Maximum likelihood estimation of length of secret message embedded using $\pm K$ steganography in spatial domain

Jessica Fridrich[a], David Soukal[b], Miroslav Goljan[a]

[a]Department of Electrical and Computer Engineering;
[b]Department of Computer Science,
SUNY Binghamton, Binghamton NY 13902-6000, USA

ABSTRACT

In this paper, we propose a new method for estimating the number of embedding changes for non-adaptive $\pm K$ embedding in images. The method uses a high-pass FIR filter and then recovers an approximate message length using a Maximum Likelihood Estimator on those stego image segments where the filtered samples can be modeled using a stationary Generalized Gaussian random process. It is shown that for images with a low noise level, such as decompressed JPEG images, this method can accurately estimate the number of embedding changes even for $K = 1$ and for embedding rates as low as 0.2 bits per pixel. Although for raw, never compressed images the message length estimate is less accurate, when used as a scalar parameter for a classifier detecting the presence of $\pm K$ steganography, the proposed method gave us relatively reliable results for embedding rates as low as 0.5 bits per pixel.

Keywords: Steganalysis, steganography, $\pm K$ embedding, MLE

1. INTRODUCTION

Steganography is the art of invisible communication. Its purpose is to hide the very presence of communication by embedding messages into innocuous-looking cover objects. Each steganographic communication system consists of an embedding algorithm and an extraction algorithm. To accommodate a secret message in a digital image, the original cover image is slightly modified by the embedding algorithm. 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 (Kerckhoffs' principle). For a more exact treatment of the concept of steganographic security, the reader is referred, for example, to Ref. 1, 2.

By far 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, a fairly large portion of steganographic programs[3] available for download on the Internet use this technique (Steganos II, S-Tools 4.0, Steghide 0.3, Contraband Hell Edition, Web Stego 3.5, EncryptPic 1.3, StegoDos, Winstorm, Invisible Secrets Pro, and many others). The popularity of LSB embedding is most likely due to its simplicity as well as the (false) early belief that modifications of LSBs in randomly selected pixels are undetectable because of the noise commonly present in digital images of natural scenes. However, flipping the bits of the LSB plane does not occur naturally. The even pixel values are either unmodified or increased by one, while odd values are either decreased by one or left unchanged. This imbalance in the embedding distortion was recently utilized to mount successful attacks.[4–6] The current state-of-the-art in detection of LSB embedding is represented by RS analysis,[5] Sample

Further author information: (Send correspondence to J. F.)
J. F.: E-mail: fridrich@binghamton.edu, Telephone: 1 607 777-2577, Fax: 1 607 777-4464; SUNY Binghamton, T. J. Watson School of Engineering, Department of Electrical and Computer Engineering, Binghamton 13902-6000, NY, USA; http://www.ws.binghamton.edu/fridrich

Pairs analysis,[6] and their improved versions.[7,8] These methods can detect stego images with an extremely high reliability and accurately estimate the number of embedding changes.

A better approach than manipulating bit planes is embedding by adding noise of specific properties. The early example of this approach is the work of Marvel,[9] Alturki,[10] and Sharp.[11] Recently, Stochastic Modulation[12] was proposed in which the act of embedding is realized by superimposing noise with an arbitrary (user-selected) probability distribution. This method attempts to mask the act of embedding as adding a device noise of specific properties. A special case of this method is, what we call, $\pm K$ embedding that is investigated in this paper. In $\pm K$ embedding, some pixel values are left unchanged, while others are either increased or decreased by K. The modifications can be either content independent or adapted to the image content.

The $\pm K$ embedding for $K = 1$ is a trivial generalization of LSB embedding. Instead of flipping the LSB, the sender increases or decreases the pixel value by one to match its LSB with the message bit. This seemingly innocent modification of the LSB embedding is significantly harder to detect because the pixel values are no longer paired. As a result, none of the existing attacks on LSB embedding can be adapted to attack ± 1 embedding.

One of the first papers on detection of embedding by noise adding is the paper by Harmsen.[13] The detection relies on the fact that adding noise to the cover image smoothes out its histogram. This method seems to work reasonably well for images that have low level of high frequency noise, such as decompressed JPEG images. It is not clear, however, if one can find a universal threshold distinguishing cover and stego images for a sufficiently wide class of images (e.g., for never compressed images, scans, or resampled images) and whether the method can reliably estimate the number of embedding modifications, which is an important piece of knowledge for the steganalyst. Also, the detection is less reliable for grayscale images.

A different method for detection of steganography based on noise adding was proposed by Westfeld.[14] Noise adding creates many (up to 26) neighbors for each color present in the cover image. In decompressed JPEG images and images with a low level of noise, each color typically has no more than 10–15 neighboring colors. However, ± 1 embedding increases the number of neighbors quite significantly even for low embedding rates. Thus, by counting the number of neighbors for each unique color in the image, one can detect the presence of ± 1 steganography. This method seems to be limited to color images and it is not known if it can estimate the number of embedding changes. Images with a large noise component, such as never compressed images, scans of photographs, or certain resampled images are often misdetected as false positives.

In this paper, we propose a new method for detection of non-adaptive $\pm K$ embedding that can also estimate the number of embedding changes, which is proportional to the length of the embedded message. Thus, this paper can be thought of as an extension of our previous work on quantitative steganalysis capable of detecting the embedded message length.[15] Also, this method works for color as well as grayscale images.

The method uses a simple denoising filter and then applies a Maximum Likelihood Estimator on pixels from those parts of the stego image where the filtered pixels can be modeled using a stationary Generalized Gaussian random process. In Section 2, we describe the new approach and in Section 3 we evaluate its performance on decompressed JPEG images and never compressed images obtained using digital cameras. The last Section 4 concludes the paper.

2. PROPOSED METHOD

Our method estimates the unknown message length p using a Maximum Likelihood Estimator. The estimator is applied to the high-pass filtered image data, which we model as a convolution of a discretized Generalized Gaussian variable with a discrete distribution of a known form (with an unknown parameter p). We derive this model in the following subsections.

2.1. Notation

A grayscale $n \times m$ image will be represented with a two-dimensional array of integers $x_{ij},\ x_{ij} \in \{0, \ldots, 255\}, i \in \{0, \ldots, n-1\}, j \in \{0, \ldots, m-1\}$. A true color 24 bit $n \times m$ image will be represented as three grayscale $n \times m$

Table 1. PM1 embedding operation.

Pixel value x	To embed bit b, modify x to	
	$b=0$	$b=1$
$0<2i<255$	$2i$	$2i+1$ or $2i-1$
$0<2i+1<255$	$2i$ or $2i+2$	$2i+1$
0	0	1
255	254	255

images r_{ij}, g_{ij}, b_{ij}. The distortion due to non-adaptive $\pm K$ embedding is modeled as an additive independent identically distributed (i.i.d.) noise signal η with the following Probability Mass Function (PMF)

$$\begin{aligned} P(\eta=0) &= 1-p/2 \\ P(\eta=K) &= P(\eta=-K) = p/4. \end{aligned} \tag{1}$$

For $K=1$, for example, this PMF corresponds to embedding a random binary bitstream of length pmn in randomly selected pixels using the embedding rule in shown Table 1.

2.2. Model Description

We model the stego image pixel s_{ij} as a sample from the random variable S_{ij}

$$S_{ij} = X_{ij} + \eta_{ij}(p), \tag{2}$$

where X_{ij} is a random variable modeling the distribution of pixels (i,j). The random variables* $\eta_{ij}(p)$ model the stego signal and are distributed according to (1). They are assumed to be independent of each other as well as of the image pixels X_{ij}. This is indeed the case for non-adaptive embedding.

Our goal is to estimate p from the stego image pixels s_{ij}. Direct estimation of the message length p from s_{ij} is very difficult because of the lack of a good model for X_{ij}. Good models are, however, available in transform domains, such as DCT, DFT, or DWT, where the samples are decorrelated and well modeled with a Generalized Gaussian distribution. Because our goal is to estimate the secret message length p, we need a transformation F that will enable us to obtain the probability density function (PDF) of $F(\eta_{ij})$ in a closed form, yet, at the same time provide decorrelation of the cover image samples. A reasonable trade-off between these requirements is a simple FIR high-pass filter $I-F_A, (A>0)$, where

$$F_A = \frac{1}{A+4}\begin{pmatrix} 0 & 1 & 0 \\ 1 & A & 1 \\ 0 & 1 & 0 \end{pmatrix}. \tag{3}$$

As explained above, the simplicity of this filter allows us to analytically derive the PMF of the filtered image from the assumed models, which we then use to obtain the ML estimate of p. Applying this filter to the stego image† yields the high-frequency "image" y_{ij}

$$y_{ij} = s_{ij} - F_A(s_{ij}) = x_{ij} - F_A(x_{ij}) + \eta_{ij}(p) - F_A(\eta_{ij}(p)) = x^F_{ij} + \eta^F_{ij},$$

due to the linearity of F_A. The first term x^F_{ij} contains the high-frequency component of the *cover* image x_{ij} and the second term η^F_{ij} is a discrete random variable with PMF described below. As already mentioned above, the high-frequency components of an image will be modeled as independent continuous random variables with the

*We use the same symbol $\eta_{ij}(p)$ to denote the random variable modeling the embedding distortion as well as a sample from the random variable.

†With the notational convenience $F_A(x_{ij}) \stackrel{\text{def}}{=} 1/(A+4)(Ax_{ij} + x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1})$.

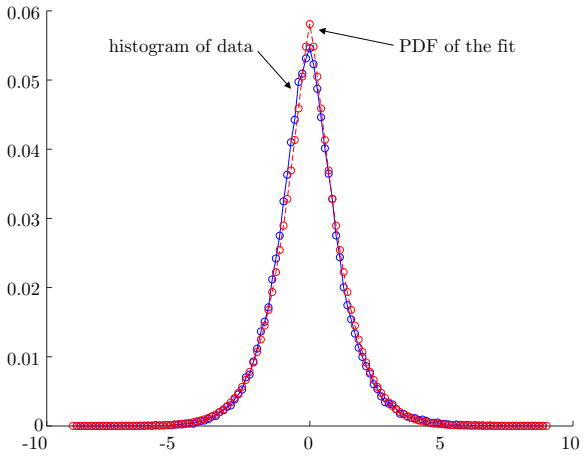

Figure 1. Histogram of the filtered image x_{ij}^F and its approximation using Generalized Gaussian distribution. The solid line is the histogram of the data and the dashed line is the PDF of the Generalized Gaussian fit.

Generalized Gaussian distribution. The probability density function of the Generalized Gaussian distribution with mean value μ, variance σ^2, and parameter $\alpha > 0$ is

$$f_{GG}(x;\mu,\sigma,\alpha) = \frac{\alpha\Gamma(3/\alpha)^{1/2}}{2\sigma\Gamma(1/\alpha)^{3/2}} \exp\left\{-\frac{|x-\mu|^\alpha}{\sigma^\alpha}\left(\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}\right)^{\alpha/2}\right\}, \tag{4}$$

where $\Gamma(\cdot)$ is the Euler Gamma function. We have derived and implemented the ML estimator using this continuous distribution and we obtained a working estimator. However, its performance was not satisfactory. During our analysis, we have discovered that modeling x_{ij}^F using a *discrete* distribution is much more appropriate because the stego image pixels are integers in the range $[0, 255]$. After filtering, x_{ij}^F take on only values that are integer multiples of $1/(A+4)$. So, we do not *actually* have real-valued samples. This observation is very important, especially in the case of ± 1 embedding, whose amplitude is comparable to that of the quantization noise.

Therefore, we have replaced the continuous distribution (4) with its *discretized* version. We have defined the "quantized" zero-mean Generalized Gaussian distribution taking on values $l\Delta$ with probability

$$f_{DGG}(l\Delta;\sigma,\alpha) = P(x = l\Delta;\sigma,\alpha) = \int_{l\Delta-\Delta/2}^{l\Delta+\Delta/2} f_{GG}(x;0,\sigma,\alpha)\,\mathrm{d}x, \tag{5}$$

for an integer l and $\Delta = 1/(A+4)$. This operation can be viewed as histogram binning at points $l\Delta$ or equivalently as quantization with the quantization step Δ. Note that the parameter σ^2 is no longer the exact variance of the quantized distribution (5). The variance of the distribution $f_{DGG}(l\Delta;\sigma,\alpha)$ is from definition

$$\mathrm{Var}\, X_{ij}^F = \sum_{l=-\infty}^{\infty} (l\Delta)^2 P(x = l\Delta;\sigma,\alpha) \approx 2\Delta^2 \sum_{l=1}^{l_{\max}} l^2 P(x = l\Delta;\sigma,\alpha), \tag{6}$$

because of the symmetry of the distribution. Theoretically, the summation must be carried over all integers, practically however, depending on the particular parameters σ, α, the probability $P(x = l\Delta;\sigma,\alpha)$ quickly approaches zero with increasing l. For typical values: $\Delta = 1/5$, $\sigma = 1$, and $\alpha = 1.6$, the probability is already less than 10^{-12} for $l_{\max} = 50$.

It can be shown that the random variable η_{ij}^F, the filtered stego signal, has a unimodal symmetric discrete distribution taking on seventeen values of the form

$$\lambda_l = lK\Delta = l\frac{K}{A+4}, \tag{7}$$

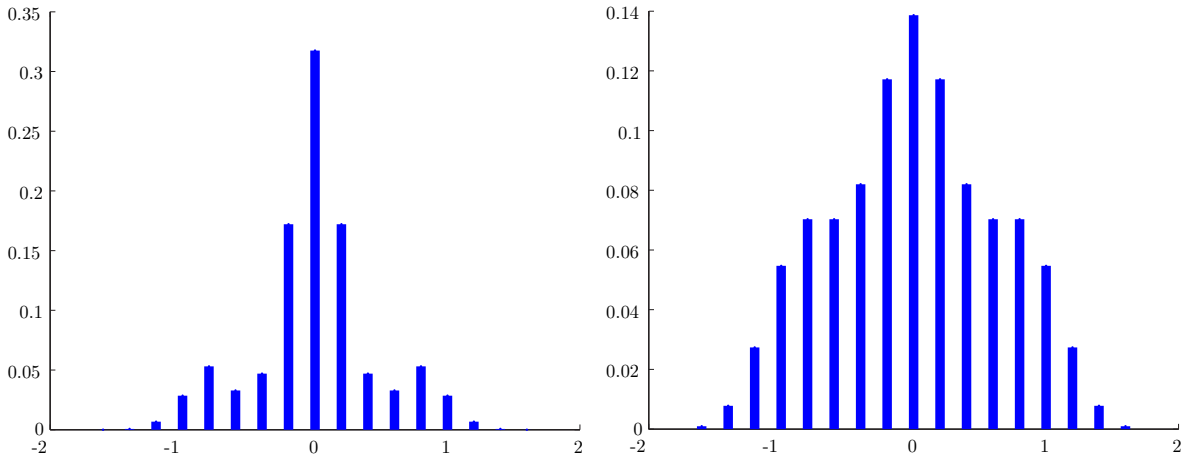

Figure 2. From left to right, examples of the distribution $\xi_l(p)$ for $p = 0.5$ and $p = 1$.

with probability

$$\xi_l(p) = P(\eta_{ij}^F = \lambda_l; p) = g(l; p), \tag{8}$$

where $l = -8, -7, \ldots, 7, 8$ and $g(-l; p) = g(l; p)$ is given by

$$g(l;p) = \begin{cases} -69/512p^5 + 115/128p^4 - 19/8p^3 + 13/4p^2 - 5/2p + 1 & l = 0, \\ 13/128p^5 - 43/64p^4 + 27/16p^3 - 2p^2 + p & l = 1, \\ -7/256p^5 + 13/64p^4 - 15/32p^3 + 3/8p^2 & l = 2, \\ -5/128p^5 + 11/64p^4 - 5/16p^3 + 1/4p^2 & l = 3, \\ 17/256p^5 - 79/256p^4 + 9/16p^3 - 1/2p^2 + 1/4p & l = 4, \\ -7/128p^5 + 15/64p^4 - 3/8p^3 + 1/4p^2 & l = 5, \\ 7/256p^5 - 3/32p^4 + 3/32p^3 & l = 6, \\ -1/128p^5 + 1/64p^4 & l = 7, \\ 1/1024p^5 & l = 8. \end{cases} \tag{9}$$

We see that the probabilities $\xi_l(p)$ are polynomials of the fifth order in p. We also see that $\xi_0(p)$ for $\lambda_0 = 0$ is the only polynomial that contains an absolute term. Figure 2 illustrates this distribution for $p = 0.5$ and $p = 1$.

To summarize our model, we represent the samples y_{ij} of the filtered stego image s_{ij} as samples from a collection of independent discrete random variables Y_{ij},

$$Y_{ij} = X_{ij}^F + \eta_{ij}^F, \tag{10}$$

where X_{ij}^F is distributed according to $f_{DGG}(l\Delta; \sigma_{ij}, \alpha_{ij})$ and η_{ij}^F is distributed according to $P(\eta_{ij}^F = \lambda_l; p)$. We also assume that X_{ij}^F and η_{ij}^F are independent, which is true for non-adaptive embedding. The distribution of Y_{ij} is then the convolution the two distributions. Thus, $Y_{ij} \propto f(\cdot; K, \Delta, \sigma_{ij}, \alpha_{ij}, p)$,

$$f(y_{ij}; K, \Delta, \sigma_{ij}, \alpha_{ij}, p) = \frac{\alpha_{ij}\Gamma(3/\alpha_{ij})^{1/2}}{2\sigma_{ij}\Gamma(1/\alpha_{ij})^{3/2}} \sum_{l=-8}^{8} \xi_l(p) \exp\left\{ -\frac{|y_{ij} - \lambda_l|^{\alpha_{ij}}}{\sigma_{ij}^{\alpha_{ij}}} \left(\frac{\Gamma(3/\alpha_{ij})}{\Gamma(1/\alpha_{ij})} \right)^{\alpha_{ij}/2} \right\}. \tag{11}$$

In this most general model, we allow the local characteristics of the image to vary from pixel to pixel—the parameters σ and α depend on the pixel position. This non-stationary model turns out to be too general and difficult to handle. Later, we will describe a simple heuristics that enables us to assume that σ_{ij} and α_{ij} are constant (the stationary model).

2.3. ML Estimator

To simplify the notation, we use a single index i instead of the two dimensional index (i, j); the index i may be interpreted as an index of some path through the image. Assuming independency and stationarity (with respect to σ_i and α_i), the joint density of the vector $\mathbf{Y} = (Y_1, \ldots, Y_N), N = nm$, can be written as

$$f_{\mathbf{Y}}(\mathbf{y}; K, \Delta, \sigma, \alpha, p) = \prod_{i=1}^{N} f(y_i; K, \Delta, \sigma, \alpha, p), \tag{12}$$

and the log-likelihood function is then (dropping the explicit dependence on *known* K, Δ)

$$\log f_{\mathbf{Y}}(\mathbf{y}; \sigma, \alpha, p) = N \log \frac{\alpha \Gamma(3/\alpha)^{1/2}}{2\sigma\Gamma(1/\alpha)^{3/2}} + \sum_{i=1}^{N} \log \sum_{l=-8}^{8} \xi_l(p) \exp\left\{ -\frac{|y_i - \lambda_l|^\alpha}{\sigma^\alpha} \left(\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)} \right)^{\alpha/2} \right\}. \tag{13}$$

The Maximum Likelihood estimate of p having observed samples $\mathbf{y}$ is then

$$(\hat{p}_{\mathrm{ML}}, \hat{\sigma}_{\mathrm{ML}}, \hat{\alpha}_{\mathrm{ML}}) \stackrel{\mathrm{def}}{=} \operatorname*{argmax}_{(p,\sigma,\alpha)} \log f_{\mathbf{Y}}(\mathbf{y}; \sigma, \alpha, p). \tag{14}$$

We see that along with the unknown message length p, we must also estimate the nuisance parameters σ, α. This maximization is a rather complex task; the function $\log f_{\mathbf{Y}}$ is not convex and has several local maxima. We decided to find the maximum using a grid search. To reduce the complexity of the search, we employed the following measures.

First of all, since the variance of $\eta_{ij}(p)$ is $\operatorname{Var} \eta_{ij}(p) = K^2 p/2$, the variance of $\eta^F_{ij}(p)$ is

$$\begin{aligned} \operatorname{Var} \eta^F_{ij}(p) &= \operatorname{Var} \left\{ \eta_{ij}(p) - \frac{1}{A+4} \left(A\eta_{ij}(p) + \eta_{i-1,j}(p) + \eta_{i+1,j}(p) + \eta_{i,j-1}(p) + \eta_{i,j+1}(p) \right) \right\} \\ &= \frac{10K^2}{(A+4)^2} p, \end{aligned} \tag{15}$$

which, in turn, means that

$$\sigma^2_Y = \operatorname{Var} Y = \operatorname{Var} X^F + \frac{10K^2}{(A+4)^2} p. \tag{16}$$

Because $0 \le p \le 1$, we can narrow the search interval for $\sigma^2 \approx \operatorname{Var} X^F$ to the interval

$$\left[\sigma^2_Y - \frac{10K^2}{(A+4)^2}, \sigma^2_Y \right]. \tag{17}$$

In reality, of course, we do not know the variance $\operatorname{Var} Y$ ahead of time, so we estimate it by the sample variance $\hat{\sigma}^2_Y$ computed from $\mathbf{y}$.

Because the parameter σ^2 of the discretized Generalized Gaussian distribution is no longer the exact variance $\operatorname{Var} X^F$, we increase the search interval for σ^2 a little bit.

Using the constraint (16) not only narrows the search interval but it also effectively reduces the dimensionality of the search from three independent variables to two. This is because when we fix α and σ, then p is calculated from (16) and (6).

Further computational optimizations are possible by iterating through α in the outer-most loop and leaving the iteration through σ as an inner-loop. This arrangement allows us to precompute the terms containing the computationally-expensive Gamma function, since they only depend on α. Here is the pseudo-code for our MLE.

1. Estimate the variance σ^2_Y by calculating the sample variance $\hat{\sigma}^2_Y$ of the sample vector $\mathbf{y}$.

2. Compute the lower bound σ_L^2 and the upper bound σ_U^2 of the search interval for σ^2 using (17). Enlarge the search window to compensate for the fact that σ^2 is not the variance of X^F by setting $\sigma_L \leftarrow \sigma_L - 1/\Delta$ and $\sigma_U \leftarrow \sigma_U + 1/\Delta$. If $\sigma_L < 0.05$, set $\sigma_L = 0.05$. Compute the search step $\delta_\sigma = (\sigma_U - \sigma_L)/N_\sigma$, where N_σ is the number of values for σ that should be examined.

3. Set the search interval for α; we have used $\alpha_L = 0.4$ and $\alpha_U = 1.6$ with the search step $\delta_\alpha = 0.05$.

4. For $\alpha \leftarrow \alpha_L$ to α_U with the step δ_α do

   For $\sigma \leftarrow \sigma_L$ to σ_U with the step δ_σ do

   i. Compute the variance $\operatorname{Var} X^F$ using (6). (The variance depends on σ and α.)

   ii. Compute p as
   $$p = \frac{(A+4)^2}{10K^2}\left(\hat{\sigma}_Y^2 - \operatorname{Var} X^F\right). \tag{18}$$

   iii. Create a look-up table for the distribution $f(y; K, \Delta, \sigma, \alpha, p)$. This is possible, because the distribution is discrete and we know the maximum a minimum value that y can take on from the data.

   iv. Evaluate the function $\log f_{\mathbf{Y}}(\mathbf{y}; \sigma, \alpha, p)$ using the look-up table and store the value in $\mathbf{f}(\sigma, \alpha)$.

5. Find the maximum value of $\mathbf{f}(\sigma, \alpha)$. The $\sigma_{\mathrm{ML}}, \alpha_{\mathrm{ML}}$ for which we obtain the maximum are, by definition, the ML estimates of σ, α. Compute the ML estimate of p from (18) with the optimal σ_{ML}, α_{ML}.

2.4. Pixel Selection

We have seen in the previous section that along with the unknown message length p we also need to estimate the parameters σ and α. If we did not have to estimate them, the estimation would be much faster, because we could treat σ and α as known constants. The fundamental problem is that we *cannot* estimate σ and α from the data without first knowing p, since we do not observe samples from X_i or X_i^F but only from S or Y, which depend on p. We have experimentally verified that the estimation of σ and α is sensitive with respect to p, which is to be expected for σ but is not so obvious for α. For large σ, the dependence on p weakens but so does our ability to reliably estimate p.

This is a fundamental problem because the parameter σ is obviously non-stationary—there are regions in the image where the simple filter F_A does a good job at removing the correlation among pixels but there are regions where this filter is not good enough. This will be reflected in the variance of the variable Y_i. The behavior of α is not clear but it is reasonable to expect that it is also non-stationary. An obvious solution is to assume non-stationarity for both and along with p estimate also σ_i and α_i for each pixel. This "solution" is obviously not feasible because we cannot reliably estimate a parameter set from a data set of the same cardinality. Therefore, we have to restrict the set of parameters to a smaller number. We could try to identify the regions in the stego image that have the same or similar structure and assume stationarity of σ and α within these regions. We have used an even simpler approach: we try to find such samples of Y_i whose local σ_i can be considered as constant and use only those samples for estimation. To find such samples, we use a heuristics described in the following algorithm.

1. First, segment the stego image to localize areas of the "same structure". The segmentation allows us to avoid pixels that lie on a boundary between two segments (pixels with a high local variance). We have used an implementation of the segmentation algorithm[16] created by one of the authors of the algorithm.

2. Calculate y_i by filtering the stego image s_i using the filter F_A.

3. Calculate the estimate $\hat{\sigma}_i^2$ of the local variances of y_i from a small square window of width B centered about the pixel i. During this estimation, only use those samples from the window that belong to the same region as the center pixel i. Calculate the number of samples in each region and denote as N_i.

4. Remove those samples y_i whose estimated local variance $\hat{\sigma}_i^2$ was computed from fewer than 90% of samples in the window, in other words those samples y_i for which $N_i < 0.9B^2$. Remove also those samples whose estimated local variance $\hat{\sigma}_i^2$ is larger than a given threshold. We have used the following threshold T,

$$T = 9 + \frac{10K^2}{(A+4)^2},$$

obtained by experimenting with the ML estimator on ideal data generated from the assumed model. This indicates that the ML estimator starts to fail to reliably estimate the message length p once the variance of X^F exceeds 9. Denote the set of pixels i that fulfill both of these conditions by I.

5. Compute the histogram $h(\cdot)$ of the set $\{\hat{\sigma}_i \mid i \in I\}$ and find all local maxima of this histogram. Denote the points where a local maximum is attained by $\sigma_l^{\max}$ for $l = 1, \ldots, L$, where L is the number of local maxima. In other words, at point $\sigma_l^{\max}$, the histogram h has a local maximum $h(\sigma_l^{\max})$, and this holds for all l.

6. Sort the local maxima in an ascending order and select the first $\sigma_l^{\max}$ with enough samples i whose estimated local standard deviation $\hat{\sigma}_i$ is in the interval

$$\hat{\sigma}_i \in (\sigma_l^{\max} - \epsilon, \sigma_l^{\max} + \epsilon),$$

where ϵ is a small constant, for example $\epsilon = 0.04$. In other words, select the smallest $\sigma_l^{\max}$ for which

$$|J| > M, \quad J = \{i \mid \hat{\sigma}_i \in (\sigma_l^{\max} - \epsilon, \sigma_l^{\max} + \epsilon)\},$$

and M is the minimum number of samples that allow reliable estimation. We have used $M = 100$.

7. Perform the ML estimation of (p, σ, α) from the data set $\{y_i \mid i \in J\}$.

In Steps 5 and 6, we try to identify pixels that have very similar local variance so that we can assume that σ_i is constant (because stationarity of $\operatorname{Var} Y_i$ implies stationarity of σ_i since the variance of $\eta(p)_i$ is fixed). We make a tacit assumption that for such pixels the parameter α_i will also be stationary. We further select those pixels whose local variance is the smallest of all. This rationale comes from the natural feeling that the smaller the local variance, the better the estimation (we verified experimentally on ideal data).

This heuristics did, indeed, lead to a significant performance improvement. Without it, the variance of the estimator p_{ML} was too high to provide any useful estimates. This also indicates that our method would not work for adaptive $\pm K$ embedding.

3. EXPERIMENTAL RESULTS

Perhaps the best performance measure would be the Cramér-Rao Lower Bound (CRLB). It is however very difficult to establish the CRLB for the distribution (12). The bound does not exist in a closed form and its numerical evaluation is also problematic because the function is not convex. This is a part of our future research. We have therefore performed several experiments to assess the performance of our estimator. We describe the experiments based on the source of the cover images.

3.1. Experiment One

The test database comprised of 180 grayscale images selected from the Greenspun database.[17] These images are stored as JPEGs with quality factor 75. In the test, we have embedded random messages with relative message length $p = 0, 0.25, 0.5, 0.75$, and 1 using ± 1, ± 2, and ± 3 embedding. The parameters of the test were as follows: the filter parameter $A = 1$ and the size of the window used for the variance estimation was $B = 13$. To speed-up the computations, the estimator used at most 25,000 samples. The minimum number of samples required for the test was set to 100.

Because of the relatively small number of tested images, we show the results of detection in Figures 3 and 4. The estimated mean and standard variation of the estimator are provided in Table 2. Note how the estimator accuracy dramatically improves with increasing amplitude K of the stego signal.

Table 2. Mean and variance of the estimator.

p	± 1		± 2		± 3	
	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\mu}$	$\hat{\sigma}$	$\hat{\mu}$	$\hat{\sigma}$
0	0.137	0.281	0.058	0.177	0.023	0.098
0.25	0.295	0.240	0.277	0.145	0.250	0.071
0.5	0.530	0.171	0.517	0.085	0.495	0.055
0.75	0.780	0.070	0.762	0.044	0.747	0.022
1	0.980	0.017	0.989	0.008	0.985	0.010

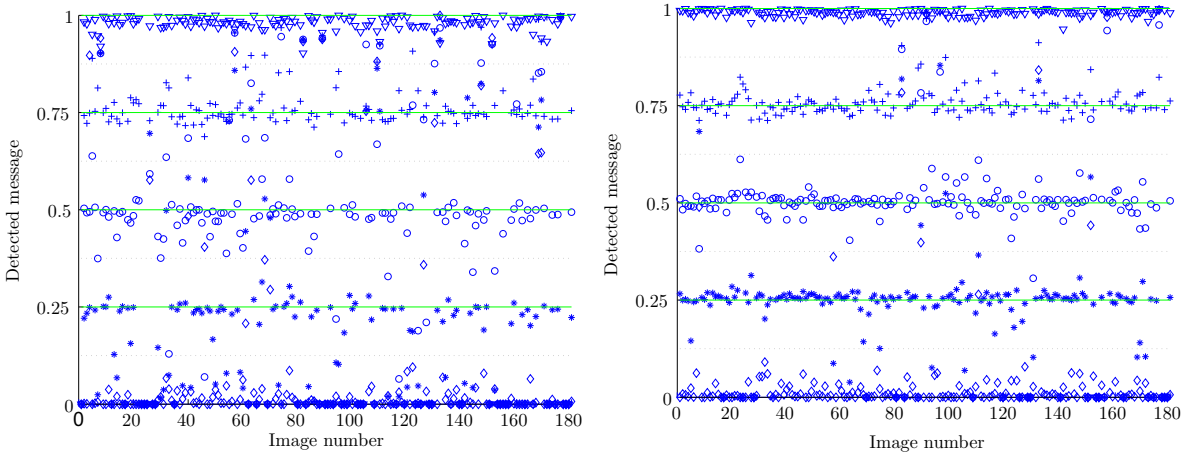

Figure 3. Estimates of message lengths for ± 1 and ± 2 embedding (on the left and right respectively) performed on a database of 180 grayscale images.

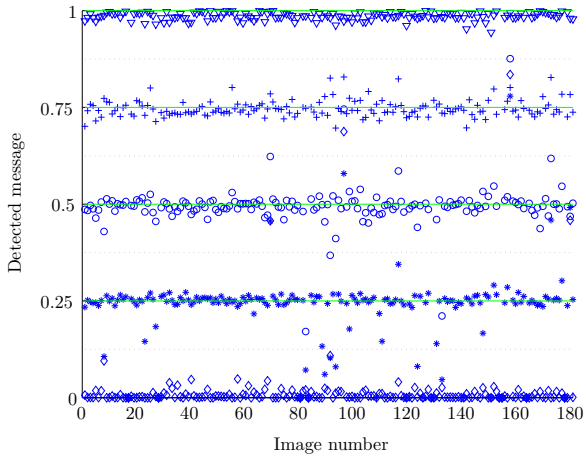

Figure 4. Estimates of message lengths for ± 3 embedding performed on a database of 180 grayscale images.

3.2. Experiment Two

The second experiment was performed on a database of images taken by three different cameras (Canon PowerShot G2, Canon PowerShot S40, and Kodak DC290) in their native raw formats (RAW and TIFF). There were 195 images from Canon PS G2, 197 images from Canon PS S40, and 195 images from Kodak DC290. The

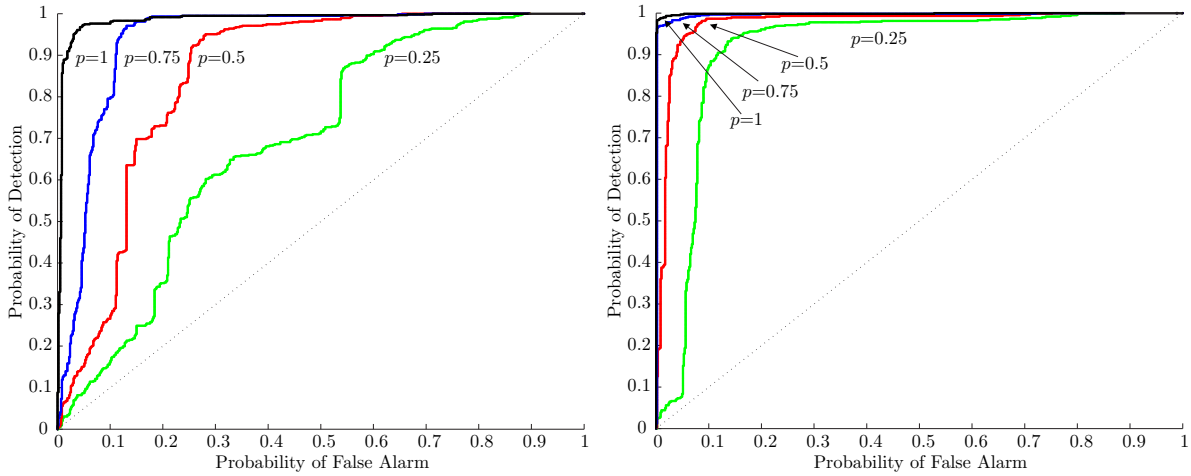

Figure 5. The ROC curves for ±1 and ±2 (on the left and right figure respectively) computed from 587 never-compressed images.

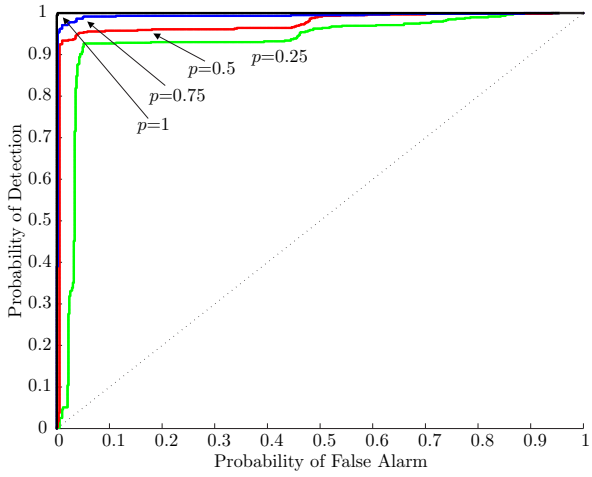

Figure 6. The ROC curves for ±3 computed from 587 never-compressed images.

images stored in the RAW format have been losslessly converted to the TIFF format. All images were converted to grayscale before the experiments.

We have again embedded random messages with $p = 0, 0.25, 0.5, 0.75$, and 1 into each image using ±1, ±2, and ±3 embedding and run the estimation. The parameters of the test were the same as in the previous test in Subsection 3.1.

As can be intuitively expected, the message length estimation is much less accurate for raw images than for decompressed JPEGs. Thus, for raw images we have decided to use the estimate as a scalar parameter to evaluate the presence of hidden data. The results are shown in Figures 5 and 6 in the form of Receiver Operating Characteristic (ROC) curves.

3.3. Experiment Three

We have applied this method also to a small set of raw, never compressed, images from a scanner. We were unable to detect any ±1 embedded message in any of the images. We attribute this behavior to a stronger noise component that is inherently present in scanned images. Detection of low-amplitude stego noise in scans or very

noisy images appears to be fundamentally difficult as none of the other detection methods works for this case either. We expect that the detection performance will improve with increasing K even for these images.

4. CONCLUSIONS

In this paper, we have proposed a new method for detection of non-adaptive $\pm K$ embedding and tested its performance on a database of images. This method can also estimate the secret message length and is applicable both to grayscale and color images. After a simple high-pass FIR filter is applied to the stego image, the unknown message length is estimated using a Maximal Likelihood estimator. We have used a simple filter to be able to derive a closed form of the PMF of the stego signal. The filtered cover image is modeled using a Generalized Gaussian distribution. The estimation is carried out only for those segments of the stego image where the distribution of the filtered cover image can be considered as stationary. On such segments, the ML estimator is used to obtain two parameters of the stationary generalized Gaussian distribution and the unknown parameter—the message length.

There is room for improvement in our analysis that we plan to investigate in our future research. One possibility would be to utilize the results of the segmentation more systematically. So far, we use the segmentation to improve the estimation of local variance by using only pixels belonging to the same region as the pixel whose variance is being estimated. The model later assumes that all the samples we use for estimation have the same variance and parameter α. It would perhaps be more realistic to assume that only pixels within the same *connected* region share the same σ and α.

Another possibility to implement our methodology is to apply linear decomposition to the image (e.g., wavelet decomposition) and perform the estimation in the transform domain. Because the wavelet transform achieves significantly better decorrelation than our simple high pass filter, we can expect an improvement in performance. The problem is that we can no longer find an analytic expression for the transformed stego message. On the other hand, based on our experiments, it appears that the wavelet transform of the stego message signal is well modelled using Generalized Gaussian distribution. Thus, we can precalculate (or tabulate) its parameters as functions of p. The approach would then proceed in the same manner as in Section 2.2. Instead of considering the wavelet transform of the stego image as a convolution of a Generalized Gaussian and a discrete distribution (9), we will now have a convolution of a Generalized Gaussian with another Generalized Gaussian.

ACKNOWLEDGMENTS

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. 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* **16(4)**, pp. 474–481, 1998.
2. C. Cachin, "An information-theoretic model for steganography," in Aucsmith,[19] pp. 306–318.
3. "Steganography software for windows." Online, http://www.stegoarchive.com.
4. A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," in *Proceedings, Information Hiding: 3th International Workshop, IH'99*, A. Pfitzmann, ed., *Lecture Notes in Computer Science* **1768**, pp. 61–75, Springer-Verlag, (Dresden, Germany), Sep 29–Oct 1 1999.
5. J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color and gray-scale images," in *Multimedia and Security Workshop*, J. Dittmann, K. Nahrstedt, and P. Wohlmacher, eds., *Proceedings of ACM*, pp. 22–28, ACM Press, (Ottawa, Ontario, Canada), Oct 5 2001.
6. S. Dumitrescu, W. Xiaolin, and Z. Wang, "Detection of LSB steganography via Sample Pair analysis," in Petitcolas,[20] pp. 355–372.

7. A. Ker, "Quantitative evaluation of Pairs and RS steganalysis," in *Security, Steganography and Watermarking of Multimedia Contents VI*, E. J. Delp III and P. W. Wong, eds., *Proceedings of SPIE* **5306**, SPIE and IS&T, (San Jose, California, USA), Jan 19–22 2004.
8. A. Ker, "Improved detection of LSB steganography in grayscale images," in *Pre-Proceedings, Information Hiding: 6th International Workshop, IH 2004*, *Lecture Notes in Computer Science*, Springer-Verlag, (Toronto, Canada), May 23–25 2004.
9. L. M. Marvel, C. G. Boncelet, and C. T. Retter, "Reliable blind information hiding for images," in Aucsmith,[19] pp. 48–61.
10. F. Alturki and R. Mersereau, "A novel approach for increasing security and data embedding capacity in images for data hiding applications," in *Proceedings of ITCC*, pp. 228–233, (Las Vegas, Nevada), 2001.
11. T. Sharp, "An implementation of key-based digital signal steganography," in *Proceedings, Information Hiding: 4th International Workshop, IHW 2001*, I. S. Moskowitz, ed., *Lecture Notes in Computer Science* **2137**, pp. 13–26, Springer-Verlag, (Pittsburgh, PA, USA), Apr 25–27 2001.
12. J. Fridrich and M. Goljan, "Digital image steganography using stochastic modulation," in Delp III and Wong,[18] pp. 191–202.
13. J. J. Harmsen and W. A. Pearlman, "Steganalysis of additive noise modelable information hiding," in Delp III and Wong,[18] pp. 131–142.
14. A. Westfeld, "Detecting low embedding rates," in Petitcolas,[20] pp. 324–339.
15. J. Fridrich, M. Goljan, D. Hogea, and D. Soukal, "Quantitative steganalysis: Estimating secret message length," in *ACM Multimedia Systems Journal, Special issue on Multimedia Security*, **9(3)**, pp. 288–302, Aug 20–24 2003.
16. P. Felzenszwalb and D. Huttenlocher, "Image segmentation using local variation," in *Proceedings of IEEE CVPR*, 1998.
17. "Greenspun database." Online, http://www.greenspun.com.
18. E. J. Delp III and P. W. Wong, eds., *Security and Watermarking of Multimedia Contents V*, *Proceedings of SPIE* **5020**, (Santa Clara, California, USA), SPIE and IS&T, Jan 21–24 2003.
19. D. Aucsmith, ed., *Proceedings, Information Hiding: 2nd International Workshop, IH'98*, *Lecture Notes in Computer Science* **1525**, (Portland, Oregon, USA), Springer-Verlag, Apr 1998.
20. F. A. P. Petitcolas, ed., *Revised Papers, Information Hiding: 5th International Workshop, IH 2002*, *Lecture Notes in Computer Science* **2578**, (Noordwijkerhout, The Netherlands), Springer-Verlag, Oct 7–9 2002.