Quantitative Steganalysis of LSB Embedding in JPEG Domain

Jan Kodovský
Binghamton University
Department of ECE
Binghamton, NY 13902-6000
jan.kodovsky@binghamton.edu

Jessica Fridrich
Binghamton University
Department of ECE
Binghamton, NY 13902-6000
fridrich@binghamton.edu

ABSTRACT

We construct new quantitative steganalyzers for steganographic techniques that hide data using LSB embedding in quantized DCT coefficients of a JPEG file. Two approaches are explored – change-rate estimation using the maximum likelihood principle with a precover model and a heuristic approach based on minimizing a penalty functional obtained from a combined analysis of the embedding operation and properties of natural images. The techniques are applied to Jsteg and its modified version called symmetric Jsteg. Experiments are used to compare the new methods with current state of the art.

Categories and Subject Descriptors

I.4.9 [**Computing Methodologies**]: Image Processing and Computer Vision—*Applications*

General Terms

Security, Algorithms, Theory

Keywords

Jsteg, Symmetric Jsteg, Quantitative Steganalysis, Maximum Likelihood, Precover, Zero Message Hypothesis

1. INTRODUCTION

Many steganographic programs that hide messages inside digital media files employ some version of the Least Significant Bit (LSB) embedding – the LSBs of individual media elements are replaced with message bits. This intuitively obvious data-hiding paradigm gained on popularity because it can be very easily implemented and provides high embedding capacity without introducing perceptible artifacts. While steganalysis of LSB embedding in images represented in the spatial domain is nowadays a mature area of research as witnessed by the well-founded structural steganalysis [12] and other approaches [13], we feel that comparatively less attention has been paid to the transform domain and its most ubiquitous form – the JPEG format. This is despite the ever-increasing interest of defense government agencies and forensic analysts to reliably detect even short messages.

The first algorithm designed to hide message bits in LSBs of quantized DCT coefficients was Jsteg [20]. Most attacks on Jsteg capitalize on the fact that its embedding disrupts the symmetry of the histogram of DCT coefficients [24, 23, 17, 16]. It is also possible to steganalyze Jsteg using approaches originally developed for LSB embedding in the spatial domain by applying such methods to the 2D array of quantized DCT coefficients [21, 1]. Currently, the most accurate quantitative steganalyzer of Jsteg uses regression in a space of features to which images are mapped [19]. Some spatial-domain LSB detectors are maximum likelihood estimators constructed from appropriate models of cover images [4, 8]. However, as noted in [11], their performance is usually weak due to lack of sufficiently accurate cover models. An appealing and quite successful option is to form a cover model by assuming that the cover is being generated from a hypothetical precover source [11]. This idea put existing structural methods on a firmer theoretical ground and outlined possible avenues for their improvement by allowing deviations from restrictive model assumptions they were based on.

In this paper, we build upon previously-proposed concepts and techniques and describe two different classes of quantitative steganalyzers and apply them to LSB embedding in JPEG domain. The goal is to improve the accuracy of existing attacks. This paper is a conference version of a journal submission [15], which is focused solely on Jsteg. Here, we extend the analysis to another embedding algorithm that we call symmetric Jsteg (sym-Jsteg). Inclusion of this algorithm allows us to better explain the strengths and weaknesses of various approaches and explore their limitations.

Algorithms that will be subjected to steganalysis are described in Section 2. Then, in Section 3, we introduce two different quantitative attacks: a maximum likelihood estimator with precover and a heuristic estimator based on a zero message hypothesis. Specific examples of attacks sorted by the cover feature model are described in Section 4. Experimental evaluation and comparison to prior art is included in Section 5. The paper is closed by providing a summary and outlining possible future directions in Section 6.

2. JSTEG AND SYMMETRIC JSTEG

In this section, we describe two versions of the LSB embedding mechanism in quantized DCT coefficients of a JPEG

MM&Sec'10, September 9–10, 2010, Roma, Italy.

file. The first algorithm is Jsteg, developed by Upham. Jsteg first decompresses the JPEG bit stream to individual quantized DCT coefficients and then replaces their LSBs with message bits. In color cover images, the message bits are embedded in both luminance and chrominance components. By design, Jsteg does not embed in coefficients from the LSB pair $(0, 1)$ because majority of DCT coefficients are zeros and when embedding in them, too many ones would be introduced by embedding, which would lead to quite perceptible artifacts. The first version of Jsteg embedded message bits sequentially, which turned out to be accurately detectable by the histogram attack [22]. An improved version of Jsteg incorporates random straddling and embeds data along a pseudo-random path determined by a shared stego key. We will refer to this randomized version of the algorithm as Jsteg.

By prohibiting the embedder from modifying the LSB pair $(0, 1)$, Jsteg embedding disrupts the symmetry of the histogram and permits construction of accurate structural attacks. A simple modification of Jsteg that preserves the histogram symmetry is obtained by redefining the LSB pairs for positive coefficients to $(1, 2)$, $(3, 4)$, etc. This way, all non-zero DCT coefficients can be used for embedding. This modified embedding operation no longer flips LSBs and is more reminiscent of LSB matching as it can change more significant bits. Besides preserving the histogram symmetry, the embedding capacity of this algorithm is also increased, which means that the same absolute payload can now be embedded with a stronger matrix embedding code, which further improves the security of this algorithm. This modification of Jsteg will be called symmetric Jsteg and abbreviated as "sym-Jsteg." There is no doubt that sym-Jsteg is still a rather poor algorithm whose security is far below more advanced algorithms, such as nsF5 [7]. The inclusion of the sym-Jsteg algorithm will give us the ability to better explain the proposed steganalysis attacks, their construction, as well as limitations.

The embedding capacity of sym-Jsteg (Jsteg) is the number of all non-zero (non-zero and non-one) DCT coefficients, which we will denote as n_0 (n_{01}). Since the ratio n_0/n_{01} varies across images, when the relative payload α (or change rate β) is fixed for one algorithm, it will vary for the other algorithm. Since we desire to evaluate quantitative steganalyzers, which are change-rate estimators, we fix β for both algorithms, knowing that by doing this we cannot fairly compare the results across the algorithms. Measuring the payload using change rate indeed makes sense since any quantitative steganalyzer can only estimate the number of embedding changes rather than the absolute number of embedded bits. When no matrix embedding is applied, the change rate is directly related to the relative payload, $\beta = \alpha/2$.

3. PROPOSED ATTACKS

In this section, we introduce two methods for estimating the change rate. The first one is a Maximum Likelihood (ML) estimator derived from a precover model, while the other approach uses a heuristically-defined penalty function constructed from some zero message hypothesis.

Throughout this paper, cover images will be represented using a feature vector $\mathbf{x} \in \mathbb{R}^d$. We comment on the choice of $\mathbf{x}$ shortly. Furthermore, it will be assumed that $\mathbf{x}$ is drawn from some prior cover distribution $P_x(\mathbf{x})$. For a fixed β and $\mathbf{x}$, the impact of embedding will be modeled with the conditional probability $P(\mathbf{x}^\beta|\mathbf{x}, \beta)$, where $\mathbf{x}^\beta$ is the feature vector of the stego image affected with change rate β. More precisely, $\mathbf{x}^\beta$ is a random variable over the pseudo-random selection of positions of embedding changes in $\mathbf{x}$.

3.1 Maximum Likelihood Steganalyzer

Given the feature of the stego image $\mathbf{y}$ and assuming the cover image and the change rate β are independent of each other,[1]

$$P(\mathbf{y}, \beta) = \int_{\mathbb{R}^d} P(\mathbf{x}, \mathbf{y}, \beta)\mathrm{d}\mathbf{x} = \int_{\mathbb{R}^d} P(\mathbf{y}|\mathbf{x}, \beta)P(\mathbf{x}, \beta)\mathrm{d}\mathbf{x}$$
$$= P(\beta)\int_{\mathbb{R}^d} P(\mathbf{y}|\mathbf{x}, \beta)P_x(\mathbf{x})\mathrm{d}\mathbf{x},$$

which leads to the following ML estimator of β:

$$\hat{\beta} = \arg\max_{\beta\geq 0} P(\mathbf{y}|\beta) = \arg\max_{\beta\geq 0} \int_{\mathbb{R}^d} P(\mathbf{y}|\mathbf{x}, \beta)P_x(\mathbf{x})\mathrm{d}\mathbf{x}. \quad (1)$$

It is apparent now that the feature $\mathbf{x}$ needs to be chosen with two things in mind. First, $\mathbf{x}$ should change predictably with embedding so that $P(\mathbf{x}^\beta|\mathbf{x}, \beta)$ can be derived. Second, we need to be able to adopt a reasonably simple model of covers $P_x(\mathbf{x})$. While obtaining the posteriors $P(\mathbf{x}^\beta|\mathbf{x}, \beta)$ is usually straightforward, modeling the cover prior $P_x(\mathbf{x})$ can be in general rather difficult. We can and should capitalize on embedding invariants to narrow down the prior. For example, because Jsteg preserves the counts of DCT coefficients in LSB bins $(2i, 2i+1)$, we can set $P_x(\mathbf{x}) = 0$ for covers whose counts violate at least one embedding invariant. As in [11], the precover model is derived by postulating that, at least for the purpose of ML estimation of β, the covers are being generated from a precover source, whose properties are determined by the side-information in the form of the observed stego image with its embedding invariants. Specific examples of this approach are given in Section 4.

3.2 Steganalysis Using Zero Message Hypothesis

Although ML change-rate estimators are theoretically well-founded, the complexity of obtaining the precover model and the embedding probabilities $P(\mathbf{x}^\beta|\mathbf{x}, \beta)$ may become quite prohibitive as one tries to utilize higher-order statistics of cover elements. This will become apparent in Section 4.3, where we discuss the precover model that reflects inter-block dependencies among DCT coefficients. Moreover, to prevent a significant loss of statistical samples, which might lead to increased estimator variance, we are often forced to adopt a simplified precover model that is not supported by experiments. This leads to model mismatch and suboptimality.

Being aware of these difficulties, as an alternative we explore the following simple heuristic approach to constructing quantitative steganalyzers. It can easily incorporate even very complex dependencies among DCT coefficients as long as they can be expressed using an appropriate penalty function $z(\mathbf{x}) \geq 0$ that satisfies

$$z(\mathbf{x}^\beta) \approx 0 \quad \text{when } \beta = 0$$
$$z(\mathbf{x}^\beta) > 0 \quad \text{when } \beta > 0.$$

[1] While this assumption can be challenged as the sender may adjust the payload size to the cover, we adopt it here anyway because it simplifies the analysis.

We adopt an additional simplifying assumption commonly made in steganalysis [5, 6, 18, 9] that the impact of embedding is equal to its expectation (which essentially means that the within-image error is ignored):

$$\mathbf{y} = E[\mathbf{x}^\beta] = \mathrm{Emb}(\mathbf{x}, \beta). \tag{2}$$

Provided the mapping Emb is invertible, an estimate of the change rate can be obtained by minimizing the penalty $\mathrm{Emb}^{-1}(\mathbf{y}, \beta)$ evokes:

$$\hat{\beta} = \arg\min_{\beta \geq 0} z(\mathrm{Emb}^{-1}(\mathbf{y}, \beta)). \tag{3}$$

The functional $z(\mathbf{x})$ can be a quantitative description of a Zero Message Hypothesis (ZMH) that captures some key property of covers violated by embedding. The minimization in (3) can be carried out either analytically or numerically by implementing a one-dimensional search over β.

Even though this framework is heuristic, when compared to the ML-based methods, its modularity, low computational complexity, and ability to easily incorporate higher-order statistical properties of covers make it worth investigating. It can also be used to convert some targeted attacks to quantitative ones [15]. Most importantly, as reported in the experimental Section 5, this approach leads to some of the most accurate change-rate estimators for Jsteg today. This approach bears similarity to the least-squares steganalysis [18, 12] and this connection can be used to design the penalty functions. In the next section, we work out both approaches for several feature vectors and apply them to both Jsteg and sym-Jsteg.

4. STEGANALYSIS OF LSB EMBEDDING IN JPEG IMAGES

The accuracy of the proposed estimators is closely tied to the choice of the feature vector. It needs to be chosen while keeping in mind the embedding mechanism of the steganographic algorithm under attack. In this section, we work with three different feature vectors: the feature vector proposed by Zhang and Ping [24], a truncated histogram of DCT coefficients, and a truncated inter-block adjacency matrix.

To avoid repeating the same argument multiple times, all binomial distributions will be approximated with the Gaussian distribution. This is justified due to the fact that typical JPEG images contain a large number of DCT coefficients. Binomial distribution with n samples and probability p will be denoted $\mathrm{Bi}(p, n)$. We will use $\varphi(x; \mu, \sigma^2) = (2\pi\sigma^2)^{-1/2} \exp(-(x-\mu)^2/(2\sigma^2))$ for the pdf of a Gaussian distribution $\mathrm{N}(\mu, \sigma^2)$.

4.1 Features of Zhang and Ping

Denoting with h_i the number of all quantized DCT coefficients in the JPEG image equal to i (the ith histogram bin), we consider the following three-dimensional feature vector originally used in [24]:

$$\mathbf{x} = [x_1, x_2, x_3] \equiv [f_0, f_1 - h_1, h_1], \tag{4}$$

where

$$f_0 = \sum_{k>0} h_{2k} + \sum_{k<0} h_{2k+1}, \tag{5}$$

$$f_1 = \sum_{k\geq 0} h_{2k+1} + \sum_{k<0} h_{2k}. \tag{6}$$

Note that while f_0 decreases with Jsteg embedding, f_1 increases, and $f_0 \approx f_1$ for cover images. We describe the attacks for Jsteg only because sym-Jsteg cannot be attacked using this feature vector due to the fact that embedding with sym-Jsteg preserves it.

4.1.1 ML Attack

Denoting the stego feature $\mathbf{x}^\beta = \left[x_1^\beta, x_2^\beta, x_3^\beta\right]$, due to the properties of LSB embedding in Jsteg, x_1^β is obtained by drawing from x_1 with probability $1 - \beta$ and from x_2 with probability β. Thus, $x_1^\beta \sim \mathrm{Bi}(1-\beta, x_1) + \mathrm{Bi}(\beta, x_2)$, which will be approximated as $x_1 \sim \mathrm{N}(\mu_1, \sigma_1^2)$,

$$\mu_1 = (1-2\beta)x_1 + \beta C, \tag{7}$$

$$\sigma_1^2 = \beta(1-\beta)C, \tag{8}$$

with $C = x_1 + x_2$. Note that the values of C and x_3 do not change during embedding and $C + x_3 = n_0$. Because the probability $P(\mathbf{x}^\beta|\mathbf{x}, \beta)$ can be expressed as

$$P(\mathbf{x}^\beta|\mathbf{x}, \beta) = P(x_3^\beta|x_2^\beta, x_1^\beta, \mathbf{x}, \beta) \cdot P(x_2^\beta|x_1^\beta, \mathbf{x}, \beta) \cdot P(x_1^\beta|\mathbf{x}, \beta),$$

due to the embedding invariants

$$\begin{aligned} P(x_3^\beta|x_2^\beta, x_1^\beta, \mathbf{x}) &= \delta(x_3^\beta = x_3), \\ P(x_2^\beta|x_1^\beta, \mathbf{x}) &= \delta(x_2^\beta = C - x_1^\beta), \end{aligned}$$

we can write

$$P(\mathbf{x}^\beta|\mathbf{x}, \beta) = \varphi\left(x_1^\beta; \mu_1, \sigma_1^2\right) \tag{9}$$

for all $\mathbf{x}^\beta$ that satisfy $x_3^\beta = x_3$ and $x_2^\beta = C - x_1^\beta$, and $P(\mathbf{x}^\beta|\mathbf{x}, \beta) = 0$ otherwise. Note that we reduced the dimensionality of the distribution by incorporating embedding invariants. By introducing the symbol "$\overset{ei}{=}$" meaning that the equality holds only when all embedding invariants are preserved and the probability density is equal to zero otherwise, (9) can be rewritten as

$$P(\mathbf{x}^\beta|\mathbf{x}, \beta) \overset{ei}{=} \varphi\left(x_1^\beta; \mu_1, \sigma_1^2\right), \tag{10}$$

without any additional comments on the values of x_2^β and x_3^β.

The prior $P_x(\mathbf{x})$ will be obtained from a hypothetical source called the precover. The cover property $f_0 \approx f_1$ indicates that a reasonable precover model should be emitting DCT coefficients from $f_0 = x_1$ or $f_1 = x_2 + x_3$ independently and equiprobably, leading to $x_1 \sim \mathrm{N}(\mu_2, \sigma_2^2)$, with $\mu_2 = \frac{1}{2}(C + x_3)$ and $\sigma_2^2 = \frac{1}{4}(C + x_3)$. Thus,

$$P_x(\mathbf{x}) \overset{ei}{=} \varphi\left(x_1; \mu_2, \sigma_2^2\right). \tag{11}$$

The normality of x_1 can be verified by inspecting the ratio $(f_0 - \mu_2)/\sigma_2$ for natural images. Experiments indicate that x_1 follows the precover model only when it is computed from histogram bins h_k with $|k| \geq 3$ (see the journal version of this paper [15]). To avoid a significant loss of data and being aware of the model mismatch, we nevertheless compute x_1 from all bins.

The ML estimator is obtained by substituting (10) and (11) into (1). The formula can be simplified because the involved integral can be evaluated analytically. We refrain from including further details of this change-rate estimator because its performance turned out to be essentially identical to the original estimator of Zhang and Ping (shortly ZP estimator) given by[2]

$$\hat{\beta} = \frac{f_1 - f_0}{2h_1}. \tag{12}$$

Despite the lack of performance gain, the exposition is valuable because it is simple enough to explain the methodology and allowed us to introduce useful notation. Interestingly, we note that by adopting an additional simplifying assumption that the stego feature x_1^{β} is equal to its expectation,

$$x_1^{\beta} = (1-2\beta)x_1 + \beta C, \tag{13}$$

the integration in (1) degenerates to a multiplication and the ML estimator reduces to the ZP estimator (12).

We do not elaborate on ZMH attacks using this feature vector as it is straightforward to show that the ZMH approach to Jsteg (Section 3.2) with the following penalty function:

$$z(\mathbf{x}) = (f_0 - f_1)^2 \equiv (x_1 - x_2 - x_3)^2 \tag{14}$$

reduces to the ZP estimator as well.

4.2 Image Histogram

In this section, we use the histogram as a feature in both proposed frameworks. First, we analyze Jsteg. Here, the histogram will be truncated to the range $[-2L, \ldots, 2R+1]$ for some positive R and L:

$$\mathbf{x} \triangleq [h_{-2L}, \ldots, h_{2R+1}]. \tag{15}$$

4.2.1 ML Attack (Jsteg)

Since the embedding changes in individual LSB pairs are independent, $P(\mathbf{x}^{\beta}|\mathbf{x},\beta)$ can be factorized:

$$\begin{aligned} P(\mathbf{x}^{\beta}|\mathbf{x},\beta) &= P\left(x_0^{\beta}|x_0,\beta\right)\cdot P\left(x_1^{\beta}|x_1,\beta\right)\cdot \qquad (16)\\ &\cdot \prod_{k\in I} P\left(x_{2k}^{\beta}, x_{2k+1}^{\beta}|x_{2k}, x_{2k+1},\beta\right), \end{aligned}$$

where $I = \{-L,\ldots,R\}\backslash\{0\}$. From embedding invariants:

$$\begin{aligned} x_k^{\beta} &= x_k \text{ for } k\in\{0,1\}, & (17)\\ x_{2k}^{\beta} + x_{2k+1}^{\beta} &= x_{2k} + x_{2k+1} \triangleq C_{2k} \text{ for } k\in I, & (18) \end{aligned}$$

(16) can be simplified to

$$P(\mathbf{x}^{\beta}|\mathbf{x},\beta) \overset{ei}{=} \prod_{k\in I} P\left(x_{2k}^{\beta}|x_{2k}, x_{2k+1},\beta\right).$$

Using the same reasoning as in Section 4.1.1, approximating the binomial distribution of x_{2k}^{β} with a Gaussian, $P(x_{2k}^{\beta}|x_{2k}, x_{2k+1},\beta) = \varphi(x_{2k}^{\beta};\mu_{2k},\sigma_{2k}^2)$ with

$$\begin{aligned} \mu_{2k} &= (1-2\beta)x_{2k} + \beta C_{2k},\\ \sigma_{2k}^2 &= \beta(1-\beta)C_{2k}, \end{aligned}$$

results in

$$P(\mathbf{x}^{\beta}|\mathbf{x},\beta) \overset{ei}{=} \prod_{k\in I} \varphi(x_{2k}^{\beta};\mu_{2k},\sigma_{2k}^2). \tag{19}$$

To derive the prior $P_x(\mathbf{x})$ from a precover model, we assume that the *unquantized* DCT coefficients are i.i.d. realizations of a random variable ξ that follows a zero-mean generalized Cauchy distribution:

$$g(x) = \frac{p-1}{2s}\left(\left|\frac{x}{s}\right| + 1\right)^{-p}. \tag{20}$$

This model gave us better results than generalized Gaussian which corresponds to the results of [23]. The positive parameters p and s were obtained from the stego image using an ML estimator, given the embedding invariants (17) and (18) as integrals of $g(x)$ over the corresponding regions. Here, we intentionally excluded zeros (the invariant $\int_{-0.5}^{0.5} g(x)\mathrm{d}x = h_0$ was ignored) because the quality of the fit at zero is irrelevant for the estimator and would only lead to a bias in the other LSB pairs.

The precover is formed by assuming that the histogram bin x_{2k} is obtained by making C_{2k} independent draws with probability $g_{2k} \triangleq P(\xi_{2k} \in [2k-0.5, 2k+0.5]\,|\,\xi_{2k} \in [2k-0.5, 2k+1.5])$,

$$g_{2k} = \left[\int_{2k-0.5}^{2k+1.5} g(x)\mathrm{d}x\right]^{-1} \cdot \int_{2k-0.5}^{2k+0.5} g(x)\mathrm{d}x.$$

We approximate the binomial distribution $\xi_{2k} \sim \mathrm{Bi}(g_{2k}, C_{2k})$ with a Gaussian, $\mathrm{N}(\bar{\mu}_{2k}, \bar{\sigma}_{2k}^2)$, with

$$\begin{aligned} \bar{\mu}_{2k} &= C_{2k} g_{2k}, & (21)\\ \bar{\sigma}_{2k}^2 &= C_{2k} g_{2k}(1-g_{2k}). & (22) \end{aligned}$$

As shown in the journal version of this paper [15], the precover model is valid for LSB pairs $[x_{2k}, x_{2k+1}]$ farther away from zero ($|k| \geq 2$). To avoid loss of statistical data, we nevertheless adopt the model for all bins. Thus,

$$P_x(\mathbf{x}) \overset{ei}{=} \prod_{k\in I} \varphi(x_{2k};\bar{\mu}_{2k},\bar{\sigma}_{2k}^2). \tag{23}$$

After substituting (19) and (23) into (1), the maximum can be found numerically. The computational complexity of this estimator is low because all the involved integrals can be evaluated analytically.

Note the difference between the proposed procedure and the attack of Yu *et al.* [23], where authors use the generalized Cauchy fit as well. In [23], the estimator is realized using the chi-square test, whereas here we use ML equation (1) that is to be solved numerically.

4.2.2 ML Attack (sym-Jsteg)

An essentially identical ML estimator can be constructed for sym-Jsteg, except it has a different set of embedding invariants:

$$\begin{aligned} x_0^{\beta} &= x_0\\ x_{2k}^{\beta} + x_{2k+1}^{\beta} &= x_{2k} + x_{2k+1} \text{ for } k\in\{-L,\ldots,-1\}\\ x_{2k-1}^{\beta} + x_{2k}^{\beta} &= x_{2k-1} + x_{2k} \text{ for } k\in\{1,\ldots,R\}. \end{aligned}$$

The generalized Cauchy fit uses integrals over the corresponding bins, which are now positioned symmetrically around zero in contrast with Jsteg. The remainder of the estimator stays the same.

[2] The authors estimated the relative payload under the tacit assumption that no matrix embedding is used. In this case, the expected value of the payload is 2β, which explains the additional 2 in the denominator of (12).

For sym-Jsteg, the histogram was truncated to the range $[-2L, \ldots, 2R]$,

$$\mathbf{x} \triangleq [h_{-2L}, \ldots, h_{2R}]. \tag{24}$$

4.2.3 ZMH Attack (Jsteg)

Because Jsteg embedding violates histogram symmetry, the ZMH framework can exploit the symmetries, $x_k \approx x_{-k}$, using the following penalty function:

$$z_{\text{sym}}(\mathbf{x}) = \sum_{k>0} w_k (x_k - x_{-k})^2, \tag{25}$$

where the weights $w_k \geq 0$ are chosen to minimize the variance of the change-rate estimator. The summation in (25) goes to $B \triangleq \min\{2L, 2R+1\}$ as we consider only the truncated histogram $[h_{-2L}, \ldots, h_{2R+1}]$. Since our next steps are essentially identical to the derivation of optimal weights for least-squares steganalysis [12, 10], we include here only a brief description of the key elements. In particular, the estimator variance is minimized only for stego images with zero payload (covers) under the (precover) assumption that x_k follows a binomial distribution with size $x_k + x_{-k}$ and probability 1/2, or $x_k \sim \mathrm{N}(\hat{\mu}_k, \hat{\sigma}_k^2)$, where $\hat{\mu}_k = (x_k + x_{-k})/2$ and $\hat{\sigma}_k^2 = (x_k + x_{-k})/4$. The weights w_k that minimize the variance of $z_{\text{sym}}(\mathbf{x})$ over cover images are

$$w_k = \frac{1}{x_k + x_{-k}}. \tag{26}$$

We note that the weights (26) are optimal only for cover images (for zero payload) and at least close to optimal for small payloads with no optimality guarantee for larger payloads because in general x_k^{β} is a poor estimate of x_k. Nevertheless, being aware of these facts and keeping in mind that the derived weights (26) can be further improved, nothing prevents us from using them in our framework with the penalty function

$$z_{\text{sym}}(\mathbf{x}) = \sum_{k=1}^{B} \frac{(x_k - x_{-k})^2}{x_k + x_{-k}}. \tag{27}$$

4.2.4 ZHM Attack (sym-Jsteg)

The sym-Jsteg algorithm preserves the histogram symmetry and thus cannot be attacked using the penalty function (27). However, it is conceivable to attack sym-Jsteg using a penalty function that measures a deviation between the histogram and its generalized Cauchy fit obtained from embedding invariants (see the details of the fitting in Section 4.2.1). The assumption is that the best fit will be obtained by $\mathrm{Emb}^{-1}(\mathbf{y}, \beta)$ for β close to the real change rate. Formally, the penalty function is written as

$$z_{\text{fit}}(\mathbf{x}) = \sum_k w_k (x_k - \hat{x}_k)^2,$$

where $\hat{x}_k$ is the histogram value obtained from the generalized Cauchy fit.

Even though the weights w_k could be adjusted, the connection to least-squares steganalysis is now illusory, since $\hat{x}_k$ is the generalized Cauchy fit and the precover cannot be formulated in a straightforward way. We experimentally verified that the choice of weights $w_k = 1/(x_k + \hat{x}_k)$ does not improve the accuracy of the estimator, and therefore used simply $w_k = 1$, i.e., the penalty function

$$z_{\text{fit}}(\mathbf{x}) = \sum_k (x_k - \hat{x}_k)^2. \tag{28}$$

We could theoretically implement the same attack for Jsteg, with the generalized Cauchy distribution fitting procedure adapted from Section 4.2.1. In Section 5, we subject this approach to tests as well.

4.3 Inter-Block Adjacency Matrix

More accurate estimators can likely be built by realizing that DCT coefficients are not i.i.d. but exhibit additional dependencies. We capture inter-block dependencies among DCT coefficients using an adjacency matrix. First, we outline a possible approach to attack Jsteg within the ML framework. We analyze the difficulties of this approach and then work out in detail the ZMH attack. The ZMH attack utilizes the violation of symmetries in the inter-block adjacency matrix. We limit the exposition to Jsteg because sym-Jsteg preserves the symmetry of the adjacency matrix.

4.3.1 ML attack (Jsteg)

Formally, for an image with $N \times M$ pixels, let us denote the array of DCT coefficients as $\mathbf{D}_{u,v}(k,l)$, where (k,l), $k,l \in \{0,\ldots,7\}$, is a DCT mode in block (u,v), $u \in \{0,\ldots,\lceil M/8 \rceil\}$, $v \in \{0,\ldots,\lceil N/8 \rceil\}$. The feature vector is the adjacency matrix $\mathbf{A} = \{a_{ij}\}$:

$$a_{ij} = \Big| \{(u,v,k,l) | \mathbf{D}_{u,v}(k,l) = i, \mathbf{D}_{u,v+1}(k,l) = j\} \Big|. \tag{29}$$

Due to the structure of LSB embedding and since Jsteg does not embed into zeros and ones, $\mathbf{A}$ naturally decomposes into disjoint groups of k coefficient pairs, $k \in \{1,2,4\}$ (see Figure 1) called *k-nodes.* Note that Jsteg embedding can move pairs freely within each node but not among the nodes. The Jsteg embedding transition probabilities for all three k-node types are shown in Figure 2.

The next step is to derive the model for $P(\mathbf{A}^{\beta}|\mathbf{A}, \beta)$ and adopt a model for $P_x(\mathbf{A})$. Following Figures 1 and 2, the probability $P(\mathbf{A}^{\beta}|\mathbf{A}, \beta)$ can be factorized into embedding transition probabilities over individual k-nodes. The probability for 1-nodes is always equal to 1. The situation for 2-nodes is similar to LSB pairs in a one-dimensional histogram. From the embedding transition probabilities shown in Figure 2 (left), a stego 4-node will follow a multinomial distribution that can be approximated by a multivariate Gaussian distribution. Furthermore, because the sum of occurrences of all four pairs in each 4-node is an embedding invariant, the dimension of the multivariate Gaussian distribution is reduced by one, resulting in a three-dimensional Gaussian distribution with an appropriate mean and covariance matrix. This way, it is possible to analytically express $P(\mathbf{A}^{\beta}|\mathbf{A}, \beta)$ as a product of low-dimensional distributions.

The complications that make this approach to change-rate estimation problematic arise when one attempts to model $P_x(\mathbf{A})$. Similarly to the one-dimensional case, the knowledge of embedding invariants can be reflected in $P_x(\mathbf{A})$ through the precover. After factorizing $P_x(\mathbf{A})$ into the probabilities over individual k-nodes, the problem reduces to finding a good parametric model for the (unquantized) cover matrix $\mathbf{A}$, given the integrals over the regions corresponding to the individual k-nodes. However, this is rather difficult because we need to reflect the dependencies between DCT coefficients into the model, otherwise we fundamen-

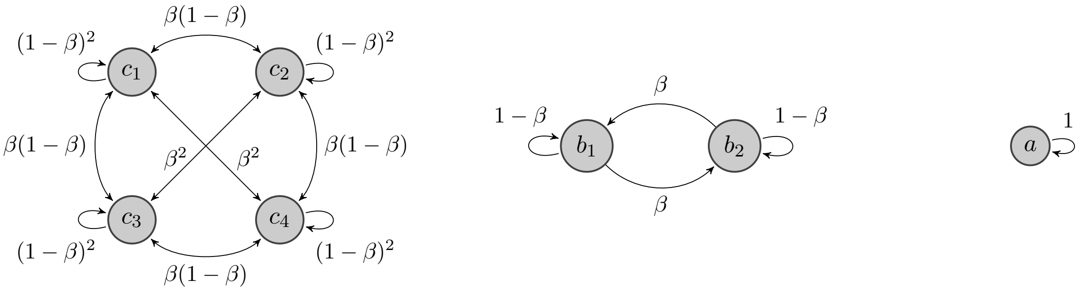

Figure 2: Embedding transition probabilities for all three k-node types. Left: 4-node $[c_1, c_2, c_3, c_4]$, Middle: 2-node $[b_1, b_2]$, Right: 1-node $[a]$.

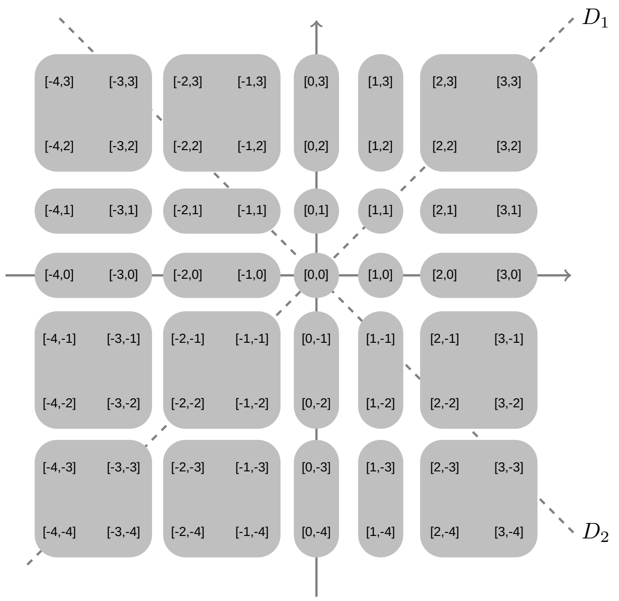

Figure 1: Graphical illustration of matrix $\mathbf{A} = \{a_{ij}\}$ defined by (29). Shaded regions represent 1-nodes, 2-nodes, and 4-nodes.

tally cannot obtain a more accurate detector than in the one-dimensional case. The matrix $\mathbf{A}$ (follow Figure (1)) will exhibit three ridges – one along each axis caused by the fact that coefficients with small absolute value are more frequent than larger ones, and one along the main diagonal, reflecting inter-block dependence of coefficients. Capturing this complicated structure requires using more complex models and estimating more parameters, which increases the complexity of the estimator substantially because this modeling process has to be executed for each analyzed stego image.

Let $g(\mathbf{A})$ be the statistical model for unquantized coefficient pairs on covers. The precover model assigns the pairs in every k-node proportionally to the integrals of $g(\mathbf{A})$ over their corresponding regions. All 2-nodes and 4-nodes will follow a binomial and multinomial distributions, respectively, while 1-nodes will be fully determined by $g(\mathbf{A})$. In principle, we can again utilize Gaussian approximations with dimensionality reduced by one thanks to the embedding invariants.

Finally, the model for $P(\mathbf{A}^\beta|\mathbf{A}, \beta)$ and $P_x(\mathbf{A})$ can be substituted into the ML estimator (1), carrying out the maximization numerically. Unfortunately, unlike in the one-dimensional case, in every step of the maximization procedure we need to numerically evaluate three-dimensional integrals for all 4-nodes, which further increases the complexity.

To summarize our insight, the complexity of the ML procedure rapidly increases due to the difficulties with modeling $P_x(\mathbf{A})$, estimating its parameters from the embedding invariants, and solving (1). We note that similar difficulties materialize when considering this approach to attack symJsteg.

4.3.2 ZMH attack (Jsteg)

In the ZMH framework, we only need to identify a property of a typical cover matrix $\mathbf{A}$ that is disturbed by embedding, which is much easier than obtaining the precover model in the ML framework. The penalty function used in this section builds upon the fact that Jsteg violates certain symmetries of $\mathbf{A}$.

First, we quantify the effect of embedding and find the inverse embedding function $\mathrm{Emb}^{-1}(\mathbf{A}^\beta, \beta)$. The embedding operation can be studied separately for different types of k-nodes (follow Figure 2):

- 1-nodes:

$$a^\beta = a,$$

- 2-nodes:

$$\begin{pmatrix} b_1^\beta \\ b_2^\beta \end{pmatrix} = \begin{pmatrix} \gamma & \beta \\ \beta & \gamma \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ b_2 \end{pmatrix},$$

- 4-nodes:

$$\begin{pmatrix} c_1^\beta \\ c_2^\beta \\ c_3^\beta \\ c_4^\beta \end{pmatrix} = \begin{pmatrix} \gamma^2 & \beta\gamma & \beta\gamma & \beta^2 \\ \beta\gamma & \gamma^2 & \beta^2 & \beta\gamma \\ \beta\gamma & \beta^2 & \gamma^2 & \beta\gamma \\ \beta^2 & \beta\gamma & \beta\gamma & \gamma^2 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}.$$

Above, we used $\gamma = 1 - \beta$. Provided $0 \leq \beta < 1/2$, all three embedding functions are linear mappings with a nonsingular matrix and thus can be easily inverted, which gives us the inverse mapping $\mathrm{Emb}^{-1}(\mathbf{A}^{\beta}, \beta)$.

To find an appropriate functional $z(\mathbf{A})$, we inspect the diagonals denoted D_1 and D_2 in Figure 1. The cover matrix $\mathbf{A}$ is symmetrical about both diagonals: $a_{i,j} \approx a_{j,i}$ (the order of DCT coefficients does not matter) and $a_{i,j} \approx a_{-j,-i}$ (the sign does not matter either).

Because of the natural shape of the distribution of DCT coefficients, there are two major ridges in $\mathbf{A}$ that correspond to the column and row passing through the origin. Note that both ridges are symmetric about both D_1 and D_2.

Since the symmetry about D_2 is disturbed by embedding (note the asymmetrical placement of k-nodes w.r.t. D_2 in Figure 1), it will be employed for structural steganalysis through the ZMH framework using the penalty functional expressed again as a weighted sum of square precover deviations:

$$z_{\mathrm{adj}}(\mathbf{A}) = \sum_{i,j=-B}^{B} \frac{(\bar{a}_{i,j} - \bar{a}_{-i,-j})^2}{\bar{a}_{i,j} + \bar{a}_{-i,-j}}. \tag{30}$$

In (30), $B = \min\{2L, 2R+1\}$ determines the size of the largest square submatrix of $\mathbf{A}$ centered at a_{00} and $\bar{a}_{i,j} = a_{i,j} + a_{j,i}$. The functional (30) is a two-dimensional analogy to the previously introduced optimally weighted one-dimensional penalty function (27). Instead of the histogram symmetry, here we exploit the symmetry of the adjacency matrix along the diagonal D_2. The optimality of weights in (30) relies on the following precover assumption: $a_{i,j} \sim \mathrm{Bi}(1/2, a_{i,j} + a_{-j,-i})$, which is to be simplified as $a_{i,j} \sim \mathrm{N}(\mu_{i,j}, \sigma_{i,j}^2)$ with

$$\begin{aligned} \mu_{i,j} &= \frac{1}{2}(a_{i,j} + a_{-j,-i}), \\ \sigma_{i,j}^2 &= \frac{1}{4}(a_{i,j} + a_{-j,-i}). \end{aligned}$$

This assumption is analyzed in [15]. Because the symmetry $a_{i,j} \approx a_{j,i}$ is preserved under embedding, adding $a_{i,j}$ and $a_{j,i}$ to form a new variable $\bar{a}_{i,j}$ increases the statistical sample and improves the performance.

5. EXPERIMENTAL RESULTS

In this section, we experimentally evaluate all quantitative steganalyzers proposed in this paper and compare their performance with current state-of-the-art estimators, including the recently proposed adaptations of spatial-domain methods [21] as well as the Support Vector Regression (SVR) feature-based approach [19]. The accuracy of the estimators will be evaluated for Jsteg and sym-Jsteg separately for change rates ranging from 0 to 0.2. In practice, an image with a negative change-rate estimate should be interpreted as a cover. However, rounding negative estimates to zero would deform the distribution of $\hat{\beta}$ for small payloads and the results would become less informative. Therefore, the minimization of the penalty function in the ZMH framework is always performed in the interval $[-1/2, 1/2]$.

Because a quantitative steganalysis technique can only estimate the change rate rather than the message length, we used simulations of Jsteg and sym-Jsteg by directly visiting DCT coefficients (along a pseudo-random path) and flipping a desired portion of them. Consequently, the estimation error due to random correlations of the message with the cover elements (e.g., see [2]) is not present in our results. We present an overall performance comparison by simulating embedding on every test image once and analyzing the compound error over the database. More detailed analysis of the within-image and between-image error for Jsteg can be found in [15].

All experiments were performed on a database of images obtained from a mother database of 6,500 JPEG images acquired by 22 different digital cameras at full resolution in a raw format and then converted to grayscale. The size of the images ranged from 1.5 to 6.0 megapixels with a median size of 3.4 megapixels. All images were resized and compressed with the JPEG quality factor 75. The resizing was carried out using bilinear interpolation so that the smaller side after resizing was 512 pixels (aspect ratio preserved).

The obtained image dataset was further randomly divided into two equal parts, each consisting of 3,250 images. The first part was used for training of the SVR-based estimator. All remaining methods were then tested on the second part, regardless of the fact of whether or not they required the first half for training. This way, all methods were evaluated on the same set of images, ensuring thus a fair comparison.

5.1 Performance Measures

The stego images were created by pseudo-randomly changing a fraction β of all non-zero non-one DCT coefficients (Jsteg) or a fraction β of all non-zero DCT coefficients (sym-Jsteg). The accuracy of all estimators is reported using the following measures:

- Median absolute error
$$\mathrm{median}_i \left\{ \left| \hat{\beta}_i - \beta \right| \right\},$$
- Median bias
$$\mathrm{median}_i \left\{ \hat{\beta}_i - \beta \right\},$$
- Interquartile range (IQR)
$$\mathrm{iqr}_i \left\{ \hat{\beta}_i \right\}.$$

5.2 Overall Performance (Jsteg)

For Jsteg, the following quantitative steganalyzers were analyzed: the estimator of Zhang and Ping (12), the histogram-based ML approach described in Section 4.2.1, the Weighted nonsteganographic Borders attack (WB) introduced in [21],[3] the ZMH-based attacks using different features and penalty functions $z_{\mathrm{sym}}(\mathbf{x})$, $z_{\mathrm{fit}}(\mathbf{x})$, and $z_{\mathrm{adj}}(\mathbf{A})$, and the SVR quantitative steganalyzer introduced in [19].

According to our experiments, the accuracy of estimators that rely on histogram (adjacency matrix) symmetry – the ZP estimator, ZMH using $z_{\mathrm{sym}}(\mathbf{x})$, and ZMH using $z_{\mathrm{adj}}(\mathbf{A})$) – decreases unless the DC terms are excluded from computing the histogram. This is because the distribution of DC terms is not symmetric around zero and thus violates the cover assumption of histogram symmetry.

Table 1 conveniently lists all quantitative steganalyzers involved in the test, together with our choices of parameters L, R and $B = \min\{2L, 2R+1\}$. The results are shown in

[3] Even though we also tested the Jpairs attack [21], the results are not included in Figure 3 because this method exhibited a markedly worse accuracy compared to the other methods and by including the results in the graphs, their visual clarity would be negatively affected.

Method		Description
ZP (J)	-	Estimator of Zhang and Ping – formula (12).
ML (J)	-	First-order ML approach described in Section 4.2.1. $L = 3,\ R = 2$.
WB (J)	-	Weighted Nonsteganographic Borders Attack [21]. We used author's code written in R.
ZMH-Sym (J)	-	First-order ZMH attack using penalty function $z_{\mathrm{sym}}(\mathbf{x})$ (27). $L = R = 4$.
ZMH-Fit (J)	-	First-order ZMH attack using penalty function $z_{\mathrm{fit}}(\mathbf{x})$ (28). $L = 3,\ R = 2$.
ZMH-Adj (J)	-	Second-order ZMH attack using penalty function $z_{\mathrm{adj}}(\mathbf{A})$ (30). $B = 3$.
SVR (J)	-	Support vector regression [19] with 548 Cartesian-calibrated Pevný features [14].
ML (S)	-	First-order ML approach for sym-Jsteg described in Section 4.2.2. $L = R = 3$.
ZMH-Fit (S)	-	First-order ZMH attack for sym-Jsteg using penalty function $z_{\mathrm{fit}}(\mathbf{x})$ (28). $L = R = 3$.
SVR (S)	-	Support vector regression [19] with 548 Cartesian-calibrated Pevný features [14].

Table 1: List of all quantitative steganalyzers involved in experiments. The first group of methods appended by (J) was used to attack Jsteg, the second group (S) to attack sym-Jsteg.

Figure 3 (left column). The following conclusions can be drawn:

1. The best performance was achieved using the SVR-based attack (—) and the ZMH approach with $z_{\mathrm{adj}}(\mathbf{A})$ (⋯⋯). Both methods have a very similar accuracy in terms of all three performance measures. The advantage of the proposed ZMH framework over the SVR is that it does not need an expensive training phase – it works solely on an image-by-image basis.

2. Among all histogram-based attacks to Jsteg, the ZMH method using $z_{\mathrm{sym}}(\mathbf{x})$ (-▲-) performs the best. This is because of the strong histogram symmetry violation of Jsteg that is effectively captured by the penalty function $z_{\mathrm{sym}}(\mathbf{x})$.

3. Even though both the ML method (-◇-) and the ZMH approach with $z_{\mathrm{fit}}(\mathbf{x})$ (-★-) use the same generalized Cauchy fit for the cover model, the ML approach delivers a better performance. This indicates that the theoretically well-founded ML method can employ this knowledge better than the heuristic ZMH approach. Unfortunately, the ML approach becomes computationally intractable when considering higher-order statistics.

4. The WB attack (-■-) has the worst performance in steganalysis of Jsteg in agreement with the previously published results [19].

5.3 Overall Performance (sym-Jsteg)

For sym-Jsteg, we tested the ML approach from Section 4.2.2, the ZMH attack with $z_{\mathrm{fit}}(\mathbf{x})$ introduced in Section 4.2.4, and the SVR approach. None of the other methods used in the previous section can be used for steganalysis of sym-Jsteg, since sym-Jsteg preserves the symmetry of both the histogram and the adjacency matrix. Figure 3 (right column) shows the results. In summary,

1. The feature-based SVR method (—) is the best sym-Jsteg change-rate estimator. The high accuracy of the attack indicates that sym-Jsteg is an easily detectable steganographic method, even though it disables most of the ML and ZMH attacks.

2. Similarly to the conclusions drawn from attacks on Jsteg, the ML method (-◇-) outperforms the ZMH attack with $z_{\mathrm{fit}}(\mathbf{x})$ (-★-) even though both approaches are based on the same cover model estimates.

3. Both first-order attacks, i.e., the ML method (-◇-) and the ZMH attack with the penalty function $z_{\mathrm{fit}}(\mathbf{x})$ (-★-) perform significantly worse than for Jsteg. This is because of the less informative embedding invariants and consequently worse estimates of the cover image histograms.[4]

Finally, we note that the journal version of this paper [15] contains detailed analysis of between-image and within-image errors for two selected attacks on Jsteg.

6. CONCLUSION

We describe several new approaches to quantitative steganalysis of LSB embedders in JPEG domain – the Jsteg algorithm and its symmetrized version sym-Jsteg. Sym-Jsteg redefines positive LSB pairs from $(2k, 2k+1)$ to $(2k, 2k-1)$, which allows it to embed into ones as opposed to Jsteg.

The new attacks can be broadly divided into two classes. The first is the theoretically well-founded maximum likelihood approach, in which the cover model is derived for each stego image from a hypothetical precover source formed from embedding invariants. Its complexity, however, quickly increases with increasing complexity of the cover model. Thus, as a simple alternative we proposed the second class of methods that uses a zero message hypothesis (ZMH) to form a non-negative penalty function that attains the minimum value on covers and increases with embedding. This latter approach, which bears some similarity to optimally weighted least squares steganalysis [12], is computationally

[4] The asymmetrical position of LSB pairs in Jsteg leads to overlapping integrals of the cover model $g(x)$ defined by (20). The quality of the fit is further improved by the embedding invariant $\int_{1/2}^{3/2} g(x)\mathrm{d}x$. None of this holds for sym-Jsteg, resulting in less accurate estimates.

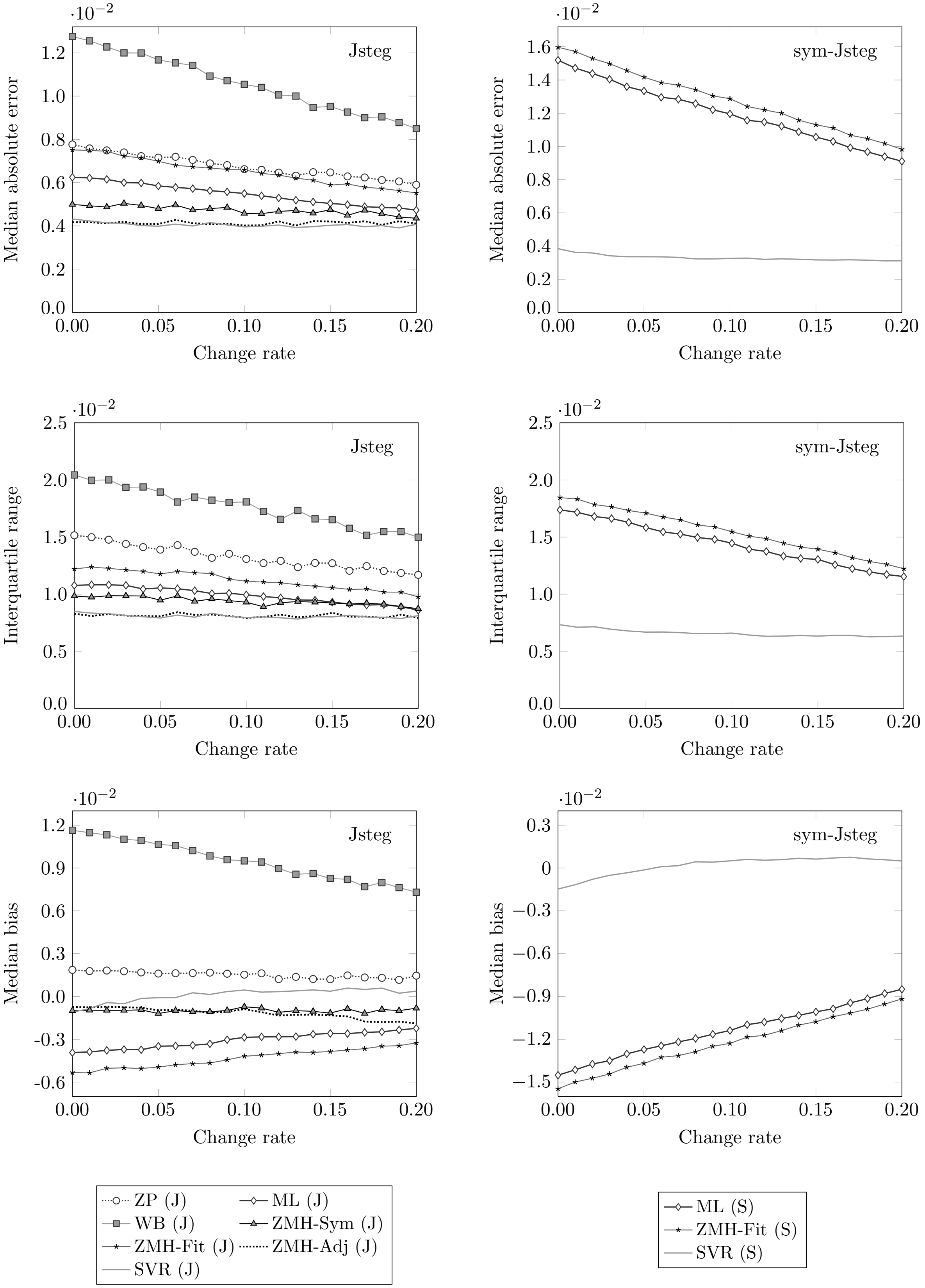

Figure 3: Median absolute error (top), interquartile range (middle), and median bias (bottom) for all attacks on Jsteg (left column) and sym-Jsteg (right column) listed in Table 1.

inexpensive and modular, and, in contrast to the maximum likelihood approach, it allows simple incorporation of even complicated higher-order properties of covers. Moreover, the ZMH approach can be used also to convert existing targeted (but not quantitative) attacks to quantitative ones [15].

All quantitative steganalyzers were experimentally evaluated on a database of 3,250 images and compared to existing attacks. Because Jsteg embedding violates symmetry of the histogram of DCT coefficients and the inter-block adjacency matrix, the impact of embedding can be well captured using properly chosen penalty functions. The most accurate ZMH estimator for Jsteg was based on symmetry of the adjacency matrix. Its accuracy was comparable to the current state-of-the-art method constructed using Support Vector Regression (SVR) with a 548-dimensional feature vector. The newly proposed method, however, offers a much simpler implementation without the need for a potentially expensive training phase.

None of the attacks that rely on symmetry-breaking can be adapted to sym-Jsteg because it preserves these symmetries. The simple modification of the embedding algorithm makes sym-Jsteg much harder to attack using structural steganalysis. None of the newly proposed attacks was able to match the accuracy of the feature-based support-vector regressor constructed from the above-mentioned 548-dimensional feature set.

Among possible future directions, we mention the possibility to further improve the accuracy of the ZMH-based change-rate estimator for Jsteg by utilizing both intra-block and inter-block dependencies between DCT coefficients by modeling the relationship between DCT coefficients with Markov chains as in [3]. Furthermore, the visually observable impact of sym-Jsteg on histogram suggests a reasonable hope of finding better penalty functions that would bring the performance of the ZMH method closer to the SVR-based steganalyzer.

7. ACKNOWLEDGMENTS

The work on this paper was supported by Air Force Office of Scientific Research under the research grant number 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.

8. REFERENCES

[1] R. Böhme. Weighted stego-image steganalysis for JPEG covers. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of *Lecture Notes in Computer Science*, pages 178–194, Santa Barbara, CA, June 19–21, 2007. Springer-Verlag, New York.

[2] R. Böhme and A. D. Ker. A two-factor error model for quantitative steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 59–74, San Jose, CA, January 16–19, 2006.

[3] C. Chen and Y. Shi. JPEG image steganalysis utilizing both intrablock and interblock correlations. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 3029–3032, May 2008.

[4] O. Dabeer, K. Sullivan, U. Madhow, S. Chandrasekaran, and B. S. Manjunath. Detection of hiding in the least significant bit. *IEEE Transactions on Signal Processing*, 52:3046–3058, 2004.

[5] S. Dumitrescu, X. Wu, and N. D. Memon. On steganalysis of random LSB embedding in continuous-tone images. In *Proceedings IEEE, International Conference on Image Processing, ICIP 2002*, pages 324–339, Rochester, NY, September 22–25, 2002.

[6] J. Fridrich, M. Goljan, D. Hogea, and D. Soukal. Quantitative steganalysis of digital images: Estimating the secret message length. *ACM Multimedia Systems Journal*, 9(3):288–302, 2003.

[7] J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities. In J. Dittmann and J. Fridrich, editors, *Proceedings of the 9th ACM Multimedia & Security Workshop*, pages 3–14, Dallas, TX, September 20–21, 2007.

[8] M. T. Hogan, N. J. Hurley, G. C. M. Silvestre, F. Balado, and K. M. Whelan. ML detection of steganography. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 16–27, San Jose, CA, January 16–20, 2005.

[9] A. D. Ker. A general framework for structural analysis of LSB replacement. In M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, editors, *Information Hiding, 7th International Workshop*, volume 3727 of *Lecture Notes in Computer Science*, pages 296–311, Barcelona, Spain, June 6–8, 2005. Springer-Verlag, Berlin.

[10] A. D. Ker. Derivation of error distribution in least squares steganalysis. *IEEE Transactions on Information Forensics and Security*, 2:140–148, 2007.

[11] A. D. Ker. A fusion of maximal likelihood and structural steganalysis. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, volume 4567 of *Lecture Notes in Computer Science*, pages 204–219, Saint Malo, France, June 11–13, 2007. Springer-Verlag, Berlin.

[12] A. D. Ker. Optimally weighted least-squares steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 6 1–6 16, San Jose, CA, January 29–February 1, 2007.

[13] A. D. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 5 1–5 17, San Jose, CA, January 27–31, 2008.

[14] J. Kodovský and J. Fridrich. Calibration revisited. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security*

Workshop, pages 63–74, Princeton, NJ, September 7–8, 2009.

[15] J. Kodovský and J. Fridrich. Quantitative structural steganalysis of Jsteg. Submitted to: *IEEE Transactions on Information Forensics and Security*, currently in review.

[16] K. Lee and A. Westfeld. Generalized category attack – improving histogram-based attack on JPEG LSB embedding. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, volume 4567 of *Lecture Notes in Computer Science*, pages 378–392, Saint Malo, France, June 11–13, 2007. Springer-Verlag, Berlin.

[17] K. Lee, A. Westfeld, and S. Lee. Category attack for LSB embedding of JPEG images. In Y.-Q. Shi, B. Jeon, Y. Shi, and B. Jeon, editors, *Digital Watermarking, 5th International Workshop*, volume 4283 of *Lecture Notes in Computer Science*, pages 35–48, Jeju Island, Korea, November 8–10, 2006. Springer-Verlag, Berlin.

[18] P. Lu, X. Luo, Q. Tang, and L. Shen. An improved sample pairs method for detection of LSB embedding. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of *Lecture Notes in Computer Science*, pages 116–127, Toronto, Canada, May 23–25, 2004. Springer-Verlag, Berlin.

[19] T. Pevný, J. Fridrich, and A. D. Ker. From blind to quantitative steganalysis. In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XI*, volume 7254, pages 0C 1–0C 14, San Jose, CA, January 18–21, 2009.

[20] D. Upham. Steganographic algorithm JSteg. http://zooid.org/~paul/crypto/jsteg.

[21] A. Westfeld. Generic adoption of spatial steganalysis to transformed domain. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of *Lecture Notes in Computer Science*, pages 161–177, Santa Barbara, CA, June 19–21, 2007. Springer-Verlag, New York.

[22] A. Westfeld and A. Pfitzmann. Attacks on steganographic systems. In A. Pfitzmann, editor, *Information Hiding, 3rd International Workshop*, volume 1768 of *Lecture Notes in Computer Science*, pages 61–75, Dresden, Germany, September 29–October 1, 1999. Springer-Verlag, New York.

[23] X. Yu, Y. Wang, and T. Tan. On estimation of secret message length in JSteg-like steganography. In F. A. P. Petitcolas, editor, *Proceedings of the 17th International Conference on Pattern Recognition*, volume 4, pages 673–676, Cambridge, UK, August 23–26, 2004.

[24] T. Zhang and X. Ping. A fast and effective steganalytic technique against Jsteg-like algorithms. In *Proceedings of the ACM Symposium on Applied Computing*, pages 307–311, Melbourne, FL, March 9–12, 2003.