New Blind Steganalysis and its Implications

Miroslav Goljan[*], Jessica Fridrich, and Taras Holotyak
Department of Electrical and Computer Engineering
SUNY Binghamton, Binghamton, NY 13902-6000, USA

ABSTRACT

The contribution of this paper is two-fold. First, we describe an improved version of a blind steganalysis method previously proposed by Holotyak et al.[1] and compare it to current state-of-the-art blind steganalyzers. The features for the blind classifier are calculated in the wavelet domain as higher-order absolute moments of the noise residual. This method clearly shows the benefit of calculating the features from the noise residual because it increases the features' sensitivity to embedding, which leads to improved detection results. Second, using this detection engine, we attempt to answer some fundamental questions, such as "how much can we improve the reliability of steganalysis given certain a priori side-information about the image source?" Moreover, we experimentally compare the security of three steganographic schemes for images stored in a raster format – (1) pseudo-random ±1 embedding using ternary matrix embedding, (2) spatially adaptive ternary ±1 embedding, and (3) perturbed quantization while converting a 16-bit per channel image to an 8-bit gray scale image.

1. INTRODUCTION

Steganography is often formulated as the prisoners' problem. Two inmates, Alice and Bob, imprisoned in two different cells are trying to secretly hatch an escape plan.[2] The only way they can communicate with each other is through a channel that is monitored by warden Wendy. If Wendy detects any encrypted messages or anything that indicates covert communication between Alice and Bob, she will throw both of them into solitary confinement. The steganography problem is: "How can Alice and Bob prepare an escape plan by communicating through the monitored channel in a manner that doesn't raise Wendy's suspicion?" The steganalysis problem is the other side of this game: "How should Wendy analyze the channel traffic so that she doesn't miss any secret messages and also doesn't throw Alice and Bob into solitary confinement if they are innocent?"

In the first part of this paper, we concentrate on Wendy's task. We assume that Alice and Bob have access to a source of digital images (*cover images*). They use a steganographic algorithm for embedding messages by altering the image content and then exchange these *stego images* to communicate the message. The algorithm requires a secret key that they agreed upon before imprisonment. Wendy does not know the images before alteration and she does not know the secret key. Often, it is assumed that Wendy knows the steganography algorithm (Kerckhoffs' Principle) and, possibly, the source of cover images.

In practice, Wendy may not know which stego algorithm is in use, in which case she is interested in a "blind" steganalysis that does not assume any a priori knowledge about the embedding algorithm. In the past, the problem of blind steganalysis was approached using feature extraction and machine learning. Construction of blind steganalysis methods starts by extracting a set of features from the cover and stego objects and then training a classifier on this data with the goal to distinguish between cover and stego objects.

It is a firm belief of the authors of this paper that the best (most sensitive) features for steganalysis are obtained when they are calculated directly in the embedding domain. Thus, for example for JPEG images, the features should be constructed from DCT coefficients rather than their spatial representation. This principle is supported by recent comparisons of different blind steganalysis methods.[3,4] Accepting this principle implies a necessity to divide the task of

[*] mgoljan@binghamton.edu; phone 1 607 777-5793; fax 1607 777-4464.

blind steganalysis according to the embedding domain (e.g., JPEG and spatial domain steganalysis). In this paper, we focus on image steganography that is performed in the spatial domain.

The concept of blind steganalysis appeared for the first time in the work of Avcibas et al.[5] Farid et al.[6,7] proposed a 72-dimensional feature space (for grayscale images) consisting of the first four statistical moments of wavelet coefficients and their prediction errors. Harmsen et al.[8] used a simple three-dimensional feature vector obtained as the center of gravity of the three-dimensional Histogram Characteristic Function (HCF[a]). While this method gives good results for steganalysis of color images with a low noise level, such as previously compressed JPEG images, its performance is markedly worse for grayscale images and raw, never compressed images from digital cameras or scanners. Ker[9] substantially improved this method by introducing the concept of calibration. Holotyak et al.[1] used an approach similar to Farid's except they calculate the features from the noise component of the image in the wavelet domain. The authors also advocate usage of high statistical moments and show that a substantial benefit can be obtained by considering higher order moments. Xuan et al.[10] proposed features calculated as the first three absolute moments of the HCF of all 9 three level subbands in a Haar decomposition.

In this paper, we focus on the choice of the features rather than the classifier part. Good features should be sensitive to embedding modifications and insensitive to the image content. We follow the philosophy introduced in Ref.,[1] in which the features are not calculated from the stego image directly but from its noise component in the wavelet domain. This improves the SNR between the stego signal and the rest of the image and leads to more reliable detection. Instead of working with very high order normalized even moments of the noise residual as in Ref.,[1] we use absolute non-normalized moments of order 1 to 9. We call this method Wavelet Absolute Moment steganalysis (WAM). A closer look reveals that the first four moments are conceptually the same as the prediction errors used by Farid.[6] The denoising filter provides predictions for wavelet coefficients. A particular difference is in the type of the wavelet transform, quality of prediction, and skipped moment normalization.

In Section 2, we describe the features and construct a classifier using Fisher Linear Discriminant (FLD). We carefully compare its performance to current state-of-the-art blind steganalyzers on exactly the same databases. In Section 3, we study how much blind steganalysis can be improved by utilizing side information about the cover image source. As one might expect, narrowing the training set to better match the source of cover images improves steganalysis. In the last experimental part of this paper (Section 4), we use the WAM classifier to compare the security of several steganographic methods. Final conclusions are drawn in Section 5.

2. WAM BLIND STEGANALYSIS

In Ref.,[1] the authors proposed a new idea to calculate the features for steganalysis only from the noise component of the stego image in the wavelet domain. The noise component was obtained using the denoising filter due to Mihcak et al.[11] We reiterate that the denoising step increases the SNR between the stego signal and the cover image, thus making the features calculated from the noise residual more sensitive to embedding and less sensitive to image content. The denoising filter is designed to remove Gaussian noise from images under the assumption that the stego image is an additive mixture of a non-stationary Gaussian signal (the cover image) and a stationary Gaussian signal with a known variance (the noise). As the filtering is performed in the wavelet domain, all our features (statistical moments) are calculated as higher order moments of the noise residual in the wavelet domain. The detailed procedure for calculating the WAM features in a gray scale image is similar as in Ref.[1] and is shown below.

We point out the main differences of WAM features when compared to Ref.[1] In particular, we use *absolute* moments that are *not normalized* by variance and we do not use a nonlinear log transform. Using absolute moments allowed us to eliminate the need for very high moments that are not very significant discriminators between cover and stego image classes. The total number of features for a grayscale image is $3\times n_{\mathrm{mom}}$. For color images, the features are calculated for each color channel, bringing the total number of features to $9\times n_{\mathrm{mom}}$. As stated above, in this paper, we use $n_{\mathrm{mom}} = 9$. We set the parameter $\sigma_0^2 = 0.5$, which is the same value as in the referenced paper[1] and corresponds to the variance of the stego signal for an image fully embedded with ±1 embedding (see (1) below).

[a] HCF is the amplitude of the Fourier transform of the color image histogram.

Algorithm: Feature calculation

Step 1. Calculate the first level wavelet decomposition of the stego image with the 8-tap Daubechies QMF.[12] Denote the vertical, horizontal, and diagonal subbands as $h(i,j)$, $v(i,j)$, $d(i,j)$, where (i,j) runs through some index set J.

Step 2. In each subband, estimate the local variance of the cover image for each wavelet coefficient using the MAP estimation for 4 sizes of a square $N\times N$ neighborhood, for $N\in\{3, 5, 7, 9\}$

$$\hat{\sigma}_N^2(i,j) = \max\left(0, \frac{1}{N^2}\sum_{(i,j)\in N} w^2(i,j) - \sigma_0^2\right), \ (i,j)\in J.$$

Take the minimum of the 4 variances as the final estimate,

$$\hat{\sigma}^2(i,j) = \min\left(\hat{\sigma}_3^2(i,j), \hat{\sigma}_5^2(i,j), \hat{\sigma}_7^2(i,j), \hat{\sigma}_9^2(i,j)\right), \ (i,j)\in J.$$

Step 3. The denoised wavelet coefficients are obtained using the Wiener filter

$$h_{\mathrm{den}}(i,j) = h(i,j)\frac{\hat{\sigma}^2(i,j)}{\hat{\sigma}^2(i,j)+\sigma_0^{\ 2}} \text{ and similarly for } v(i,j), \text{ and } d(i,j), \ (i,j)\in J.$$

Step 4. Calculate the noise residual in each subband

$$r_h(i,j) = h(i,j) - h_{\mathrm{den}}(i,j) \text{ and similarly } r_v \text{ and } r_d \text{ for } v(i,j), \text{ and } d(i,j), \ (i,j)\in J.$$

Step 5. Denoting the mean value with a bar, calculate the absolute central moments of each noise residual for $p = 1,2,\ldots, n_{\mathrm{mom}}$

$$m_p = \frac{1}{|J|}\sum_{(i,j)\in J} |\, r_h(i,j) - \overline{r}_h \,|^p \ .$$

2.1 Comparing the WAM classifier to other blind classifiers

We now compare the performance of the proposed WAM classifier with the results reported for current state-of-the-art classifiers[1,9,14]. To make the comparison fair, we always compare on the same image database and the same testing methodology.

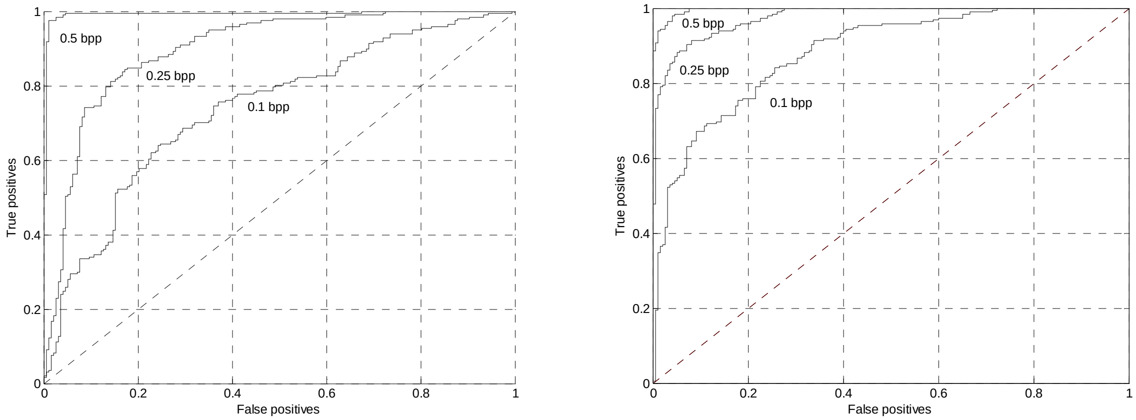
We start with the classifier by Holotyak et al.[1] and use the same classifier (FLD) and embedding method – random ±1 embedding. This trivial embedding method is arguably the simplest possible embedding that imposes the smallest distortion (pixel values are modified by at most one). Later, in Section 4, we use random ±1 embedding as a baseline with respect to which other embedding methods are compared.

We will assume that the cover and stego images are vectors of integers in a fixed range (e.g., for grayscale images this range is the set {0, 1, …, 255}). In random ±1 embedding, one message bit is communicated at pixel x by applying the embedding operation Emb1 to x, obtaining thus the pixel value y from the stego image

$$y = \mathrm{Emb1}(x) = \begin{cases} x-1 & \text{if } (r>0 \text{ or } x=255)\ \&\ b\neq \mathrm{LSB}(x) \\ x & \text{if } \ b = \mathrm{LSB}(x) \\ x+1 & \text{if } (r<0 \text{ or } x=0)\ \&\ b\neq \mathrm{LSB}(x), \end{cases} \tag{1}$$

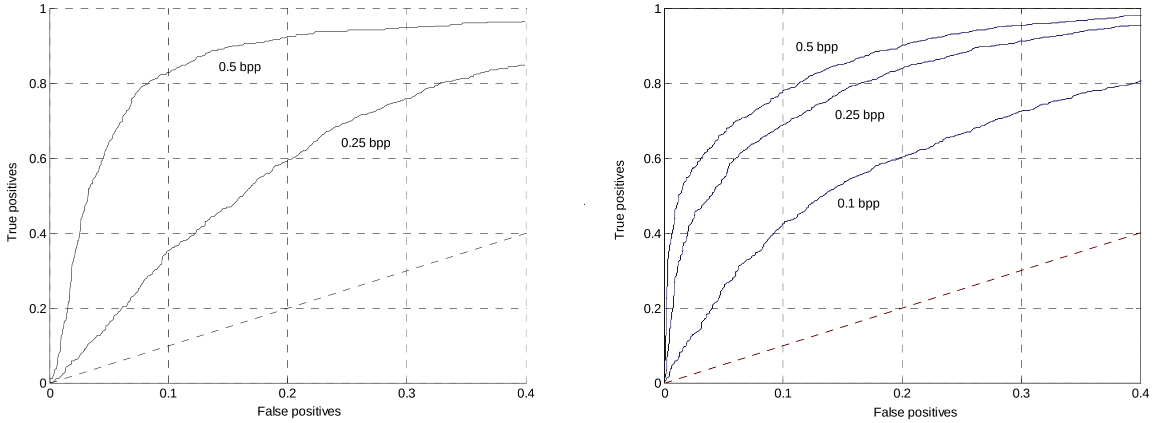
where r is an i.i.d. random variable with uniform distribution on {–1,1}, b is the message bit, and LSB(x) is the least significant bit of x. The word “random” in “random ±1 embedding” emphasizes that the pixels are selected (pseudo) randomly using a shared stego key for embedding. Note that for a given payload, the total energy of the stego signal is the same as for LSB embedding. However, steganalysis of ±1 embedding is much more difficult than for LSB embedding, for which astonishingly accurate detectors exist.[13]

In Figure 1, we show ROC curves representing the percentage of correctly detected stego images as a function of false positives (cover images detected as stego). The three curves correspond to relative payloads of 0.1, 0.25, and 0.5 bits/pixel (bpp). The FLD classifier with 3×9=27 features was trained separately for each payload size on 390 images divided into 195 cover and 195 stego images. All 195 source images were never compressed grayscale images taken with the Kodak DC290 camera. Note the significant reduction of false alarms for small payloads 0.1 and 0.25 bpp compared to the approach[1]. As in Ref.[1], the ROC shows the results of the training phase only – the separability of cover and stego images in the feature space.



a) **b)**

Figure 1. Class separability from Ref.[1] (a) and the WAM classifier (b) on raw images from Kodak DC290 camera.



a) **b)**

Figure 2. Class separability from Ref.[1] a) and the WAM classifier b) for a database of never compressed 2375 raw images from 22 digital cameras.

We now compare the detection results for a much more diverse source of cover images. Figure 2 shows the performance comparison of the WAM classifier with the method of Ref.[1] on 2375 never compressed images originated from 22 different digital cameras (see Appendix A). This database was partially obtained from[b] and partially from our own resources. It includes a wide variety of images, indoor/outdoor scenes, scenes taken with a without a flash, close-ups, landscapes, and images taken under different light conditions, temperature, etc. Those images that were originally taken in the 16-bit raw format were converted to 8-bit grayscale images. This database, which we call CAMERA_RAW in this paper, is also used in our experiments in Section 3 and 4.

We see again a radical improvement for small payloads and an overall reduction of false alarms. Also, by comparing Figure 1 with Figure 2, we note that the steganalyzer constructed for a specific image source (images from Kodak

[b] The Digital Forensic Image Library (DFIL), http://www.cs.dartmouth.edu/cgi-bin/cgiwrap/isg/imagedb.py.

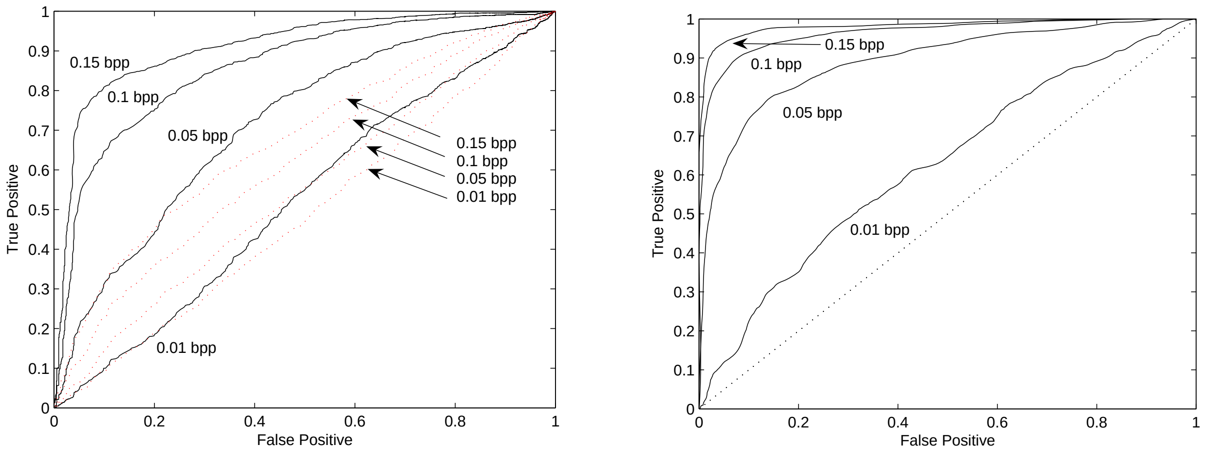
DC290) performs substantially better than for a more diverse source. This indicates that if prior information about the image source is available, training the classifier on a narrower source can improve the reliability of blind steganalysis by a large factor.

The third database, on which we compared the results, is a set of 2375 raw scans of negatives[c]. This is the most difficult class of images on which we test because raw scans of analog photographs or films are inherently very noisy and the detection of steganography is the most difficult. This is why the detection results for this class of images are the worst. In Table **1**, we include the comparison with Holotyak et al.[1] and also with Ker[9]. Instead of showing the ROCs, we report the false positives for 50% and 80% detection rates (as in Ker[9]). The results for the two methods referenced above are taken from Ref.[1]

	False positives at 50% detection rate	False positives at 80% detection rate
WAM	1.77	7.45
Blind Statistical Steganalysis in Ref.[1]	3.45	16.25
Steganalysis of LSB matching in Ref.[9]	7	27

Table 1. Percentage of false positives at 50% and 80% true detection rates compared to two published methods.

We now compare the WAM classifier to the classifier based on Binary Similarity Measures (BSM).[14] Since the authors of Ref.[14] show comparison of their results with Farid's classifier,[6] we obtain at the same comparison for both Ref.[14] and Ref.[6] We use the same "Greenspun" image database[d] and preprocess the images as in Ref.[14]. The images were converted to grayscale, black borders around them were cropped, and finally the images were recompressed with a quality factor of 75. We randomly chose 720 images and prepared an even mix of 4 times 180 stego-images embedded with ±1 embedding with relative payloads 0.01, 0.05, 0.1, and 0.15 bpp for training the classifier. The remaining set of 1080 images was randomly partitioned for testing to 540 original images and 540 stego for each payload. Figure 3a shows the ROC for Ref.[14,6] and WAM (b) averaged over 50 tests. We remark that Figure 3a was obtained using support vector machine classifier while our WAM classifier was implemented using a simple FLD classifier. Thus, further performance boost is expected after incorporating a better classifier.



a) **b)**

Figure 3. a) ROC for detection of ±1 embedding with four different payloads for the classifier based on BSMs[14] (solid line) and Farid's classifier[6] (dashed lines), b) Performance of the WAM with 3×9 features (data for figure 3a is provided courtesy of Mehdi Kharrazi from[14]).

[c] NRCS Photo Gallery, http://photogallery.nrcs.usda.gov

[d] Images downloaded from http://philip.greenspun.com/

3. WAM CLASSIFIER WITH SIDE INFORMATION ABOUT COVER SOURCE

When designing a blind steganalyzer in practice, when no prior information about the stego method or the cover image source is available, the best strategy is to train the classifier on as diverse image database as possible. However, in some cases, we may have some side information available about the source of covers. For example, we may know that the images are coming from a digital camera of a certain model. Obviously, we can incorporate this side information by training the classifier on images from a camera of the same make and model. Quick comparison of Figures 1 and 2 shows that detection of steganography in images produced by a specific Kodak DC290 camera is substantially more accurate if the classifier is only trained on images from the same camera. Intuitively, using side information about the cover source to narrow the training database should improve the detection results.

When dealing with images taken with digital cameras, the stego image itself is a source of additional information that might be taken into account, such as its size, format, color histogram, power spectrum, semantic content (e.g., indoor/outdoor scene, city/country), camera settings (e.g., with/without flash, ISO), camera sensor type (CCD/CMOS), camera make and model, etc.

3.1 Experiment

In this section, we use our WAM classifier to experimentally investigate how much improvement in steganalysis one can obtain by training the classifier on a narrower database determined by specific side-information about the cover image source. In particular, we ask the following questions

- If we know the camera make and model for the cover image source, how much improvement in detection reliability can be expected by training the classifier on images from the same camera model?
- How much does the detection improve if we train the classifier on images from the exact same camera that took the stego image compared to detection using classifier trained on images from a mixture of different cameras of the same model?

For this experiment, we used the CAMERA_RAW database of 2375 never compressed digital camera images because detection of spatial-domain steganography in raw images is much harder than in images previously JPEG compressed (description of the image database is in Section 2). Among the images from 22 cameras were 300 images from Olympus C765#1. To complete the experiments, we prepared another 400 images from another camera of the same model (Olympus C765#2) that contained mostly the same or similar scenes. All images were converted to grayscale.

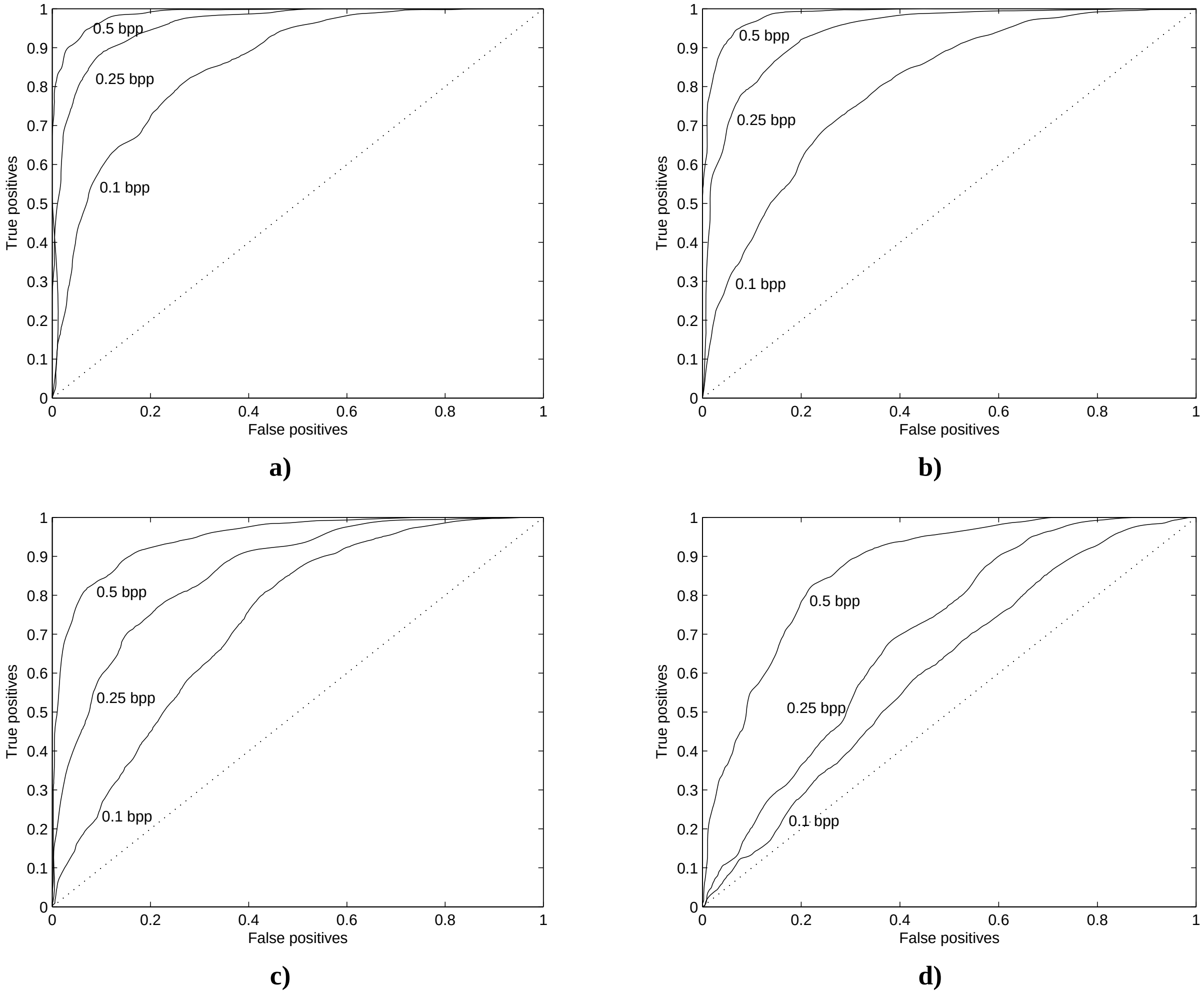
In Figure 4a), we show the ROC for an FLD classifier trained on 300 cover images from Olympus C765#2 and their embedded forms (300 stego images). The testing was performed on the remaining 100 images divided into 50 cover images and 50 stego images from Olympus C765#2. The test was carried out separately for three relative payloads, 0.1, 0.25, and 0.5 bpp. The ROC curves in Figure 4 were averaged over 50 random partitions of the test set. The remaining three experiments shown in Figure 4 b)–d) were performed using the same steps for different sets of the training and testing database.

Figure 4b) shows the FLD classifier trained on 400 test images from Olympus C765#1 and tested on 400 images from Olympus C765#2. Figure 4c) is the ROC for the FLD classifier trained on 400 test images from Olympus C765#2 and tested on 400 images from Olympus C765#1. In Figure 4d), the classifier was trained on all 2567 images, including Olympus C765#1, and tested on 400 images from Olympus C765#2.

The experimental results suggest that there is a significant gain when the WAM classifier is trained on images from the same camera make and model compared to the classifier trained on a much diverse database that included images from 22 other cameras (compare d) with c) or b) in Figure 4). For example, we have 1.1% false alarms for 50% detection when trained on the same camera at 0.25 bpp compared to 28.5% when trained on many other cameras that included Olympus C765#1 (the only Olympus in the set) but tested on Olympus C765#2 (see Table 2**)**. On the other hand, the difference between the results for a classifier trained on images from the very same camera and the classifier trained on the same make and model (but different camera) is smaller (compare rows (a) and (b) in Table 2). More experiments need to be carried out to verify this result on more camera models, however.

False positives at 50% detection	0.1 bpp	0.25 bpp	0.5 bpp
a) training and testing on Olympus C765#2	7.02	1.10	0.00
b) training on Olympus C765#1, testing on Olympus C765#2	13.75	1.59	0.03
c) training on Olympus C765#2, testing on Olympus C765#1	22.55	7.11	0.99
d) training on 22 cameras, testing on Olympus C765#2	36.09	28.51	8.81

Table 2. Percentage of false positives at 50% true detection for all experiments reported in Figure 4.



a)

b)

c)

d)

Figure 4. ROC curves for detection of random ±1 embedding with relative payloads 0.1, 0.25, and 0.5 bpp. a) training and testing on Olympus C765#2, b) training on Olympus C765#1, testing on Olympus C765#2, c) training on Olympus C765#2, testing on Olympus C765#1, d) training on images from 22 cameras, testing on Olympus C765#2.

4. USING BLIND CLASSIFIER AS BENCHMARK FOR STEGANOGRAPHY

Given a blind steganalyzer, we can use it to compare the security of different steganographic schemes as well as an oracle for constructing the least detectable embedding scheme. Of course, this verdict is dependent on the steganalyzer and may change when a different, more advanced steganalyzer is used. Nevertheless, some authors advocate that the concept of steganographic security should be relative to a steganalyzer.[15] In any case, it is quite intriguing to ask the question "what is the most secure steganographic method given the best current steganalysis engine." In this section, we use the WAM classifier to investigate the security of three advanced embedding techniques. We now explain each technique one by one.

4.1 Ternary ±1 embedding combined with matrix embedding

The ±1 embedding can be obviously improved because we can embed a ternary symbol t per each pixel x as $t = x \bmod 3$, rather than a bit, using the same distortion. Moreover, for small relative payloads we can apply matrix embedding[16,17] and further substantially decrease the number of embedding changes. We now briefly explain the principles of matrix embedding realized using a ternary [13, 10, 3] Hamming code with a parity check matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 2 & 1 & 2 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 2 & 1 & 2 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 2 & 1 & 0 & 1 & 1 & 2 & 1 \end{bmatrix}. \qquad (2)$$

The message is first transcoded from bits to ternary symbols {0,1,2}. The cover image is then divided into disjoint blocks of 13 pixels $\boldsymbol{x} = (x_1, x_2, \ldots, x_{13})$. We will embed 13–10=3 ternary message symbols $\boldsymbol{m}$ in each block by changing at most one x_i by 1, obtaining the modified block of pixels $\boldsymbol{y}$. The message symbols $\boldsymbol{m}$ are communicated as the syndrome $\boldsymbol{m}^\mathsf{T} = \mathbf{H}\boldsymbol{t}_y^\mathsf{T}$ of the vector $\boldsymbol{t}_y = \boldsymbol{y} \bmod 3$. Because the covering radius of the Hamming code is 1, the coset leader of every non-zero syndrome has Hamming weight 1 and thus it is guaranteed that we will never have to change more than one pixel in each block.

To give a specific example, let us assume that $\boldsymbol{t}_x = \boldsymbol{x} \bmod 3 = [2\ 1\ 1\ 0\ 0\ 1\ 0\ 2\ 1\ 0\ 2\ 1\ 0]$ and that we wish to embed the following three message symbols $\boldsymbol{m} = [1\ 0\ 2]$. We first calculate the difference $\boldsymbol{m}^\mathsf{T} - \mathbf{H}\boldsymbol{t}_x^\mathsf{T} = [2\ 0\ 1]^\mathsf{T}$, where "$^\mathsf{T}$" denotes transposition. Because the parity check matrix of the Hamming code contains all possible non-zero triples of symbols (up to a multiplication by a non-zero scalar), we can find in **H** a column that is a multiple of $[2\ 0\ 1]^\mathsf{T}$. Indeed, the 8th column is actually exactly equal to this vector. Thus, to adjust the syndrome $\mathbf{H}\boldsymbol{t}_x$ to the required message triple $\boldsymbol{m}$, we modify x_8 to $y_8 = x_8+1$ and set $y_i = x_i$ otherwise. Thus, the block of pixels $\boldsymbol{y}$ in the stego image is such that $\boldsymbol{t}_y = \boldsymbol{y} \bmod 3 = [2\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 2\ 1\ 0]$ (because 2 + 1 = 0 in ternary arithmetic). The recipient will simply read the three ternary message symbols as $\boldsymbol{m}^\mathsf{T} = \mathbf{H}\boldsymbol{t}_y^\mathsf{T}$. Note that in the lucky case when $\boldsymbol{m}^\mathsf{T} = \mathbf{H}\boldsymbol{t}_x^\mathsf{T}$, no embedding change is necessary. Assuming the message is a random stream of ternary symbols, this will happen with probability 26/27. Thus, the average number or changes per message *bit* is $26/27\ (3\log_2 3)^{-1} \approx 0.2025$. Note that the regular binary ±1 embedding (1) needs 0.5 changes per embedded bit.

In general, for ternary ±1 embedding realized using matrix embedding with ternary Hamming codes $[(3^r-1)/2, (3^r-1)/2-r, 3]$, $r = 1, 2, \ldots$, we embed $r\log_2 3$ bits per $(3^r-1)/2$ pixels using on average $1-3^{-r}$ changes. Thus, for relative message length $\dfrac{2r\log_2 3}{3^r-1}$ we embed $\dfrac{r\log_2 3}{1-3^{-r}}$ bits per embedding change or, equivalently, we make on average $\dfrac{2\left(1-3^{-r}\right)}{3^r-1}$ changes per pixel. Table 3 shows the expected number of modifications for several relative payloads for different values of r.

	r=2	r=3	r=4	r=5
Relative payload (bpp)	0.7925	0.3658	0.1585	0.0655
Embedding changes per pixel	0.2222	0.0741	0.0247	0.0082
Theoretical lower bound for embedding changes	0.1595	0.0557	0.0196	0.0068

Table 3. Relative payload and expected number of embedding changes per pixel as a function of the parameter *r* in matrix embedding using ternary Hamming codes.

We note that the theoretical lower bound for the average relative number of changes for any ternary code is the inverse ternary entropy of the relative payload p, $H_3^{-1}(p)$.[19] This bound is asymptotically approached with almost all $[n, n(1–p)]$ codes as their length approaches infinity. A version of this statement for binary codes is in Ref.[20] in Theorem 12.3.5 on page 325.

4.2 Adaptive ternary ±1 embedding

The second embedding method is a simple version of adaptive ±1 embedding. Following the intuitive argument that embedding changes are more difficult to detect in textured or noisy areas of the image, we select the pixels that carry message symbols as pixels with the largest variance in their 3×3 local neighborhood. Note that it is necessary to use wet paper codes (WPC)[19] because the local variance will change after embedding and the recipient may not identify exactly the same set of pixels from the stego image. Obviously, we can again use ternary alphabet for the message to improve the embedding efficiency. However, since we only embed in pixels with the largest variance, we now do not have any space for matrix embedding as in the case of random ±1 embedding.

The embedding function is Emb3 (3). Values $x = 0$ and $x = 255$ are changed to 1 and 254 before applying Emb3 to avoid under and over-flowing.

$$y = \text{Emb3}(x) = \begin{cases} x-1 & \text{if } t = (x-1 \bmod 3) \\ x & \text{if } t = (x \bmod 3) \\ x+1 & \text{if } t = (x+1 \bmod 3), \end{cases} \tag{3}$$

where t is a ternary message symbol. Note that we are now embedding $\log_2 3$ bits per pixel and making an embedding change with probability 2/3. Thus, we need on average $2/3 / \log_2 3 \approx 0.4206$ changes per embedded bit.

4.3 Perturbed quantization while decreasing the color depth

The third method is a version of perturbed quantization (PQ).[21] In this approach, we assume that the sender has access to a more accurate or detailed version of the cover image and embeds data while processing the image using an operation that includes quantization (e.g., resizing, JPEG compression, AD conversion, color depth reduction, etc.). In this paper, we use the operation of decreasing the bit depth of the cover image from 16 bits per color channel to 8 bits per channel. Indeed, some cameras allow storing their digital images with higher bit-depths. According to the methodology of PQ, the sender selects those pixels whose 16-bit values are in the middle of the lattice determined by 8 bit values. Wet paper codes (codes for memories with defective cells) must be used to communicate a message to the recipient because it is not possible to determine from the 8 bit per channel image which pixels were selected by the sender.

Note that PQ methods cannot benefit from ternary coding because each sample can only be rounded to two values.[21]

4.4 Embedding distortion

It is illustrative to calculate the expected embedding distortion for all three methods. We evaluate the embedding distortion as the mean square error (MSE) between the stego image and its raw, 16-bit form. To obtain closed-form expressions, we model each color channel of the raw 16-bit image as i.i.d. realizations of a real-valued random variable uniformly distributed in the interval [0,255]. For regular (binary) random ±1 embedding with relative payload p bpp, we have

$$MSE_{\pm 1} = 2\int_0^{0.5} p/4(1+x)^2 + (1-p/2)x^2 + p/4(1-x)^2 dx = \frac{1}{12} + \frac{p}{2}. \tag{4}$$

The distortion for both ternary ±1 embedding methods is obtained from (4) by substituting for p the equivalent length of random message that would incur $\frac{2\left(1-3^{-r}\right)}{3^r-1}$ embedding changes per pixel using binary ±1 embedding. Thus, for ternary ±1 embedding using ternary Hamming codes, $r = 2, 3, \ldots$

$$MSE_{\pm 1TH}\left(p\right) = \frac{1}{12} + 2\frac{2\left(1-3^{-r}\right)}{3^r-1}\frac{1}{2} = \frac{1}{12} + 2\frac{1-3^{-r}}{3^r-1}, \text{ where } p = \frac{2r\log_2 3}{3^r-1} \text{ bpp is the relative message length.} \tag{5}$$

Because the adaptive ternary ±1 embedding is a special case of matrix embedding with Hamming codes with $r = 1$, from (5), we can write

$$MSE_{\pm 1T}\left(p\right) = \frac{1}{12} + \frac{2}{3}\frac{p}{\log_2 3}\,, \tag{6}$$

The embedding distortion for PQ embedding is

$$MSE_{PQ}(p) = 2\int_0^{p/2} x^2 dx + \int_{1/2-p/2}^{1/2+p/2} \frac{x^2}{2} + \frac{(1-x)^2}{2} dx = \frac{1}{12} + \frac{p^2}{4}\,. \tag{7}$$

Figure 5 shows the plot of MSE as a function of the relative payload p for all three methods. As expected, the adaptive embedding has the largest distortion while PQ has the smallest distortion. Because matrix embedding imposes a limit on the possible length of the message that can be embedded, we only compared the embedding methods for two payloads of 0.3658 and 0.1585 bpp, which correspond to $r = 3$ and $r = 4$ (see Table 3). We did not run the test for the higher payload ($r = 2$) because PQ embedding should not be used at such high payload. For the lower available payloads ($r > 4$), the steganalyzer does not perform too well and thus no meaningful conclusions can be drawn.

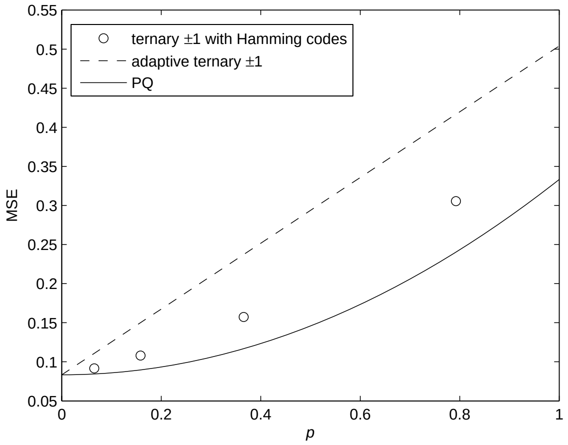
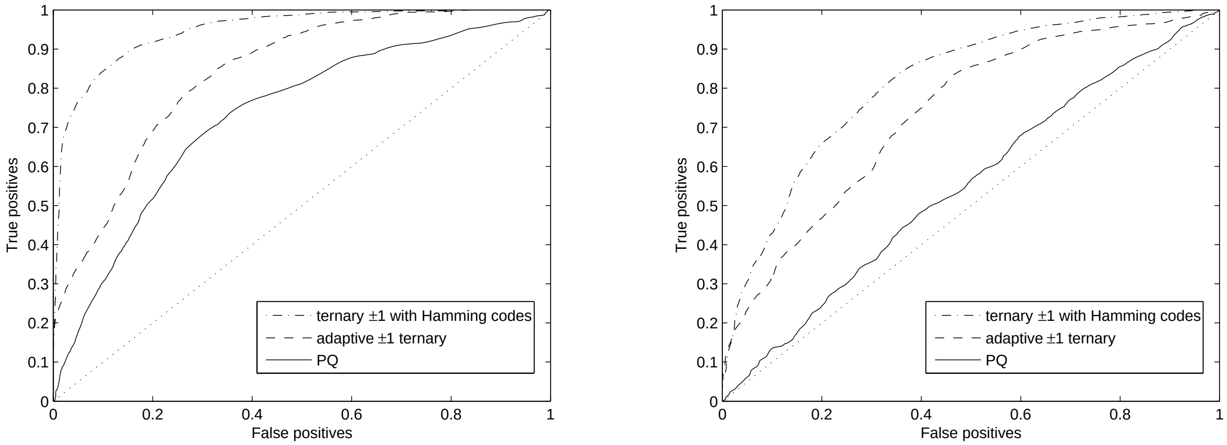


Figure 5. Embedding distortion as MSE for three embedding methods.

4.5 Experiment

We trained a WAM classifier for all three embedding methods on a subset of the CAMERA_RAW database consisting of 1520 raw 16-bit per channel camera images (because the selected version of PQ requires the 16-bit images, for a meaningful comparison we had to constrain all three methods to images that were originally 16-bit per channel; for their list see Appendix). Before applying both ±1 embedding methods, the 16-bit per channel cover images were converted to 8 bits per channel. The PQ embedding was applied directly to the 16-bit images.

The experiment produced ROC curves in Figure 6. Referring to this figure, Perturbed Quantization is the least detectable of all three methods but we need to realize that PQ needs substantial amount of side information – the 16-bit image, which the other two methods do not require. Adaptive ±1 embedding with ternary coding placed the second despite the fact that it imposes the largest embedding distortion. This confirms the well-known fact that embedding distortion is a poor indicator of steganographic security. This experiment also supports the principle that adaptive steganography generally improves steganographic security. This result can be intuitively expected because it is more difficult to distinguish the stego signal from content in textured areas. On the other hand, adaptive methods are a double-edged sword because the stego image provides information to an attacker about the placement of embedding changes. The attacker may be able to use areas that were most likely not used for embedding for calibration of certain statistics and compare them to areas likely used for embedding. Thus, adaptive embedding rules open up space for targeted attacks. Indeed, the leakage of information provided by content-adaptive selection channel has been used in the past to construct successful attacks.[22]



a) **b)**

Figure 6. ROC for all three embedding methods and relative payload a) *p* = 0. 3658, b) *p* = 0.1585.

Finally, we note that random ±1 method with ternary matrix embedding has a potential to be further improved with better codes (see the theoretical lower bound in the columns for $r = 3, 4$ in Table 3), which may change the conclusions reached in this section.

5. CONCLUSION

In this paper, we built an improved blind WAM steganalyzer for detection of embedding in raster image formats and compared its performance to four previously proposed blind steganalyzers. The comparison, which was performed on the same image databases under the same testing conditions, indicates that the proposed blind steganalysis offers improved performance over a wide class of images. The new WAM steganalyzer uses a set of 27 features (or 3×27 features for color images) calculated as absolute moments of noise residual in the wavelet domain.

We used the WAM classifier to study some fundamental issues in steganography. First, we analyzed how much improvement one can expect in detection if some a priori side-information is available about the cover image source, such as the model of the camera used to take the cover images. We found that training the classifier on an image database narrowed to images taken by a different camera of the same model significantly improved the detection results when compared to a classifier trained on a mixture of 22 cameras. As expected, training on images coming from the exact same camera as the camera used for test images produced the best results, although the difference when training on images from the same camera model (but not the same camera) was not very big.

It is likely that any side information about the image under investigation should be exploited in the classifier training phase. This may include information about the image quality (JPEG lossy compression), image processing (re-sampling, double compression), image size, and image source. We leave the investigation of such side informed steganalysis for our future research.

The second issue we investigated was careful comparison of steganographic security of three advanced embedding paradigms in the spatial domain – the random ±1 embedding combined with ternary matrix embedding (to decrease the number of embedding changes), locally adaptive ternary ±1 embedding, and perturbed quantization while converting a 16-bit per channel image to an 8-bit grayscale image. We used the WAM classifier as an oracle to decide which technique is the most secure. The tests were done using a linear classifier on 1520 16-bit raw cover images. The verdict was that the perturbed quantization was the least detectable, followed by adaptive ternary ±1 embedding, and random ternary ±1 embedding with matrix embedding. The adaptive embedding was better than the random ±1 embedding despite the fact that it imposes the biggest distortion measured as MSE. This confirms the intuition that adaptive embedding rules improve steganographic security.

Our future work will focus on other open problems, such as how to utilize the inter channel correlation in an optimal way for blind steganalysis or the little researched problem of improving steganographic security by trading the number of changes for their amplitude.

ACKNOWLEDGMENT

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under the research grants number F30602-02-2-0093 and FA8750-04-1-0112. 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. T. Holotyak, J. Fridrich, S. Voloshynovskiy, "Blind Statistical Steganalysis of Additive Steganography Using Wavelet Higher Order Statistics," *9th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security*, LNCS vol. 3677, Springer-Verlag, Berlin, pp. 273–274, 2005.
   Full paper available at http://www.ws.binghamton.edu/fridrich/publications.html
2. G. J. Simmons, "The prisoners' problem and the subliminal channel," in Advances in Cryptology: *Proceedings of Crypto 83* (D. Chaum, ed.), Plenum Press, pp. 51–67, 1984.
3. T. Pevny and J. Fridrich, "Towards Multi-Class Blind Steganalyzer for JPEG Images," International Workshop on Digital Watermarking, LNCS vol. 3710, Springer-Verlag, pp. 39–53, 2005.
4. M. Kharrazi, H.T. Sencar, N.D. Memon: "Benchmarking Steganographic and Steganalytic Techniques," *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, vol. 5681, San Jose, CA, January 16–20, pp. 252–263, 2005.
5. I. Avcıbaş, N. Memon, and B. Sankur, "Steganalysis Using Image Quality Metrics," in E. Delp et al. (eds.): *Proc. SPIE Electronic Imaging, Security and Watermarking of Multimedia Contents II*, vol. 4314, pp. 523–531, 2001.
6. H. Farid and S. Lyu: "Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines," in F.A.P. Petitcolas (ed.): *5th International Workshop on Information Hiding*, LNCS vol. 2578, Springer-Verlag, Berlin-Heidelberg, New York, pp. 340–354, 2002.
7. S. Lyu and H. Farid, "Steganalysis Using Color Wavelet Statistics and One-Class Support Vector Machines," *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI*, vol. 5306, San Jose, CA, January 19–22, pp. 35–45, 2004.
8. J. J. Harmsen and W. A. Pearlman, "Steganalysis of Additive Noise Modelable Information Hiding," in E. Delp at al. (eds.): *Proc. SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents V*, pp. 131–142, 2003.
9. Andrew D. Ker, "Steganalysis of LSB Matching in Grayscale Images." *IEEE Signal Processing Letters*, vol. **12**(6), pp. 441–444, 2005.
10. G. Xuan, Y.Q. Shi, J. Gao, D. Zou, C. Yang, Z. Zhang, P. Chai, C. Chen, and W. Chen, "Steganalysis Based on Multiple Features Formed by Statistical Moments of Wavelet Characteristic Functions," *Proc. 7th Information Hiding Workshop*, Barcelona, Spain, June 6–8, 2005.
11. M. K. Mihcak, I. Kozintsev, and K. Ramchandran, "Spatially Adaptive Statistical Modeling of Wavelet Image Coefficients and its Application to Denoising," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 6, Phoenix, AZ, pp. 3253–3256, 1999.
12. D. Donoho, M. R. Duncan, and X. Huo, WaveLab Version 802, http://www-stat.stanford.edu/~wavelab/
13. A. Ker, "General Framework for Structural Steganalysis of LSB Replacement, in M. Barni et al. (eds.): *7th International Workshop on Information Hiding*, LNCS vol. 3727, Springer-Verlag, Berlin, pp. 296–311, 2005.

14. I. Avcıbaş, M. Kharrazib, N. Memon, B. Sankur, "Image Steganalysis with Binary Similarity Measures," EURASIP JASP, No. 17, pp. 2749–2757, 2005.
15. R. Chandramouli, M. Kharrazi, N. D. Memon, "Image Steganography and Steganalysis: Concepts and Practice," in T. Kalker et al. (eds.): Digital Watermarking. International Workshop. LNCS vol. 2939, Springer-Verlag, Berlin, pp. 35–49, 2003.
16. R. Crandall, "Some Notes on Steganography," posted on Steganography Mailing List, 1998. http://os.inf.tu-dresden.de/~westfeld/crandall.pdf
17. M. van Dijk and F. Willems, "Embedding Information in Grayscale Images," in *Proc. of the 22nd Symposium on Information and Communication Theory in the Benelux*, Enschede, The Netherlands, May 15–16, pp. 147–154, 2001.
18. J. Fridrich, M. Goljan, P. Lisoněk, and D. Soukal, "On Embedding Efficiency in Steganography," submitted to *8th International Workshop on Information Hiding*, Washington, D.C., July 10–12, 2006.
19. J. Fridrich, M. Goljan, P. Lisoněk, and D. Soukal, "Writing on Wet Paper," *IEEE Trans. on Sig. Proc.*, Special Issue on Media Security, Eds. T. Kalker and P. Moulin, vol. **53**, pp. 3923–3935, October 2005.
20. G. D. Cohen, I. Honkala, S. Litsyn, and A. Lobstein, *Covering Codes*, vol. 54. Elsevier, North-Holland Mathematical Library, 1997.
21. J. Fridrich, M. Goljan and D. Soukal, "Perturbed Quantization Steganography with Wet Paper Codes," *Proc. ACM Multimedia Security Workshop*, Magdeburg, Germany, September 20–21, pp. 4–15, 2004.
22. A. Westfeld and R. Böhme, "Exploiting Preserved Statistics for Steganalysis," in J. Fridrich (ed.): *6th International Workshop on Information Hiding*, LNCS vol. 3200, Springer-Verlag Berlin Heidelberg, pp. 82–96, 2005.

6. APPENDIX A

List of digital cameras used to produce the CAMERA_RAW image database. The second and third columns correspond to the number of images taken with each camera and the image color depth (in bits).

Camera model	#	bits/channel
Canon EOS D30	149	16
Canon EOS D60	119	16
Canon PowerShot G3	75	16
Canon PowerShot G5	141	16
Canon PowerShot Pro90IS	73	16
Canon PowerShot S100	80	16
Canon PowerShot S50	59	16
CanPS G2	195	8
CanPS S40	197	8
Kodak DC290	195	8
Nikon CoolPix 5700	107	16
Nikon CoolPix 990	25	16
Nikon CoolPix SQ	32	16
Nikon D10	143	16
Nikon D100	27	16
Nikon D100	160	8
Nikon D1H	33	16
Nikon D1X	115	16
Olympus C765	300	8
Sony CyberShot DSC F505V	46	16
Sony CyberShot DSC F707	112	16
Sony CyberShot DSC S75	117	16
Sony CyberShot DSC S85	67	16
Total	2567	