Design of Adaptive Steganographic Schemes for Digital Images

Tomáš Filler and Jessica Fridrich

Department of Electrical and Computer Engineering
SUNY Binghamton, Binghamton, NY 13902-6000, USA

ABSTRACT

Most steganographic schemes for real digital media embed messages by minimizing a suitably defined distortion function. In practice, this is often realized by syndrome codes which offer near-optimal rate–distortion performance. However, the distortion functions are designed heuristically and the resulting steganographic algorithms are thus suboptimal. In this paper, we present a practical framework for optimizing the parameters of additive distortion functions to minimize statistical detectability. We apply the framework to digital images in both spatial and DCT domain by first defining a rich parametric model which assigns a cost of making a change at every cover element based on its neighborhood. Then, we present a practical method for optimizing the parameters with respect to a chosen detection metric and feature space. We show that the size of the margin between support vectors in soft-margin SVMs leads to a fast detection metric and that methods minimizing the margin tend to be more secure w.r.t. blind steganalysis. The parameters obtained by the Nelder–Mead simplex-reflection algorithm for spatial and DCT-domain images are presented and the new embedding methods are tested by blind steganalyzers utilizing various feature sets. Experimental results show that as few as 80 images are sufficient for obtaining good candidates for parameters of the cost model, which allows us to speed up the parameter search.

Keywords: Steganography, minimal-distortion embedding, steganography design.

1. INTRODUCTION

Most steganographic schemes[10] for real digital media embed messages by small perturbations of the original cover object. This form of steganography allows utilizing highly complex cover sources without knowing their exact probability distributions. If precise knowledge of the underlying probability distribution is available, perfectly secure[4] stegosystems can be implemented by merely sampling from the cover source.[1, 26, 30] Unfortunately, such knowledge is often available only for artificial cover sources and not for real digital media, which is an example of an "empirical source." Böhme even argues that the distribution of real digital media is incognizable [3, Chapter 3]. Thus, we study steganographic schemes that embed by minimizing a given distortion function instead of preserving the ever elusive cover distribution. Of course, such schemes are not perfectly secure and fall under the square root law of steganography,[9] which means that the statistical detectability of embedding changes increases with the payload. Thus, by optimizing the embedding we understand minimizing the detectability for a given payload size and for as wide a cover source as possible. The object of optimization is the choice of the distortion function and its parameters and not the actual embedding itself because the problem of embedding with minimal distortion has been already resolved elsewhere for almost arbitrary distortion functions.[7, 8]

To better explain our objective in a precise manner, we now introduce a few technical concepts. In this paper, we use terms "image" and "pixel" mainly to keep the description specific. Applications to other forms of digital media than digital images are certainly possible. We denote by $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X} = \{\mathcal{I}\}^n$ a cover image composed of n pixels with values from the dynamic range $\mathcal{I}$. For example, $\mathcal{I} = \{0, \ldots, 255\}$ for 8-bit grayscale images. Before embedding into $\mathbf{x}$, the sender first defines the range $\mathcal{I}_i \subset \mathcal{I}$ into which each cover pixel x_i can be changed. We call $\mathcal{I}_i$ the support of the embedding operation. An embedding algorithm is called binary and ternary if $|\mathcal{I}_i| = 2$ and $|\mathcal{I}_i| = 3$ for all i, respectively. Given a specific message, the sender strives to find a stego image $\mathbf{y} = (y_1, \ldots, y_n) \in \mathcal{Y} \triangleq \mathcal{I}_1 \times \cdots \times \mathcal{I}_n$ carrying the message with the least possible cost (distortion) $D(\mathbf{x}, \mathbf{y})$.

E-mail: tomas.filler@gmail.com, fridrich@binghamton.edu; T.F.: http://dde.binghamton.edu/filler

For a fixed cover $\mathbf{x}$, the relationship between the minimum expected[*] distortion needed to embed a payload of a fixed size will be referred to as the rate–distortion bound.

Minimal-distortion steganography is often implemented in practice with an additive cost function

$$D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \rho_i(\mathbf{x}, y_i), \tag{1}$$

where $\rho_i(\mathbf{x}, y_i) \in \mathbb{R}$ is the cost of changing the ith cover pixel x_i to y_i. This cost depends only on the original cover image $\mathbf{x}$ and y_i, but not on the other values y_j, $j \neq i$. This choice makes the embedding changes mutually independent.

For example, embedding algorithms may minimize the number of changed cover elements, such as in the nsF5 algorithm,[12] or costs related to the quantization error as in MMx[18, 27] or Perturbed Quantization.[11] In spatial domain, the embedding operation can be ternary, such as in LSB matching, where the color is changed by ±1 randomly. In some algorithms,[24, 29] only the embedding change leading to the smaller distortion is chosen to modify a pixel's LSB. This choice allows the receiver to extract the message from LSBs, but effectively reduces the embedding operation to binary, which limits the maximum possible per-pixel payload to 1 bit instead of $\log_2(3) \approx 1.56$ bits.

In Ref. 8, the authors provide a practical framework allowing the steganographer to minimize an additive distortion function (1) while embedding a near-maximal payload even for embedding operations with a larger support. The framework allows the sender to minimize an additive distortion described by the set of local costs $\rho_i(\mathbf{x}, y_i)$, $i \in \{1, \ldots, n\}$, without having to share them with the receiver. In order to read the message, the only information the receiver needs is the size of the message to be extracted. This freedom opens up the possibility of learning ρ_i from the cover source. By letting $\rho_i(\mathbf{x}, y_i) \to \infty$, the framework can prohibit modifications of the ith pixel – an option often used with zero AC DCT coefficients in JPEG images.[12] It is our belief that further substantial increase in secure payload can be achieved by properly designing the cost function instead of improving the coding algorithm.

The key question is how to derive the cost function D so that minimizing D corresponds to more secure algorithms. In practice, most distortion functions are obtained heuristically and do not generalize well to other cover sources. Even though in this article we limit ourselves to independent embedding changes, the design of single-pixel cost functions ρ_i for an additive D is an important problem. It is the first step leading towards more general solutions, such as the Gibbs construction,[7] that work with non-additive distortion functions that are additive over larger (and possibly overlapping) groups of cover elements of which (1) is a special case. The Gibbs construction generalizes the above framework by minimizing cost functions that can model dependencies among embedding changes.

Our motivation for solving the problem of the cost-function design comes from the HUGO algorithm[24] that assigns the costs of individual changes based on the pixel neighborhood. Unfortunately, this approach does not easily generalize to other cover sources, such as JPEG or color bitmap images, neither is it clear how to optimize the design. In this paper, we open the question of the cost-function design and propose a practical methodology for learning the costs from a set of training cover images using a set of steganalytic features. We also strive for a robust approach that generalizes well to unseen cover images and unseen steganalytic features to avoid overfitting to a particular cover source and feature space. For example, the Feature Correction Method,[19] which is a heuristic approach to embed while approximately preserving the cover-image feature vector, is known to be overly sensitive to the chosen feature set and does not generalize or scale well.

The rest of this paper is organized as follows. In Section 2, we introduce the minimal-distortion embedding framework and its practical implementation. All embedding algorithms introduced in this paper will follow this framework. Section 3 casts the cost-design problem into function optimization and introduces two new design criteria and a methodology for learning the costs from training images. The methodology developed in Section 3 is then applied to grayscale spatial-domain images in Section 4. Application to grayscale JPEG images is considered in Section 5. The paper concludes in Section 6 with a discussion of possible future directions on how to apply and improve the proposed methodology for designing adaptive embedding schemes.

[*] The expectation is over different messages.

2. MINIMAL-DISTORTION EMBEDDING FRAMEWORK

This section summarizes the minimal-distortion embedding framework as described in Ref 8[†]. All quantities derived in this section depend on the chosen cover object $\mathbf{x}$. Let $\mathcal{I}_i \subset \mathcal{I}$ be (possibly different) embedding operations defined for every $i \in \{1, \ldots, n\}$. The sender will embed a message by minimizing the introduced cost (distortion), which we assume to be additive over individual pixels (1). We remind that the distortion is described by the set of local cost functions ρ_i.

We assume that the stego image is a random variable over $\mathcal{I}_1 \times \cdots \times \mathcal{I}_n$ with distribution $\pi_{\mathbf{x}}$, i.e., the probability of sending the stego object $\mathbf{y}$ is $Pr(\mathbf{Y} = \mathbf{y}|\mathbf{x}) = \pi_{\mathbf{x}}(\mathbf{y})$. Without having to share the cover $\mathbf{x}$ or $\pi_{\mathbf{x}}$ with the receiver, the sender can send up to $H(\pi_{\mathbf{x}})$ bits while introducing expected distortion $E_{\pi_{\mathbf{x}}}[D]$, where

$$H(\pi_{\mathbf{x}}) = -\sum_{\mathbf{y}\in\mathcal{Y}} \pi_{\mathbf{x}}(\mathbf{y}) \log_2 \pi_{\mathbf{x}}(\mathbf{y}) \qquad \text{and} \qquad E_{\pi_{\mathbf{x}}}[D] = \sum_{\mathbf{y}\in\mathcal{Y}} \pi_{\mathbf{x}}(\mathbf{y}) D(\mathbf{x},\mathbf{y}).$$

One possible formulation of the embedding problem called the *payload-limited sender* calls for finding $\pi_{\mathbf{x}}$ that achieves the smallest $E_{\pi_{\mathbf{x}}}[D]$ while sending m bits, i.e.,

$$\underset{\pi_{\mathbf{x}}}{\text{minimize}}\ E_{\pi_{\mathbf{x}}}[D] \qquad \text{subject to } H(\pi_{\mathbf{x}}) = m. \tag{2}$$

The solution of this embedding problem is in the form of a Gibbs distribution

$$\pi_{\mathbf{x}}(\mathbf{y}) = \frac{\exp(-\lambda D(\mathbf{x},\mathbf{y}))}{Z(\lambda)} \overset{(a)}{=} \prod_{i=1}^{n} \frac{\exp(-\lambda \rho_i(\mathbf{x}, y_i))}{Z_i(\lambda)} \triangleq \prod_{i=1}^{n} \pi_{\mathbf{x},i}(y_i), \tag{3}$$

where the parameter $\lambda \geq 0$ is obtained by solving the payload constraint in (2),[‡] and $Z(\lambda) = \sum_{\mathbf{y}\in\mathcal{Y}} \exp(-\lambda D(\mathbf{x},\mathbf{y}))$, $Z_i(\lambda) = \sum_{y_i\in\mathcal{I}_i} \exp(-\lambda\rho_i(\mathbf{x}, y_i))$ are the corresponding partition functions. Step (a) follows from the additivity of D, which also leads to mutual independence of individual stego pixels y_i given $\mathbf{x}$. The best possible embedding algorithm implementing the payload-limited sender can be simulated in practice by first solving (2) for λ and then by sampling the ith stego pixel independently from $\pi_{\mathbf{x},i}(y_i)$. This method is particularly useful for testing the algorithm since it allows us to simulate the statistical impact of embedding a random message. The resulting stego objects can then be subjected to steganalysis.

The relationship between the costs, $\rho_i(\mathbf{x}, y_i)$, and the probabilities, $\pi_{\mathbf{x},i}(y_i)$, $y_i \in \mathcal{I}_i$, given by (3) can be inverted so that a given set of probabilities $\pi_{\mathbf{x},i}(y_i)$, $y_i \in \mathcal{I}_i$, leads to costs $\rho_i(\mathbf{x}, y_i)$ unique up to an affine transformation.[§] Using this equivalence, minimal-distortion embedding can be interpreted as a particular case of model-based steganography[28] with one important difference – in our case the model (the cost functions) does not need to be shared with the receiver.

The performance of practical embedding algorithms will be evaluated using the *coding loss* defined as the relative decrease in payload due to practical coding:

$$l(D_\epsilon) = \frac{m_{\text{MAX}} - m}{m_{\text{MAX}}}. \tag{4}$$

In (4), m is the payload embedded by a given algorithm and m_{MAX} is the maximal payload embeddable with distortion not exceeding D_ϵ. The payload-limited sender can be realized in practice using Syndrome-Trellis Codes (STCs),[8] for which the loss l is typically between 7% to 14% depending on the complexity parameter (the constraint height).

[†] For C++ and Matlab implementation, see `http://dde.binghamton.edu/download/syndrome/`.

[‡] A simple binary search is sufficient since $H(\pi_{\mathbf{x}})$ is monotone w.r.t. λ.

[§] Costs for the same i can be multiplied and/or shifted by a common constant without changing the solution of (2).

3. EMPIRICAL DESIGN OF COST FUNCTIONS

In this section, we focus on designing adaptive embedding schemes for the payload-limited sender subjected to sequential steganalysis. In this regime, the sender decides on the number of bits he wants to hide in a given cover object, embeds his payload, and sends the stego object through a passively monitored channel. In sequential steganalysis,[17] the Warden has to decide whether a given image is cover or stego solely based on a single object. We deliberately omit the possibility of intentionally spreading the payload into a group of cover images – a technique known as the batch steganography. This mode can improve the security of the scheme, however, it should no longer be tested with sequential steganalysis. The Warden should use pooled steganalysis[17] that allows her to pool the results over a larger group of objects. We leave this direction open for a future research.

A common way of testing steganographic schemes is to report a chosen detection metric (ROC curve, accuracy, minimum error probability under equal priors P_{E}, etc.) empirically estimated from a database of cover and stego images where each stego image carries a fixed relative payload. Whenever possible, we report results obtained from cover images of roughly the same size to reduce the effect of the square root law.[9]

Our goal is to design a set of functions ρ_i, $i \in \{1,\ldots,n\}$, which, given the original cover image, assign the cost of changing individual cover elements to their new values. For digital images, the dependence between two cover pixels rapidly decreases with their distance. In case of grayscale spatial-domain digital images, the cost of changing a single pixel should mainly depend on its immediate neighborhood. For this reason, we constrain ρ_i to be a real-valued function Θ with small support, $\rho_i(\mathbf{x},y_i)=\Theta(x_{\sigma(i)},y_i)$, where $x_{\sigma(i)}$ denotes cover pixels spatially close to pixel i.

From practical experiments, it is possible to identify the quantity that should drive the costs. For example, pixels in busy regions can be changed more frequently (and by a larger amount) than those in smooth regions because they are generally harder to predict (model). On the other hand, pixels in saturated areas should not be modified at all. However, giving exact relationship between predictability of a pixel change given a small neighborhood, i.e., finding a good Θ is not an easy task. For simplicity, we allow Θ to depend on a vector-valued parameter $\theta \in \mathbb{R}^k$ and use our prior knowledge about the cover source to suitably parametrize Θ. With a real-valued measure of statistical detectability (such as the P_{E} error), the problem of finding the best ρ_i's is transformed to an optimization problem over the parameter space of θ – a problem which can be solved by numerical methods.

In the rest of this section, we review several detectability metrics and discuss their suitability for designing the cost function based on the dimensionality of θ. We will illustrate each optimization criterion on a simple problem of designing an adaptive embedding scheme for grayscale spatial-domain digital images with a single-parameter search space. All experiments described in this section were carried out with 10800 512×512 grayscale images from the BOWS2 database[2] described in Section 4.

Inverse single-difference cost model: Let $\theta \geq 0$ and $\mathcal{N}_i=\{x_{i,\rightarrow},x_{i,\nearrow},x_{i,\uparrow},\ldots,x_{i,\searrow}\}$ be a set of eight pixels from the 3×3 neighborhood of the ith pixel. We use the ±1 embedding operation, $\mathcal{I}_i=\{x_i-1,x_i,x_i+1\}\cap\mathcal{I}$, and define

$$\rho_i(\mathbf{x},y_i)=\Theta(\mathcal{N}_i,y_i)=\begin{cases}0 & \text{if } y_i=x_i,\\ \infty & \text{if } y_i\notin\mathcal{I}_i,\\ \sum_{z\in\mathcal{N}_i}(1+\theta|z-x_i|)^{-1}+(1+\theta|z-y_i|)^{-1} & \text{otherwise.}\end{cases} \tag{5}$$

At the image boundary, the set of neighboring pixels $\mathcal{N}_i$ is reduced accordingly. This cost assignment penalizes changes in textured areas less than those in smooth regions depending on the differences between neighboring pixels.

3.1 Blind steganalysis

The only way of evaluating the security of steganographic schemes for empirical covers is to subject them to a steganalysis test. According to Kerckhoffs' principle, we allow the Warden to know all elements of the stegosystem (the cover source, the embedding algorithm and the size of the possible payload) except for the (possibly encrypted) message. Given a single image, the Warden has to decide whether it is cover or stego. In this simple binary hypothesis test, the Warden can make two types of errors – either detect the cover image as stego (false alarm) or recognize the stego image as cover (missed detection). The corresponding probabilities are denoted P_{FA} and P_{MD}, respectively. The relationship between these two errors is completely described by the ROC curve obtained by plotting $1 - P_{\mathrm{MD}}(P_{\mathrm{FA}})$ as a function of P_{FA}. Unfortunately, ROC curves cannot be directly used for evaluating steganalyzers (embedding algorithms) as they cannot be ordered (they may overlap). Thus, we reduce the ROC curve into a scalar detection measure called the *minimum error probability under equal priors:*

$$P_E = \min_{P_{FA}} \frac{1}{2}\big(P_{FA} + P_{MD}(P_{FA})\big). \tag{6}$$

Due to the lack of exact probability distributions for real digital media covers, practical steganalyzers for such empirical cover sources are constructed by training a binary classifier on a set of cover and stego images obtained by embedding a pseudo-random message. Prior to training, the dimensionality of cover objects is reduced by extracting a feature vector from them. The final steganalyzer can be implemented, for example, using Support Vector Machines[5, 6] (SVM). The features serve here as a lower-dimensional model for the object under study and often capture the dependencies between individual cover pixels (DCT coefficients). Many feature sets were proposed in the literature for grayscale digital images represented either in the DCT or the spatial domain (see Ref. 21 and the references therein). In this paper, we use the second-order SPAM features[23] with $T = 3$ for spatial-domain images, while JPEG images will be represented using the Cartesian-Calibrated Pevný features (CC-PEV) with calibration implemented via cropping by 4×4 pixels.[20] The merger of both sets is called the Cross-Domain Feature set[21] (CDF) and we will use it in both domains.[¶] With regards to machine learning, we use soft-margin SVMs with a Gaussian kernel of width γ implemented using LIBSVM.[5] The database of cover images was randomly divided into two halves – one for training and one for testing. The SVM hyper-parameters C and γ were found using a grid-search with five-fold cross-validation over the set $(C,\gamma) \in \big\{(10^k, 2^j) | k \in \{-3,\dots,4\}, j \in \{-L-3,\dots,-L+3\}\big\}$, where $L = \log_2 d$ is the binary logarithm of the feature dimensionality.

Even though blind steganalysis provides the most trustworthy measure of detectability in practice, it requires a large number of images for training and a separate set of images for testing. In practice, many thousands of images are usually processed by the embedding algorithm to create the stego images and extract the features. Since the training can also be very time consuming, evaluating detectability of a specific embedding algorithm at a given payload using machine learning can be prohibitively expensive. For this reason, only a small number of parameters θ can be evaluated and thus this method is impractical for optimizing a high dimensional θ. This complexity issue is the main motivation for developing alternative and much faster optimization criteria. We used the error P_{E} estimated using an SVM-based classifier mainly for validating the results obtained from other optimization criteria or for performing the grid search over a small region of the search space.

3.2 `L2R_L2LOSS` - soft-margin optimization criterion

Although there exist many algorithms for binary classification, SVMs are popular for their good ability to generalize to unseen data samples. The success of SVMs lies in the optimization criterion which, for the case of a linear classifier, looks for the separating hyperplane maximizing the distance (often called *margin*) between itself and the closest data points. Intuitively, the larger the margin between two classes, the better they can be separated and the smaller the P_{E} error becomes. We use the *size of the margin* for a linear SVM as the optimization criterion. It is described and studied below.

Let $\mathcal{C}$ be the set of N cover images and $\mathcal{S}$ the set of N stego images obtained from $\mathcal{C}$ by embedding a pseudo-random message into each image. By extracting a d-dimensional feature from each image, we obtain a set of $2N$ vectors $\{\mathbf{f}_i \in \mathbb{R}^d | i \in \{1,\dots,2N\}\}$. We also define the labels g_i, $i \in \{1,\dots,2N\}$, as $g_i = -1$ if $\mathbf{f}_i$ was obtained from a cover image and $g_i = +1$ otherwise. Furthermore, we normalize all cover feature vectors $\mathbf{f}_i$ so that the sample variance of each element is 1. This scaling is then applied to stego features as well. SVMs with a linear kernel[16] classify a new sample $\mathbf{f}$ as cover if $\mathbf{w}^T\mathbf{f} < 0$, where $\mathbf{w} \in \mathbb{R}^d$ is the normal vector of the decision hyperplane obtained by solving the optimization problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{2N} \xi(\mathbf{w}; \mathbf{f}_i, g_i). \tag{7}$$

[¶] Spatial-domain images are JPEG compressed with quality factor 100 before CC-PEV features are extracted.

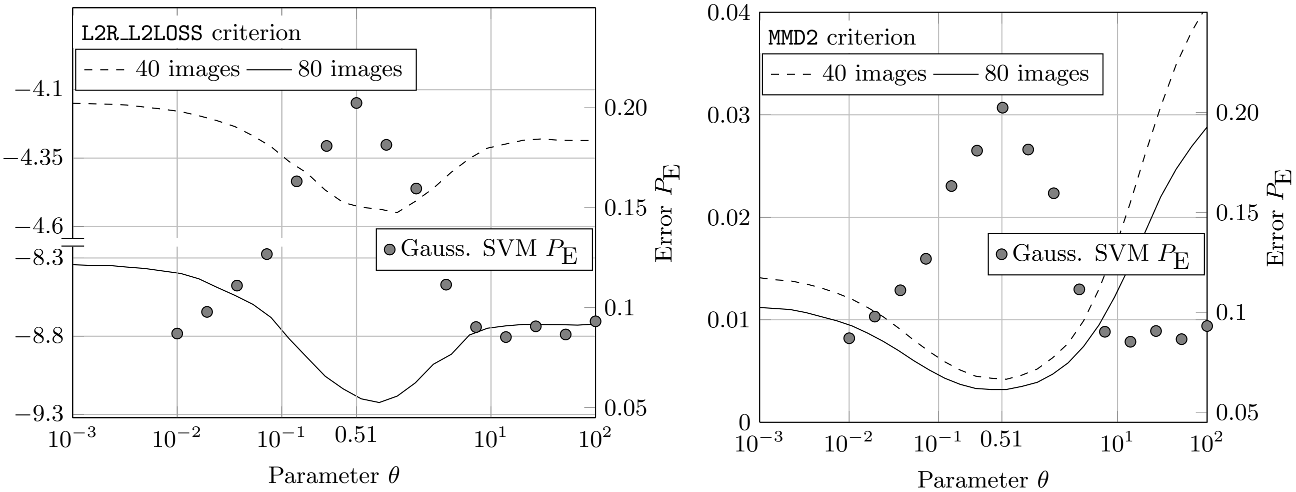

Figure 1. Comparison of different cost assignments in the inverse single-difference cost model (5) with a payload-limited sender embedding 0.5 bpp using the L2R_L2LOSS (left) and MMD2 (right) optimization criteria. The results are compared with the P_E error obtained from an SVM-based classifier. All results were produced using the CDF set and the BOWS2 database of 512 × 512 grayscale images.

Here, $\xi(\mathbf{w}; \mathbf{f}_i, g_i)$ is a loss function and $C > 0$ is a penalty parameter. By minimizing (7), we maximize the margin while penalizing the misclassified samples. We focus on the so-called L2-SVM penalty function $\xi(\mathbf{w}; \mathbf{f}_i, g_i) = \max(1 - g_i \mathbf{w}^T \mathbf{f}_i, 0)^2$. The optimization problem (7) can also be formulated in its dual form:[16]

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^{2N}} \quad & h(\boldsymbol{\alpha}) = \frac{1}{2} \boldsymbol{\alpha}^T \bar{\mathbb{Q}} \boldsymbol{\alpha} - \sum_{i=1}^{2N} \alpha_i \\ \text{subject to} \quad & 0 \le \alpha_i, \forall i \in \{1, \ldots, 2N\}, \end{aligned} \tag{8}$$

where $\bar{\mathbb{Q}} = \mathbb{Q} + \mathbb{D}$, $\mathbb{D}$ being a diagonal matrix with $D_{ii} = (2C)^{-1}$, and $Q_{ij} = g_i g_j \mathbf{f}_i^T \mathbf{f}_j$, $i, j \in \{1, \ldots, 2N\}$. Given $\boldsymbol{\alpha}$, the solution to (7) is $\mathbf{w} = \sum_{i=1}^{2N} g_i \alpha_i \mathbf{f}_i$. From the duality, the value $-h(\boldsymbol{\alpha})$, for any $\boldsymbol{\alpha}$ with $\alpha_i \ge 0$, bounds the optimal solution to the primal problem from below. We call the optimal value of $h(\boldsymbol{\alpha})$ from (8), the L2R_L2LOSS (L_2-regularized L_2-loss) criterion. The smaller the value of this criterion, the larger the optimal value of (7) is, and the smaller the possible margin between cover and stego samples becomes. Therefore, steganographers should be interested in *minimizing* L2R_L2LOSS.

We used a dual coordinate descent method[16] with 10^4 iterations, $C = 0.1$, and $\epsilon = 0.1$ as implemented in the LIBLINEAR[6] package to calculate L2R_L2LOSS. Evaluating L2R_L2LOSS with second-order SPAM features took 1–2 seconds for $N = 80$ 512×512 cover images on a cluster of 40 CPUs when the message-embedding and feature-extraction parts were distributed using OpenMPI.

When optimizing θ using L2R_L2LOSS, we fix the set of cover images $\mathcal{C}$ and the set of pseudo-random messages we will be embedding. We did this by fixing the seeds used for choosing the cover images and the seed used by the embedding simulator. Although L2R_L2LOSS may have different values when evaluated across different sets $\mathcal{C}$, the minimum w.r.t. θ stays approximately the same. Figure 1(left) shows the value of the L2R_L2LOSS criterion based on the CDF set when evaluated for different values of $\theta \ge 0$ in (5) and the number of images in $\mathcal{C}$. We can see that even with 40 images, the optimal value of θ is close to the value obtained from the SVM-based classifier.

Because the L2R_L2LOSS criterion can be evaluated quickly, it can be minimized using numerical methods even for a high dimensional θ. Unfortunately, for higher dimensional θ, the surface obtained by this criterion w.r.t. θ is not smooth enough for gradient-based optimization methods to be used efficiently. Instead, we used the Nelder–Mead simplex-reflection method (exactly as described in [22, Chapter 9.5]) with elements of the initial simplex generated uniformly at random in $[0, 1]$. Due to the non-smooth nature of the optimization criterion, we cannot guarantee that we reached a global minimum (in fact, the solution will be most likely a local minimum).

3.3 Other optimization criteria and their relevance to cost design

Due to the non-smooth optimization surface, we may be interested in other metrics. Metrics leading to a smooth optimization surface may produce an embedding algorithm whose cost assignments may be easier to interpret. Here, we present one such metric – the Maximum Mean Discrepancy (MMD).[14, 25] MMD has been used for comparison of steganographic methods[25] and other machine learning problems, such as feature selection.[13] Originally, MMD was designed as a statistical test for the two-sample problem – to decide whether two data sets were obtained from the same distribution. The theoretical derivation of MMD appears in Ref. 25. Here, we only review the connection between MMD and binary hypothesis testing.

Let $\mathcal{C}'$ and $\mathcal{S}'$ be the sets of N' cover and stego images, respectively. We require the set of cover images used for creating $\mathcal{S}'$ to be disjoint with $\mathcal{C}'$. Let $\mathbf{c}_i, \mathbf{s}_i \in \mathbb{R}^d$, $i \in \{1, \dots, N'\}$, be the feature vectors representing the ith cover and stego image, respectively. As in Section 3.2, we normalize $\mathbf{c}_i$ and $\mathbf{s}_i$ to unit variance obtained from the cover features. An unbiased estimate of MMD^2 is

$$\mathrm{MMD}(\mathcal{C}', \mathcal{S}')^2 = \frac{1}{N'(N'-1)} \sum_{i \neq j} k_\lambda(\mathbf{c}_i, \mathbf{c}_j) - k_\lambda(\mathbf{c}_i, \mathbf{s}_j) + k_\lambda(\mathbf{s}_i, \mathbf{s}_j) - k_\lambda(\mathbf{s}_i, \mathbf{c}_j), \tag{9}$$

where $k_\lambda(\mathbf{c}, \mathbf{s}) = \exp(-\gamma \|\mathbf{c} - \mathbf{s}\|_2^2)$ is the Gaussian kernel with parameter $\gamma \geq 0$. We set the width of the Gaussian kernel to $\lambda = 10^{-3}$, which closely corresponds to the "median rule".[14] In practice, we used the set of $N \geq 2N'$ cover images from which $\mathcal{C}'$ and $\mathcal{S}'$ were derived using a pseudo-random permutation. For a given set of N cover images, we define the `MMD2` criterion as the sample mean of $\mathrm{MMD}(\mathcal{C}', \mathcal{S}')^2$ calculated over M pseudo-random partitions. For the 1234-dimensional CDF set, evaluating `MMD2` using $N = 80$ 512×512 cover images with $N' = 40$ and $M = 10^5$ took 4 seconds on a 40-CPU computer cluster when all operations were parallelized using OpenMPI.

The `MMD2` criterion is related to binary classification using Parzen windows [15, Chapt. 6.6]. A simple binary hypothesis testing problem (deciding whether a given image is cover or stego) can be solved optimally using the Likelihood Ratio Test (LRT) once the exact probability distributions of cover, P_C, and stego feature vectors, P_S, are available. Given an unknown feature vector $\mathbf{f}$, the LRT calls $\mathbf{f}$ cover if $P_C(\mathbf{f}) > P_S(\mathbf{f})$ and stego otherwise. Because neither P_C or P_S are available, one may want to estimate them from a set of N cover and N stego training samples $\mathbf{f}_i \in \mathbb{R}^d$ with labels g_i, $i \in \{1, \dots, 2N\}$. The Parzen estimate of $P_C(\mathbf{f})$ defined as

$$\hat{P}_C(\mathbf{f}) = \frac{1}{N} \sum_{g_i = -1} K_\lambda(\mathbf{f}_i, \mathbf{f}) \tag{10}$$

"counts" the number of training vectors that are close to $\mathbf{f}$. Here, $K_\lambda(\mathbf{f}_i, \mathbf{f})$ is a kernel giving larger weights to vectors closer to $\mathbf{f}$. A popular choice for K_λ is the Gaussian kernel $K_\lambda(\mathbf{f}_i, \mathbf{f}) = k_\lambda(\mathbf{f}_i, \mathbf{f}) = \exp(-\gamma \|\mathbf{f}_i - \mathbf{f}\|_2^2)$. The Parzen estimate of $P_S(\mathbf{f})$, denoted $\hat{P}_S(\mathbf{f})$, is defined in a similar way. When we substitute $\hat{P}_C(\mathbf{f})$ and $\hat{P}_S(\mathbf{f})$ into the LRT, we obtain the Parzen window classifier. Therefore, $\mathrm{MMD}(\mathcal{C}', \mathcal{S}')^2$ calculates a finite-sample estimate of the average detection criterion with equal-priors:

$$\mathrm{MMD}(P_C, P_S)^2 = E_{\mathbf{f}, \mathbf{f}_{-1} \sim P_C, \mathbf{f}_{+1} \sim P_S} \left[k_\lambda(\mathbf{f}, \mathbf{f}_{-1}) - k_\lambda(\mathbf{f}, \mathbf{f}_{+1}) \right] + E_{\mathbf{f}_{-1} \sim P_C, \mathbf{f}, \mathbf{f}_{+1} \sim P_S} \left[k_\lambda(\mathbf{f}, \mathbf{f}_{+1}) - k_\lambda(\mathbf{f}, \mathbf{f}_{-1}) \right] \tag{11}$$

obtained using the leave-one-out cross-validation [15, Chapt. 7.10]. Due to the Gaussian kernel k_λ, $\mathrm{MMD}(P_C, P_S)^2 \geq 0$ and $\mathrm{MMD}(P_C, P_S)^2 = 0$ if and only if $P_C = P_S$. For this reason, the steganographer should *minimize* the `MMD2` criterion, which is a bootstrapped version of (9).

Figure 1(right) compares the `MMD2` criterion when calculated from $N = 80$ and $N = 40$ cover images using $N' = N/2$ and $M = 10^5$ over different values of $\theta \geq 0$. The results obtained from the SVM-based classifier are plotted for reference. Due to bootstrapping, the `MMD2` criterion results in a smooth optimization surface even for a high-dimensional θ. We used a simple gradient descent-based optimization technique to minimize `MMD2`.

4. APPLICATION TO SPATIAL-DOMAIN DIGITAL IMAGES

In this section, we apply the proposed optimization criteria to the problem of optimizing the cost models for grayscale spatial-domain digital images. We first compare the `L2R_L2LOSS` and the `MMD2` criteria on a high-dimensional cost model and validate the results using an SVM-based steganalyzer. `L2R_L2LOSS` is then used for optimizing models similar in nature to those used in the HUGO algorithm.[24]

We use the BOWS2 image database[2] containing approximately 10800 grayscale images of size 512 × 512. Images in this database were obtained by rescaling high-resolution photographs of different scenes originally stored as JPEGs and then converted to grayscale. The database was not processed to remove images containing areas with saturated pixels. For comparison, we also use the BOSSBase[‖] image database with 9074 grayscale images originally taken by seven different camera models in a RAW format (CR2 or DNG) and converted/resized to grayscale images of size 512 × 512. This database was intentionally formed to not contain images with large regions of saturated pixels.

4.1 Comparing the `L2R_L2LOSS` and `MMD2` criteria for high-dimensional search space

In the single-difference cost model (5), the cost of changing the ith pixel was forced to follow the inverse model driven by the scalar parameter θ. We now generalize this and associate one parameter with each value of a pixel difference.

Generalized single-difference cost model: Since most pixel differences are concentrated around zero, we define $\boldsymbol{\theta}=(\theta_{-\Delta},\theta_{-\Delta+1},\ldots,\theta_{\Delta-1},\theta_{\Delta},\theta_{\bullet})\in\mathbb{R}^{2\Delta+2}$ to be a $2\Delta+2$-dimensional vector, for some fixed parameter $\Delta\in\mathbb{N}$. Again, let $\mathcal{N}_i=\{x_{i,\rightarrow},x_{i,\nearrow},x_{i,\uparrow},\ldots,x_{i,\searrow}\}$ be a set of eight pixels in the 3×3 neighborhood of the ith pixel. Given $\boldsymbol{\theta}$, the cost of changing the ith pixel by ± 1, $\mathcal{I}_i=\{x_i-1,x_i,x_i+1\}\cap\mathcal{I}$, is

$$\rho_i(\mathbf{x},y_i)=\Theta(\mathcal{N}_i,y_i)=\begin{cases}0 & \text{if } y_i=x_i,\\ \infty & \text{if } y_i\notin\mathcal{I}_i,\\ \sum_{z\in\mathcal{N}_i}\theta^2_{z-x_i}+\theta^2_{z-y_i} & \text{otherwise,}\end{cases} \tag{12}$$

where $\theta_j=\theta_{\bullet}$ when $|j|>\Delta$. We require $\rho_i(\mathbf{x},y_i)\geq 0$ and enforce this by squaring. Allowing $\rho_i(\mathbf{x},y_i)<\rho_i(\mathbf{x},x_i)$ would lead to cases where it is actually beneficial to make the change instead of keeping the original value. We do not consider such a case here.

Figure 2 shows the progress of optimizing the generalized single-difference cost model (12) using the `MMD2` (left) and `L2R_L2LOSS` (right) criteria when embedding a fixed relative payload of 0.5 bpp. We used a simple gradient-descent and the Nelder–Mead simplex-reflection algorithms utilizing the CDF set to minimize `MMD2` and `L2R_L2LOSS` over a fixed set of 80 images, respectively. Selected values of the parameter $\boldsymbol{\theta}$ were also tested using a Gaussian SVM-based steganalyzer utilizing the CDF set. For the final solution, the `L2R_L2LOSS` criterion provides a more secure embedding algorithm (a higher P_{E} error) than those obtained from `MMD2`. As can be seen from the left figure, optimizing the cost assignments w.r.t. the `MMD2` criterion does not lead to increasing the P_{E} error of the SVM-based steganalyzer. Although the final solution obtained from the `L2R_L2LOSS` criterion does not achieve the best known result (see the leftmost point achieving $P_{\mathrm{E}}=26\%$ in the left graph), we consider it to be better connected to the P_{E} error and use it for all experiments in this paper. The discrepancy between the P_{E} error and the `MMD2` criterion may be due to the strong relationship between `MMD2` and the non-parametric Parzen window classifier, which is believed to be worse than a Gaussian SVM-based steganalyzer. The fact that `L2R_L2LOSS` does not achieve the maximal known P_{E} is because solution was a local minimum. Restarting the Nelder–Mead algorithm with a different initial simplex lead to different solutions achieving different `L2R_L2LOSS` values. The gap between the current and optimal solution may be closed in the future using other optimizing criteria or more involved optimization methods.

[‖] The latest version of the image database used in the BOSS contest `http://boss.gipsa-lab.grenoble-inp.fr/`.

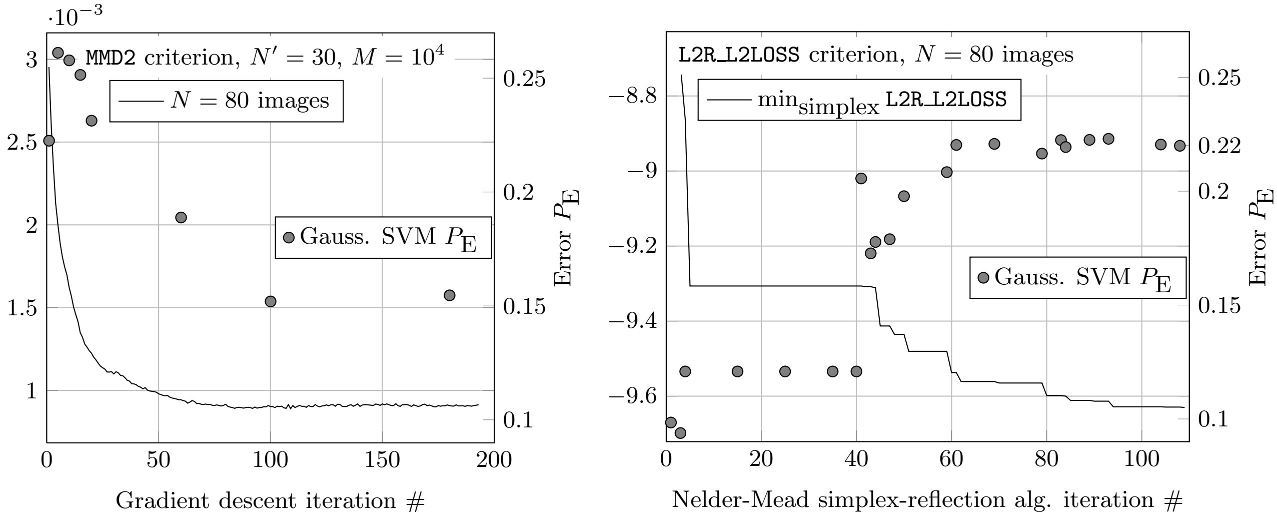

Figure 2. The value of the optimization criteria MMD2 (left) and L2R_L2LOSS (right) when optimized by their respective algorithms using the generalized single-difference cost model (12) embedding 0.5 bpp. Selected cost assignments are validated with the P_{E} error obtained from the SVM-based classifier. All results were produced using the CDF set and the BOWS2 database of 512 × 512 grayscale images. These results are explained in Section 4.1.

4.2 Cost models based on pixel differences

We further generalize the single-difference cost model by allowing the cost to depend on a larger neighborhood via two or three pixel differences. For better clarity, we represent the cover image $\mathbf{x}$ in a matrix form, where $x_{i,j} \in \mathcal{I}$ denotes the pixel in ith row and jth column.

Two-difference cost model: Let $\mathcal{D}^{\rightarrow}_{i,j}(z) = \{(x_{i,j-2} - x_{i,j-1}, x_{i,j-1} - z), (x_{i,j-1} - z, z - x_{i,j+1}), (z - x_{i,j+1}, x_{i,j+1} - x_{i,j+2})\}$ be a set of two-element vectors describing the differences around the i, jth pixel in the horizontal direction when $x_{i,j}$ is replaced by $z \in \mathcal{I}$. We define $\mathcal{D}_{i,j}(z) = \mathcal{D}^{\rightarrow}_{i,j}(z) \cup \mathcal{D}^{\nearrow}_{i,j}(z) \cup \mathcal{D}^{\uparrow}_{i,j}(z) \cup \mathcal{D}^{\nwarrow}_{i,j}(z)$, where the last three sets are defined similarly as $\mathcal{D}^{\rightarrow}_{i,j}(z)$ except with a different orientation. The cost model is described by $\boldsymbol{\theta} \in \mathbb{R}^{(2\Delta+1)^2+1}$ consisting of $\theta_{k,l} \in \mathbb{R}$ for $-\Delta \leq k, l \leq \Delta$ (this models the cost of disturbing the difference vector (k, l)) and $\theta_{\bullet} \in \mathbb{R}$ for all other values outside Δ. Given $\boldsymbol{\theta}$, the cost of changing the i, jth pixel by ±1, $\mathcal{I}_{i,j} = \{x_{i,j} - 1, x_{i,j}, x_{i,j} + 1\} \cap \mathcal{I}$, is

$$\rho_{i,j}(\mathbf{x}, y) = \Theta(y) = \begin{cases} 0 & \text{if } y = x_{i,j}, \\ \infty & \text{if } y \notin \mathcal{I}_{i,j}, \\ \sum_{\mathbf{d} \in \mathcal{D}_{i,j}(x_{i,j})} \theta^2_{\mathbf{d}} + \sum_{\mathbf{d} \in \mathcal{D}_{i,j}(y)} \theta^2_{\mathbf{d}} & \text{otherwise}, \end{cases} \tag{13}$$

where $\theta_{\mathbf{d}} = \theta_{\bullet}$ whenever any element of $\mathbf{d} \in \mathbb{N}^2$ is larger than Δ. We reduce the sum in (13) accordingly when the i, jth pixel is close to the image boundary.

Three-difference cost model: We extend $\mathcal{D}^{\rightarrow}_{i,j}(z)$ to include all three-element vectors one may obtain from four pixels in the horizontal direction containing $x_{i,j}$, i.e., $|\mathcal{D}^{\rightarrow}_{i,j}(z)| = 4$ and define a $(2\Delta + 1)^3 + 1$-dimensional cost model in the same fashion as above.

Figure 3 compares the performance of algorithms based on two and three-difference cost models with $\Delta = 4$ optimized using the L2R_L2LOSS criterion for payloads $\alpha' = 0.2$ and $\alpha' = 0.5$ bpp. Both algorithms were simulated on their respective rate–distortion bounds. The performance of a practical implementation of the scheme for $\alpha' = 0.5$ is rather close to the simulated scheme when implemented using the multi-layered STCs.[8] The costs were minimized using the second-order SPAM features with $T = 3$ and tested with a Gaussian SVM-based steganalyzer with the CDF set. This shows the ability of the optimization procedure to produce cost assignments that are not overtrained to a specific feature set despite the fact that the dimensionality of the search space for the three-difference cost model was $(2\Delta + 1)^3 + 1 = 730$. As can be seen from the figure, the

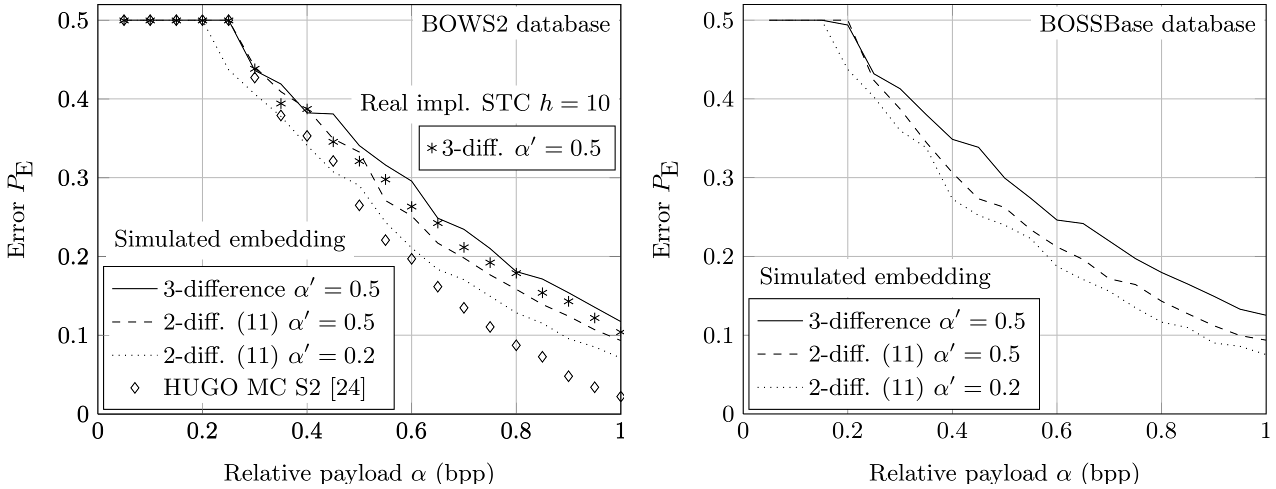

Figure 3. Performance of embedding algorithms optimized using the L2R_L2LOSS criterion with second-order SPAM features with $T = 3$, payload α' bpp, and 80 random images from the BOWS2 database. All algorithms were tested using a Gaussian SVM-based steganalyzer utilizing the CDF set with training and testing images from BOWS2 (left) and BOSSBase (right). Results from the HUGO algorithm[24] when simulated on the rate–distortion bound are shown for comparison.

algorithm designed for $\alpha' = 0.5$ bpp achieved better results for larger payloads. Increasing the design payload above 0.5 bpp did not bring any further improvement. All algorithms achieve better performance than HUGO,[24] because they better utilize the ternary embedding operation for large payloads.

5. APPLICATION TO DIGITAL IMAGES IN DCT DOMAIN

Most adaptive embedding schemes for JPEG images[8, 18, 27] embed message bits while quantizing the DCT coefficients during JPEG compression and minimize an additive distortion function (1) derived from the rounding errors. This approach utilizes the side-information in the form of a never-compressed image, which may not always be available. In this section, we focus on designing adaptive embedding schemes that start directly from a JPEG image and derive the costs of changing a single DCT coefficient from its neighborhood.

We used a mother database of 6500 images obtained from 22 different cameras at their full resolution in a raw format from which a database of 6500 grayscale JPEG cover images was created. Each raw image was first converted to grayscale, resized to a smaller size of 512 pixels using bilinear interpolation while preserving the aspect ratio, and finally JPEG compressed using quality factor 75.

A common way of expressing the payload in DCT-domain steganography is the number of bits embedded per non-zero AC DCT coefficient,[12] which we denote as "bpac." This is because essentially all embedding schemes for DCT domain never change zero coefficients and some even avoid changing DC coefficients due to their high impact on statistical detectability. According to,[12] the most secure algorithm that does not rely on any side-information is the nsF5, which minimizes the number of changed non-zero AC DCT coefficients. Using our terminology, the nsF5 uses a binary embedding operation that decreases the absolute value of a non-zero AC DCT coefficient, i.e., $\mathcal{I}_i = \{x_i, x_i - sign(x_i)\}$ whenever $x_i \neq 0$ is an AC coefficient, and $\mathcal{I}_i = \{x_i\}$ otherwise. Figure 4 shows the performance of nsF5 when simulated as described in Section 2. The detection was implemented using the CDF set with a Gaussian SVM-based steganalyzer.

Similar to the spatial domain, we design the costs based on the differences between DCT coefficients either from neighboring blocks or from similar DCT modes in the same 8×8 block. This allows us to express the context in which a single change is made. We represent a JPEG image $\mathbf{x}$ in a matrix notation, where $x_{i,j} \in \mathcal{I} \triangleq \{-1024, \ldots, 1024\}$ denotes the DCT element of mode $(i \bmod 8, j \bmod 8)$ in the $\lceil i/8 \rceil, \lceil j/8 \rceil$th block. The set $\{x_{i,j} | i \bmod 8 \neq 0 \vee j \bmod 8 \neq 0\}$ describes all AC DCT coefficients in $\mathbf{x}$. We define the following cost model, which we use with a ternary embedding operation.

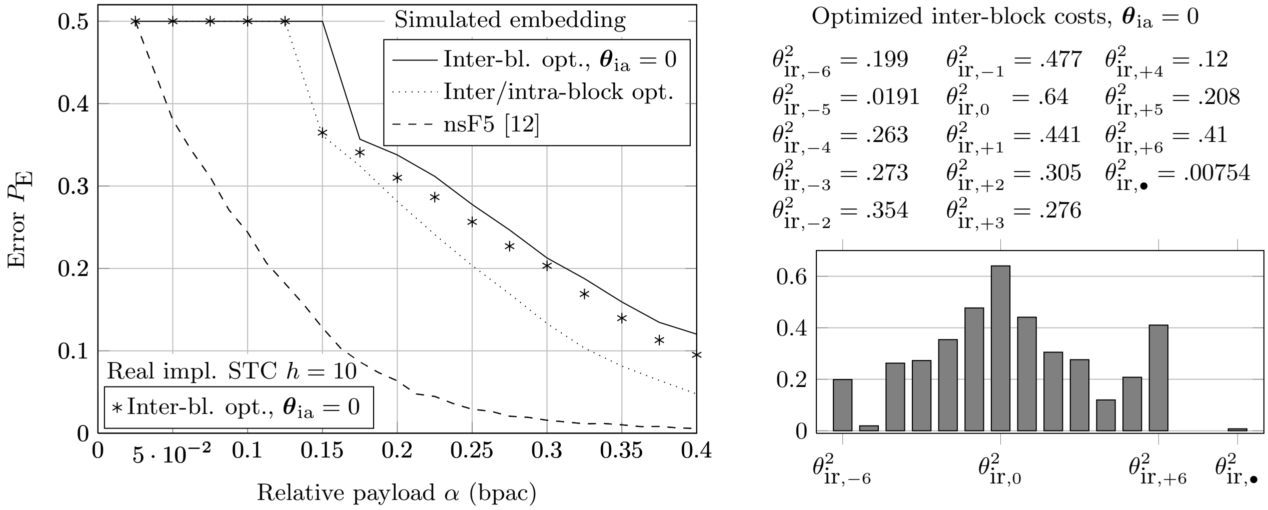

Figure 4. (Left) Detectability of embedding algorithms for the DCT domain based on the inter/intra-block cost model (14) optimized using the `L2R_L2LOSS` criterion and CC-PEV features for the payload of 0.5 bpac. The error P_{E} was measured using a Gaussian SVM-based steganalyzer with the CDF set. (Right) The values of $\boldsymbol{\theta}_{\mathrm{ir}}$ for the optimized inter-block model used to generate the plot on the left.

Inter/intra-block cost model: Let $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\mathrm{ir}}, \boldsymbol{\theta}_{\mathrm{ia}}) \in \mathbb{R}^{(2\Delta+1)+1} \times \mathbb{R}^{(2\Delta+1)+1}$ be the model parameters describing the cost of disturbing inter- and intra-block dependencies with $\boldsymbol{\theta}_{\mathrm{ir}} = (\theta_{\mathrm{ir},-\Delta}, \dots, \theta_{\mathrm{ir},\Delta}, \theta_{\mathrm{ir},\bullet})$ and $\boldsymbol{\theta}_{\mathrm{ia}} = (\theta_{\mathrm{ia},-\Delta}, \dots, \theta_{\mathrm{ia},\Delta}, \theta_{\mathrm{ia},\bullet})$. The cost of changing *any* (even zero) AC DCT coefficient $x_{i,j}$ to $y \in \mathcal{I}_{i,j} \triangleq \{x_{i,j} - 1, x_{i,j}, x_{i,j} + 1\} \cap \mathcal{I}$ is

$$\rho_{i,j}(\mathbf{x}, y) = \Theta(y) = \begin{cases} 0 & \text{if } y = x_{i,j}, \\ \infty & \text{if } y \notin \mathcal{I}_{i,j}, \\ \sum_{z\in\mathcal{N}_{\mathrm{ia}}} \theta^2_{\mathrm{ia},x_{i,j}-z} + \sum_{z\in\mathcal{N}_{\mathrm{ir}}} \theta^2_{\mathrm{ir},x_{i,j}-z} & \text{otherwise}, \end{cases} \tag{14}$$

where $\mathcal{N}_{\mathrm{ir}} = \{x_{i+8,j}, x_{i,j+8}, x_{i-8,j}, x_{i,j-8}\}$ and $\mathcal{N}_{\mathrm{ia}} = \{x_{i+1,j}, x_{i,j+1}, x_{i-1,j}, x_{i,j-1}\}$ are inter- and intra-block neighborhoods, respectively. As before, $\theta_{\mathrm{ia},z} = \theta_{\mathrm{ia},\bullet}$ and $\theta_{\mathrm{ir},z} = \theta_{\mathrm{ir},\bullet}$ whenever $|z| > \Delta$. We reduced the sum in (14) accordingly when the required element falled outside of the image boundary.

Figure 4 (left) compares the performance of embedding algorithms based on the above inter/intra-block cost model when optimized using the `L2R_L2LOSS` criterion with CC-PEV features and payload 0.5 bpac. We report the performance of two algorithms for $\Delta = 6$. In the first version, both $\boldsymbol{\theta}_{\mathrm{ir}}$ and $\boldsymbol{\theta}_{\mathrm{ia}}$ were optimized, while in the second version only the inter-block part $\boldsymbol{\theta}_{\mathrm{ir}}$ was optimized while $\boldsymbol{\theta}_{\mathrm{ia}} = (0, \dots, 0)$. To show that the optimized algorithms are not over-trained to the CC-PEV features calibrated by cropping by 4 × 4 pixels, we report the P_{E} error obtained from a Gaussian SVM-based steganalyzer utilizing the CDF set. Similar performance results were obtained using the CC-PEV feature set with calibration by cropping by 2 × 4 pixels, which suggests that the algorithms are not over-trained to a specific feature set. Unfortunately, the algorithm optimized w.r.t. both inter- and intra-block parts did not achieve a better performance than the algorithm with $\boldsymbol{\theta}_{\mathrm{ia}} = 0$, which is just a special case. This is due to the fact that the Nelder–Mead algorithm converged to a local minimum (the `L2R_L2LOSS` criterion was smaller for the case with $\boldsymbol{\theta}_{\mathrm{ia}} = 0$). When compared with the non-adaptive nsF5 algorithm, both versions increased the payload for the same level of security more than twice. All algorithms can be implemented using the multi-layered STCs[8] in practice. Figure 4 shows that the loss introduced by such a practical implementation is small when implemented using STCs with constraint height $h = 10$.

We found out experimentally that it is more effective to optimize the cost functions w.r.t. larger payloads. Methods optimized for smaller payloads, such as 0.1 bpac, did not achieve as high performance for higher payloads as methods optimized for larger payloads.

6. CONCLUSION

Minimal-distortion steganography is a general principle for building embedding schemes for empirical cover sources, such as digital media, for which the embedding cannot be designed to preserve the cover source distribution simply because epistemiological arguments can be made that such a distribution may not even exist. The basic premise behind steganography designed to embed while minimizing a certain distortion function is that the distortion is related to statistical detectability. In the past, steganographers used heuristically defined distortion functions and focused on the problem of embedding with minimal distortion while no attempt was made to justify the choice of the distortion function or optimize its design. Since the problem of embedding with minimal distortion has been resolved in a near-optimal fashion using clever coding methods, what remains to be done and where the biggest gain in steganographic security lies is the form of the distortion function.

The main contribution of this paper is a practical methodology using which one can optimize the distortion to design steganographic schemes with improved security. We do so by representing images in a feature space in which we define a criterion evaluating the separability between the sets of cover and stego features. The distortion function is parametrized and the parameters are found by optimizing them w.r.t. the chosen criterion on a set that is relatively small – 80 cover and stego images. The result is validated on various cover sources using blind steganalyzers. We intentionally use steganalyzers that utilize different feature spaces than the one in which we optimize to demonstrate that our optimized design generalizes to other feature sets as well cover sources.

We work with additive distortion functions that can be written as a sum of costs defined for each pixel, while each pixel cost depends on neighboring cover pixels. After investigating three different choices for the criterion, we selected the margin of a linear SVM as the most suitable one that is computationally efficient yet still closely tied to detectability as determined by a binary classifier trained on a large set of images.

The merit of the proposed work is demonstrated by incorporating the optimized cost for the ±1 embedding operation in the spatial domain and the ±1 operation for the DCT domain. The improvement over current state of the art is especially apparent in the DCT domain where the methods with optimized costs can embed more than twice as large payloads for the same detectability as the nsF5 algorithm. The costs are robust in the sense that the improvement can be observed even when the new method is tested with steganalyzers using a different feature set and even on a slightly different cover source.

Without any doubts, better parametric models for the distortion in the DCT domain can and should be considered. For example, the cost parameters should be dependent on the spatial frequency of DCT coefficients. This would substantially increase the dimensionality of the parameter space which would need to be balanced out by a corresponding increase of the number images. This appears to be a mere issue of increased complexity rather than one that would render our approach inapplicable and we might consider it in our future work. Embedding simulators used in this paper can be downloaded from `http://dde.binghamton.edu/download/stego_design/`.

ACKNOWLEDGMENTS

The work on this paper was supported by Air Force Office of Scientific Research under the research grant number FA9550-08-1-0084. 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.

REFERENCES

1. R. Anderson. Stretching the limits of steganography. In R. J. Anderson, editor, *Information Hiding, 1st International Workshop*, volume 1174 of Lect. Notes in Computer Sc., pages 39–48, Cambridge, UK, May 30–June 1, 1996. Springer-Verlag, Berlin.
2. P. Bas and T. Furon. BOWS-2. `http://bows2.gipsa-lab.inpg.fr/BOWS2OrigEp3.tgz`, July 2007.
3. R. Böhme. *Advanced statistical steganalysis.* Springer-Verlag, Heidleberg, 2010.

4. C. Cachin. An information-theoretic model for steganography. In D. Aucsmith, editor, *Information Hiding, 2nd International Workshop*, volume 1525 of Lect. Notes in Computer Sc., pages 306–318, Portland, OR, April 14–17, 1998. Springer-Verlag, New York.
5. Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
6. R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, (9):1871–1874, 2008. Software available at `http://www.csie.ntu.edu.tw/~cjlin/liblinear`.
7. T. Filler and J. Fridrich. Gibbs construction in steganography. *IEEE Trans. on Information Forensics and Security*, 5(4):705–720, December 2010.
8. T. Filler, J. Judas, and J. Fridrich. Minimizing additive distortion in steganography using Syndrome-Trellis Codes. *IEEE Trans. on Information Forensics and Security*, 2010. Submitted. See `http://dde.binghamton.edu/filler/publications.php`.
9. T. Filler, A. D. Ker, and J. Fridrich. The Square Root Law of steganographic capacity for Markov covers. In *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XI*, volume 7254, pages 08 1–08 11, San Jose, CA, January 18–21, 2009.
10. J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications.* Cambridge University Press, 2009.
11. J. Fridrich, M. Goljan, and D. Soukal. Perturbed quantization steganography. *ACM Multimedia System Journal*, 11(2):98–107, 2005.
12. J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities. In *Proceedings of the 9th ACM Multimedia & Security Workshop*, pages 3–14, Dallas, TX, September 20–21, 2007.
13. K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, (5):73–99, 2004.
14. A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 513–520. MIT Press, Cambridge, MA, 2007.
15. T. Hastie, Tibshirani R., and Friedman J. *The elements of statistical learning: data mining, inference, and prediction.* Springer-Verlag, Heidleberg, 2nd edition edition, 2009.
16. C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. K., and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 408–415. ACM, 2008.
17. A. D. Ker. Batch steganography and pooled steganalysis. In *Information Hiding, 8th International Workshop*, volume 4437 of Lect. Notes in Computer Sc., pages 265–281, Alexandria, VA, July 10–12, 2006.
18. Y. Kim, Z. Duric, and D. Richards. Modified matrix encoding technique for minimal distortion steganography. In *Information Hiding, 8th International Workshop*, volume 4437 of Lect. Notes in Computer Sc., pages 314–327, Alexandria, VA, July 10–12, 2006.
19. J. Kodovský and J. Fridrich. On completeness of feature spaces in blind steganalysis. In *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 123–132, Oxford, UK, September 22–23, 2008.
20. 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.
21. J. Kodovský, T. Pevný, and J. Fridrich. Modern steganalysis can detect YASS. In *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XII*, volume 7541, pages 02–01–02–11, January 17–21, 2010.
22. J. Nocedal and S. Wright. *Numerical Optimization.* Springer, 2nd edition edition, 2006.
23. T. Pevný, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Trans. on Information Forensics and Security*, 5(2):215–224, 2010.

24. T. Pevný, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. In *Information Hiding, 12th International Conference*, Lect. Notes in Computer Sc., Calgary, Alberta, Canada, June 28–30 2010.
25. T. Pevný and J. Fridrich. Benchmarking for steganography. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of Lect. Notes in Computer Sc., pages 251–267, Santa Barbara, CA, June 19–21, 2008. Springer-Verlag, New York.
26. B.Y. Ryabko and D.B. Ryabko. Asymptotically optimal perfect steganographic systems. *Problems of Information Transmission*, 45(2):184–190, 2009.
27. V. Sachnev, H. J. Kim, and R. Zhang. Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding. In *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 131–140, Princeton, NJ, Sept. 2009.
28. P. Sallee. Model-based steganography. In T. Kalker, I. J. Cox, and Y. Man Ro, editors, *Digital Watermarking, 2nd International Workshop*, volume 2939 of Lect. Notes in Computer Sc., pages 154–167, Seoul, Korea, October 20–22, 2003. Springer-Verlag, New York.
29. C. Wang, X Li, B. Yang, X. Lu, and C. Liu. A content-adaptive approach for reducing embedding impact in steganography. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, pages 1762–1765, March 2010.
30. Y. Wang and P. Moulin. Perfectly secure steganography: Capacity, error exponents, and code constructions. *IEEE Transactions on Information Theory, Special Issue on Security*, 55(6):2706–2722, June 2008.