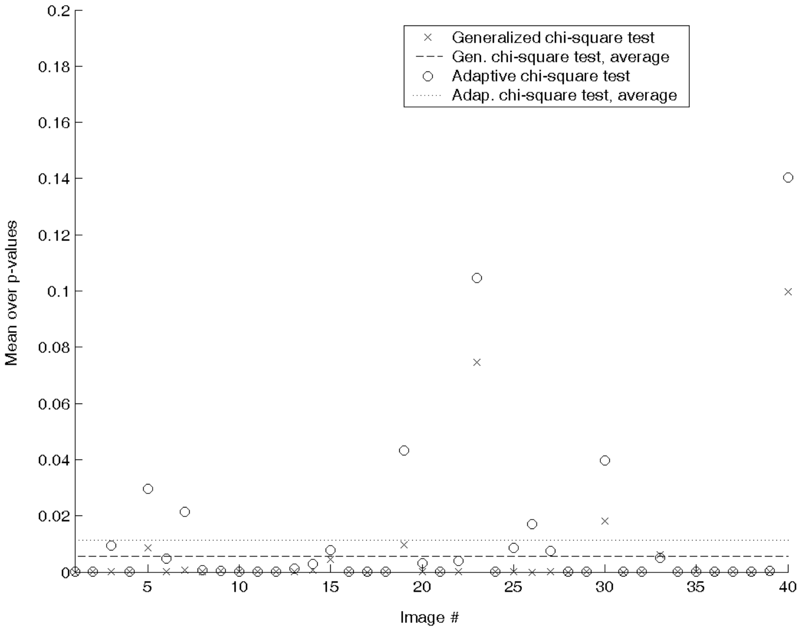


Figure 3: The mean *p*-value for 40 cover fractal images in the GIF format for the generalized chi-square attack and its adaptive version. Note that the adaptive version gives slightly larger overall response.

3.3 Performance evaluation of the chi-square attacks

It is not clear how this methodology can be used to approximate the message length. The *p*-value is a very sensitive measure and it typically exhibits large variations as we slide the window along the image. Also the value itself does not seem to reliably capture the amount of embedded data. Of course, the more data is embedded, the higher the *p*-value is, in general. However, some images react sensitively even for a small amount of data and, on the contrary, some images exhibit areas of almost no response to the test. Thus, only a very rough message length estimate could be obtained.

Another problem of the generalized chi-square test arises from the fact that the calibration process depends on the threshold *T* whose value does not have proper justification. Also, we note that any steganalytic technique based on the analysis of sample counts (the histogram) will be easy to circumvent. Provos himself shows how one can design a JPEG embedding technique (OutGuess 0.2) that will preserve the original counts of samples in their PoVs and thus avoid message detection using the chi-square attack or its generalized versions.

Before we describe the details of the proposed Pairs Analysis, we acknowledge the work by Farid et al.[1] who have also reported a successful detection of the original EzStego algorithm with sequential embedding. For a false detection rate less than 1%, the authors detect about 77% of all stego images with a close-to-100% message length embedded. With shorter messages, the detection rate quickly falls to small values. Given the fact that this algorithm is blind to the stego method, this is a remarkable result. However, targeted approaches, such as the Pairs Analysis reported in the next section, provide significantly higher detection rate while giving an accurate estimate of the secret message length.

4. PAIRS ANALYSIS

In this section, we describe a new statistical attack for the embedding archetype that embeds messages in palette images as LSBs of indices to palette colors along a random walk. The new method is more reliable and accurate than previously proposed approaches (see the previous section and Section 5) and can accurately estimate the secret message length. The EzStego algorithm[4], Steganos[8], and Hide&seek[8] are examples of programs that embody this embedding archetype. Before we describe the principle of Pairs Analysis, we analyze the impact of EzStego embedding on cover images.

Let (c_1, c_2) be a color pair from E (see Expression (1)). We extract the colors c_1, c_2 from the whole image for example by scanning it by rows and/or columns. This sequence of colors is then converted to a binary vector by associating c_1 with a "0" and c_2 with a "1". We denote this vector $Z(c_1, c_2)$ and call it the *color cut* for the pair (c_1, c_2). Because palette images have a small number of colors, Z will show considerable structure for most pairs (c_1, c_2). The embedding process will disturb this structure and increase the entropy of Z. Finally, when the maximal length message has been embedded in the cover image (1 bit per pixel), the entropy of Z will be maximal corresponding to a random binary sequence.

Now, we will look at what happens during embedding to color cuts for "shifted" color pairs from the set E' (see Expression (2)). Let us look at one fixed color pair ($c_{\pi(2k-1)}$, $c_{\pi(2k)}$) from E'. During embedding, the colors $c_{\pi(2k-1)}$ and $c_{\pi(2k-2)}$ are exchanged for each other and so are the colors $c_{\pi(2k)}$ and $c_{\pi(2k+1)}$. Even after embedding the maximal message (each pixel modified with probability ½), the color cut $Z(c_{\pi(2k-1)}, c_{\pi(2k)})$ will still show some residual structure. To see this, imagine a binary sequence W that was formed from the cover image by scanning it by rows and associating a "0" with the colors $c_{\pi(2k-1)}$ and $c_{\pi(2k-2)}$ and a "1" with the colors $c_{\pi(2k)}$ and $c_{\pi(2k+1)}$. Convince yourself that the color cut $Z(c_{\pi(2k-1)}, c_{\pi(2k)})$ after embedding a maximal pseudo-random message in the image is the same as starting with the sequence W and skipping each element of W with probability ½. Now, because W showed structure in the cover image, most likely long runs of 0's and 1's, we see that randomly chosen subsequences of W will show some residual structure as well.

Having presented our arguments in the paragraph above, we describe our new detection method. We concatenate color cuts for all pairs in E into one vector $Z = Z(c_{\pi(0)}, c_{\pi(1)})$ & $Z(c_{\pi(2)}, c_{\pi(3)})$ & … & $Z(c_{\pi(P-2)}, c_{\pi(P-1)})$ and all color cuts for shifted pairs E' into the vector $Z' = Z(c_{\pi(1)}, c_{\pi(2)})$ & $Z(c_{\pi(3)}, c_{\pi(4)})$ & … & $Z(c_{\pi(P-1)}, c_{\pi(0)})$. Next, we define the quantity that will be used to measure the structure in the bit-streams Z and Z'. Let $E_2(Z)$ be the second-order entropy

$$E_2(Z) = \sum_{i=1}^{4} - p_i \log p_i \ ,$$

where p_i are the probabilities of occurrence of the bit-pairs '00', '01', '10', and '11' in Z. To simplify the calculations, instead of using E_2 directly, we simply count the number of "homogenous" bit-pairs ('00', '11') in Z. Let $R(p)$ denote the expected number of homogeneous bit-pairs in Z after flipping the LSBs of indices of p×100% of randomly chosen pixels, $0 \le p \le 1$, divided by n – the length of Z. Similarly, let $R'(p)$ be the relative number of homogeneous bit-pairs in Z'. For p < ½, this number of modifications corresponds to embedding a random message of length $2pMN$ bits ($2p$ bits per pixel), where M and N are image dimensions.

In Theorem 1 below, we prove that $R(p)$ is a parabola with its vertex at p = ½ and $R(1/2)$ = ½ (see Figure 4). Because we can calculate $R(q)$ from the stego image with an unknown message length q, the points (q, $R(q)$) and (1/2, $R(1/2)$) uniquely determine $R(p)$. Note that $R(q) = R(1-q)$.

It appears that $R'(p)$ is well modeled using a parabola, as well, although we have no formal proof of this statement. The value of $R'(1/2)$ can be derived from Z' (see Theorem 2), while the values $R'(q)$ and $R'(1-q)$ can be calculated from the stego image and the stego image with all colors flipped, respectively. Thus, we can fit a second-degree polynomial through the points (q, $R'(q)$), (1/2, $R'(1/2)$), and (q, $R'(1-q)$) to obtain $R'(p)$.

Finally, we accept one additional assumption $R(0) = R'(0)$, which says that the number of homogenous pairs in Z and Z' must be the same if no message has been embedded. This is, indeed, intuitive because there is no reason why the color cuts of pairs in E and E' should have different structures. The accuracy with which this assumption is satisfied determines the accuracy of the message length estimate. Examples of images for which this assumption is not satisfied are discussed in the next section.

Figure 4 shows $R(p)$ and $R'(p)$ as functions of p – the Pairs-Diagram – for a typical test image. After denoting $D(p) = R(p)-R'(p) = ap^2 + bp + c$, with a, b, and c yet undetermined constants, we can write $D(0) = c = 0$ and $D(1/2) = R(1/2)-R'(1/2) = a/4 + b/2$. Also, $D(q) = aq^2 + bq$ and $D(1-q) = a(1-q)^2 + b(1-q)$. Eliminating the unknown parameters a, b leads after simple algebra to the following quadratic equation for q:

$$4D(1/2)q^2+[D(1-q) - D(q) - 4D(1/2)]q+D(q) = 0. \tag{3}$$

Because the coefficients of this quadratic equation are known, we can solve it for the unknown q. The root that is smaller is our approximation to the unknown message length q.

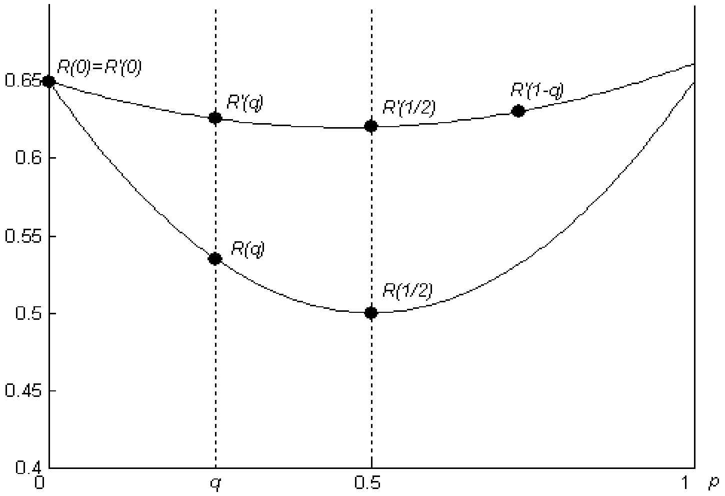


Figure 4: Typical Pairs-Diagram showing the relative number of homogenous bit-pairs in Z and Z' as functions of the number of pixels p with flipped color indices.

Theorem 1. *The expected value of* $R(p)$, $p\in[0,1]$, *is a parabola with its minimum at* 0.5. *In particular,* $R(1/2) = (n-1)/2n \approx 1/2$ for large n.

Proof: We repeat that $R(p)$ is the relative number of homogenous pairs in Z after embedding a message of relative length p. We can write Z as a concatenation of binary segments consisting of consecutive 0's or 1's with lengths of the segments $k_1, k_2, \ldots, k_r$, $k_1+k_2+\ldots+k_r = n$, $k_i > 0$, where n is the length of Z. For example, for $Z = 001110110\ldots$, we have $k_1 = 2$, $k_2 = 3$, $k_3 = 1$, …. For $p = 0$, we have $nR(0) = \sum_{i=1}^{r}(k_i - 1) = n-r$. After embedding a message of relative length p, in a segment consisting of k bits, the probability that a given pair of consecutive bits will be homogenous is $p^2+(1-p)^2$ (both are changed or none are changed). Because we have $k-1$ consecutive pairs, the expected number of homogenous pairs is $[p^2+(1-p)^2](k-1)+2p(1-p)$, where the last term comes from the right end of the segment (an additional pair will be formed at the boundary if the last bit in the segment flips and the first bit of the next segment does not flip, or vice versa). Thus, the total number of homogenous pairs is a sum over all segments except for the last one, which lacks the boundary term $2p(1-p)$:

$$n\,R(p) = \sum_{i=1}^{r}[(k_i - 1)(p^2 + (1-p)^2) + 2p(1-p)] - 2p(1-p) = 2p^2(n-2r) - 2p(n-2r) + n - r - 2p(1-p)\,.$$

We see that $R(p)$ is a parabola in p with its vertex at $p = ½$, $R(0) = R(1) = (n-r)/n$, and $R(1/2) = (n-1)/2$, which concludes the proof. □

Theorem 2. *Let* $Z'=\{b_i\}^n_{i=1}$ *be the binary vector defined in the text above. The expected value of* $R'(1/2)$ *is* $\sum_{k=1}^{n-1} 2^{-k} h_k$, *where* h_k *is the number of homogenous pairs in the sequence of pairs* b_1b_{1+k}, b_2b_{2+k}, b_3b_{3+k}, …, $b_{n-k}b_n$.

Proof: Let $W = W_1\&W_2\& \ldots \&W_{P/2}$ be the concatenation of binary sequences W_j formed from the stego image by scanning it by rows and associating a "0" with the colors $c_{\pi(2j-1)}$ and $c_{\pi(2j-2)}$ and a "1" with the colors $c_{\pi(2j)}$ and $c_{\pi(2j+1)}$. The color cut $Z'(c_{\pi(2j-1)}, c_{\pi(2j)})$ after embedding a maximal message in the cover image is the same as starting with the sequence W_j and skipping each element of W with probability ½. Imagine you are going through Z' while skipping each element with probability ½. Then, the probability of skipping exactly k–1 elements in a row is 2^{-k}, k = 1, 2, …. Because there are h_k homogenous pairs in the sequence of pairs b_1b_{1+k} , b_2b_{2+k} , b_3b_{3+k} , …, $b_{n-k}b_n$, the expected number of homogenous pairs separated by k–1 elements is 2^{-k} h_k. The formula of Theorem 2 is obtained by summing these contributions from k=1 to the maximal separation k–1 = n–2. □

5. EXPERIMENTAL RESULTS

We have performed tests on a database of 180 color GIF images. The images were obtained using four different digital cameras and were originally stored as high quality JPEG images. For our test purposes, we resampled them to 800×600 pixels using Corel Photo-Paint 9 (with anti-alias option) and converted to palette images using with the following options: optimized palette, ordered dithering. All images were embedded with 0, 10, 20, 40, 60, 80, and 100% messages (100% message corresponding to 1 bit per pixel) and processed using our detection method. The results are shown in Figure 5. Figure 6 shows the distribution of detected message length around the true message length (assuming the distribution is Gaussian). Note that the RS Steganalysis gives very similar results (see Figure 7) but may occasionally produce an outlier. Pairs Analysis exhibits more stable behavior.

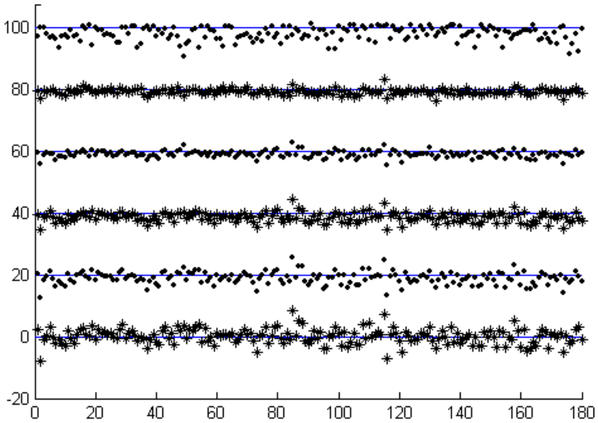


Figure 5: Estimated message length (in % of maximal message length) for a database of 180 GIF images.

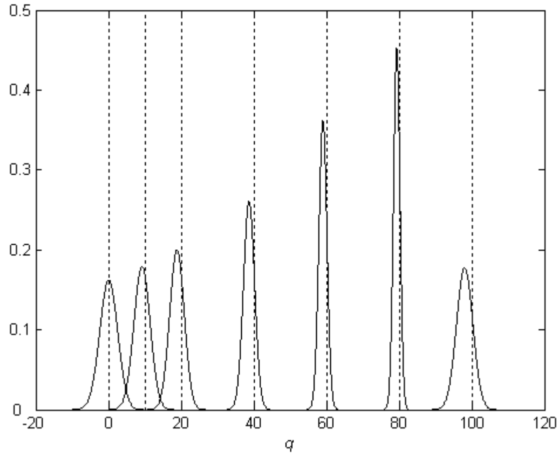


Figure 6: Distribution of estimated message length q compared to the true message length (vertical lines) for 180 GIF images.

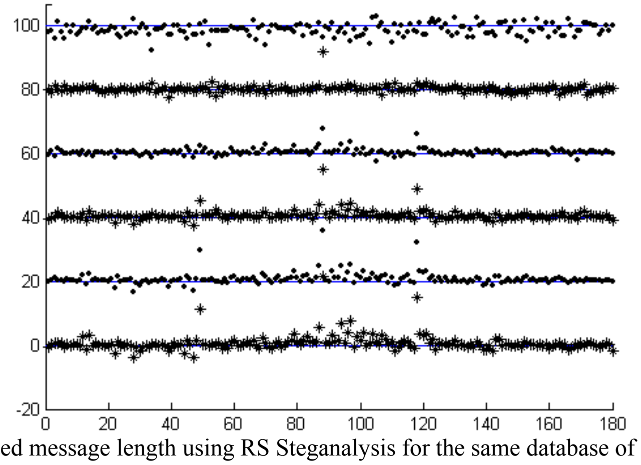


Figure 7: Estimated message length using RS Steganalysis for the same database of 180 GIF images.

Figures 8 and 9 show similar results for a database of 24 cartoon images and Figures 10 and 11 the results for computer-generated images (fractals). We can see that the accuracy of the message length estimate is somewhat lower for cartoons and fractals when compared to GIFs converted from true-color images. This is to be expected because artificial images will more often have palettes with unusual color distribution and unusual dithering patterns that will violate the assumptions of Pairs Analysis. There is one outlier among the cartoon images (image No. 5) that produces a very large false positive. This phenomenon is investigated in the next section, where we also show how Pairs Analysis can be adjusted to give more stable results for artificial images and eliminate such outliers.

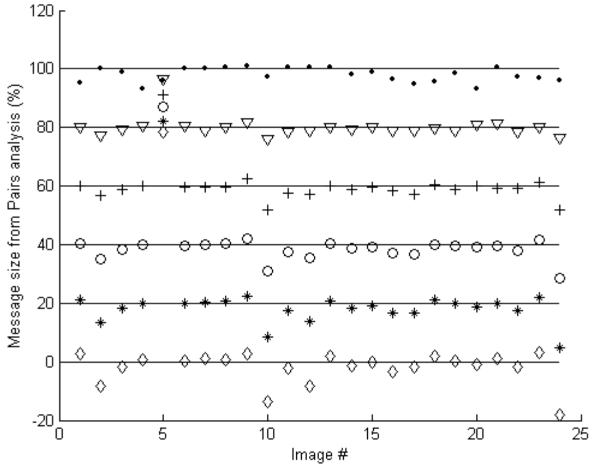


Figure 8: Estimated message length for 24 cartoon images using Pairs analysis.

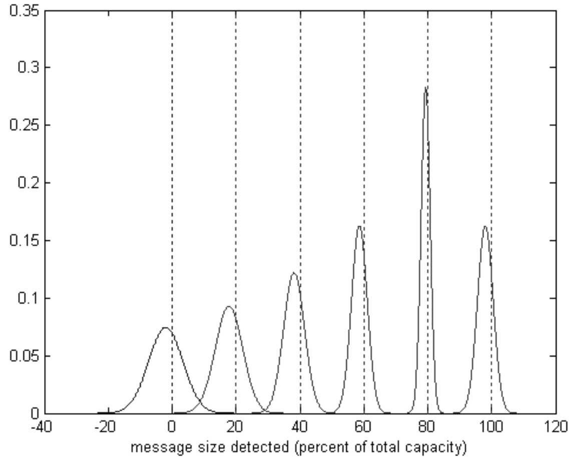


Figure 9: Distribution of estimated message length around the true message length.

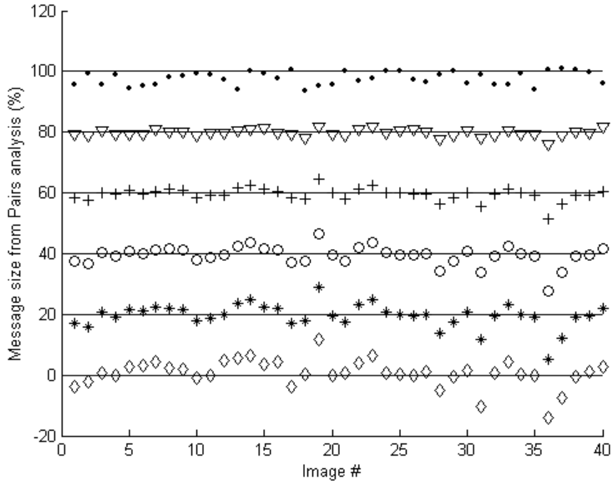


Figure 10: Estimated message length for fractal images using Pairs analysis.

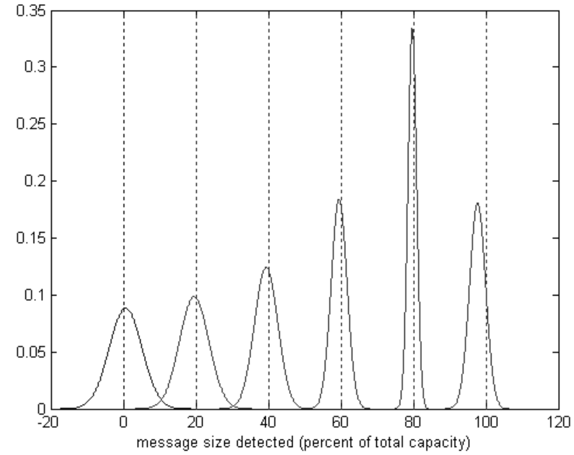
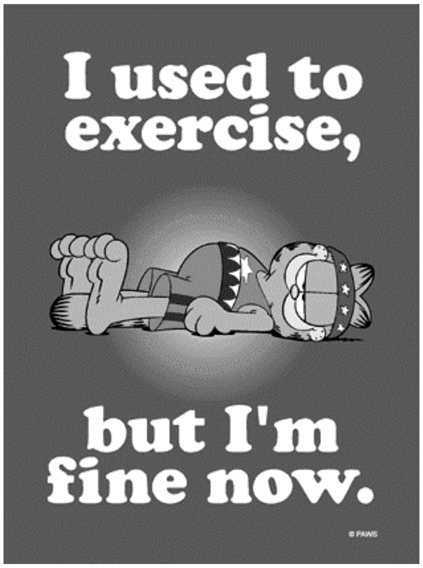


Figure 11: Distribution of estimated message length around the true message length.

6. PAIRS ANALYSIS FOR ARTIFICIAL IMAGES

Overall, the accuracy of Pairs Analysis is mostly influenced by the “initial bias”, which is the message length (in percents) detected in the cover image. The fact that this initial bias may not be exactly zero reflects the fact that the assumption $R(0) = R'(0)$ in Section 4 is an approximate experimental fact. Images with a low number of unique colors, such as cartoons and computer-generated images, may have some singular structure that causes our assumption to become invalid and, consequently, a large error in message length estimation. For example, some dithering patterns between two colors (e.g., in a uniform background) may be misinterpreted as a false message. This is much more likely to occur in artificial images, such as cartoons or computer art, than in natural images where the number of colors is large. Also, a histogram with a very high number of pixels for one or two colors next to each other may in some exceptional cases negatively influence the detection accuracy.

In this section we describe a modification of the Pairs Analysis that exhibits more stable behavior for artificial images that may contain some of the singularities mentioned in the paragraph above. We explain in detail why image No. 5 in the cartoon database (see Figure 12) became an outlier. This image has two similar colors ($c_{\pi(k)}$, $c_{\pi(k+1)}$) with a high occurrence in the cover image. The colors form the purple background. In order to approximate the true background color, the dithering process alternates both colors. Consequently, the color cut $Z_{k,k+1} = (c_{\pi(k)}, c_{\pi(k+1)})$ has a relatively low number of homogenous pairs. However, the color cuts of shifted pairs $Z_{k-1,k} = (c_{\pi(k-1)}, c_{\pi(k)})$ and $Z_{k+1,k+2} = (c_{\pi(k+1)}, c_{\pi(k+2)})$ will likely have a large number of homogenous pairs because of the high frequency of the colors $c_{\pi(k)}$ and $c_{\pi(k+1)}$. Due to high frequency of occurrence of both purple colors, the contribution from the color cuts $Z_{k-1,k}$, $Z_{k,k+1}$, and $Z_{k+1,k+2}$ is the dominant factor in evaluating $R(0)$ for Z and $R'(0)$ for Z'. This is interpreted by the algorithm as a large false message.



"Garfield" character and poster: Copyright Paws, Incorporated. All Rights Reserved. Used by Permission.

Figure 12: Image No. 5 is an outlier in Figure 7 because the dithering pattern of two similar purple colors in the background has been misinterpreted as a large false message.

To prevent misinterpretation of dithering as message embedding, Pairs Analysis has been modified. In particular, in order to avoid the case described above, we select the color pairs for evaluating both *R* and *R*' based on the following criteria:

1. Start with the sets *E* and *E*' as defined above. Then, add more color pairs (c_k, c_l) to the set *E*' based on the following criteria:
   a) The distance between both colors is less than the average distance between every two colors in *E*
   b) The length of the color cut $Z_{k,l}$ for ($c_{\pi(k)}$, $c_{\pi(l)}$), satisfies $|Z_{k,l}| >$ *min_length* (we take *min_length* = 100)
   c) Add at most |*E*| pairs with the shortest distance
   d) Do not add pairs that are already in *E*, $E \cap E' = \varnothing$.

2. Reduce the number of color pairs in both *E* and *E*' retaining only those pairs whose cuts have a relatively large number of homogenous pairs *and* take up a large area in the image. This is achieved by first sorting the pairs in *E* by the following quantity that combines both requirements

$$R_{Z_{k,l}}{}^2 + w\left[\frac{length(Z_{k,l})}{length(Z)}\right]^2,$$

and then retaining only the first quarter of those pairs. Then, we sort *E*' by the same value and retain the same number of pairs as in *E*. The factor *w* is a weighting constant that adjusts the importance of the two factors (in our experiments, $w \approx 1/9$ gave the best results).

The modified Pairs Analysis exhibits much more stable behavior for artificial images than the original version (Figures 13–16). This is true especially when the embedded message is small or when no message has been embedded. In particular, image No. 5 is no longer an outlier. The results are worse, however, for large message lengths. Thus, it seems that the best solution would be to combine both the original Pairs Analysis and its modified version. For images obtained through dithering and color quantization of natural images, we should use the original method as described in Section 4. For artificial images, we can calculate both values and compare them. If there is a large discrepancy between both values, use the result obtained from the modified Pairs Analysis. When both results are close and larger than 0.5, the result from the original algorithm should be used because the modified version is less accurate for large messages.

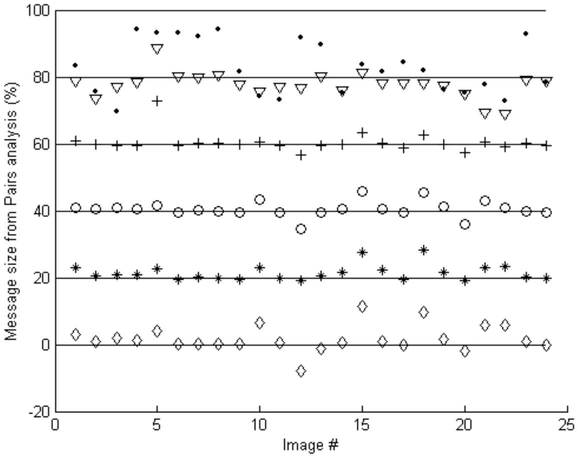


Figure 13: Estimated message length in 25 cartoon images using Pairs Analysis.

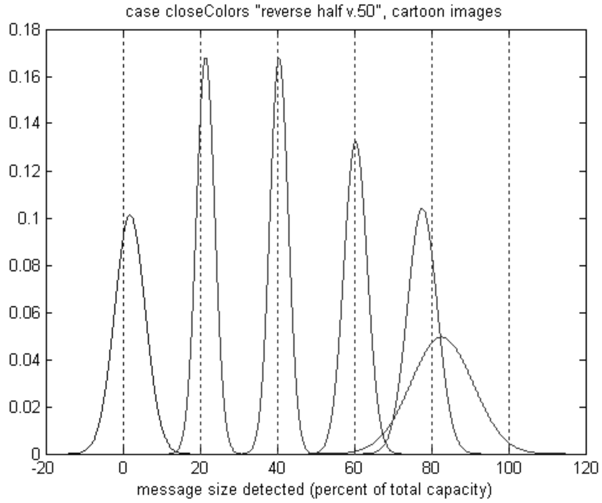


Figure 14: Distribution of estimated message length around the true message length.

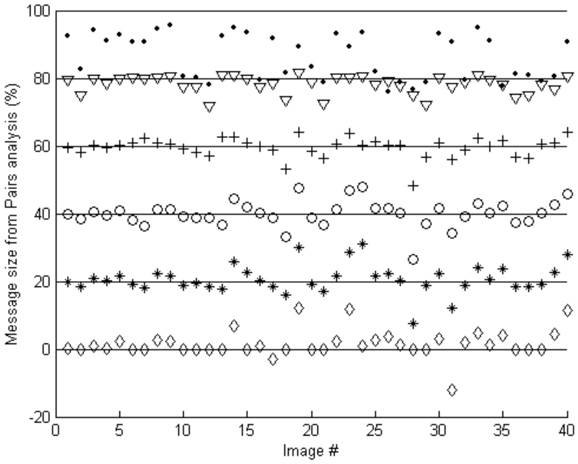


Figure 15: Estimated message length in 40 Fractal images using Pairs Analysis.

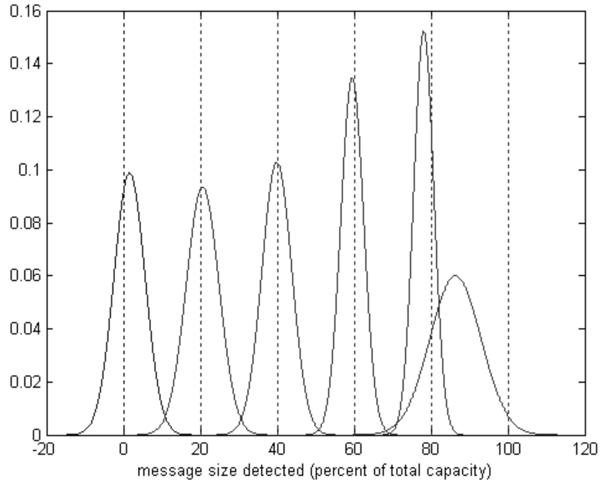


Figure 16: Distribution of estimated message length around the true message length.

7. SUMMARY AND FUTURE DIRECTIONS

In this paper, we present a new higher-order steganalytic technique for palette images (Pairs Analysis). The technique uses large-scale patterns formed by pairs of colors (color cuts) to estimate the length of the secret message. The structure of the color cuts is measured using an entropy-like quantity R. The value R is evaluated for color pairs that are swapped during embedding and for the shifted (unrelated) color pairs. We prove that R is a quadratic function of the secret message length and, based on this fact, we can estimate the unknown message length from the stego image.

Although the basic idea behind Pairs Analysis can be applied to many steganographic paradigms and different image formats, it is ideally suited for palette images when the message bits are embedded as LSBs of indices to a pre-ordered palette. As an example of this embedding paradigm, we took the EzStego algorithm, version 2.0b4, and modified the embedding step so that message bits are embedded along a pseudo-random walk. We note that in this version of EzStego, the palette is pre-ordered to minimize color differences between neighboring colors (by solving the traveling salesman problem).

Pairs Analysis for the modified EzStego algorithm was tested on three databases of images in the GIF format: a) 180 images taken by a digital camera and converted to the GIF format, b) 24 cartoon images downloaded from various web sites, and c) 40 computer-generated fractal images (see Sections 5 and 6). The method as described in Section 4 had to be slightly modified for analysis of artificial images, such as cartoons or computer art (in Section 6). The experimental results are reported for each database separately. In conclusion, Pairs Analysis significantly outperforms previously proposed attacks that are based on the chi-square attack (Provos[5], Section 3). Also, for palette images, it produces more reliable results than the RS steganalysis[2].

Overall, Pairs Analysis can detect messages as small as 10% of the image capacity (100% = 1 bit per pixel) while accurately estimating the message length. It could also in principle be used for BMP images, but according to our tests, it does not perform as well as RS Steganalysis[2], that was specifically designed for detection of LSB embedding in 8-24 bit BMPs. RS Steganalysis also performs slightly better than Pairs-Analysis for gray scale images. Both methods are highly accurate for high-resolution good quality images. When combined, the two approaches cover both bitmap and palette image formats.

As part of our future effort, we plan to further investigate the modification as described in Section 6. Artificial images can have very singular histograms and, may, on a rare occasion, be misjudged by Pairs Analysis. Although manual inspection of the stego image can usually reveal the cause of the observed outlier, it would be useful to identify outliers automatically without relying on human intervention. Investigating higher-order entropies or more careful selection of the shifted pairs are possible avenues that deserve further study.

ACKNOWLEDGEMENTS

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under a research grant number F30602-02-2-0093. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Air Force Research Laboratory, or the U. S. Government.

REFERENCES

1. H. Farid and L. Siwei, “Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines”, *Pre-proceedings 5th Information Hiding Workshop*, Noordwijkerhout, Netherlands, Oct. 7–9, 2002.
2. J. Fridrich, M. Goljan, and R. Du, “Detecting LSB Steganography in Color and Gray-Scale Images”, *Magazine of IEEE Multimedia*, *Special Issue on Security*, October-November issue, 2001, pp. 22–28.
3. N. F. Johnson and S. Jajodia, “Steganalysis of Images Created Using Current Steganography Software,” Lecture Notes in Computer Science, vol.1525, Springer-Verlag, Berlin, 1998, pp. 273–289.
4. R. Machado, EzStego, http://www.stego.com.
5. N. Provos and Peter Honeyman, “Detecting Steganographic Content on the Internet”, *CITI Technical Report 01-11,* August 2001, submitted for publication.
6. A. Westfeld and A. Pfitzmann, "Attacks on Steganographic Systems," Lecture Notes in Computer Science, vol.1768, Springer-Verlag, Berlin, 2000, pp. 61–75.
7. A. Westfeld, Detecting Low Embedding rates. In: Petitcolas et al. (eds.): *Preproceedings 5th Information Hiding Workshop*, Noordwijkerhout, Netherlands, Oct. 7–9, 2002.
8. Steganography software for Windows, http://members.tripod.com/steganography/stego/software.html.