Steganalysis Using Higher-Order Image Statistics

Siwei Lyu, *Student Member, IEEE,* and Hany Farid, *Member, IEEE*

***Abstract*— Techniques for information hiding (steganography) are becoming increasingly more sophisticated and widespread. With high-resolution digital images as carriers, detecting hidden messages is also becoming considerably more difficult. We describe a universal approach to steganalysis for detecting the presence of hidden messages embedded within digital images. We show that, within multi-scale, multi-orientation image decompositions (e.g., wavelets), first- and higher-order magnitude and phase statistics are relatively consistent across a broad range of images, but are disturbed by the presence of embedded hidden messages. We show the efficacy of our approach on a large collection of images, and on eight different steganographic embedding algorithms.**

I. Introduction

The goal of steganography is to embed within an innocuous looking cover medium (text, audio, image, video, etc.) a message so that casual inspection of the resulting medium will not reveal the presence of the message (see, e.g., [1]–[4] for general reviews). For example, with plain text as a cover medium, a German spy, during World War I, sent the following message:

> *Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils.*

which upon casual inspection seems fairly harmless. When the second letter of each word is extracted, however, this text is seen to be a carrier for the following message:

> *Pershing sails from NY June 1.*

With the advent of the Internet and the broad dissemination of large amounts of digital media, digital images have become a popular cover medium for steganography tools. At the time of this paper's publication there are more than 100 freely available steganography software tools for embedding messages within digital images. In addition to being nearly ubiquitous on most web pages, digital images are well suited as a cover medium. An uncompressed color image of size 640×480, for example, can hide approximately $100,000$ characters of text. A simple method for embedding a message into a digital image is to change the least significant bit (LSB) of the image pixels, so that the LSB of consecutive pixels encode a message. In so doing, the perceptual distortion to the cover image is nearly negligible and unlikely to be detected by simple visual inspection.

It is not surprising that with the emergence of steganography, that the development of a counter-technology, steganalysis, has also emerged (see [5] for a review). The goal of steganalysis is to determine if an image (or other carrier) contains an embedded message. As this field has developed, determining the length of the message [6] and the actual contents of the message are also becoming an active area of research. Current steganalysis methods fall broadly into one of two categories: embedding specific (e.g., [7]–[12]) or universal (e.g., [13]–[17]). While universal steganalysis attempts to detect the presence of an embedded message independent of the embedding algorithm and, ideally, the image format, embedding specific approaches to steganalysis take advantage of particular algorithmic details of the embedding algorithm. Given the ever growing number of steganography tools, universal approaches are clearly necessary in order to perform any type of generic, large-scale steganalysis.

We have previously described a universal approach to steganalysis. Specifically, in [18], we showed how a statistical model based on first- and higher-order magnitude statistics extracted from a wavelet decomposition, coupled with a linear discriminant analysis (LDA), could be used to detect steganography in grayscale images. In [13], we replaced the LDA classifier with a non-linear support vector machine (SVM), affording better classification accuracy. And in [19] we extended the statistical model to color images, and described a one-class SVM that simplified the training of the classifier. In this culminating paper we summarize these results and extend the statistical model to include phase statistics. We show the efficacy of our approach on a large collection of images, and on eight different steganography embedding algorithms. We examine the general sensitivity and robustness of our approach to message size, false-positive rate, JPEG compression, the specific components of the statistical model, cover image format, and to the choice of classifier.

II. Statistical Model

The decomposition of images using basis functions that are localized in spatial position, orientation and scale (e.g., wavelets) have proven extremely useful in image compression, image coding, noise removal and texture synthesis. One reason is that such decompositions exhibit statistical regularities that can be exploited. From such decompositions, our statistical model extracts first- and higher-order magnitude and phase statistics. Before describing the details of this model, we motivate our choice of image representations over others.

A. Choosing an Image Representation

At the core of our statistical model is the choice of decomposing images using basis functions that are localized in spatial position, orientation and scale. There are, of course, many other possible representations to choose from. The simplest

S. Lyu (lsw@cs.dartmouth.edu) and H. Farid (farid@cs.dartmouth.edu) are with the Department of Computer Science at Dartmouth College, 6211 Sudikoff Lab, Hanover NH 03755.

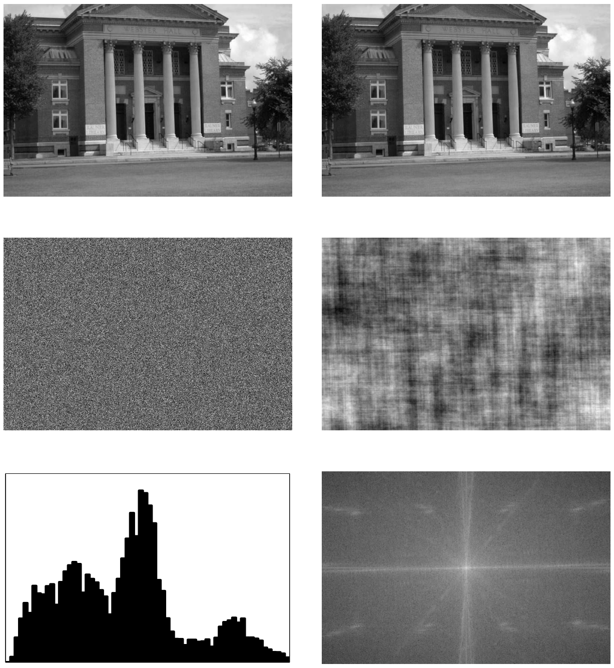


Fig. 1. Shown in the first column are a pair of images with identical intensity histograms (bottom panel). Shown in the second column are a pair of images with identical Fourier magnitudes (bottom panel).

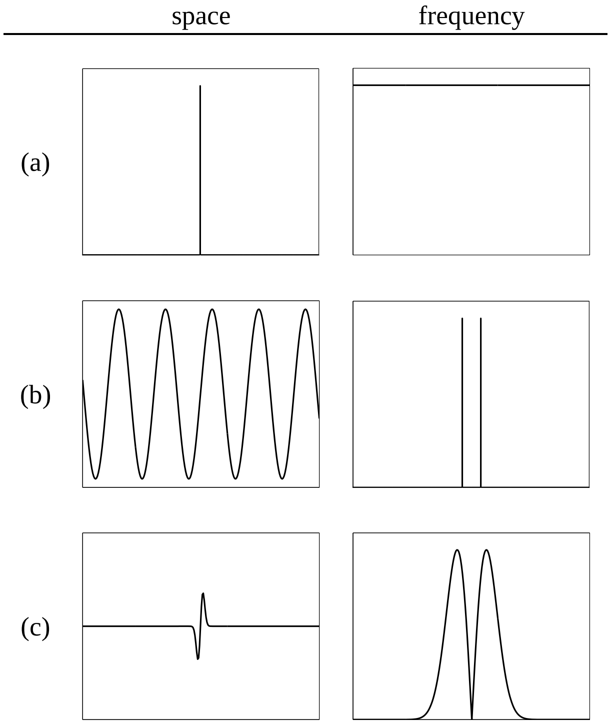


Fig. 2. Shown are 1-D space and frequency (magnitude) representations of (a) pixel, (b) Fourier, and (c) wavelet-like basis functions.

representation, for example, would be a pixel-based approach, where the representation is simply the original intensity values. In this representation an $n \times n$ grayscale image is considered as a point in a n^2-dimensional space, where the i^{th} coordinate is determined by the intensity value of the i^{th} pixel (a color RGB image is represented by a point in a $3n^2$-dimensional space). From such a representation, the most standard model is based on the histogram of intensity values. Shown in the first column of Fig. 1 are two images with exactly the same intensity histograms (bottom panel). This example shows that such a pixel-based representation is not sufficiently powerful to even discriminate between an image and a noise pattern.

Another popular representation is that based on a global Fourier decomposition. In this representation, an image is represented as a sum of sines and cosines of varying frequency and orientation: $F(\omega_x, \omega_y) = \sum_x \sum_y I(x,y) e^{-j(\omega_x x + \omega_y y)}$, where $I(x,y)$ is a grayscale image, and $F(\omega_x, \omega_y)$ is its Fourier transform (each channel of a color image is independently represented in the same way). It has been observed that the power spectrum, $|F(\omega_x, \omega_y)|$ of natural images are often well modeled with an exponential fall-off: $|F(\omega_x, \omega_y)| \approx (\omega_x^2 + \omega_y^2)^{-p}$, where the exponent p determines the rate of the fall-off [20]. Shown in the right column of Fig. 1 are two images with exactly the same Fourier magnitude (bottom panel). This example shows that such a Fourier-based representation is not sufficiently powerful to discriminate between an image and a “fractal-like” pattern.

The pixel- and Fourier-based representations are, in some ways, at opposite ends of a spectrum of representations. The basis functions for the pixel-based representation are perfectly localized in space, but are infinite in terms of their frequency coverage. On the other hand, the basis functions for a Fourier-based representation are perfectly localized in frequency, but are infinite in the spatial domain. Image representations with basis functions partially localized in both space and frequency (e.g., wavelets), offer a compromise between these representations, Fig. 2. As a result, these representations are generally better than pixel- or Fourier-based representations at capturing local image structure. To this end, we employ a wavelet decomposition and a local angular harmonic decomposition from which we extract a statistical feature vector for differentiating between clean and stego images. We empirically show that these representations and subsequent statistical measurements capture certain fundamental properties of an image which are disturbed in the presence of steganography.

B. *Image Representation*

We describe two image decompositions that localize image structure in both space and frequency – a wavelet decomposition and a local angular harmonic decomposition. From the former we extract magnitude statistics and from the latter we extract phase statistics.

1) Wavelet Decomposition: The image decomposition employed here is based on separable quadrature mirror filters (QMFs) [21]–[23]. We choose this specific decomposition over more traditional wavelets (e.g., Haar or Daubechies) because, unlike other wavelets, QMFs minimize spatial aliasing within the decomposition subbands. On the other hand, QMFs do not afford perfect reconstruction of the original image – though reconstruction errors can be minimized with a careful filter design [23].

The separable QMFs consist of a pair of one-dimensional low-pass, $l(\cdot)$, and high-pass, $h(\cdot)$, filters. The first scale of the decomposition, consisting of a vertical, horizontal, and diagonal subband, is generated by convolving each color channel, $c \in \{r, g, b\}$, of the intensity image, $I^c(x, y)$, with these filters:

$$V_1^c(x, y) \quad = \quad I^c(x, y) \star h(x) \star l(y), \tag{1}$$

$$H_1^c(x, y) \quad = \quad I^c(x, y) \star l(x) \star h(y), \tag{2}$$

and

$$D_1^c(x, y) \quad = \quad I^c(x, y) \star h(x) \star h(y), \tag{3}$$

where $\star$ is the convolution operator. Subsequent scales are generated by creating a low-pass subband:

$$L_1^c(x, y) \quad = \quad I^c(x, y) \star l(x) \star l(y), \tag{4}$$

which is down-sampled by a factor of two and filtered in the same way as above, to yield $V_2^c(x, y)$, $H_2^c(x, y)$, $D_2^c(x, y)$ and $L_2^c(x, y)$. This entire process is repeated to create as many scales, $V_i^c(x, y)$, $H_i^c(x, y)$, $D_i^c(x, y)$ and $L_i^c(x, y)$, as desired, or as is possible given the image size. Shown in Fig. 3, for example, is a three-scale QMF decomposition.

2) Local Angular Harmonic Decomposition: It is possible to model local phase statistics from a complex wavelet decompostion [24], affording a unified underlying image representation with the wavelet decomposition described in the previous section. We have found, however, that a local angular harmonic decomposition (LAHD) affords more accurate estimates of local phase [25]. The LAHD decomposes local image structure by projecting onto a set of angular Fourier basis functions. The p^{th}-order LAHD, a two-dimensional complex valued quantity, is given by:

$$A_p^c(x, y) \quad = \quad \int_r \int_\theta I_{x,y}^c(r, \theta) g(r) e^{jp\theta} dr d\theta, \tag{5}$$

where $j = \sqrt{-1}$, $g(r)$ is an integrable radial function, and $I_{x,y}^c(r, \theta)$ is the polar parameterization of the color channel, $c \in \{r, g, b\}$, centered at location (x, y).

The p^{th}-order LAHD can be computed efficiently by convolving the image with derivatives of a differentiable radial filter (e.g., a Gaussian):

$$A_p^c(x, y) = I^c(x, y) \star \left(\sum_{k=0}^{p} \binom{p}{k} j^{p-k} \frac{\partial^p G(x, y)}{\partial x^k \partial y^{p-k}} \right). \tag{6}$$

For example:

$$\begin{aligned}
A_1^c(x, y) &= I^c(x, y) \star \left(\frac{\partial G}{\partial x} + j \frac{\partial G}{\partial y} \right) \\
A_2^c(x, y) &= I^c(x, y) \star \left(\frac{\partial^2 G}{\partial x^2} - \frac{\partial^2 G}{\partial y^2} + 2j \frac{\partial^2 G}{\partial x \partial y} \right) \\
A_3^c(x, y) &= I^c(x, y) \star \left(\frac{\partial^3 G}{\partial x^3} - 3 \frac{\partial^3 G}{\partial x \partial y^2} + j \left(3 \frac{\partial^3 G}{\partial x^2 \partial y} - \frac{\partial^3 G}{\partial y^3} \right) \right).
\end{aligned}$$

Note that, as with the basis functions of the previous section, these basis functions, sums of derivatives of a low-pass filter, are also localized in space and frequency.

C. Magnitude Statistics

Given the QMF decomposition described in Section II-B.1, the first component of the statistical model is composed of the mean, variance, skewness and kurtosis of the subband coefficients at each orientation, scale and color channel. While these statistics characterize the basic coefficient distributions, they are unlikely to capture the strong correlations that exist across space, orientation, scale and color. For example, edges tend to extend spatially and across multiple scales. As such, if a large coefficient is found in a horizontal subband, then it is likely that its left and right spatial neighbors in the same subband will also have a large value. Similarly, a large coefficient at scale i might indicate a large value for its "parent" at scale $i + 1$.

In order to capture some of these higher-order statistical correlations, we collect a second set of statistics that are based on the errors in a linear predictor of coefficient magnitude [26]. For the purpose of illustration, consider first a vertical band of the green channel at scale i, $V_i^g(x, y)$. A linear predictor for the magnitude of these coefficients in a subset[1] of all possible spatial, orientation, scale, and color neighbors is given by:

$$\begin{aligned}
|V_i^g(x, y)| &= w_1 |V_i^g(x-1, y)| + w_2 |V_i^g(x+1, y)| \\
&+ w_3 |V_i^g(x, y-1)| + w_4 |V_i^g(x, y+1)| \\
&+ w_5 |V_i^g(x/2, y/2)| + w_6 |D_i^g(x, y)| \\
&+ w_7 |D_{i+1}^g(x/2, y/2)| + w_8 |V_i^r(x, y)| \\
&+ w_9 |V_i^b(x, y)|,
\end{aligned} \tag{7}$$

where $|\cdot|$ denotes absolute value and w_k are the weights. This linear relationship can be expressed more compactly in matrix form as:

$$\vec{v} \quad = \quad Q\vec{w}, \tag{8}$$

where $\vec{v}$ contains the coefficient magnitudes of $V_i^g(x, y)$ strung out into a column vector, and the columns of the matrix Q contain the neighboring coefficient magnitudes as specified in Equation (7), and $\vec{w} = (w_1 \ \ldots \ w_9)^T$. Only magnitudes greater than 1 are considered, with the intensity values in the range $[0, 255]$. Low magnitude coefficients are ignored when constructing the linear predictor because we expect them to be less predictable from their neighbors, and therefore less informative in capturing statistical regularities. The weights $\vec{w}$ are determined by minimizing the following quadratic error function:

$$E(\vec{w}) \quad = \quad [\vec{v} - Q\vec{w}]^2. \tag{9}$$

This error function is minimized by differentiating with respect to $\vec{w}$:

$$\frac{dE(\vec{w})}{d\vec{w}} \quad = \quad 2Q^T(\vec{v} - Q\vec{w}), \tag{10}$$

setting the result equal to zero, and solving for $\vec{w}$ to yield the least-squares estimate:

$$\vec{w} \quad = \quad (Q^T Q)^{-1} Q^T \vec{v}. \tag{11}$$

[1] The particular choice of neighbors was motivated by the observations of [26] and modified to include non-casual neighbors.

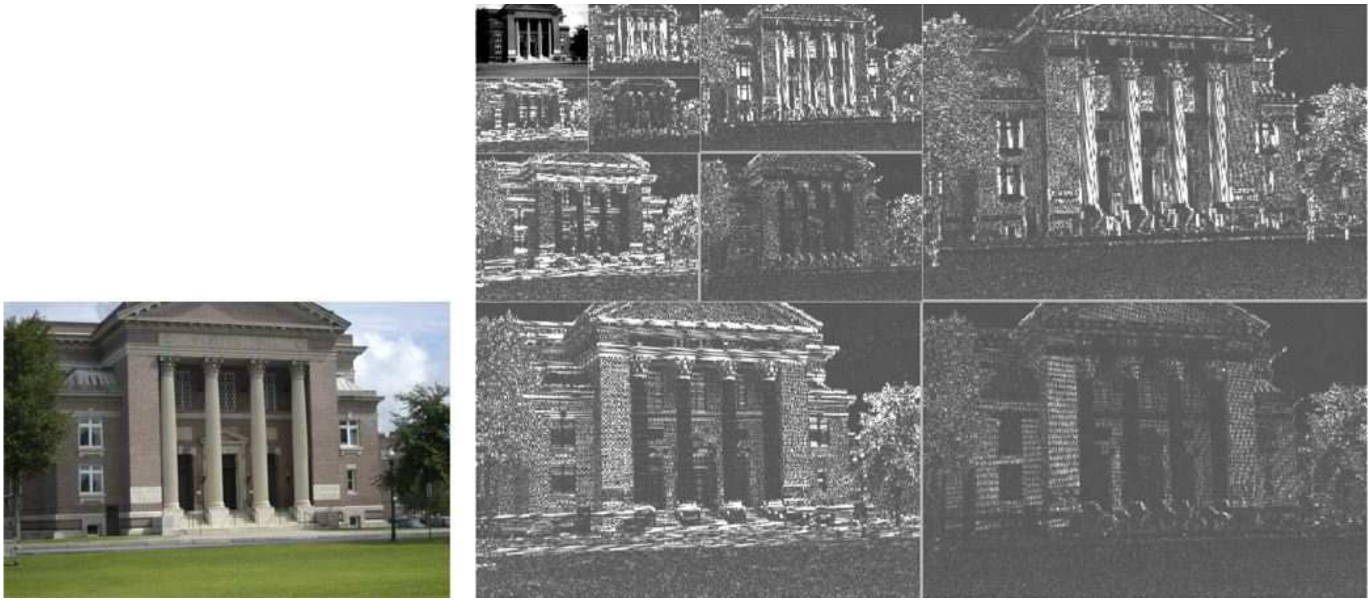


Fig. 3. A three-scale, three-orientation decomposition of the green channel of the image shown to the left. Shown are, starting from the lower-left and moving counter-clockwise, the horizontal, diagonal, and vertical subbands at each of three scales, with the residual low-pass subband in the upper left corner.

Given the large number of constraints (one per pixel) in only nine unknowns, it is generally safe to assume that the 9×9 matrix $Q^T Q$ will be invertible.

Given the linear predictor, the log error between the actual coefficient and the predicted coefficient magnitudes is:

$$\vec{p} \;=\; \log(\vec{v}) - \log(|Q\vec{w}|), \tag{12}$$

where the $\log(\cdot)$ is computed point-wise on each vector component. It is from this error that additional statistics are collected, namely the mean, variance, skewness and kurtosis. This process is repeated for scales $i = 1, ..., n-1$, and for the subbands V_i^r and V_i^b, where the linear predictors for these subbands are of the form:

$$\begin{aligned} |V_i^r(x,y)| &= w_1|V_i^r(x-1,y)| + w_2|V_i^r(x+1,y)| \\ &+ w_3|V_i^r(x,y-1)| + w_4|V_i^r(x,y+1)| \\ &+ w_5|V_i^r(x/2,y/2)| + w_6|D_i^r(x,y)| \\ &+ w_7|D_{i+1}^r(x/2,y/2)| + w_8|V_i^g(x,y)| \\ &+ w_9|V_i^b(x,y)|, \end{aligned} \tag{13}$$

and

$$\begin{aligned} |V_i^b(x,y)| &= w_1|V_i^b(x-1,y)| + w_2|V_i^b(x+1,y)| \\ &+ w_3|V_i^b(x,y-1)| + w_4|V_i^b(x,y+1)| \\ &+ w_5|V_i^b(x/2,y/2)| + w_6|D_i^b(x,y)| \\ &+ w_7|D_{i+1}^b(x/2,y/2)| + w_8|V_i^r(x,y)| \\ &+ w_9|V_i^g(x,y)|. \end{aligned} \tag{14}$$

A similar process is repeated for the horizontal and diagonal subbands. As an example, the predictor for the green channel takes the form:

$$\begin{aligned} |H_i^g(x,y)| &= w_1|H_i^g(x-1,y)| + w_2|H_i^g(x+1,y)| \\ &+ w_3|H_i^g(x,y-1)| + w_4|H_i^g(x,y+1)| \\ &+ w_5|H_i^g(x/2,y/2)| + w_6|D_i^g(x,y)| \\ &+ w_7|D_{i+1}^g(x/2,y/2)| + w_8|H_i^r(x,y)| \\ &+ w_9|H_i^b(x,y)|, \end{aligned} \tag{15}$$

$$\begin{aligned} |D_i^g(x,y)| &= w_1|D_i^g(x-1,y)| + w_2|D_i^g(x+1,y)| \\ &+ w_3|D_i^g(x,y-1)| + w_4|D_i^g(x,y+1)| \\ &+ w_5|D_i^g(x/2,y/2)| + w_6|H_i^g(x,y)| \\ &+ w_7|V_i^g(x,y)| + w_8|D_i^r(x,y)| \\ &+ w_9|D_i^b(x,y)|. \end{aligned} \tag{16}$$

For the horizontal and diagonal subbands, the predictor for the red and blue channels are determined in a similar way as was done for the vertical subbands, Equations (13)-(14). For each oriented, scale and color subband, a similar error metric, Equation(12), and error statistics are computed.

For a decomposition with scales $i = 1, ..., n$, the total number of basic coefficient statistics is $36(n-1)$ ($12(n-1)$ per color channel), and the total number of error statistics is also $36(n-1)$, yielding a total of $72(n-1)$ statistics. These statistics form the first half of the feature vector to be used to discriminate between clean and stego images.

D. Phase Statistics

Given the local angular harmonic decomposition (LAHD) of Section II-B.2, a measure of relative phase, as described in [25], is computed as follows:

$$\phi_{p,q}^{c_1,c_2}(x,y) \;=\; \angle\{[A_q^{c_1}(x,y)]^p, [A_p^{c_2}(x,y)]^q\}, \tag{17}$$

where $\angle\{.,.\}$ is the angle between two complex numbers, and $c_1, c_2 \in \{r, g, b\}$. From the relative phase, the following rotation invariant signature, as described in [25], is computed:

$$\begin{aligned} s_{p,q}^{c_1,c_2}(x,y) &= \sqrt{|A_q^{c_1}(x,y)||A_p^{c_2}(x,y)|} \\ &\times \ \exp(j\phi_{p,q}^{c_1,c_2}(x,y)), \end{aligned} \tag{18}$$

where $|\cdot|$ denotes magnitude and $j = \sqrt{-1}$.

From the 1^{st}- through N^{th}-order LAHDs, $6N(N-1)$ signatures are collected from within and across color channels (there are $N(N-1)/2$ combinations of LAHD orders, 6 ordered combinations of color channels, and 2 statistics per combination, yielding $6N(N-1)$). The phase statistics are collected from the two-dimensional distribution of these signatures in the complex plane. Specifically, assuming zero-mean data, we consider the covariance matrix:

$$M_{p,q}^{c_1,c_2} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}, \tag{19}$$

where:

$$m_{11} = \frac{1}{S}\sum_{x,y}\Re(s_{p,q}^{c_1,c_2}(x,y))^2 \tag{20}$$

$$m_{12} = \frac{1}{S}\sum_{x,y}\Re(s_{p,q}^{c_1,c_2}(x,y))\Im(s_{p,q}^{c_1,c_2}(x,y)) \tag{21}$$

$$m_{22} = \frac{1}{S}\sum_{x,y}\Im(s_{p,q}^{c_1,c_2}(x,y))^2 \tag{22}$$

$$m_{21} = m_{12}, \tag{23}$$

where S is the total number of signatures, and $\Re(\cdot)$ and $\Im(\cdot)$ correspond to the real and imaginary components of a complex quantity. The structure of this covariance matrix is captured by the measures:

$$\mu_1 = \frac{\min(m_{11}, m_{22})}{\max(m_{11}, m_{22})}, \tag{24}$$

and,

$$\mu_2 = \frac{m_{12}}{\max(m_{11}, m_{22})}. \tag{25}$$

Considering this distribution as a scaled and rotated Gaussian distribution, the first measure corresponds to the relative scales along the minor and major axes, and the second of these measures to the orientation of the distribution.

In order to capture these phase statistics at various scales, this entire process is repeated for several levels of a Gaussian pyramid decomposition of the image [27]. These statistics form the second half of the feature vector to be used to discriminate between clean and stego images.

E. Summary

Here we summarize the construction of the statistical feature vector from a color (RGB) image.

1) Build a n-level, 3-orientation QMF pyramid for each color channel, Equations (1)-(4).
2) For scales $i = 1, \ldots, n-1$ and for orientations V, H and D, across all three color channels, $c \in \{r, g, b\}$, compute the mean, variance, skewness, and kurtosis of the subband coefficients. This yields $36(n-1)$ statistics.
3) For scales $i = 1, \ldots, n-1$, and for orientations V, H and D, across all three color channels, $c \in \{r, g, b]\}$, build a linear predictor of coefficient magnitude, Equation (11). From the error in the predictor, Equation (12), compute the mean, variance, skewness, and kurtosis. This yields $36(n-1)$ statistics.
4) Build a n-level Gaussian pyramid for each color channel. For each level of the pyramid, compute the 1^{st}- through N^{th}-order LAHD, Equation (6). Compute the relative phases, Equation (17). Compute the rotation invariant signature, Equation (18), across all color channels and LAHD orders, from which the covariance matrix, Equation (19), and subsequent phase statistics are extracted, Equation (24) and (25). This yields $6N(N-1)n$ statistics.

III. Classification

From the measured statistics of a training set of clean and stego images, the goal is to determine whether a test image contains a hidden message. To this end, we employ support vector machines (SVM) [28]–[30]. In the next section, we consider the effectiveness of linear, non-linear and one-class SVMs – see [13], [19] for the full details on the construction of these classifiers.

IV. Results

We have collected $40,000$ natural images[2]. These color images span a range of indoor and outdoor scenes, are JPEG compressed with an average quality of 90%, and typically are 600×400 pixels in size (on average, 85.7 kilobytes). To contend with slight variations in image size, only the central 256×256 region of each image was considered in the analysis. Statistics, as described in Section II, were collected from each image, yielding a 432-D feature vector of magnitude and phase statistics. For the magnitude statistics, a four-level, three-orientation QMF pyramid was constructed using 9-tap filters, from which 108 marginal and 108 error statistics are collected. For the phase statistics a three-level Gaussian pyramid was constructed (using the 5-tap binomial filter $[1\ 4\ 6\ 4\ 1]/16$), from which the 1^{st}- through 4^{th}-order LAHDs are computed, to yield 216 phase statistics [3].

Next, $40,000$ stego images were generated by embedding messages of various sizes into the full-resolution cover images. The messages consisted of central regions of randomly chosen images from the same image database with sizes 6.0, 4.7, 1.2, 0.3 kilobytes (K), corresponding to 100%, 78%, 20% and 5% of total cover capacity. The total cover capacity is defined to be the maximum size of a message that can be embedded by the embedding algorithm. Since this quantity will vary depending on the cover image, we compute the average capacity across $1,000$ cover images. The steganography capacity is then the

[2] All natural images were downloaded from www.freefoto.com - all images were photographed with a range of different films, cameras, and lenses, and digitally scanned.

[3] Since the N^{th}-order LAHD requires an N^{th}-order discrete derivative, the computation of higher-order LAHDs requires similarly higher-order derivatives. While computing higher-order discrete derivatives can be challenging, we find that we are able to compute stable LAHDs up to 4^{th}-order.

ratio between the size of the embedded message and the total cover capacity. Messages were embedded using Jsteg [31], Outguess [32], Steghide [33], Jphide [34] and F5 [35]. Each stego image was generated with the same quality factor as the original cover image so as to minimize double JPEG compression artifacts. The same statistical feature vector as described above was computed from the central 256×256 region of each stego image. In all of the results presented below, $32,000$ of the clean and stego images were used to train a SVM, and the remaining $8,000$ images were used in testing – throughout, results from the testing stage are presented.

Shown in Fig. 4 is the detection accuracy for each of five steg embedding programs, four different message sizes, and for the following types of SVMs (a) linear, (b) non-linear (RBF kernel) and (c) one-class with six hyperspheres (fewer hyperspheres led to poor generalization, and more hyperspheres led to over-fitting). All SVM parameters were optimized by a grid search that optimized the SVM training and testing accuracy. Specifically, the classifier was trained on $30,000$ cover and $30,000$ stego images, and then tested on $2,000$ cover and $2,000$ stego images. After this search was completed, the parameter set resulting in the best performance, while keeping the false positive rate fixed, was used as the starting point for the next round of training that reduced the granularity of the parameter search. The left-most gray bar in Fig. 4 corresponds to the detection accuracy of clean images which is, on average, greater than 99.0% (the false-positive rate, a clean image incorrectly classified as stego, is 100 minus this value). For the linear SVM, the average detection accuracy is 44.7%, 26.7%, 11.2% and 1.0% for embeddings at capacities 100%, 78%, 20% and 5%, with a maximum/minimum detection accuracy of $61.4\%/31.1\%$, $31.2\%/16.5\%$, $12.3\%/8.3\%$, and $2.3\%/0.3\%$. For the non-linear SVM[4], the average detection accuracy is 78.2%, 64.5%, 37.0% and 7.8% with a maximum/minimum detection accuracy of $91.1\%/66.4\%$, $76.4\%/51.8\%$, $43.2\%/31.3\%$ and $11.2\%/4.8\%$. For the one-class SVM[4], the average detection accuracy is 76.9%, 61.5%, 30.3% and 5.4%, with a maximum/minimum detection accuracy of $92.4\%/64.4\%$, $79.6\%/49.2\%$, $42.3\%/15.8\%$ and $8.9\%/2.7\%$. The non-linear SVM gives a clear benefit over the linear SVM, while the one-class SVM results in only a modest degradation in detection accuracy, while affording a simpler training stage. For point of comparison, the results for the non-linear SVM (Fig. 4(b)) are annotated with the detection accuracy for the linear SVM (Fig. 4(a)), and the detection accuracy for the one-class SVM (Fig. 4(c)) are annotated with the detection accuracy for the non-linear SVM (Fig. 4(b)).

Shown in Fig. 5(a) is the detection accuracy for a non-linear SVM[4] with a classification accuracy of 99.9% (0.1% false-positive rate). The average detection accuracy is 70.7%, 56.5%, 27.7% and 3.9% for embeddings at capacities 100%, 78%, 20% and 5%, with a maximum/minimum detection accuracy of $86.3\%/58.3\%$, $71.2\%/42.1\%$, $37.8\%/14.6\%$ and $7.1\%/1.3\%$. For point of comparison, these results are annotated with the detection accuracy for the non-linear SVM

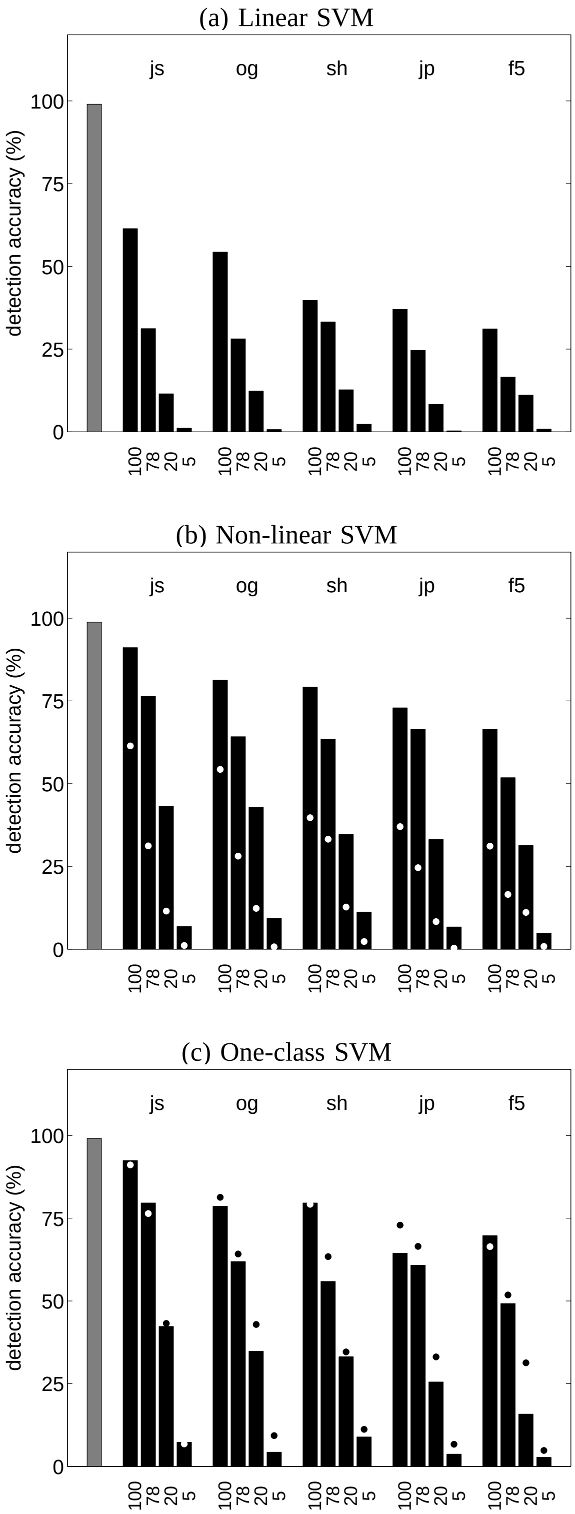


Fig. 4. Classification accuracy for (a) linear, (b) non-linear, and (c) one-class SVMs. The left-most gray bar corresponds to the classification accuracy (the false-positive rate, a clean image classified as stego, is 100 minus this value). Each group of four bars corresponds to different steg embedding programs (jsteg (js); outguess (og); steghide (sh); jphide (jp); and F5 (f5)). The numeric values on the horizontal axes correspond to the message size (as a percentage of cover capacity). For point of comparison, the dots in panel (b) correspond to the detection accuracy of panel (a); and the dots in panel (c) correspond to the detection accuracy of panel (b).

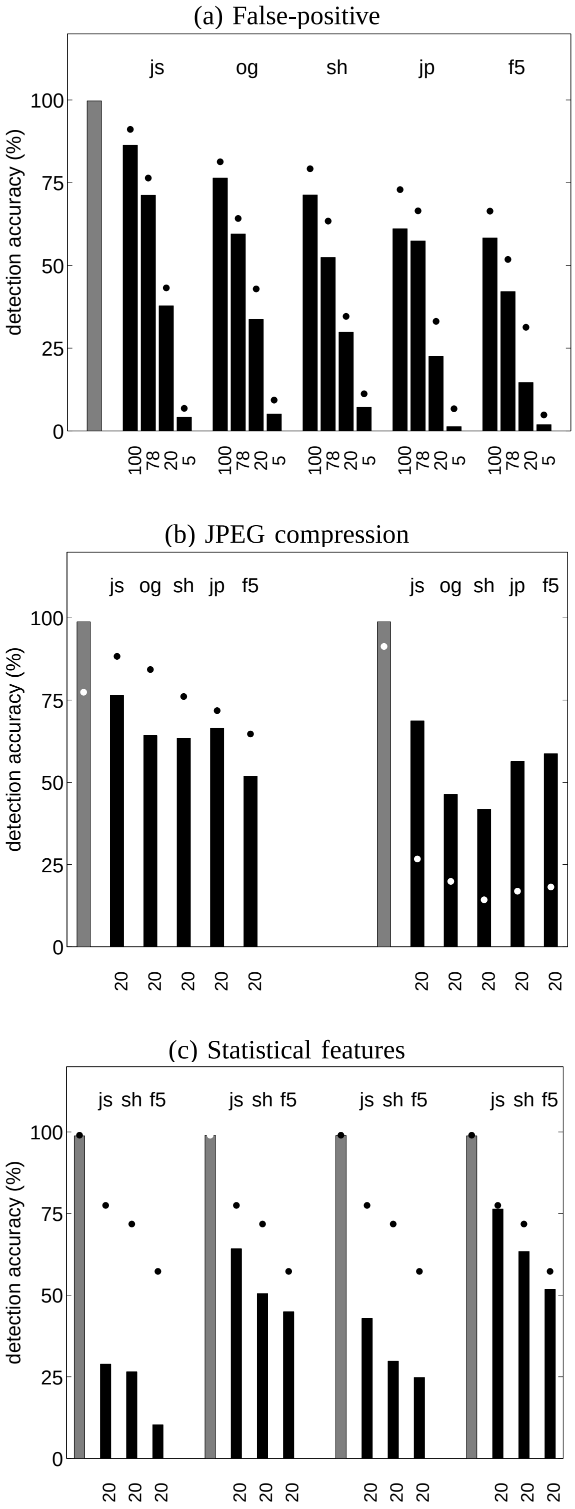
[4] *LIBSVM* [36], with a radial basis kernel, was used as the underlying SVM algorithm.

with a 99.0% detection accuracy (1.0% false-positive rate), Fig. 4(b). Note that an order of magnitude lower false-positive rate results in a relatively small degradation in detection accuracy.

Shown in Fig. 5(b) is the detection accuracy for a non-linear SVM[4] trained on JPEG images with quality factor 90 (left) or 70 (right) and then tested on JPEG images with quality 90 and 70. The classifier trained and tested on images with quality factor 90, and a message embedded at a capacity of 20%, achieves an average detection accuracy of 64.5% with a classification accuracy of 98.8% (1.2% false-positive rate). When tested on images of quality factor 70, this same classifier achieves an average detection accuracy of 77.0%. This higher accuracy seems, at first glance, to be a bit puzzling, but note that the classification accuracy decreases to 77.4% (22.6% false-positive rate), rendering this classifier largely useless for image qualities other than those near to the training images. The classifier trained and tested on images with quality factor 70 achieves an average detection accuracy of 54.4% with a classification accuracy of 98.8% (1.2% false-positive rate). When tested on images of quality factor 90, this same classifier achieves an average detection accuracy of only 19.2%, with a classification accuracy of 91.3% (8.7% false-positive rate), again rendering this classifier largely useless for image qualities other than those near to the training images. For point of comparison the dots on the left-most bars correspond to testing on quality factor 70 after training on quality factor 90, and the dots on the right-most bars correspond to testing on quality factor 90 after training on quality factor 70. These results show that our classifiers do not generalize well to new JPEG quality factors, but that individually trained classifiers, on several JPEG quality factors, are able to detect steg in carrier images of varying compression factors.

Shown in Fig. 5(c), from left to right, is the detection accuracy for a non-linear SVM[4] trained with magnitude marginal statistics only, magnitude error statistics only, phase statistics only, and magnitude marginal and error statistics. For point of comparison, the dots correspond to a non-linear SVM trained on the complete set of statistics. These results show that the combined magnitude and phase statistics provide for better detection accuracy than only a subset of the statistics. The phase statistics, however, provide only an incremental improvement in overall accuracy – see below for a further discussion on this.

While the previous results were based on JPEG cover images, the results presented next are for TIFF and GIF format images (converted from their original JPEG format – performance on these previously compressed images may not be perfectly representative of true TIFF or GIF images). For the TIFF cover images, messages were embedded using a generic least significant bit (LSB) algorithm. These messages were of sizes $84.6K$, $75.0K$, $18.8K$ and $4.6K$ corresponding to embedding capacities of, approximately, 100%, 89%, 22% and 5%. For the GIF cover images, messages were embedded using EZStego [37]. These messages were of sizes $26.2K$, $22.7K$, $6.7K$ and $1.6K$ corresponding to embedding capacities of, approximately, 100%, 85%, 25% and 6%. Shown in Fig. 6 is the detection accuracy for non-linear SVMs[4] separately trained


(a) False-positive

(b) JPEG compression

(c) Statistical features

Fig. 5. Classification accuracy for (a) a non-linear SVM with 0.1% false-positives. For point of comparison, the dots correspond to the detection accuracy for a non-linear SVM with 1.0% false-positives; (b) a non-linear SVM trained on JPEG images with quality factor 90 (left) or 70 (right). The dots on the left-most bars correspond to testing on quality factor 70, and the dots on the right-most bars correspond to testing on quality factor 90; and (c) a non-linear SVM trained on (from left to right) magnitude marginal statistics only, magnitude error statistics only, phase statistics only, and magnitude marginal and error statistics. The dots correspond to a non-linear SVM trained on the complete set of statistics. The gray bars correspond to the classification accuracy (the false-positive rate, a clean image classified as stego, is 100 minus this value). Each group of bars corresponds to different steg embedding programs (jsteg (js); outguess (og); steghide (sh); jphide (jp); and F5 (f5)). The numeric values on the horizontal axes correspond to the message size (as a percentage of cover capacity).

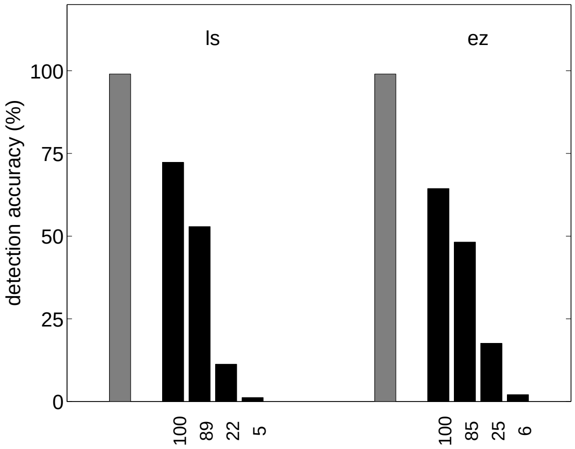


Fig. 6. Classification accuracy for non-linear SVM on TIFF and GIF format images. The gray bars correspond to the classification accuracy (the false-positive rate, a clean image classified as stego, is 100 minus this value). Each group of four bars corresponds to different steg embedding programs (a generic LSB embedding in TIFF (ls) and EzStego in GIF (ez). The numeric values on the horizontal axis correspond to the message size (as a percentage of cover capacity).

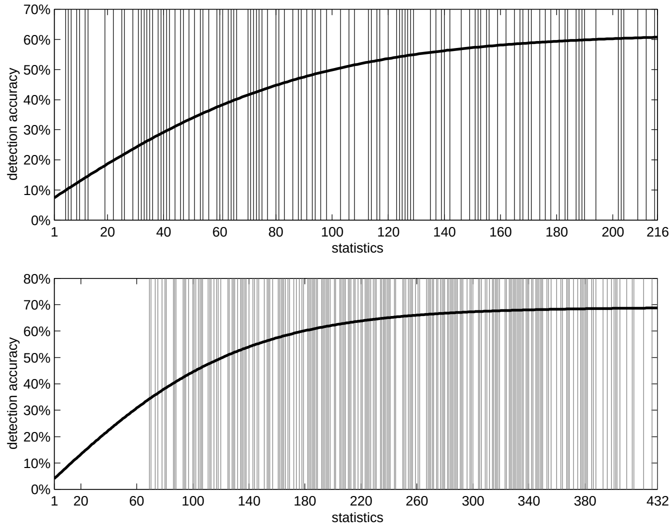


Fig. 7. Shown is the detection accuracy of linear classifiers as a function of (top) the number and category of feature (coefficient or error); and (bottom) the number and category of feature (magnitude or phase). The horizontal axis corresponds to the number of statistics incorporated, and the vertical axis corresponds to the detection accuracy in percentage. In the top panel the white and gray stripes correspond to error and coefficient statistics, respectively. In the bottom panel the white and gray stripes correspond to the magnitude and phase statistics, respectively.

on the TIFF and GIF images. Each group of bars corresponds to a different embedding algorithm: from left to right, LSB (ls) and EZStego (ez). The gray bars correspond to a detection accuracy of 99.0% (1.0% false-positive rate). For the TIFF images, the detection accuracy is 72.3%, 52.9%, 11.3% and 1.2%.

Note that the overall detection accuracy for these TIFF carriers is lower than for the JPEG carriers, which may, at first, seem surprising given that the uncompressed TIFF images provide a considerably larger cover medium. We suspect that the reason for this difference is that the TIFF embedding is LSB-based, while the JPEG embedding is block DCT-based. As a result, changes to the DCT coefficients affect an entire 8×8 pixel block, whereas changes to the LSB only affect a single pixel. For the GIF images, the detection accuracy is 64.4%, 48.2%, 17.6% and 2.1%. In terms of embedding capacity, these detection rates are slightly lower than the detection accuracy for JPEG cover images.

We wondered which set of statistics, coefficient, error or phase, were most crucial for the classifier. Shown in Fig. 7 (top panel) is the accuracy of classifiers plotted against the number and category of feature (coefficient or error) for the linear classifier [5]. We began by choosing the single feature, out of the 216 possible coefficient and error features, that gave the best classification accuracy. This was done by building 216 classifiers each based on a single feature, and choosing the feature that yielded the highest accuracy (the feature was the variance in the error of the green channel's diagonal band at the second scale). We then choose the next best feature from the remaining 215 components. This process was repeated until all features were selected. The solid line in Fig. 7 (top panel) is the accuracy as a function of the number of features. The white and gray regions correspond to error and coefficient features, respectively. That is, if the feature included on the i^{th} iteration is a coefficient then we denote that with a vertical gray line at the i^{th} position on the horizontal axis. Note that the coefficient and error statistics are interleaved, showing that both sets of statistics are important for classification. Shown in Fig. 7 (bottom panel) is the accuracy of the classifier plotted against the number and category of feature (magnitude or phase). In this case, it is clear that the magnitude statistics (coefficient and error) are far more important than the phase statistics. That is, the first 70 categories belong to the magnitude statistics (for which we don't differentiate between coefficient or error). We were surprised that the phase statistics did not provide a larger boost to the overall detection accuracy. There are several possible reasons for this: (1) our specific statistical model for phase simply fails to capture the relevant phase statistics of natural images; (2) our phase statistics do capture the relevant phase statistics, but the steg embedding algorithms do not disturb these statistics; or (3) what we think most likely, the magnitude error statistics implicitly capture similar properties of the phase statistics – that is, geometric regularities (e.g., edges) are explicitly captured by the phase statistics through correlations between the angular harmonics, while these same regularities are implicitly captured by the error statistics through correlations of the magnitude across space and scale.

V. Related Work

There are, of course, many steganalysis techniques that have emerged over the past few years. While many of these are specific to individual embedding programs, a few are universal, or near-universal approaches. In this section we attempt to compare the effectiveness of our approach to that

[5] This analysis was performed only on the linear classifier because the computational cost of retraining $23,220 = 216 + \cdots + 1$ non-linear classifiers was prohibitive. We expect a similar pattern of results for the non-linear SVM.

of Fridrich [17], as it has clearly emerged has one of the most effective techniques.

Fridrich extracted statistical measurements based on marginal and joint DCT statistics, from clean and stego images. A Fisher linear discriminant classifier was then trained and tested on a collection of $1,800$ images. While there are some obvious high-level similarities to our approaches, a direct comparison is difficult since (1) Fridrich's approach was specifically designed to detect steganography in JPEG images while our approach was applied to JPEG, GIF, and TIFF formats; (2) Fridrich employed a linear classifier while we employed linear and non-linear classifiers; (3) Fridrich tested her approach on $1,800$ grayscale images, while we tested ours on $40,000$ color images; and (4) Fridrich employed only 23 statistical features, while we employed a considerably larger 432 features.

With these caveats in mind, we compared the performance of our approaches on OutGuess and F5. For a 1% false-positive rate and an embedding rate for Outguess of 0.05 and 0.1 bpc [6] (bits per non-zero DCT coefficient), our detection accuracies (non-linear SVM) were 53.8% and 71.3% while those of Fridrich were 31.1% and 99.1%. For a 1% false-positive rate and an embedding rate for F5 of 0.05 and 0.1 bpc, our detection accuracies were 10.7% and 26.3% while those of Fridrich were 2.6% and 7.2%. While our approach seems to be more effective at lower embedding rates, Fridrich's approach is more effective at higher embedding rates. This is particularly impressive given the low-dimensional feature vector and the use of only a linear classifier.

VI. Discussion

We have described a universal approach to steganalysis that relies on building a statistical model of first- and higher-order magnitude and phase statistics extracted from multi-scale, multi-orientation image decompositions. We have shown that these statistics are relatively consistent across a broad range of images, but are disturbed by the presence of hidden messages. We are able to reliably detect, with a fairly low false-positive rate, the presence of hidden messages embedded at or near the full capacity of the underlying cover image. As the message size becomes smaller, the chance of detection falls – messages utilizing approximately 5% of the cover are unlikely to be detected. We expect that as universal steganalysis continues to improve, steganography tools will simply embed their messages into smaller and smaller portions of the cover image. As a result, hidden messages will continue to be able to be transmitted undetected, but high-throughput steganography will become increasingly more difficult to conceal.

Acknowledgment

This work was supported by an Alfred P. Sloan Fellowship, an NSF CAREER Award (IIS99-83806), a gift from Adobe Systems, Inc. and under a grant (2000-DT-CX-K001) from the U.S. Department of Homeland Security, Science and Technology Directorate (points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security or the Science and Technology Directorate).

References

[1] D. Kahn, "The history of steganography," in *Proceedings of Information Hiding, First International Workshop*, Cambridge, UK, 1996.

[2] R. Anderson and F. Petitcolas, "On the limits of steganography," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 474–481, 1998.

[3] N. Johnson and S. Jajodia, "Exploring steganography: seeing the unseen," *IEEE Computer*, vol. 31, no. 2, pp. 26–34, 1998.

[4] E. Petitcolas, R. Anderson, and M. Kuhn, "Information hiding - a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[5] J. Fridrich and M. Goljan, "Practical steganalysis: State of the art," in *SPIE Photonics West, Electronic Imaging*, San Jose, CA, 2002.

[6] J. Fridrich, M. Goljan, D. Hogea, and D. Soukal, "Quantitative steganalysis of digital images: Estimating the secret message length," *ACM Multimedia Systems Journal, Special issue on Multimedia Security*, vol. 9, no. 3, pp. 288–302, 2003.

[7] N. Johnson and S. Jajodia, "Steganalysis of images created using current steganography software," *Lecture notes in Computer Science*, vol. 1525, pp. 273–289, 1998.

[8] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," in *Proceedings of Information Hiding, Third International Workshop*, Dresden, Germany, 1999.

[9] N. Provos and P. Honeyman, "Detecting steganographic content on the internet," University of Michigan, Tech. Rep. CITI 01-1a, 2001.

[10] J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG images: Breaking the F5 algorithm," in *5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, 2002.

[11] A. Westfeld, "Detecting low embedding rates," in *5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, 2002.

[12] X. Wu, S. Dumitrescu, and Z. Wang, "Detection of LSB steganography via sample pair analysis," in *5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, 2002.

[13] S. Lyu and H. Farid, "Detecting hidden messages using higher-order statistics and support vector machines," in *5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, 2002.

[14] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 221–229, 2002.

[15] J. Fridrich, M. Goljan, and D. Hogea, "New methodology for breaking steganographic techniques for JPEGs," in *SPIE Symposium on Electronic Imaging*, Santa Clara, CA, 2003.

[16] J. Harmsen and W. Pearlman, "Steganalysis of additive noise modelable information hiding," in *SPIE Symposium on Electronic Imaging*, Santa Clara, CA, 2003.

[17] J. Fridrich, "Feature-based steganalysis for jpeg images and its implications for future design of steganographic schemes," in *6th International Workshop on Information Hiding*, Toronto, Canada, 2004.

[18] H. Farid, "Detecting hidden messages using higher-order statistical models," in *International Conference on Image Processing*, Rochester, New York, 2002.

[19] S. Lyu and H. Farid, "Steganalysis using color wavelet statistics and one-class support vector machines," in *SPIE Symposium on Electronic Imaging*, San Jose, CA, 2004.

[20] E. P. Simoncelli, "Statistical modeling of photographic images," in *Handbook of Image and Video Processing*, A. Bovik, Ed. Academic Press, 2005, ch. 4.7, 2nd edition.

[21] P. Vaidyanathan, "Quadrature mirror filter banks, M-band extensions and perfect reconstruction techniques," *IEEE ASSP Magazine*, vol. 4, no. 3, pp. 4–20, 1987.

[22] M. Vetterli, "A theory of multirate filter banks," *IEEE Transactions on ASSP*, vol. 35, no. 3, pp. 356–372, 1987.

[6] Our detection accuracies are given with respect to the total cover capacity, defined to be the maximum size of a message that can be embedded by the embedding algorithm. Comparable bpc values for these embedding rates were determined to allow for a direct comparison to Fridrich's results. For OutGuess, a bpc value of 0.05 and 0.1 corresponds to an embedding capacity of 44.2% and 88.5%, respectively. For F5, a bpc value of 0.05 and 0.1 corresponds to an embedding capacity of 7.8% and 15.7%, respectively.

[23] E. Simoncelli and E. Adelson, *Subband image coding*. Kluwer Academic Publishers, 1990, ch. Subband transforms, pp. 143–192.

[24] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49–71, 2000.

[25] E. P. Simoncelli, "A rotation-invariant pattern signature," in *International Conference on Image Processing*, 1996.

[26] R. Buccigrossi and E. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 8, no. 12, pp. 1688–1701, 1999.

[27] J. Ogden, E. Adelson, J. Bergen, and P. Burt, "Pyramid-based computer graphics," *RCA Engineer*, vol. 30, no. 5, pp. 4–15, 1985.

[28] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[29] ——, *Statistical Learning Theory*. John Wiley and Sons, 1998.

[30] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[31] D. Upham, *Jsteg*, software available at `ftp.funet.fi`.

[32] N. Provos, *Outguess*, software available at `www.outguess.org`.

[33] S. Hetzl, *Steghide*, software available at `steghide.sourceforge.net`.

[34] A. Latham, *Jpeg Hide-and-Seek*, software available at `linux01.gwdg.de/alatham/stego`.

[35] A. Westfeld, *F5*, software available at `wwwwrn.inf.tu-dresden.de/westfeld/f5`.

[36] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at `http://www.csie.ntu.edu.tw/˜cjlin/libsvm`.

[37] R. Machado, *EZStego*, software available at `www.ezstego.com`.