From Blind to Quantitative Steganalysis

Tomáš Pevný, Jessica Fridrich, *Member, IEEE*, and Andrew D. Ker, *Member, IEEE*

***Abstract*—A quantitative steganalyzer is an estimator of the number of embedding changes introduced by a specific embedding operation. Since for most algorithms the number of embedding changes correlates with the message length, quantitative steganalyzers are important forensic tools. In this paper, a general method for constructing quantitative steganalyzers from features used in blind detectors is proposed. The core of the method is a support vector regression, which is used to learn the mapping between a feature vector extracted from the investigated object and the embedding change rate. To demonstrate the generality of the proposed approach, quantitative steganalyzers are constructed for a variety of steganographic algorithms in both JPEG transform and spatial domains. The estimation accuracy is investigated in detail and compares favorably with state-of-the-art quantitative steganalyzers.**

***Index Terms*—Blind steganalysis, message length estimation, quantitative steganalysis, regression.**

I. Introduction

WHILE the objective of steganalysis is to detect the mere presence of hidden messages in a communication, in practice the steganalyst will clearly want to achieve more. For example, an estimate of the number of modifications introduced by steganography provides information about the length of the embedded secret message. Steganalyzers designed to estimate the relative number of embedding changes (the change rate) are called quantitative. Their design typically requires full knowledge of the embedding algorithm. The steganalyzer is built using clever tricks and heuristic principles combined with experience and intuition. Because of the lack of a general approach, the vast majority of current quantitative steganalyzers attack only least significant bit (LSB) embedding schemes (see, e.g., [1]–[8]). Although there exist a few quantitative steganalyzers for other embedding operations, such as LSB matching (also called ±1 embedding) in the spatial domain [9], the embedding operation of F5 [1], and the model-based steganography [10], quantitative steganalyzers are missing for most steganographic algorithms. This is rather surprising as essentially all of these algorithms can be reliably detected using blind steganalyzers by representing images in an appropriate feature space [11]–[17].

This paper presents a general methodology for designing quantitative steganalyzers that does not depend on a detailed knowledge of the embedding algorithm. Instead, all that is required is a set of stego objects embedded with a range of relative payloads and a set of steganographic features *changing predictably* with the payload. If the latter requirement is fulfilled, the separation boundary between cover and stego images is a deterministic function of the change rate, and we build a change-rate estimator by mathematically describing the relationship between the feature vector and its position in the feature space. Regression tools are used to learn the relationship between the features' location and the number of embedding changes. In this work, we explore ordinary linear least square regression and a kernelized variation called support vector regression, essentially a data-driven method similar in spirit to a support vector machine.

The most important advantage of this approach to quantitative steganalysis over previous art is that it may be possible to design a quantitative steganalyzer even without any knowledge of the embedding mechanism. In fact, all that is required is the access to a database of images embedded with a range of known payloads. These images could be generated if the steganalyst has an access to the embedding algorithm but not necessarily to its inner workings (e.g., if only an executable file is available). A second requirement is that there must exist a feature set sensitive to the embedding, an assumption that is satisfied for almost all currently known steganographic schemes for digital images. The accuracy of the resulting quantitative steganalyzer depends on the sensitivity of the features to the embedding changes.

Our previous conference contribution on this topic [18] dealt with a small set of steganographic algorithms for JPEG images. In this paper, we present a more comprehensive evaluation of the presented methodology by constructing quantitative steganalyzers for algorithms hiding in both JPEG and spatial domains and for different feature sets [14], [15]. We also investigate for both domains the errors due to image content and message placement within the image (the so-called between- and within-image errors).

The paper is organized as follows. Section II presents the general methodology for constructing quantitative steganalyzers from features. The methodology is evaluated experimentally in Sections III and IV, where we report the accuracy of message-length estimators for eight steganographic schemes for

Manuscript received January 13, 2011; revised September 13, 2011; accepted September 28, 2011. Date of publication November 15, 2011; date of current version March 08, 2012. The work of T. Pevný was supported by the Air Force Office of Scientific Research under the Research Grant FA8655-11-1-3035 and by Czech Ministry of Education Grant ME10051: AMVIS-AnomalyNET and Grant MEB111008. The work of J. Fridrich was supported by the Air Force Office of Scientific Research under the Research Grant FA9550-08-1-0084 and Research Grant FA9550-09-1-0147. 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 AFOSR or the U.S. Government. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Bangalore S. Manjunath.

T. Pevný is with the Agent Technology Center, Czech Technical University, 121 35 Prague 2, Czech Republic (e-mail: pevnak@gmail.com).

J. Fridrich is with the Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY 13902 USA (e-mail: fridrich@binghamton.edu).

A. D. Ker is with Oxford University Computing Laboratory, Oxford University, Oxford, OX1 3QD, U.K. (e-mail:adk@comlab.ox.ac.uk).

Digital Object Identifier 10.1109/TIFS.2011.2175918

1556-6013/$26.00 © 2011 IEEE

JPEG images and for LSB steganography in the spatial domain. Section V contains detailed analysis of the estimator error for Jsteg, nsF5, and LSB matching, decomposing it into the within-image and between-image components. In Section VI, the estimation accuracy is compared with state-of-the-art quantitative steganalyzers for Jsteg, LSB matching, and LSB replacement. The paper is concluded in Section VII.

II. Approach

Before explaining the general approach to the construction of quantitative steganalyzers, we would like to stress that it is only *changes* to the cover which can ever be detected, and so any quantitative steganalyzer necessarily estimates the *number of embedding changes* rather than the message length. To obtain an estimate of the message length, one may have to take into account the effect of matrix embedding [19], [20] and source coding (data compression applied to the message prior to embedding) incorporated in the embedding algorithm. Although we explain the methodology on the example of digital images, it can be readily applied to other digital media objects, such as audio or video files.

The process of building a quantitative steganalyzer starts with extracting steganographic features from an image. Formally, this is captured with a mapping $\mathbf{f} : \mathcal{C} \mapsto \mathbb{R}^d$ from the space of all images $\mathcal{C}$ to a d-dimensional Euclidean feature space. The map $\mathbf{f}$ is usually scalable so that it can be applied to images of arbitrary size. Everywhere in this paper, we will work with $\mathcal{C}$ being the set of all gray-scale images in either the raster or JPEG format. Our quantitative steganalyzer will be in the form of a function $\psi : \mathbb{R}^d \mapsto [0,1]$ revealing the relationship between the features' location and the *change rate*. By change rate, we denote the number of embedding modifications divided by the number of cover elements. Depending on the type of the cover, its elements could be pixels (in a gray-scale raster image) or nonzero quantized DCT coefficients (in a JPEG file).

To formalize the problem, let $\mathbf{X} = \{(\mathbf{x}_i, y_i) \mid i \in \{1,\dots,l\}\}$ denote l samples consisting of feature vectors $\mathbf{x}_i = \mathbf{f}(c_i) \in \mathbb{R}^d$ computed from l images c_i embedded with relative number of embedding changes $y_i \in [0,1]$. Our goal is to construct a quantitative steganalyzer by finding a function $\hat{\psi} : \mathbb{R}^d \mapsto [0,1]$ that minimizes the error on $\mathbf{X}$, or

$$\hat{\psi} = \arg\min_{\psi\in\mathcal{F}} \frac{1}{l}\sum_{i=1}^{l} e\left(\psi(\mathbf{x}_i), y_i\right) \tag{1}$$

where $e : \mathbb{R}\times\mathbb{R} \mapsto \mathbb{R}_0^+$ is an error function (also called a loss function) and $\mathcal{F}$ is an appropriately chosen class of functions $\psi : \mathbb{R}^d \mapsto [0,1]$.

The error function $e(\hat{y}, y)$ and the class of functions $\mathcal{F}$ influence the accuracy of the resulting estimator $\hat{\psi}$. It is possible that a desired accuracy is not achieved for a given feature set simply because of a wrong combination of e and $\mathcal{F}$. In this work, we consider two ways to solve the regression problem (1): ordinary linear least-square regression (OLLSR) and support vector regression (SVR) with a Gaussian kernel.

A. Linear Least-Squares Regression

In linear regression, the class $\mathcal{F}$ consists of linear functionals $\psi(\mathbf{x}_i) = \mathbf{a}\cdot\mathbf{x}_i + b$ for $\mathbf{a} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, and it typically uses the square loss function $e(\hat{y}, y) = (\hat{y} - y)^2$. The regression problem (1) can then be solved directly using linear operations. This OLLSR is very simple, intuitive, and has a low computational complexity, but it cannot find nonlinear dependencies between the features and the target variable. We assume that the reader is already familiar with OLLSR.

B. Support Vector Regression

Support vector regression solves the regression problem by a technique analogous to the support vector machine (SVM) [21] approach to classification. In the simplest version, the class $\mathcal{F}$ still consists of linear functionals $\psi(\mathbf{x}_i) = \mathbf{a}\cdot\mathbf{x}_i + b$, but the loss function combines an ϵ-insensitive error with the norm of $\mathbf{a}$

$$e_\epsilon(\hat{y}, y) = \begin{cases} \frac{1}{2}\|\mathbf{a}\|^2 + C(|\hat{y}-y|-\epsilon), & \text{if } |\hat{y}-y| > \epsilon \\ \frac{1}{2}\|\mathbf{a}\|^2, & \text{otherwise.} \end{cases}$$

The first term is a measure of complexity, with less complex functionals given preference to prevent overfitting. The second is a measure of loss which ignores the error of near-correct estimates. The latter causes the optimization problem (1) to become sparse and only a few of the training instances become the *support vectors* which influence the outcome: the result is better generalization and faster estimation. Furthermore, we will see in Section V that estimation is subject to a few extreme outliers; replacing a square loss function with one which is linear (above the threshold ϵ) may help counterbalance this. The parameter C controls how the two terms are balanced.

In this work we will combine the SVR technique with the "kernel trick" [22] which replaces the usual scalar product $x_i^T x_j$ with, in our case, the Gaussian kernel $k(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|_2^2)$. Kernelized SVR can reveal more complicated nonlinear dependencies at the cost of increased computational complexity. For more details we refer to the tutorial [21].

There are three hyper-parameters that need to be set prior to training [solving(1)]: the penalization parameter C, the width of the Gaussian kernel γ, and the insensitivity of the loss function ϵ. The choice of the hyper-parameters has a significant influence on the ability of the estimator to generalize (to accurately estimate the change rate on samples not in the training set). Since there is no optimal method to set them, in experiments presented in Sections III and IV, we used a search on a predefined set of triplets (C, γ, ϵ), on which the generalization was estimated by five-fold cross-validation over the training set. To decrease the computational complexity, the search used two phases.

In the first phase, the generalization was measured on the following grid:

$$\begin{aligned}(C,\gamma,\epsilon) \in \mathcal{S}_1 = \{&\left(10^i, 2^j, 0.005\cdot k\right) \mid \\ &i \in \{-3,\dots,4\},\ j \in \{-11,\dots,-5\},\ k \in \{1,2,3,4\}.\end{aligned}$$

The triplet $(C_1, \gamma_1, \epsilon_1)$ with the least error on $\mathcal{S}_1$ was used to seed the search in the second phase, which was performed on the grid

$$\mathcal{S}_2 = \left\{\left(10^i, 2^j, 0.005\cdot k\right) \mid i,j \in \mathbb{Z}, k \in \mathbb{N}\right\}.$$

In each iteration, the point with the least generalization error was checked to see whether it lay on the grid boundary. If so, the error was estimated on the neighboring points from the set $\mathcal{S}_2$ and the check was repeated. If not, the search was stopped and the triplet $(C_2, \gamma_2, \epsilon_2)$ with the least estimated generalization error was used for training.

The two-phase search is used to ensure that the point with the least estimated generalization error is not the boundary point of the explored set. Under the assumption that the generalization error surface is convex, which generally holds for the vast majority of practical problems, this algorithm keeps the number of explored points relatively low, while returning a suitable set of hyper-parameters.

III. Steganalysis in Transform Domain

In this section, the proposed method is evaluated by constructing quantitative steganalyzers for eight steganographic algorithms for JPEG images: JP Hide&Seek (JPHS) [23], Jsteg [24], Model Based Steganography without deblocking (MBS1) [25], MMx [26], F5 with shrinkage removed by wet paper codes with matrix embedding turned off (nsF5) [27], OutGuess [28], Perturbed Quantization [29] (PQ), and Steghide [30]. The chosen steganographic algorithms employ a variety of different embedding mechanisms. PQ and MMx use side information in the form of the uncompressed image during embedding.

A. Setup of Experiments

The image sets for experiments reported here, and in the next section, were all derived from a mother database called the CAMERA database. This database contains approximately 9200 images taken by 23 different digital cameras in their native resolution in raw format (no in-camera JPEG compression). The size of the images ranges from one to six megapixels.

For the purpose of steganalyzing JPEG images in this section, all CAMERA images were first converted to gray-scale and then single-compressed with JPEG quality factor 80 (the MMx algorithm requires the uncompressed gray-scale image). The only exception were images used in the experiments with Perturbed Quantization [29] (PQ) where the cover images were double-compressed with primary quality factor 85 and secondary quality factor 70.[1] These two quality factors were chosen in order to maximize the capacity of PQ.

All images were divided into two sets of equal size (approximately 4600 images per set). One set was used exclusively for training the estimator, while the other set was used for evaluating its accuracy. The stego images were created by embedding a random message of (uniform) random length between 0 and $m_{\max}$, where $m_{\max}$ is the maximum embeddable payload for each combination of the embedding algorithm and the cover image.

As a feature set $\mathbf{f}$, we used the 274-PEV feature set from [14]. Since the features are sensitive not only to the payload, but also to the image size (i.e., they are not properly normalized), we have augmented the features with the number of nonzero DCT coefficients n_0. The additional 275th feature improves the accuracy of the steganalyzer, helping it to adjust to different values of features on images of different size.

All 275-PEV features were normalized to have zero mean and unit variance. The normalization coefficients were always calculated on the training set of cover images.

B. Experimental Results

Two quantitative steganalyzers trained on the same training set were created for each steganographic algorithm: one created using OLLSR, the other one was constructed using kernelized SVR as outlined earlier.

Fig. 1 shows a scatter plot of the change rates estimated by SVR steganalyzers versus the true values. Because the error distribution of quantitative steganalyzers often exhibits heavy tails [31] (and Section V confirms this observation for our steganalyzers as well), the performance is evaluated using robust statistics. Table I displays the estimator bias, defined as the mean observed error, and two measures of estimator dispersion: interquartile range of observed error (denoted IQR) and mean absolute observed error (denoted μAE). The most robust measure is IQR, which is completely insensitive to outlier estimates; μAE retains some sensitivity to outliers but does not suffer from the same leverage as, for example, sample variance.

We can see most quantitative steganalyzers for transform domain steganography have good performance, with IQR of the estimation of relative change-rate of the order of 10^{-3} and an order of magnitude lower bias (recall that the quantity estimated is the change rate, which is on a scale from 0 to 1). All estimators show a few outlier values. Despite the fact that the attacked steganographic schemes employ very different embedding operations and strategies, the steganalyzers provide rather accurate estimates. However, the estimator for the PQ algorithm is accurate only for small payloads (less than 0.2 bpac). For larger payloads, the estimator basically fails despite the fact that a binary classifier for the presence or absence of stego data, based on the same feature set, works quite well for all payloads [27].

We carefully investigated this phenomenon: it arises because the cluster of stego-image feature vectors seriously deforms with increasing payload rather than being moved rigidly in one direction by a vector whose length depends on the change rate. This phenomenon makes it difficult for the estimator to learn the relationship between cover and stego features as a function of the change rate. We confirmed this by measuring the average distance between the clusters of cover and stego images as well as the distances between the cover image and its corresponding stego image in the feature space. While each image appears to have been shifted by a vector whose length monotonically increases with the change rate, the difference between the means of cover and stego features stops increasing at around 0.2 bpac.

The OLLSR estimator has a slightly higher dispersion, but exhibits a lower bias for several embedding algorithms, than the SVR estimator. The fact that the OLLSR accuracy is of the same order as the corresponding SVR estimator suggests that the features shift almost linearly with the number of embedding changes. Despite the slightly higher dispersion, the OLLSR regression offers an attractive choice, because of its low computational complexity: training the SVR, which includes the search

[1] PQ embeds messages while recompressing the cover JPEG image with a different quality factor.

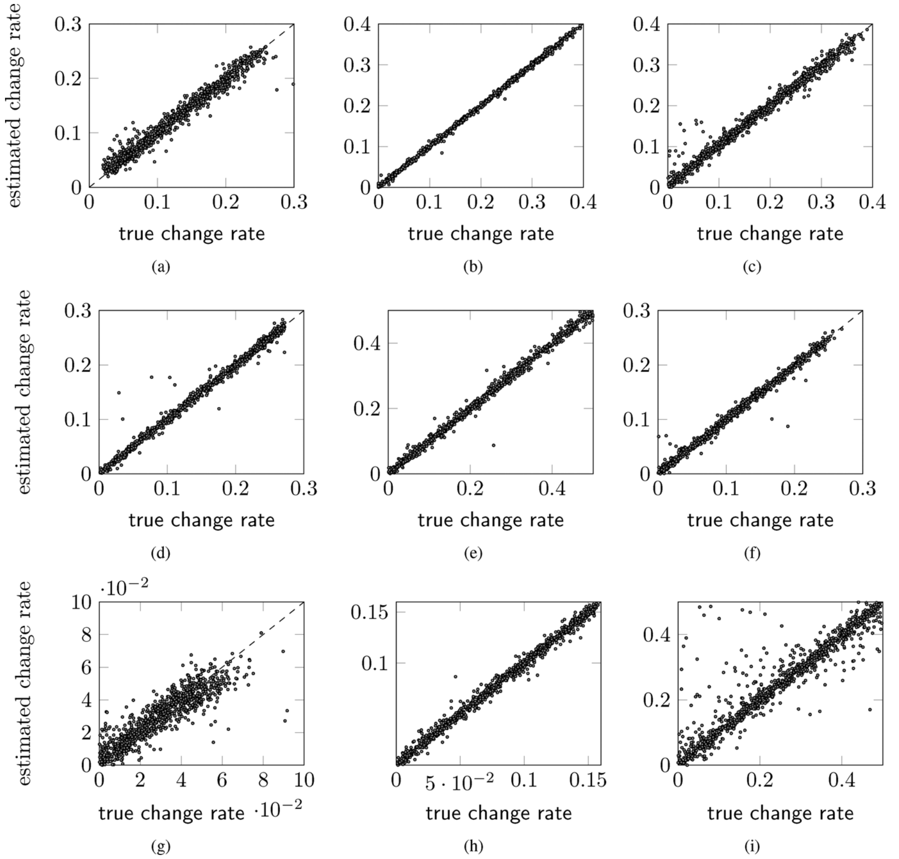

Fig. 1. Scatter plot showing the estimated change rate with respect to the true change rate for eight JPEG domain embedding algorithms, and one spatial domain embedding algorithm. All estimates were made on images from the testing set, using SVR with Gaussian kernel. The dashed line corresponds to perfect estimation. (a) JPHS. (b) Jsteg. (c) MBS1. (d) MMx. (e) nsF5. (f) Outguess. (g) PQ. (h) Steghide. (i) LSB matching.

TABLE I
Interquartile Range (IQR), Mean Absolute Error (μAE), and Bias, for the OLLS Estimator and SVR With Gaussian Kernel, on Eight JPEG Domain and Two Spatial Domain Steganographic Algorithms

Embedding Algorithm	Feature Set	OLLSR			SVR		
		IQR	μAE	Bias	IQR	μAE	Bias
JPHS	275-PEV	$1.58 \cdot 10^{-2}$	$1.05 \cdot 10^{-2}$	$-1.70 \cdot 10^{-4}$	$1.03 \cdot 10^{-2}$	$8.15 \cdot 10^{-3}$	$2.41 \cdot 10^{-4}$
Jsteg	275-PEV	$5.51 \cdot 10^{-3}$	$3.85 \cdot 10^{-3}$	$2.40 \cdot 10^{-4}$	$3.87 \cdot 10^{-3}$	$2.91 \cdot 10^{-3}$	$2.59 \cdot 10^{-4}$
nsF5	275-PEV	$1.68 \cdot 10^{-2}$	$1.12 \cdot 10^{-2}$	$-5.29 \cdot 10^{-4}$	$9.67 \cdot 10^{-3}$	$6.97 \cdot 10^{-3}$	$-2.51 \cdot 10^{-4}$
MBS1	275-PEV	$1.81 \cdot 10^{-2}$	$1.20 \cdot 10^{-2}$	$3.86 \cdot 10^{-5}$	$1.32 \cdot 10^{-2}$	$9.20 \cdot 10^{-3}$	$-1.63 \cdot 10^{-4}$
MMx	275-PEV	$6.52 \cdot 10^{-3}$	$4.30 \cdot 10^{-3}$	$1.58 \cdot 10^{-4}$	$5.37 \cdot 10^{-3}$	$3.82 \cdot 10^{-3}$	$1.08 \cdot 10^{-4}$
Steghide	275-PEV	$5.10 \cdot 10^{-3}$	$3.51 \cdot 10^{-3}$	$1.51 \cdot 10^{-4}$	$4.10 \cdot 10^{-3}$	$2.86 \cdot 10^{-3}$	$1.80 \cdot 10^{-4}$
PQ	275-PEV	$9.96 \cdot 10^{-3}$	$7.06 \cdot 10^{-3}$	$-4.44 \cdot 10^{-4}$	$8.69 \cdot 10^{-3}$	$5.81 \cdot 10^{-3}$	$3.61 \cdot 10^{-4}$
OutGuess	275-PEV	$6.50 \cdot 10^{-3}$	$4.40 \cdot 10^{-3}$	$2.60 \cdot 10^{-4}$	$4.97 \cdot 10^{-3}$	$3.57 \cdot 10^{-3}$	$3.67 \cdot 10^{-4}$
LSBM	SPAM	$2.59 \cdot 10^{-1}$	$6.32 \cdot 10^{-2}$	$1.57 \cdot 10^{-1}$	$3.04 \cdot 10^{-2}$	$2.99 \cdot 10^{-2}$	$-1.41 \cdot 10^{-3}$
LSBR	SPAM	$2.65 \cdot 10^{-1}$	$5.85 \cdot 10^{-2}$	$1.51 \cdot 10^{-1}$	$2.90 \cdot 10^{-1}$	$2.55 \cdot 10^{-2}$	$-3.70 \cdot 10^{-4}$

for the hyper-parameters, takes about one day on a 64-bit AMD Opteron 2.4-GHz computer, but OLLSR regression on the same machine takes less than 1 min.

Using this "cookie-cutter" approach, we were able to construct quantitative steganalyzers for algorithms such as JPHS and MMx, where none previously existed in the literature.[2] Moreover, as will be explored in more detail in Section VI in the case of Jsteg, where previous quantitative steganalyzers do exist, the estimator built from the 275-PEV feature set comfortably outperforms them.

IV. Steganalysis in Spatial Domain

In Section III, the newly proposed methodology for constructing quantitative steganalyzers was demonstrated on algorithms that embed in JPEG images. To further prove its utility, in this section we apply the same approach to algorithms that embed in the spatial domain. We will concentrate on LSB matching (LSBM, also called ±1 embedding), the steganographic method that hides message bits in LSBs of pixels by

[2] A quantitative steganalyzer for MBS1 has been constructed in [10]. The construction was essentially the same as the one used in this paper. The only difference is in the used regression algorithm.

randomly modifying their values by ±1. Despite its simplicity, LSB matching has proved to be difficult to reliably detect even at relatively large payloads of 0.1 bits per pixel (bpp). Although there exist some feature-based steganalyzers detecting LSB matching [15], [17], [32], [33], to the best of our knowledge there is only one quantitative steganalyzer, reported in [9], and it has a rather poor accuracy.

As before, we construct the quantitative steganalyzer by means of the SVR following the method described in Section II. As a feature set $\mathbf{f}$, we used the "second-order SPAM features" [15], which have dimension 686, augmented by the number of pixels in the image as an additional 687th feature. We chose this feature set due to its popularity and ability to detect LSB matching.

The images for our experiments were taken from the CAMERA database and converted to gray-sale by the convert program from ImageMagick package [34]. Using LSB matching, a random message of random length between 0 and $m_{\max}$ ($m_{\max}$ is the number of pixels in the image) was embedded in each image. Half of the images were used to train the estimator, with the other half used to evaluate its accuracy.

Fig. 1(i) shows the estimated change rates against true change rates, and the estimator bias and dispersion appear in Table I. The estimator is less accurate than those for embedding in JPEG images. This is most likely due to the fact that the embedding changes in the spatial domain are well masked by noise already present in digital images. Because the noise component is largely suppressed in JPEG coefficients due to quantization, it is also easier to detect the pseudorandom changes made to the quantized coefficients. Previously published studies confirm the difficulty of detecting LSB matching in the spatial domain over JPEG steganography [15], [27]. Table I shows that the errors of steganalyzers in spatial domain are approximately one magnitude larger than of steganalyzers for DCT domain. Table I also shows the performance of the same estimator trained and tested on LSB replacement (LSBR) embedding, which has been shown to be substantially weaker than LSB matching [4], [5], [35]; these results show that the SPAM features are not able to make much use of the additional weaknesses in the LSB replacement embedding.

V. Detailed Error Analysis

Motivated by the presence of outlier estimates visible in Fig. 1, this section presents a breakdown of the errors in the quantitative steganalyzers for nsF5 and Jsteg (using 275-PEV features), and LSB matching (using SPAM features). We chose those two JPEG algorithms because their simple embedding mechanism allows precise control of the number of embedding changes. We are interested in the extent of the outliers and how variation in cover, payload, and their random correlations, contribute to estimation error.

In general, the payload size estimation error can be decomposed into three parts, as first described in [31] and extended in [36]. When a payload is embedded, because the number of embedding changes depends on random correlations with the cover, the changes do not indicate exactly the size of the payload. In our experiments, we have eliminated this deviation by working directly with the number of embedding changes. This error, however, may have a nonnegligible effect when the estimator is applied to genuine stego images, and we call it *change-rate uncertainty* (CRU). The remaining error can be partly attributed to random placement of the payload within the cover, the so-called *within-image error* (WIE), and the rest to the properties of the cover itself, called the *between-image error* (BIE). These errors are not independent, but can be approximately separated and compared by repeatedly embedding different payloads in each cover.

We selected six embedding change rates, $\beta \in \{0, 0.025, 0.05, 0.125, 0.25, 0.375\}$, and embedded 200 random payloads into each of the approximately 4600 images in the training set, using Jsteg, nsF5, and LSB matching. We term each combination of the embedding algorithm, change rate, and cover image, a *cell*, so that each cell contains estimates of 200 equally sized but differently located payloads (except for cells with no payload, for which there is only one possible object per cover).

First, we consider the shape of the within- and between-image errors: picking a single cell of the Jsteg steganalyzer, we display a log-log empirical cumulative distribution function (cdf) plot for the 200 estimates in Fig. 2. The data has been normalized to zero mean, and the Gaussian fit is selected to match the sample variance: it appears to be excellent, and we see similar results across all steganalyzers, images, and embedding rates. A summary of these fits is found in the first columns of Table II: we computed Shapiro-Wilk tests [37] for normality in every cell, and display the proportion of cells with p values above 0.1. If the cells are truly Gaussian, we would expect that 90% of cells would pass this test; here, any deviation from normality is small enough to be undetectable with 200 samples per cell.

The cell means describe the between-image error, and we plot log-log empirical cdfs for one particular embedding rate in Fig. 2. These data are clearly not Gaussian, but there is a good fit with the Student t-distribution. These results accord closely with what was observed for heuristic LSB replacement estimators in [36] and [31] (it is surprising that quantitative steganalysis of JPEG embedding via SVR displays the same characteristics as structural steganalysis of spatial-domain embedding, given that their modes of operation are so different). An important consequence of this observation is that it is unsound to measure estimator variance, standard deviation, or mean square error: the true estimator variance may be infinite, or even if finite the sample statistic will converge only very slowly to the true value.

Finally, we compare the magnitudes of the within- and between-image errors, also including the theoretical predictions for embedding change rate uncertainty which is given by a simple Binomial distribution whose dispersion depends on the number of embedding locations. For this analysis, bias is discounted. Because of the heavy tails in the between-image error, we use interquartile range (IQR) as a highly robust measure of spread. For six embedding change rates, the IQRs of these three error factors are displayed in Table II. Because the CRU and WIE vary a small amount between covers, the table displays the average values for these IQRs.

The magnitude of the CRU is generally negligible. For BIE and WIE, the behavior of the JPEG steganalyzers differs from

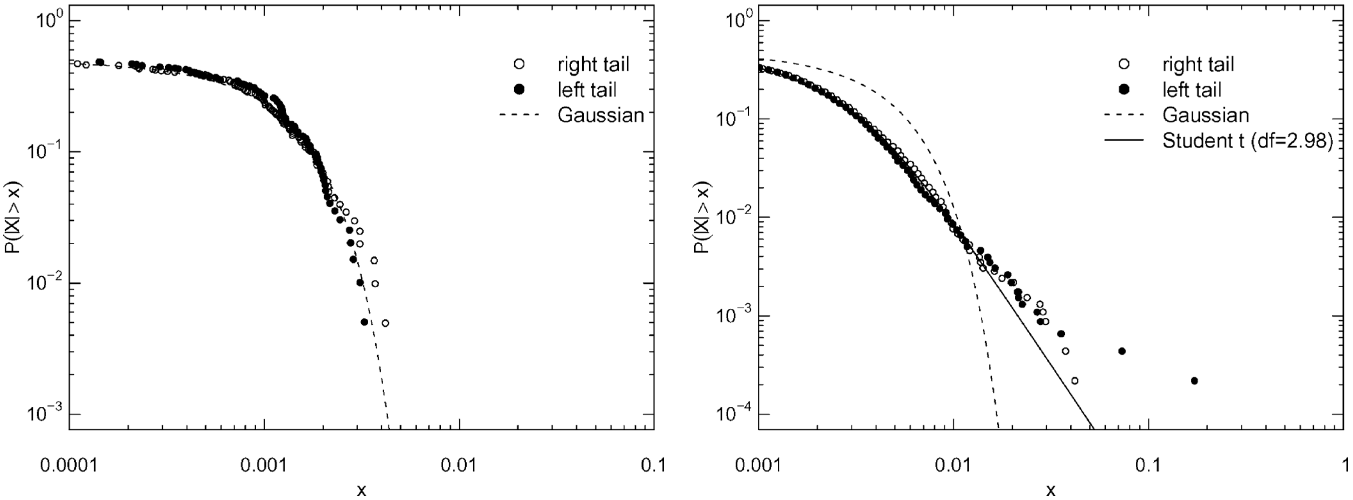

Fig. 2. Log-log tail plots of empirical distributions of (left) within- and (right) between-image errors, for the Jsteg steganalyzer. Gaussian and Student-t fits are shown.

TABLE II
PROPORTION OF CELLS PASSING A SHAPIRO-WILK (S-W) TEST FOR NORMALITY OF WITHIN-IMAGE ERROR, AT 10% SIGNIFICANCE; COMPARISON OF MAGNITUDES OF BETWEEN-IMAGE ERROR (BIE), WITHIN-IMAGE ERROR (WIE), AND CHANGE RATE UNCERTAINTY (CRU), MEASURED BY INTER-QUARTILE RANGE, FOR SIX EMBEDDING CHANGE RATES (β)

	Jsteg/PEV-275				nsF5/PEV-275				LSBM/SPAM			
β	S-W $p > 0.1$	BIE	WIE	CRU	S-W $p > 0.1$	BIE	WIE	CRU	S-W $p > 0.1$	BIE	WIE	CRU
0.000	–	3.63	0.00	0.00	–	7.74	0.00	0.00	–	47.6	0.00	0.00
0.025	90.2%	3.23	1.52	0.28	93.9%	6.99	2.81	0.29	90.2%	43.4	6.95	0.28
0.050	89.9%	3.02	1.91	0.39	93.9%	6.79	3.52	0.41	90.1%	39.7	9.46	0.39
0.125	90.2%	2.79	2.57	0.59	93.7%	6.93	4.78	0.62	89.1%	31.0	13.5	0.59
0.250	89.8%	2.87	3.25	0.78	94.2%	8.31	6.77	0.81	90.2%	23.4	16.8	0.78
0.375	90.3%	3.69	3.56	0.87	94.2%	10.6	8.47	0.91	89.3%	25.3	18.6	0.87
		$\cdot 10^{-3}$	$\cdot 10^{-3}$	$\cdot 10^{-3}$		$\cdot 10^{-3}$	$\cdot 10^{-3}$	$\cdot 10^{-3}$		$\cdot 10^{-3}$	$\cdot 10^{-3}$	$\cdot 10^{-3}$

the spatial-domain case: for the latter, the dispersion of WIEs are not negligible, even for fairly small embedding rates. This is also in contrast to the structural steganalyzers considered in [36] and [31]. Also, the BIE for JPEG domain steganalysis remains stable or increases at larger embedding rates, whereas the opposite holds for spatial-domain estimators.

VI. Comparison With Prior Art

This section compares the SVR-based quantitative steganalyzers of Jsteg, LSB matching, and LSB replacement with their heuristic-based counterparts from the literature. Because of the lack of accurate quantitative steganalyzers, we could not make comparison with other steganographic algorithms for JPEG images.[3]

A. Jsteg

Among the multitude of methods described in [8], we selected Jpairs and Weighted Nonsteganographic Borders Attack (WB) and compared their performance with our quantitative SVR steganalyzer. According to [8], the Jpairs quantitative steganalyzer was one of the most accurate quantitative steganalyzers for Jsteg. The algorithms were compared on the approximately 4600 images in the testing set, by bias and IQR, at 21 embedding change rates from the set $\beta \in \{0, 0.025, 0.05, \ldots, 0.475, 0.5\}$.

Fig. 3 shows that the quantitative steganalyzer constructed by SVR has almost always better performance than both Jpairs and WB attacks. Moreover, its performance is more stable with respect to the change rate. Contrary to the conclusion reached in [8], we found that the WB attack was more precise than Jpairs attack; this discrepancy could be caused by us using a different database of images. Note, though, that Fig. 3 overstates the accuracy of JPairs, because the JPairs method sometimes fails to produce an estimate at all. This happens most often for large embedding rates: for $\beta = 0.375$, as many as one third of estimates fail. The SVR and WB methods never fail to produce an estimate.

B. LSB Matching

To the best of our knowledge, the only quantitative steganalyzer of LSB matching is based on maximum likelihood principle [9]. We compared the accuracy of this detector with our solution based on SVR and SPAM features. Since we did not possess the implementation of the maximum likelihood estimator, we mimicked the testing conditions published in [9, Sec.

[3] The heuristic quantitative steganalyzer of F5, presented in [1], is based on an essentially the same idea (regression). It uses a 2d-feature vector (two histogram bins) for which an analytic expression for the stego feature vector as a function of change rate can be derived. Thus, by definition, it will be less accurate, because the used 275-PEV feature vector is a superset of this 2d vector.

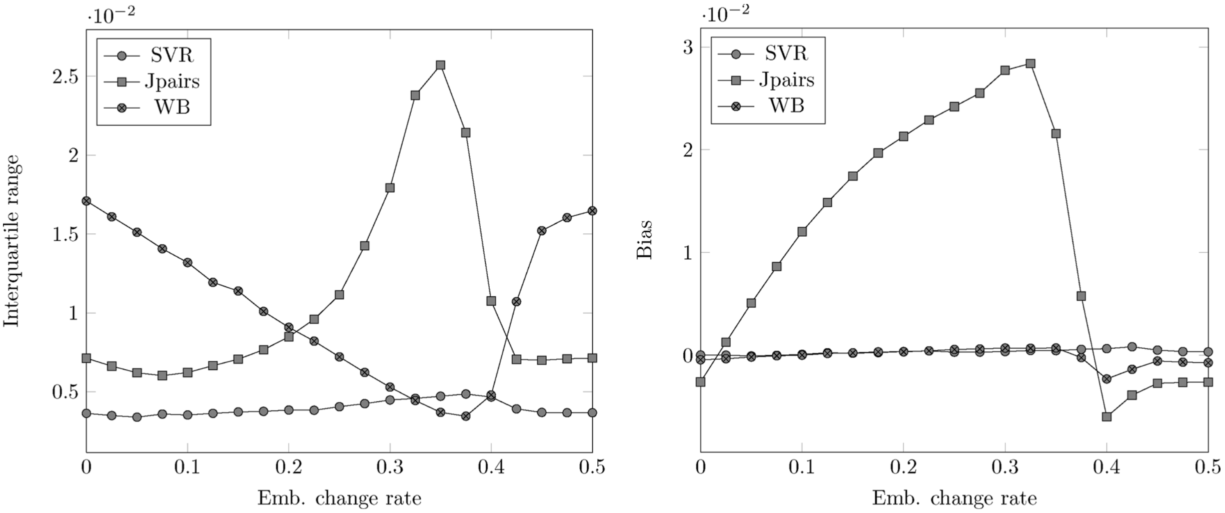

Fig. 3. Comparison with prior art: Jsteg. (Left) Interquartile range and (right) bias of Jpairs, WB, and SVR quantitative steganalyzers.

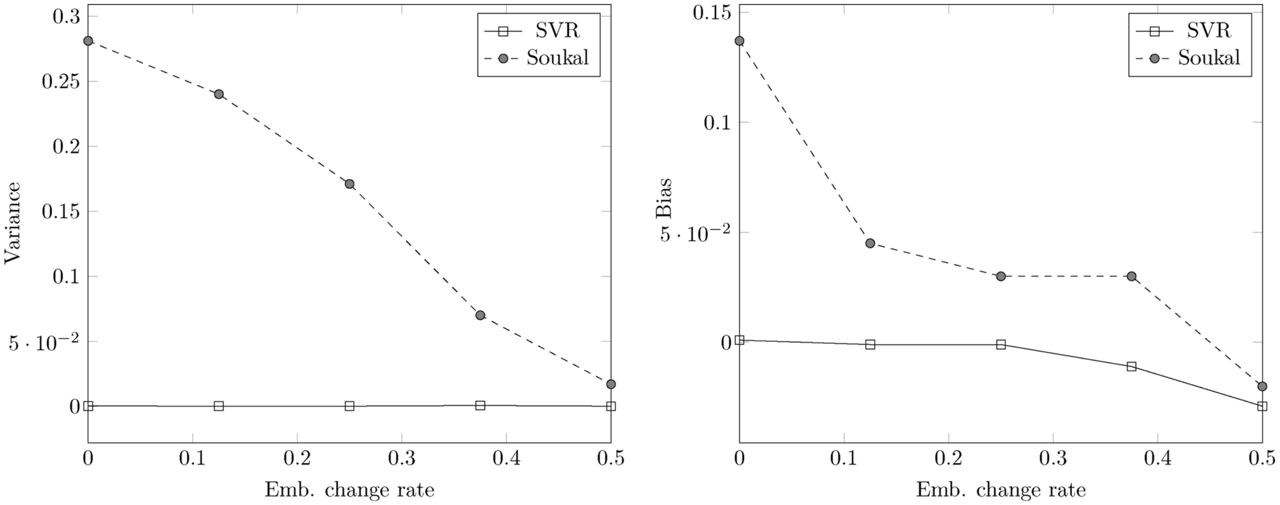

Fig. 4. Comparison with prior art: LSB matching. Bias and variance of Soukal's estimator and SVR steganalyzer at five embedding rates.

3.1]. This means that we have used the same image database (Greenspun database [38]), embedding rates, number of images for testing (180 images), and the same evaluation criteria (bias and variance[4]). We have been also careful to avoid using testing images for training the SVR estimator.

Results for five different payloads are summarized in Fig. 4 and Table III. We can see that our estimator has approximately three orders of magnitude lower variance than Soukal's, and one order of magnitude smaller bias. Here, we need to point out that images in Greenspun database used in this experiment were JPEG compressed (at quality factor 75), which significantly simplifies the steganalysis in spatial domain.

TABLE III
COMPARISON WITH PRIOR ART: LSB MATCHING. BIAS AND VARIANCE OF SOUKAL'S ESTIMATOR AND SVR STEGANALYZER AT FIVE EMBEDDING RATES

	Soukal		SVR	
β	Bias	Variance	Bias	Variance
0.000	0.137	0.281	0.001	$2.6 \cdot 10^{-4}$
0.125	0.045	0.240	-0.001	$1.4 \cdot 10^{-4}$
0.250	0.030	0.171	-0.001	$1.3 \cdot 10^{-4}$
0.375	0.030	0.070	-0.011	$6.4 \cdot 10^{-4}$
0.500	-0.020	0.017	-0.029	$1.2 \cdot 10^{-3}$

C. LSB Replacement

Unlike quantitative steganalyzers for LSB matching, state-of-the-art quantitative steganalyzers for LSB replacement are very accurate because they exploit an asymmetry in the parity structure of the embedding process. The SPAM features we have used for spatial domain steganalysis do not expose this asymmetry. We compared the accuracy of SVR-based steganalyzer of LSB matching presented in Section IV to Sample Pairs analysis (SPA) [2] and improved WS estimator [39]. According to [39], the improved WS method is the most accurate estimator for LSB replacement in the spatial domain. Fig. 5 compares bias and IQR on 21 different embedding change rates $\beta \in \{0, 0.025, 0.05, \ldots, 0.475, 0.5\}$.

[4] Although we stated in Section V that variance is not good for evaluation of the quality of the estimator, we made an exception here, because Soukal's work reports errors by bias and variance.

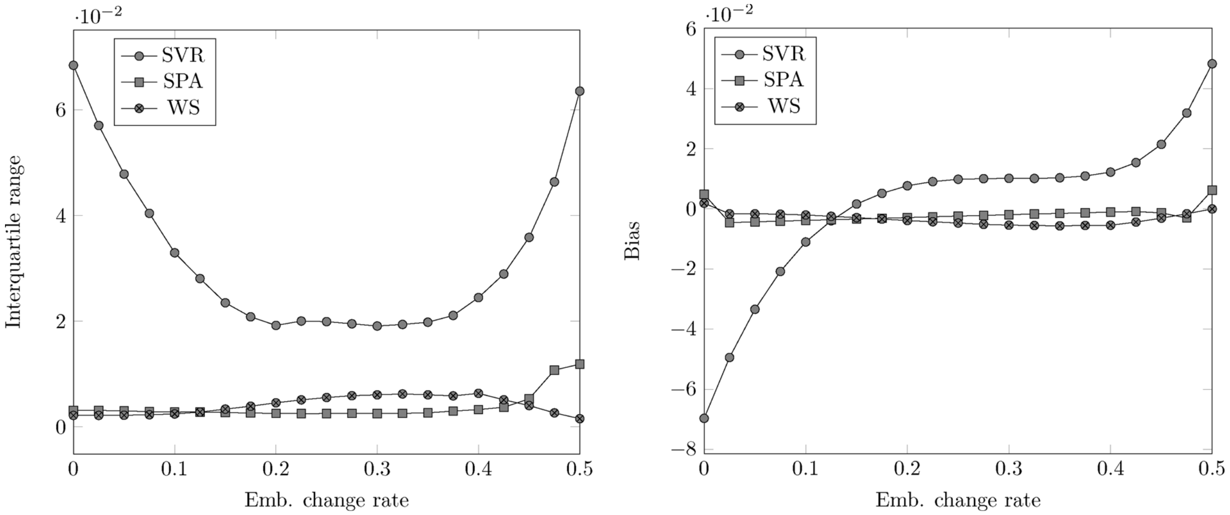

Fig. 5. Comparison with prior art: LSB replacement. (Left) Interquartile range and (right) bias of WS, Sample Pairs, and SVR quantitative steganalyzer.

As expected, Sample Pairs analysis and improved WS estimators offer an order of magnitude higher accuracy than the SVR-based estimator. We strongly believe that this is only due to the fact that SPAM features do not exploit the parity asymmetric embedding operation of LSB replacement.

VII. Conclusion

Quantitative steganalyzers were so far available only for a small set of specific embedding methods, because their design was inherently very difficult. Until now, their design was driven by heuristics and the intuition of the steganalyst, and it required a complete knowledge of the attacked steganographic scheme. A solid foundation enabling easy construction of quantitative steganalyzers for an arbitrary scheme was missing.

This paper presented a method to construct quantitative steganalyzers in a fashion similar to blind steganalyzers, based on the combination of steganographic features and a pattern recognition algorithm. The main idea is to use steganographic features and learn the relationship between the features' location and the change rate using regression.

The presented method assumes that the steganalyst possesses steganographic features that react predictably to the number of embedding changes. On the example of seven out of eight steganographic algorithms in the JPEG domain, as well as LSB matching and LSB replacement in the spatial domain, we have successfully demonstrated that the assumption holds for a wide variety of steganographic schemes: it failed for one JPEG steganographic scheme (Perturbed Quantization), which allowed only small payloads to be estimated. Using the proposed method, we were able to construct quantitative steganalyzers for stegosystems for which no quantitative attacks existed. Because of this lack of prior art, we could compare the performance only to a limited set of steganalysis methods for Jsteg, LSB matching, and LSB replacement. Similar to previously proposed quantitative steganalyzers, the within-image error of the proposed steganalyzers is significant in magnitude and the between-image error exhibits heavy tails. This means that care must be exercised to use robust measures of accuracy: variance and mean square error are unsound in such a circumstance.

We believe that the application of the presented method in steganalysis is vast. The new approach may provide a better control of the false-positive rate in targeted blind steganalysis (blind steganalyzer trained as targeted) due to the fact that the estimated change rate is a scalar quantity. Another tempting possibility is to combine existing quantitative LSB estimators, such as Triples [5], SPA [35], and WS [39], and use them together in the proposed framework to construct a new quantitative steganalyzer with higher accuracy.

Acknowledgment

The authors would like to thank A. Westfeld for providing code and support for WS and Jpairs attacks on Jsteg and R. Böhme for the original code used to produce log-log empirical cdf plots.

References

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

[2] S. Dumitrescu, X. Wu, and Z. Wang, "Detection of LSB steganography via sample pair analysis," in *Proc. 5th Int. Workshop Information Hiding*, F. A. P. Petitcolas, Ed., Noordwijkerhout, The Netherlands, Oct. 7–9, 2002, vol. 2578, Lecture Notes in Computer Science, pp. 355–372, Springer-Verlag, New York.

[3] T. Zhang and X. Ping, "A fast and effective steganalytic technique against JSteg-like algorithms," in *Proc. ACM Symp. Applied Computing*, Melbourne, FL, Mar. 9–12, 2003, pp. 307–311.

[4] J. Fridrich and M. Goljan, "On estimation of secret message length in LSB steganography in spatial domain," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI*, E. J. Delp and P. W. Wong, Eds., San Jose, CA, Jan. 19–22, 2004, vol. 5306, pp. 23–34.

[5] A. D. Ker, "A general framework for structural analysis of LSB replacement," in *Proc. 7th Int. Workshop Information Hiding*, M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, Eds., Barcelona, Spain, Jun. 6–8, 2005, vol. 3727, Lecture Notes in Computer Science, pp. 296–311, Springer-Verlag, Berlin.

[6] K. Lee and A. Westfeld, "Generalized category attack—Improving histogram-based attack on JPEG LSB embedding," in *Proc. 9th Int. Workshop Information Hiding*, T. Furon, F. Cayre, G. Doërr, and P. Bas, Eds., Saint Malo, France, Jun. 11–13, 2007, Lecture Notes in Computer Science, pp. 378–392, Springer-Verlag.

[7] R. Böhme, "Weighted stego-image steganalysis for JPEG covers," in *Proc. 10th Int. Workshop Information Hiding*, K. Solanki, Ed., Santa Barbara, CA, Jun. 19–21, 2008, Lecture Notes in Computer Science, pp. 178–194, Springer-Verlag, New York.

[8] A. Westfeld, "Generic adoption of spatial steganalysis to transformed domain," in *Proc. 10th Int. Workshop Information Hiding*, K. Solanki, Ed., Santa Barbara, CA, Jun. 19–21, 2008, Lecture Notes in Computer Science, pp. 161–177, Springer-Verlag, New York.

[9] D. Soukal, J. Fridrich, and M. Goljan, "Maximum likelihood estimation of secret message length embedded using $\pm k$ steganography in spatial domain," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, E. J. Delp and P. W. Wong, Eds., San Jose, CA, Jan. 16–20, 2005, vol. 5681, pp. 595–606.

[10] Q. Guan, J. Dong, and T. Tan, "Blind quantitative steganalysis based on feature fusion and gradient boosting," in *Proc. 9th Int. Conf. Digital Watermarking (IWDW'10)*, Berlin, Heidelberg, 2011, pp. 266–279, Springer-Verlag.

[11] S. Lyu and H. Farid, "Steganalysis using higher-order image statistics," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 1, pp. 111–119, Mar. 2006.

[12] G. Xuan, Y. Q. Shi, J. Gao, D. Zou, C. Yang, Z. Z. P. Chai, C. Chen, and W. Chen, "Steganalysis based on multiple features formed by statistical moments of wavelet characteristic functions," in *Proc. 7th Int. Workshop Information Hiding*, M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, Eds., Barcelona, Spain, Jun. 6–8, 2005, vol. 3727, Lecture Notes in Computer Science, pp. 262–277, Springer-Verlag, Berlin.

[13] Y. Q. Shi, C. Chen, and W. Chen, "A Markov process based approach to effective attacking JPEG steganography," in *Proc. 8th Int. Workshop Information Hiding*, J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, Eds., Alexandria, VA, Jul. 10–12, 2006, vol. 4437, Lecture Notes in Computer Science, pp. 249–264, Springer-Verlag, New York.

[14] T. Pevný and J. Fridrich, "Merging Markov and DCT features for multi-class JPEG steganalysis," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, E. J. Delp and P. W. Wong, Eds., San Jose, CA, Jan./Feb. 29 – 1, 2007, vol. 6505, pp. 3 1–3 14.

[15] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," in *Proc. 11th Workshop on Multimedia & Security*, J. Dittmann, J. Fridrich, and S. Craver, Eds., New York, Sep. 7–8, 2009, pp. 75–84, ACM.

[16] J. Kodovský and J. Fridrich, "Influence of embedding strategies on security of steganographic methods in the JPEG domain," in *Proc. SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, E. J. Delp and P. W. Wong, Eds., San Jose, CA, Jan. 27–31, 2008.

[17] D. Zo, Y. Q. Shi, W. Su, and G. Xuan, "Steganalysis based on Markov model of thresholded prediction-error image," in *Proc. IEEE Int. Conf. Multimedia and Expo.*, Toronto, Canada, Jul. 9–12, 2006, pp. 1365–1368.

[18] T. Pevný, J. Fridrich, and A. D. Ker, "From blind to quantitative steganalysis," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents XI*, E. Delp and P. Wong, Eds., Jan. 19–22, 2009.

[19] J. Fridrich and D. Soukal, "Matrix embedding for large payloads," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, E. J. Delp and P. W. Wong, Eds., San Jose, CA, Jan. 16–19, 2006, vol. 6072, pp. W1–W10.

[20] A. Westfeld, "High capacity despite better steganalysis (F5 – A steganographic algorithm)," in *Proc. 4th Int. Workshop Information Hiding*, I. S. Moskowitz, Ed., Pittsburgh, PA, Apr. 25–27, 2001, vol. 2137, Lecture Notes in Computer Science, pp. 289–302, Springer-Verlag, New York.

[21] A. J. Smola and B. Schölkopf, A Tutorial on Support Vector Regression NeuroCOLT2 Tech. Rep. NC2-TR-1998-030, 1998.

[22] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. Cambridge, MA: MIT Press, 2001.

[23] A. Latham, Steganography [Online]. Available: http://linux01.gwdg.de/~alatham/stego.html

[24] D. Upham, JSteg source [Online]. Available: http://zooid.org/~paul/crypto/jsteg/

[25] P. Sallee, "Model-based steganography," in *Proc. 2nd Int. Workshop Digital Watermarking*, T. Kalker, I. J. Cox, and Y. M. Ro, Eds., Seoul, Korea, Oct. 20–22, 2003, vol. 2939, Lecture Notes in Computer Science, pp. 154–167, Springer-Verlag, New York.

[26] Y. Kim, Z. Duric, and D. Richards, "Modified matrix encoding technique for minimal distortion steganography," in *Proc. 8th Int. Workshop Information Hiding*, J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, Eds., Alexandria, VA, Jul. 10–12, 2006, vol. 4437, Lecture Notes in Computer Science, pp. 314–327, Springer-Verlag, New York.

[27] J. Fridrich, T. Pevný, and J. Kodovský, "Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities," in *Proc. 9th ACM Multimedia & Security Workshop*, J. Dittmann and J. Fridrich, Eds., Dallas, TX, Sep. 20–21, 2007, pp. 3–14.

[28] N. Provos, "Defending against statistical steganalysis," in *Proc. ACM Symp. Applied Computing, 10th USENIX Security Symp.*, Aug. 13–17, 2001.

[29] J. Fridrich, M. Goljan, and D. Soukal, "Perturbed quantization steganography," *ACM Multimedia Syst. J.*, vol. 11, no. 2, pp. 98–107, 2005.

[30] S. Hetzl and P. Mutzel, "A graph-theoretic approach to steganography," in *Proc. 9th IFIP TC-6 TC-11 Int. Conf. Communications and Multimedia Security (CMS 2005)*, J. Dittmann, S. Katzenbeisser, and A. Uhl, Eds., Salzburg, Austria, Sep. 19–21, 2005, vol. 3677, Lecture Notes in Computer Science, pp. 119–128.

[31] R. Böhme and A. D. Ker, "A two-factor error model for quantitative steganalysis," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, E. J. Delp and P. W. Wong, Eds., San Jose, CA, Jan. 16–19, 2006, vol. 6072, pp. 59–74.

[32] G. Cancelli, G. Doërr, I. Cox, and M. Barni, "Detection of ± 1 steganography based on the amplitude of histogram local extrema," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, San Diego, CA, Oct. 12–15, 2008.

[33] M. Goljan, J. Fridrich, and T. Holotyak, "New blind steganalysis and its implications," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, E. J. Delp and P. W. Wong, Eds., San Jose, CA, Jan. 16–19, 2006, vol. 6072, pp. 1–13.

[34] [Online]. Available: http://www.imagemagick.org

[35] S. Dumitrescu and X. Wu, "LSB steganalysis based on higher-order statistics," in *Proc. 7th ACM Multimedia & Security Workshop*, A. M. Eskicioglu, J. Fridrich, and J. Dittmann, Eds., New York, NY, Aug. 1–2, 2005, pp. 25–32.

[36] R. Böhme, "Improved Statistical Steganalysis using Models of Heterogeneous Cover Signals," Ph.D. degree, Technische Universität Dresden, Dresden, Germany, Sep. 2008.

[37] P. Royston, "An extension of Shapiro and Wilk's test for normality to large samples," *Appl. Statist.*, vol. 31, pp. 115–124, 1982.

[38] A. Greenspun [Online]. Available: http://www.greenspun.com

[39] A. D. Ker and R. Böhme, "Revisiting weighted stego-image steganalysis," in *Proc. SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, E. J. Delp and P. W. Wong, Eds., San Jose, CA, Jan. 27–31, 2008, vol. 6819, pp. 5-1–5-17.

Tomáš Pevný received the M.S. degree in computer sciences from Czech Technical University, Prague, in 2003, and the Ph.D. degree in computer sciences from the State University of New York at Binghamton in 2008.

He holds the position of researcher at Czech Technical University, Prague, Czech Republic. In 2008–2009, he spent one year in Gipsa-lab in Grenoble, France, as a postdoctoral researcher. His main research interests are in nonparametric statistics with focus on steganography, steganalysis, network security, and intrusion detection.

Jessica Fridrich (M'05) received the MS degree in applied mathematics from Czech Technical University, Prague, Czech Republic, in 1987, and the Ph.D. degree in systems science from Binghamton University, in 1995.

She holds the position of Professor of Electrical and Computer Engineering at Binghamton University (SUNY), Binghamton, NY. Her main interests are in steganography, steganalysis, digital watermarking, and digital image forensic.

Dr. Fridrich's research work has been generously supported by the U.S. Air Force and AFOSR. Since 1995, she received 19 research grants totaling over $7.5 mil for projects on data embedding and steganalysis that lead to more than 120 papers and 7 U.S. patents. She is a member of ACM.

Andrew D. Ker (M'06) was born in Birmingham, U.K., in 1976. He received the B.A. degree in mathematics and computer science and the D.Phil. degree in computer science from Oxford University, Oxford, U.K., in 1997 and 2001, respectively.

He is Fellow and Praelector in Computer Science at University College, Oxford, and a University Lecturer at the Department of Computer Science, Oxford University. He has published widely on steganography and steganalysis, in practice and theory.

Dr. Ker is also a member of the ACM.