

On Estimation of Secret Message Length in LSB Steganography in Spatial Domain

Jessica Fridrich* and Miroslav Goljan

Department of Electrical and Computer Engineering
SUNY Binghamton, Binghamton, NY 13902-6000

ABSTRACT

In this paper, we present a new method for estimating the secret message length of bit-streams embedded using the Least Significant Bit embedding (LSB) at random pixel positions. We introduce the concept of a weighted stego image and then formulate the problem of determining the unknown message length as a simple optimization problem. The methodology is further refined to obtain more stable and accurate results for a wide spectrum of natural images. One of the advantages of the new method is its modular structure and a clean mathematical derivation that enables elegant estimator accuracy analysis using statistical image models.

1. INTRODUCTION

The purpose of steganography¹ is to hide the very presence of communication by embedding messages into innocuous-looking cover objects, such as digital images. The secret message is embedded in the original *cover* image by making slight modifications to it. As a result, the *stego image* is obtained.

The most important requirement for a steganographic system is *undetectability*: stego images should be statistically indistinguishable from cover images. In other words, there should be no artifacts in the stego image that could be detected by an attacker with probability better than random guessing, given the full knowledge of the embedding algorithm, including the statistical properties of the source of cover images, except for the stego key (Kerckhoff's principle).

The most popular and frequently used steganographic method is the Least Significant Bit embedding (LSB). It works by embedding message bits as the LSBs of randomly selected pixels. The pixel selection is usually determined by a secret stego key shared by the communicating parties. Today, the vast majority of steganographic programs⁹ available for download on the Internet use this technique (Steganos II, S-Tools 4.0, Steghide 0.3, Contraband Hell Edition, Wb Stego 3.5, Encrypt Pic 1.3, StegoDos, Wnstorm, Invisible Secrets Pro, and many others). The popularity of the LSB embedding is most likely due to its simplicity as well as the [false] early belief that modifications of pixel values by 1 in randomly selected pixels are undetectable because of the noise commonly present in all digital images of natural scenes.

During the last three years, many powerful steganalytic methods²⁻⁸ capable of detecting LSB embedding were proposed. Current state-of-the-art in detection of LSB embedding is represented by RS analysis⁶ and Sample Pairs analysis⁷. Both methods can detect stego images with an extremely high reliability. Moreover, both methods can very accurately estimate the number of changes due to embedding and thus estimate the secret message length.

In this paper, we present a new method for estimating the number of embedding changes due to LSB steganography in spatial image formats. One of the advantages of the new method is its clean and quite simple mathematical derivation and the possibility to engage statistical image models for obtaining upper bounds on the performance of the message length estimator. The detection algorithm is introduced in a series of theorems in Sec. 2. The same section also contains experimental results on databases of test images. In Sec. 3, we compare the performance of the new method with RS analysis and Sample Pairs analysis and then conclude the paper.

* fridrich@binghamton.edu; phone 1 607 777-2577; fax 1 607 777-4464; <http://www.ws.binghamton.edu/fridrich>

2. METHOD DESCRIPTION

2.1 Message length estimation as a minimization problem

Let $X = \{x_i\}_{i=1}^n$ be a column vector of integers in the range $[0, 255]$ representing a grayscale cover image with $n = M_x \times N_x$ pixels. The value of x_i after flipping its LSB will be denoted as \bar{x}_i , $\bar{x}_i = x_i + 1 - 2(x_i \bmod 2)$. Let $S = \{s_i\}$ denote the stego image after embedding qn bits, $0 \leq q \leq 1$, using LSB embedding in qn pixels randomly selected from the cover image X . Our task is to estimate q from the stego image S . Theorem 1 below forms the basis for constructing our message length estimators.

Theorem 1. *Let $S = \{s_i\}$ be the image X after flipping the LSBs of exactly $qn/2$ pixels from X (this is what happens on average after embedding qn random bits in X). Let $S^{(p)} = \{s_i^{(p)}\}$ be the “weighted” stego image, $s_i^{(p)} = s_i + p/2(\bar{s}_i - s_i)$, $0 \leq p \leq 1$, $i = 1, \dots, n$. Then,*

$$q = \arg \min_p E_0(p),$$

$$E_0(p) = \frac{1}{n} \sum_{i=1}^n (s_i^{(p)} - x_i)^2. \quad (1)$$

Proof.

$$\begin{aligned} E_0(p) &= \frac{1}{n} \sum_{i=1}^n (s_i^{(p)} - x_i)^2 = \frac{1}{n} \sum_{i=1}^n [s_i - x_i + p/2(\bar{s}_i - s_i)]^2 = \frac{1}{n} \sum_{s_i=x_i} p^2/4 + \frac{1}{n} \sum_{s_i=\bar{x}_i} (1-p/2)^2 = \\ &= \frac{1}{n} (n - qn/2) p^2/4 + \frac{1}{n} (1-p/2)^2 nq/2 = p^2(1-q/2)/4 + q(1-p/2)^2/2. \end{aligned}$$

Thus, $dE_0(p)/dp = p/2 - q/2$, which proves the fact that $E_0(p)$ reaches its unique minimum at $p=q$. \square

Theorem 1 essentially says that $S^{(q)}$ is the closest weighted stego image to X in the least square sense (in L^2 norm^{*}) among all weighted stego images $S^{(p)}$. This gives us an idea that we could estimate the unknown message length q from the stego image as a minimization problem. However, because the cover image pixel values x_i are not available for detection, we need to use an estimate of x_i as a function of s_i and its neighbors $N(s_i)$.

Assuming the one dimensional index i of the vector $\{x_i\}$ corresponds to the pixel x_{kj} in the image, $1 \leq k \leq M_x$, $1 \leq j \leq N_x$, the neighborhood N of x_i could be formed, for example, by x_{kj} together with its four closest pixels $N(x_{kj}) = \{x_{kj}, x_{k-1,j}, x_{k+1,j}, x_{k,j-1}, x_{k,j+1}\}$. If the central pixel $x_i = x_{kj}$ is excluded from the neighborhood, we use the subscript 0 to indicate this fact, $N_0(x_{kj}) = \{x_{k-1,j}, x_{k+1,j}, x_{k,j-1}, x_{k,j+1}\}$. Moreover, in this paper we accept the convention that $N(x_i)$ is the neighborhood of x_i in X and $N(s_i)$ is the neighborhood of s_i in S .

Any function $F: N(x) \rightarrow \mathbf{R}$, where \mathbf{R} denotes the set of real numbers, will be called a local estimator of x . One of the simplest estimators is the FIR spatially uniform filter:

$$F(N(x_{kj})) = \sum_{r=-L}^L \sum_{s=-L}^L c_{rs} x_{k+r,j+s}, \quad (2)$$

where c_{rs} are constants and L is a small positive integer. The simplest version of this local predictor is the arithmetic average of the four closest neighbors of x

$$F(N_0(x_{kj})) = 1/4(x_{k+1,j} + x_{k-1,j} + x_{k,j-1} + x_{k,j+1}). \quad (3)$$

This local estimator will be used in all our experiments.

* The L_2 norm is important because $S^{(q)}$ is *not* the closest in L_1 norm (in fact, $S^{(0)} = S$ is the closest in L_1)

Theorem 2 below is an analog of Theorem 1 with the values of the cover image x_i replaced with its local estimate calculated from the neighborhood $N_0(s_i)$ in the stego image. This theorem is suitable for practical calculations and its performance is evaluated in Sec. 2.1.1.

Theorem 2. *Keeping the notation from Theorem 1, let F be a local estimator of x on $N_0(x)$ and let $S=\{s_i\}$ be the image X after flipping the LSBs of exactly $qn/2$ pixels from X . Then*

$$\bar{q} = \arg \min_p E_1(p) = -\frac{2}{n} \sum_{i=1}^n [s_i - F(N(s_i))](\bar{s}_i - s_i) \quad (4)$$

is an estimate of q satisfying $\bar{q} = q + r(X, S)$, where

$$E_1(p) = \frac{1}{n} \sum_{i=1}^n [s_i^{(p)} - F(N_0(s_i))]^2 \quad (5)$$

and $r(X,S)$ is a composite error term

$$r(X, S) = r_1(X, S) + r_2(X, S) = \frac{1}{n} \sum_{i=1}^n [x_i - F(N_0(x_i))](\bar{s}_i - s_i) + \frac{1}{n} \sum_{i=1}^n [F(N_0(x_i)) - F(N_0(s_i))](\bar{s}_i - s_i).$$

Proof.

$E_1(p) = \frac{1}{n} \sum_{i=1}^n [s_i^{(p)} - F(N(s_i))]^2 = \frac{1}{n} \sum_{i=1}^n [s_i + p/2(\bar{s}_i - s_i) - F(N(s_i))]^2$. Thus, the minimum of $E_1(p)$ is reached for p that satisfies the following equation

$$dE_1(p) / dp = \frac{2}{n} \sum_{i=1}^n [s_i + p/2(\bar{s}_i - s_i) - F(N(s_i))](\bar{s}_i - s_i) / 2 = 0,$$

which gives (4) after using the fact that $(\bar{s}_i - s_i)^2 = 1$ for all i . To prove the rest, we write (4) as

$$\begin{aligned} \bar{q} &= -\frac{2}{n} \sum_{i=1}^n [s_i - x_i + x_i - F(N_0(x_i)) + F(N_0(x_i)) - F(N_0(s_i))](\bar{s}_i - s_i) = \\ &= q + \frac{1}{n} \sum_{i=1}^n [x_i - F(N_0(x_i))](\bar{s}_i - s_i) + \frac{1}{n} \sum_{i=1}^n [F(N_0(x_i)) - F(N_0(s_i))](\bar{s}_i - s_i). \end{aligned} \quad (6)$$

In the last expression, we used the fact that for $(1-q/2)n$ pixels $s_i=x_i$ and for $qn/2$ pixels $s_i=\bar{x}_i$. \square

Next, we estimate both error terms based on some assumptions about natural images. We will show that both error terms are small for typical images of natural scenes. Thus, Equation (4) can be used to estimate the number of flipped pixels $qn/2$.

The term $e_i = x_i - F(N_0(x_i))$ is the error of the local estimator F . Because natural images contain noise, e_i should have little correlation with the LSB of x_i . Let us model both e_i and $\text{LSB}(x_i)$ as i.i.d. random variables that are independent of each other. Because e_i is the result after applying to the cover image a filter that has a high-pass character, the Probability Distribution Function (PDF) of e_i can be well modeled using the generalized Gaussian distribution with zero mean and variance $\sigma_e^2 = \text{Var}(e_i)$. Assuming the message bits and the $\text{LSB}(x_i)$ are independent of each other, the variable $\bar{s}_i - s_i$ is also a binary random variable with values from the set $\{-1, 1\}$ that is independent of e_i . Thus, by the Central Limit Theorem

$$E(r_1) = 1/n \times \sum_i E(e_i(\bar{s}_i - s_i)) = 0, \text{ and } \text{Var}(r_1) = \text{Var}(1/n \times \sum_i e_i(\bar{s}_i - s_i)) = \sigma_e^2/n.$$

For our database of 60 test images, we observed $\sigma_e \leq 18$.

The second error term is a correlation between $\bar{s}_i - s_i$ and $m_i = F(N_0(x_i)) - F(N_0(s_i))$. Assuming the local estimator F is linear, $\{m_i\}$ is the filtered stego signal. Again, since the message bits are expected to be independent of the image features, we

can make a feasible assumption that $\bar{s}_i - s_i$ and m_i are independent. If F is of the type (2), m_i will have a symmetric PDF with variance $Var(m_i) < 1$. Thus,

$$E(r_2) = 1/n \times \sum_i E(m_i(\bar{s}_i - s_i)) = 0, \quad Var(r_2) = Var(1/n \times \sum_i m_i(\bar{s}_i - s_i)) < 1/n.$$

Finally, we have an estimate for the error term $r(X, S) < (\sigma_e^2 + 1)/n$.

2.1.1 Experimental results

To evaluate the proposed estimator, we have prepared a database of 60 images all taken with the Canon G2 digital camera at the highest JPEG quality setting. The images were resampled from their original 2272×1704 resolution to 800×600 and converted to grayscale. The database is very diverse containing images of scenes with rich textures, images with a little texture and large flat areas (e.g., a night shot of fireworks, or a few images with a flat sky), one out-of-focus image, and images taken at different zoom levels (landscape panoramas and close-ups). The reason for down sampling the images was twofold – first to remove possible interference of JPEG compression⁵ and to speed up the experimental tests.

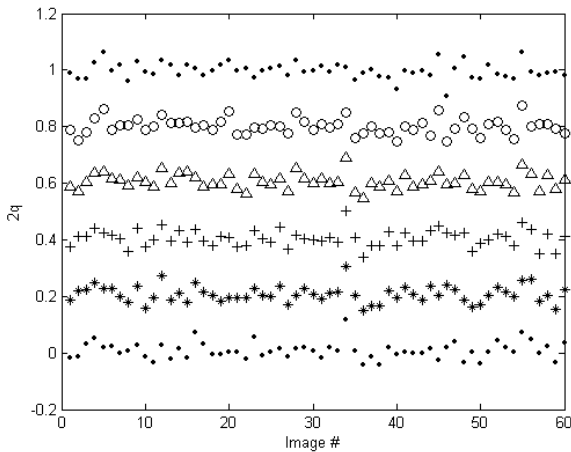
For each image, we have generated five stego images by randomizing the LSBs of a specified number of randomly selected pixels (10%, 20%, 30%, 40%, and 50%, corresponding to $q = 0.2, 0.4, 0.6, 0.8, 1.0$). So, for example, for $q = 1.0$, the modifications correspond to embedding a random bit-stream of maximal length 1 bit per pixel (bpp) into the whole image (i.e., randomizing 50% of pixels). We note that the random walk was generated using the Matlab PRNG and was the same for each image.

To evaluate the performance of the estimator, we calculated the mean $\mu(q)$ and the standard deviation $\sigma(q)$ for each q over all 60 stego images. In other words, let $p(i, q)$ be the estimate of q for the i -th image. Then,

$$\mu(q) = \frac{1}{60} \sum_{i=1}^{60} p(i, q),$$

$$\sigma^2(q) = \frac{1}{60} \sum_{i=1}^{60} [p^2(i, q) - \mu^2(q)].$$

In our experiments in this paper, we have used the local predictor (3). At this point, we would like to stress that it is important that the local predictor $F(x_i)$ does not depend on x_i itself because otherwise the local estimator error e_i would be correlated with $\bar{s}_i - s_i$, which would cause a significant increase of the error r_1 (we have experimentally observed this phenomenon).



q	$\mu(q)$	$\sigma(q)$
0	0.0078	0.0300
0.2	0.2076	0.0297
0.4	0.4058	0.0295
0.6	0.6063	0.0273
0.8	0.8003	0.0282
1.0	0.9969	0.0279

Fig. 1. Estimates of the number of changed pixels on a database of 60 images.

The result of the experiments is shown in Fig. 1. We can see that the variance of the estimate is approximately the same for all q . This variance is mostly due to the first error term r_1 . The more texture the cover image has, the larger this error becomes because the local estimator F will perform increasingly poorly. It turns out that the error term r_1 (and thus the estimator variance) can be made smaller by introducing local weights into expression (5). This issue is investigated in the next section.

2.2 Introducing local weights

As mentioned in the previous section, the local estimator F will be more accurate in areas of the image that are flat with less texture and less accurate in highly textured areas with many edges. Thus, it can be expected that better results might be obtained by introducing weights into the expression for $E_i(p)$:

$$E_2(p) = \sum_{i=1}^n w_i [s_i^{(p)} - F(N_0(s_i))]^2, \quad (7)$$

where w_i are weights calculated for each pixel s_i from the stego image and $\sum_i w_i = 1$. At this point, it is not clear how the weights will influence the minimization problem and how to choose them to construct an accurate message length estimator. Before we investigate this issue, we state an auxiliary lemma.

Lemma 1. *Let $Z = \{z_1, \dots, z_n\}$ be a set of n real numbers and $I = \{1, \dots, n\}$. For a fixed $k \leq n$, let ω_k be a random variable defined as the sum of k randomly selected elements in Z :*

$$\omega_k = \sum_{\substack{i \in I_k, I_k \subset I \\ I_k \text{ random}, |I_k|=k}} z_i.$$

Then,

$$E(\omega_k) = \frac{k}{n} \sum_{i=1}^n z_i \quad (8)$$

$$\text{Var}(\omega_k) = \frac{k \left(1 - \frac{k}{n}\right)}{n-1} \left(n \sum_{i=1}^n z_i^2 - \left(\sum_{i=1}^n z_i \right)^2 \right). \quad (9)$$

Furthermore, assuming $\sum_{i=1}^n z_i = 1$, the variance $\text{Var}(\omega_k)$ is minimal when $z_i = 1/n$ for all i .

Proof.

The probability $P(I_k)$ that a given subset $I_k \subset I$ of k indices is selected is $[n(n-1)\dots(n-k+1)]^{-1}$. Thus,

$$\begin{aligned} E(\omega_k) &= \sum_{I_k} P(I_k) \sum_{i \in I_k} z_i = \sum_{I_k} \frac{1}{n(n-1)\dots(n-k+1)} \sum_{i \in I_k} z_i = \\ &= \frac{1}{n(n-1)\dots(n-k+1)} \sum_{i=1}^n z_i k(n-1)\dots(n-k+1) = \frac{k}{n} \sum_{i=1}^n z_i, \end{aligned}$$

because each index i belongs to $k(n-1)\dots(n-k+1)$ different ordered tuples I_k . To calculate the variance, we write

$$\begin{aligned}
E(\omega_k^2) &= \sum_{I_k} P(I_k) \left(\sum_{i \in I_k} z_i \right)^2 = \frac{1}{n(n-1)\dots(n-k+1)} \sum_{I_k} \left(\sum_{i \in I_k} z_i^2 + \sum_{\substack{k, j \in I_k \\ k \neq j}} z_k z_j \right) = \\
&= \frac{k}{n} \sum_{i=1}^n z_i^2 + \frac{1}{n(n-1)\dots(n-k+1)} \sum_{\substack{k, j \in I_k \\ k \neq j}} z_k z_j k(k-1)(n-2)\dots(n-k+1) = \\
&= \frac{k}{n} \sum_{i=1}^n z_i^2 + \frac{k(k-1)}{n(n-1)} \sum_{\substack{k, j \in I_k \\ k \neq j}} z_k z_j = \frac{k}{n} \sum_{i=1}^n z_i^2 + \frac{k(k-1)}{n(n-1)} \left[\left(\sum_{i=1}^n z_i \right)^2 - \sum_{i=1}^n z_i^2 \right] = \\
&= \left(\frac{k}{n} - \frac{k(k-1)}{n(n-1)} \right) \sum_{i=1}^n z_i^2 - \frac{k}{n} \left(\frac{k}{n} - \frac{k-1}{n-1} \right) \left(\sum_{i=1}^n z_i \right)^2 + (E(\omega_k))^2 = \\
&= \frac{k}{n} \frac{n-k}{n-1} \sum_{i=1}^n z_i^2 - \frac{k}{n} \frac{n-k}{n(n-1)} \left(\sum_{i=1}^n z_i \right)^2 + (E(\omega_k))^2 = \\
&= \frac{\frac{k}{n} \left(1 - \frac{k}{n} \right)}{n-1} \left(n \sum_{i=1}^n z_i^2 - \left(\sum_{i=1}^n z_i \right)^2 \right) + (E(\omega_k))^2,
\end{aligned}$$

which proves the expression for variance. To prove the rest, we use the method of Lagrange multipliers and find the minimum of $Var(\omega_k)$ under the condition $\sum_{i=1}^n z_i = 1$. Thus, assuming we have a function $f(z, \lambda)$

$$f(z, \lambda) = n \sum_{i=1}^n z_i^2 - 1 - \lambda \left(\sum_{i=1}^n z_i - 1 \right), \text{ we can write}$$

$$\frac{\partial f}{\partial z} (z, \lambda) = 2nz_i - \lambda, \text{ which implies } z_i = \lambda / 2n \text{ for all } i, \text{ thus proving Lemma 1. } \square$$

Theorem 3. Keeping the notation from Theorem 2, let F be a local estimator of x on $N_0(x)$ and let $S = \{s_i\}$ be the image X after flipping the LSBs of exactly $qn/2$ pixels from X . Furthermore, let $\{w_i\}$ be the set of n non-negative weights with $\sum_i w_i = 1$. Then

$$\bar{q} = \arg \min_p E_2(p) = -2 \sum_{i=1}^n w_i [s_i - F(N(s_i))] (\bar{s}_i - s_i) \quad (10)$$

is an estimate of q satisfying $\bar{q} = 2 \sum_{x_i = \bar{s}_i} w_i + r(X, S)$, where $r(X, S)$ is a composite error term

$$r(X, S) = r_1(X, S) + r_2(X, S) = \sum_{i=1}^n w_i [x_i - F(N_0(x_i))] (\bar{s}_i - s_i) + \sum_{i=1}^n w_i [F(N_0(x_i)) - F(N_0(s_i))] (\bar{s}_i - s_i).$$

Proof.

Equation (10) is obtained in the same way as in the proof of Theorem 2 by differentiating E_2 and solving for its minimum. To derive the expression for the error term r , we write (10) as

$$\begin{aligned}
\bar{q} &= -2 \sum_{i=1}^n w_i [s_i - x_i + x_i - F(N_0(x_i)) + F(N_0(x_i)) - F(N_0(s_i))] (\bar{s}_i - s_i) = \\
&= 2 \sum_{x_i = \bar{s}_i} w_i + \sum_{i=1}^n w_i [x_i - F(N_0(x_i))] (\bar{s}_i - s_i) + \sum_{i=1}^n w_i [F(N_0(x_i)) - F(N_0(s_i))] (\bar{s}_i - s_i).
\end{aligned}$$

Because there are exactly $nq/2$ pixels with flipped LSBs in the stego image (with $s_i = \bar{x}_i$), by Lemma 1, the expected value of the first term (over different sets of randomly selected modified pixels) is $2 \frac{nq/2}{n} \sum_{i=1}^n w_i = q$. \square

Intuitively, the local weights w_i should be smaller in highly textured areas and larger in smoother areas where the local estimator F has worse performance. On the other hand, to keep the variance of the estimate (10) low, by Lemma 1 we desire the weights to have as little variance as possible. To resolve these conflicting requirements, we should take into consideration statistical properties of natural images in small neighborhoods. Because the term $s_i - F(N_0(s_i))$ is well modeled using a generalized Gaussian distribution, we have experimented with the following form of the weights in order to “normalize” the residual signal $s_i - F(N_0(s_i))$:

$$w_i = \frac{A}{1 + \sigma_i^\alpha},$$

where σ_i is the local variance at pixel x_i calculated from its 4 closest neighbors and α is a parameter. The constant A is a normalization factor needed to satisfy $\sum_i w_i = 1$. The additive constant 1 in the denominator prevents obtaining large values of weights in flat areas of the image. In our experiments, the best performance over all images and message lengths was obtained for $\alpha = 1$.

Similar to the argument for the design of the local estimator F , we have observed that it is very important that the local weights w_i not depend on the central pixel x_i , otherwise correlation between the local estimator error e_i and $\bar{s}_i - s_i$ would be created. This correlation would make the error terms r_1 and r_2 quite significant, which in turn would lead to inaccurate results (this has been experimentally confirmed).

2.2.1 Experimental results

We have tested the estimator (10) on the same image database of 60 images as in Sec. 2.1.1. The results are shown in Fig. 2. We can clearly see that the estimates now have a significantly smaller variance than before. The standard deviation is smaller by a factor of 3 for $q=1$ and by a factor of 2 for $q=0$. The increased positive estimation bias is due to three outliers (image No. 2, 11, and 13). The estimates for these images exhibit quite a large error for $q=0.2, 0.4, 0.6,$ and 0.8 . Without these three outliers, the estimator bias becomes better than for the case without weights.

Overall, the results of the weighted estimator are more stable and exhibit smaller variance. Thus, to further improve the results, we investigate the three outliers and try to identify a way to correct them to obtain more reliable results for all images.

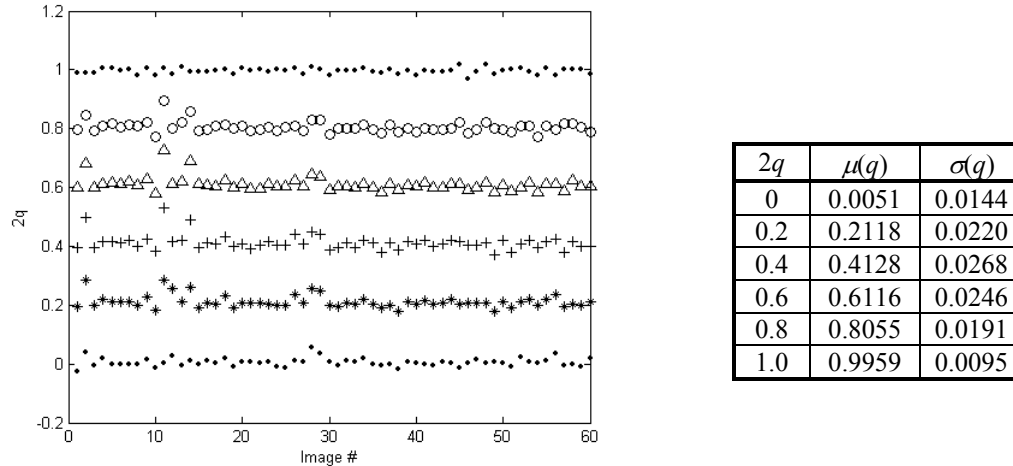


Fig. 2. Local weights. Estimates of the number of changed pixels on a database of 60 images.

2.3 Outlier analysis

In this section, we analyze the outliers from Fig. 2 and design a procedure that improves the estimates for such images. We have chosen the biggest outlier, image No. 11, for our analysis. The analysis of the errors r_1 and r_2 revealed a surprising fact. The estimation error was due to the error term r_2 .

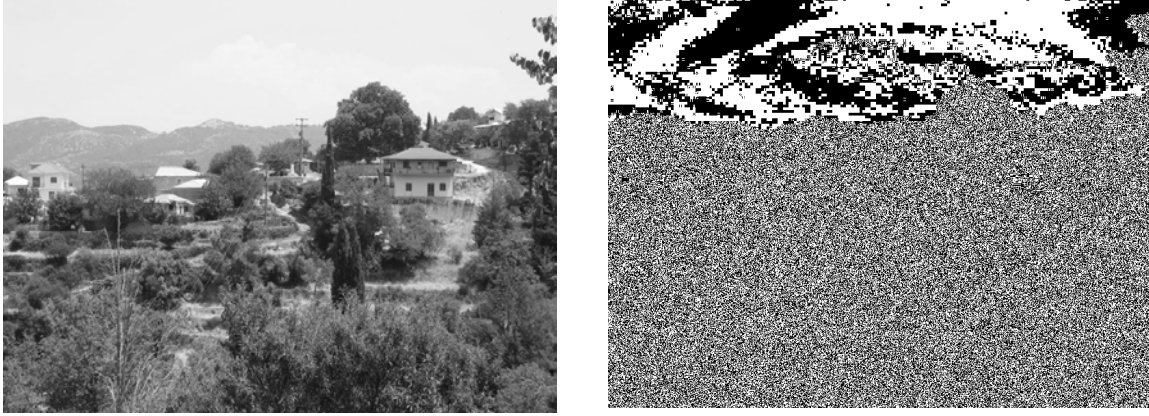


Fig. 3. Outlier image and its LSB plane.

Image No. 11 has a large area in the sky where there is very little variation in pixel values. Although it is rather unlikely that such large areas of constant color will be found in a natural image, they may arise due to color saturation in over exposed parts of the image, such as the bright sky in image No. 11. Why such areas cause an increase in the second error term r_2 is explained next.

Let us call a given pixel x_{kj} *flat* if its closest neighboring pixels have the same values as x_{kj} , $x_{kj} = x_{k+1j} = x_{k-1j} = x_{kj-1} = x_{kj+1}$. The set of all flat pixels in the cover image will be denoted as M_0 , $|M_0| = m_0$ is the cardinality of M_0 . Let us now assume that $qn/2$ pixels are flipped in the stego image. Our task is to estimate that part of the error term r_2 that involves flat pixels:

$$\sum_{i=1}^n w_i [F(N_0(x_i)) - F(N_0(s_i))] (\bar{s}_i - s_i). \quad (11)$$

Assuming the local estimator F is given by (3), for a flat pixel x_{kj} we have

$$F(N_0(x_{kj})) - F(N_0(s_{kj})) = 1/4(x_{k+1j} - s_{k+1j} + x_{k-1j} - s_{k-1j} + x_{kj-1} - s_{kj-1} + x_{kj+1} - s_{kj+1}).$$

Thus $F(N_0(x)) - F(N_0(s)) = a\xi$, where $\xi \in \{0, 1/4, 1/2, 3/4, 1\}$ and $a = x - \bar{x}$. Because the local predictor F and the weights w_i do not depend on the central pixel, we can assume that $F(N_0(x_i)) - F(N_0(s_i))$ and $\bar{s}_i - s_i$ are independent random variables

$$\begin{aligned} E\{w_i [F(N_0(x)) - F(N_0(s))] (\bar{s} - s)\} &= E\{w_i [F(N_0(x)) - F(N_0(s))]\} \times E(\bar{s} - s) = \\ &= E\{w_i [F(N_0(x)) - F(N_0(s))]\} \times [-a(1 - q/2) + aq/2] = (q-1) E\{w_i [F(N_0(x)) - F(N_0(s))]\}. \end{aligned}$$

The variable ξ attains its values with the following probabilities and weights w_i

ξ	Probability	$w(\xi)$
0	$(1 - q/2)^4$	A
1/4	$4 \times q/2 (1 - q/2)^3$	$A / (1 + (3/16)^{1/2})$
1/2	$6 \times (q/2)^2 (1 - q/2)^2$	$A / (1 + (4/16)^{1/2})$
3/4	$4 \times (q/2)^3 (1 - q/2)$	$A / (1 + (3/16)^{1/2})$
1	$(q/2)^4$	A

Table 1. The probabilities and values of ξ .

To explain how the table was obtained, we elaborate on the second row. The value $\xi=1/4$ occurs when exactly one neighbor in $N_0(x)$ is flipped. Thus, $F(N_0(x))-F(N_0(s)) = 1/4(x+x+x+x) - 1/4(\bar{x}+x+x+x) = 1/4a$. This happens with probability $4 \times q/2(1-q/2)^3$ (one pixel flipped and three unchanged). To obtain the local weight, we calculate the local variance $\sigma^2 = Var(\{\bar{x}, x, x, x\}) = Var(\{\bar{x}-x, 0, 0, 0\}) = 1/4(-a)^2 - (-1/4a)^2 = 3/16$. The local weight at x is $w=A/(1+\sigma)=A/(1+(3/16)^{1/2})$.

Finally, we can write (11) as

$$E \left(\sum_{x_i \in M_0} w_i [F(N_0(x_i)) - F(N_0(s_i))](\bar{s}_i - s_i) \right) = m_0(q-1) \sum_{\xi} w(\xi) \xi P(\xi),$$

where the variable ξ goes through all its five values listed in Table 1. After some algebra, we obtain

$$E \left(\sum_{x_i \in M_0} w_i [F(N_0(x_i)) - F(N_0(s_i))](\bar{s}_i - s_i) \right) = m_0 q(q-1)(c_1 + c_2 q + c_3 q^2 + c_4 q^3),$$

where

$$\begin{aligned} c_1 &= 1/2w(1/4) \\ c_2 &= 3/4[-w(1/2)+w(1/4)] \\ c_3 &= 3/4[w(1/2)-2w(1/4)+w(3/4)] \\ c_4 &= 1/16[-w(1/2)+3w(1/4)-3w(3/4)+w(1)]. \end{aligned}$$

Now, we can clearly see why the second error term r_2 may become large in images with many flat pixels. In flat areas, the weights w_i are the largest, which makes r_2 more significant than for uniform (no) weights. We can also see that (11) is zero for $q=0$ or $q=1$ and largest for $q \approx 0.5$, which is in agreement with Fig. 2.

This analysis also gives us an idea how to correct our estimation process to eliminate the outliers caused by large areas of flat pixels. If the cover image contains m_0 flat pixels, after flipping $nq/2$ pixels, only a certain fraction of flat pixels will remain. A flat pixel stays flat if and only if either none of the five pixels forming the flat area is flipped or when all are flipped. Thus, the number of flat pixels remaining from the flat area M_0 is

$$m_q = m_0[(1-q/2)^5 + (q/2)^5].$$

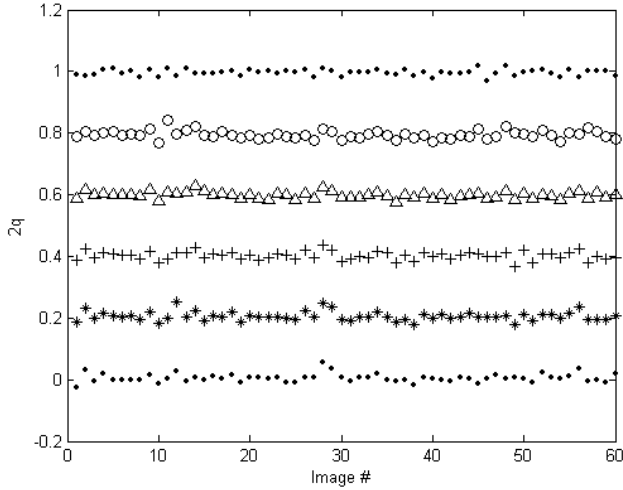
Knowing the number of flat pixels in the stego image m_q , we can estimate m_0 as $m_q / [(1-p/2)^5 + (p/2)^5]$, where p is the estimate obtained using (10). The contribution r_f of flat pixels to r_2 can then be estimated as

$$r_f = \frac{m_q p(p-1)(c_1 + c_2 p + c_3 p^2 + c_4 p^3)}{(1-p/2)^5 + (p/2)^5}.$$

Thus, to obtain the corrected value of the estimate p from (10), we add the contribution r_f

$$p := p + r_f. \quad (12)$$

We acknowledge that some flat pixels in the stego image may be created by the embedding process from pixels that were ‘‘nearly flat’’ in the cover image. So, the correction term r_f may not be completely accurate. However, according to our experiments, this contribution is negligible and the corrected estimate (12) successfully removes the outliers. In Fig. 4, we show the experimental results for our test database of 60 images. The correction term r_f has been added for each image. The estimator bias is now very small and the standard deviation of the estimates is also improved.



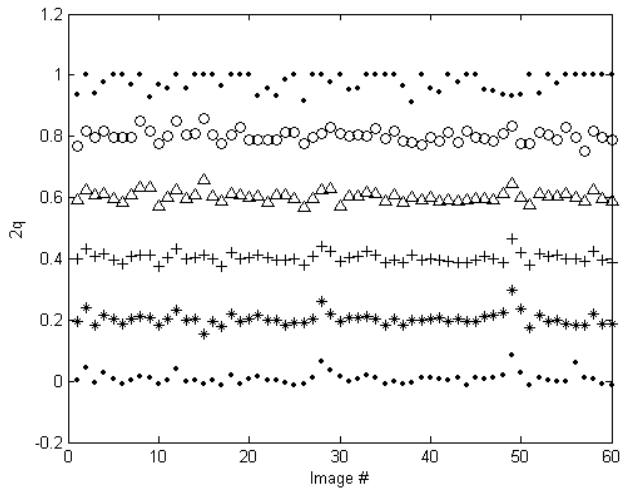
$2q$	$\mu(q)$	$\sigma(q)$
0	0.0048	0.0138
0.2	0.2056	0.0150
0.4	0.4021	0.0135
0.6	0.5987	0.0108
0.8	0.7944	0.0134
1.0	0.9956	0.0102

Fig. 4. Local weights with correction. Estimates of the number of changed pixels on a database of 60 images.

3. COMPARISON AND CONCLUSIONS

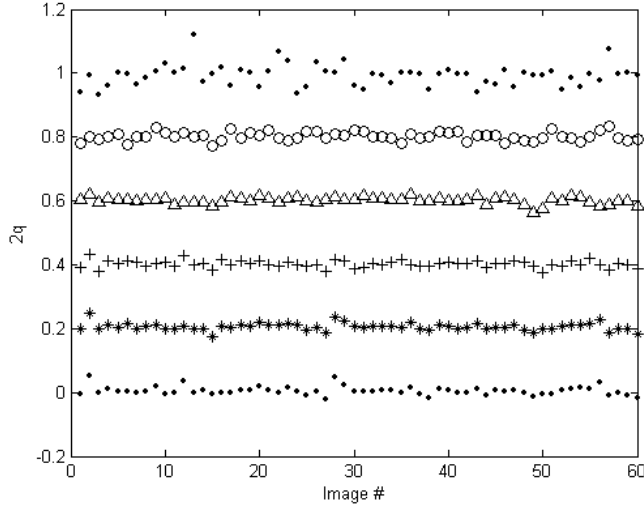
To evaluate the performance of the new proposed detection method, we have compared it to the two most accurate and reliable techniques available today – the RS analysis⁶ (RSA) and Sample Pairs analysis² (SPA). Both methods are related in the sense that the early version of SPA was, in fact, a special case of RSA. SPA also provided more insight into RSA and justified some of its experimental assumptions.

First, we have applied both detection techniques to the same set of 60 grayscale test images (see Fig. 5. for SPA and Fig. 6. for RSA). Comparing the detected average message length and its standard deviation, we can conclude that the new method has a better performance than SPA for all message lengths. RSA gives comparable or slightly better results for messages shorter than 80% of capacity. For messages close to 100%, the new method has significantly better performance than both RSA and SPA.



$2q$	$\mu(q)$	$\sigma(q)$
0	0.0081	0.0193
0.2	0.2031	0.0214
0.4	0.4035	0.0161
0.6	0.6031	0.0172
0.8	0.8019	0.0205
1.0	0.9762	0.0281

Fig. 5. Results of Sample Pairs analysis (for comparison).



$2q$	$\mu(q)$	$\sigma(q)$
0	0.0041	0.0135
0.2	0.2058	0.0116
0.4	0.4024	0.0110
0.6	0.6021	0.0104
0.8	0.8025	0.0135
1.0	0.9920	0.0348

Fig. 6. Results of RS analysis (for comparison).

For a practical steganalytic method, it is important how it performs for short messages and how it evaluates cover images rather than its performance for images that are almost fully embedded. It is obviously less serious if we detect 99% message instead of a 95% message than when a steganalytic routine indicates a presence of a 5% message in a cover image. Thus, we have looked at the performance of RSA, SPA, and the new method for 5% messages (0.05 bits per sample). The test images were grayscale images from the database of Philip Greenspun (www.greenspun.com) consisting of 1816 24-bit color images stored in the JPEG format. The images were converted to grayscale and slightly cropped to remove the black frame that was probably added for esthetic reasons. The images were embedded with a 5% message and fed into all three detectors. The results were evaluated using the Receiver Operating Characteristic (ROC) curve that shows the detection percentage as a function of false positives. As can be seen from Fig. 7., all three detection methods have excellent and quite similar performance. To better see the performance for short messages, we have plotted the ROC curve in a semi-logarithmic fashion in Fig. 8. The figure clearly shows that for very short messages (less than 3%) the new method performs better than RSA or SPA.

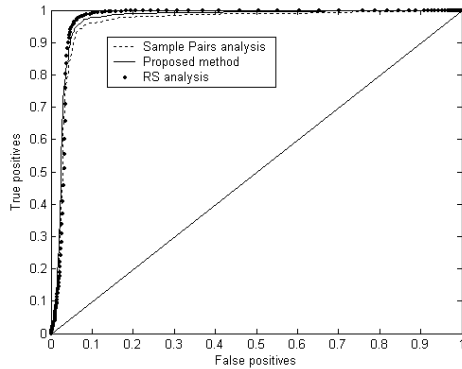


Fig. 7. ROC curve for 5% messages (0.05 bits per pixel) for RS analysis, Sample Pairs analysis, and the proposed method for 1800 test images (Greenspun database).

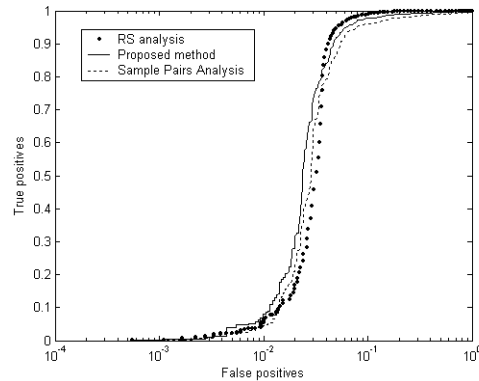


Fig. 8. Semi-logarithmic plot of the ROC curve from Fig. 7. to see the methods' performance for short messages and a low false positive rate.

In conclusion, we summarize that in this paper, we proposed a new method for detection and message length estimation for LSB embedding. The methodology is well founded and built through a series of improvements. One of the advantages of the new method is its clean and quite simple mathematical derivation and the possibility to engage statistical image models for obtaining upper bounds on the performance of the message length estimator. We acknowledge that there are many elements in the detection algorithm that can be changed or replaced with other elements.

As opposed to RSA or SPA, where we did not see much space for further improvement, the new detection scheme offers at least two areas that might lead to even more accurate and reliable detection – the local predictor and the local weights. We have experimented with other linear predictors as well as predictors based on the Wiener filter. It was a rather surprising finding that more sophisticated predictors did not in general lead to better detection schemes. The local weights should be derived from local statistical models of images.

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. Special thanks belong to Xiaolin Wu and Zhe Wang for kindly providing their code for Sample Pairs analysis and to Hany Farid for providing the Greenspun image database.

REFERENCES

1. R.J. Anderson and F.A.P. Petitcolas, “On the Limits of Steganography”, *IEEE Journal of Selected Areas in Communications*, Special Issue on Copyright and Privacy Protection, vol. **16**(4), pp. 474–481, 1998.
2. R. Chandramouli and N. Memon, “Analysis of LSB Based Image Steganography Techniques”, *Proc. of ICIP 2001*, Thessaloniki, Greece, October 7–10, 2001.
3. S. Dumitrescu, Wu Xiaolin, and Zhe Wang, “Detection of LSB Steganography via Sample Pair Analysis”, In *LNCS* vol. 2578, Springer-Verlag, New York, pp. 355–372, 2003.
4. H. Farid and L. Siwei, “Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines”, In *LNCS* vol. 2578, Springer-Verlag, New York, pp. 340–354, 2003.
5. J. Fridrich, M. Goljan, and R. Du, “Steganalysis based on JPEG compatibility”, *SPIE Multimedia Systems and Applications IV*, Denver, CO, August 20–24, 2001.
6. 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, pp. 22–28, 2001.
7. N. Provos and P. Honeyman, “Detecting Steganographic Content on the Internet”, CITI Technical Report 01-11, 2001.
8. A. Westfeld and A. Pfitzmann, “Attacks on Steganographic Systems”, In: *LNCS* vol.1768, Springer-Verlag, Berlin, pp. 61–75, 2000.
9. Steganography software for Windows, <http://members.tripod.com/steganography/stego/software.html>.