Optimizing Pixel Predictors for Steganalysis

Vojtěch Holub and Jessica Fridrich

Department of ECE, SUNY Binghamton, NY, USA {vholub1,fridrich}@binghamton.edu

ABSTRACT

A standard way to design steganalysis features for digital images is to choose a pixel predictor, use it to compute a noise residual, and then form joint statistics of neighboring residual samples (co-occurrence matrices). This paper proposes a general data-driven approach to optimizing predictors for steganalysis. First, a local pixel predictor is parametrized and then its parameters are determined by solving an optimization problem for a given sample of cover and stego images and a given cover source. Our research shows that predictors optimized to detect a specific case of steganography may be vastly different than predictors optimized for the cover source only. The results indicate that optimized predictors may improve steganalysis by a rather non-negligible margin. Furthermore, we construct the predictors sequentially – having optimized k predictors, design the $k+1$st one with respect to the combined feature set built from all k predictors. In other words, given a feature space (image model) extend (diversify) the model in a selected direction (functional form of the predictor) in a way that maximally boosts detection accuracy.

Keywords: Steganalysis, steganography, pixel predictor, optimization, classifier.

1. INTRODUCTION

Steganalysis features for digital images represented in the spatial domain are typically computed as joint or conditional probabilities of adjacent samples from a noise residual obtained using a pixel predictor. The purpose of working with the noise residual is to increase the SNR between the stego signal and the original image by suppressing the image content and to narrow the dynamic range of the resulting signal to allow its description using higher-order co-occurrence matrices. Even the early steganalysis algorithms, then called blind detectors, utilized predictors. The very first feature-based steganalyzer proposed in 2000 by Avcibas et al.[1] employed image quality measures whose values are largely dependent on the image noise component computed by subtracting from the image its low-pass filtered version (here, interpreted as a cover prediction). Farid[7] used a shift-invariant linear predictor of wavelet coefficients and formed the features as higher-order moments of marginals of the predictor error. Here, the predictor was chosen to minimize the mean square prediction error. Higher-order moments of noise residual obtained using a denoising filter were used as features by Holotyak[18] and Goljan[16] in WAM. The SPAM feature vector[26] as well as the features proposed in Ref. 30 were formed as Markov transition probabilities of differences between neighboring pixels, which are noise residuals obtained using a very simple predictor – the value of its immediate neighbor. Recently, the authors of Refs. 14, 15, 17 pointed out the importance of forming features from a wider class of noise residuals computed using many different pixel predictors that employed mostly local polynomial models. Pixel predictor is also the cornerstone of the quantitative weighted-stego LSB detector.[2, 6, 12, 19]

It is thus only natural to ask whether detection performance can be improved and by how much by optimizing the predictor within a given detection framework. To this end, we need to narrow down the set of predictors within which the optimization will operate. One possibility is to use off-the-shelf denoising filters and optimize w.r.t. their parameters, such as the variance of the Gaussian noise being removed. Such predictors, however, put great weight on the central pixel being predicted, which leads to predictions biased by the stego signal. The subsequent subtraction of the stego image when forming the residual thus undesirably suppresses the stego signal. The prediction should only utilize the immediate neighborhood excluding the pixel being predicted.

A tempting idea is to fit a Markov random field model[29] to a given cover source and use as predictors the local characteristics, which are conditional probabilities of pixel values given their neighborhood. A simpler approach would be to minimize the prediction error on covers when measured in some appropriate manner, such as in the least square sense. However, predictors built only by considering the cover source and not the

embedding may not perform well for detecting steganography, which is a binary decision problem rather than cover source modeling. Indeed, one could conceivably create an embedding method tailored to be undetectable (or only weakly detectable) in a given feature space, for example using the concept of feature correction[5,20] or the paradigm introduced in Ref. 8 by suitably defining the distortion function. Optimal predictor will thus surely be a function of both the cover source and the embedding algorithm.*

In this paper we restrict ourselves to particularly simple predictors in the form of a shift-invariant linear filter. The predictor will be applied globally to all images and will thus be dependent on a given cover source but not on the individual images. By parametrizing the predictor kernel,† we determine such values of the parameters that give the most accurate detection for a given cover source, steganography method, and detection framework. We are interested in how much the detection can be improved over previously proposed constructs, such as kernels designed to minimize the square prediction error and the heuristically derived predictors introduced in Ref. 14, 15. Improved pixel predictors will lead to more accurate steganalysis for a given feature dimensionality and will enable construction of more compact rich models obtained by merging several diverse feature sets.[13] To this end, we study "conditional design" of the predictors to increase the diversity by optimizing the predictor w.r.t. a collection of existing ones.

The authors do not necessarily view the fact that the predictor will be optimized w.r.t. a given source and stego method as a deficiency. Studying steganalysis in a given source may provide useful insight and even improve methods that aspire to be universal as such systems may be built as a collection of steganalyzers designed for selected classes of cover sources supplied with a source classifier. Moreover, we argue that in the past numerous if not all steganalysis features were designed for a specific embedding paradigm and source even though the authors may not have openly stated this fact. For example, the SPAM feature vector[27] was seemingly proposed from a pure cover model but in reality it is driven by a specific case of steganography as well as cover source. The choice of the order of the Markov process as well as the threshold T and the local predictor were driven by observing the detection performance on ± 1 embedding on a fixed database of images. Had the authors used HUGO[28] instead of ± 1 embedding or larger images, different parameters of the SPAM feature would have been selected. In particular, in light of the recent work,[14,15,17] the predictor would have probably used second- or third-order differences instead of the first-order difference and the order of the Markov chain model might have been four instead of three to leverage longer-range pixel dependencies.

The paper has the following structure. In Section 2, we describe the steganalysis features used in this paper as well as the performance criterion used to evaluate detection accuracy. This same section also explains the parametrization of the predictor kernels and a method for optimizing the detection performance w.r.t. to the kernel parameters. The first set of results appear in Section 3, where we report the detection performance of kernels optimized w.r.t. four different cover sources and three steganographic algorithms. In Section 4, we present and interpret the results of kernel optimization w.r.t. a collection of existing predictors. The paper is concluded in Section 5 with a summary of the achievements and plans for future research together with a discussion about the limitations and possible applications of the proposed predictor optimization method.

We will use boldface symbols for vectors and matrices, such as $\mathbf{x}, \mathbf{y} \in \{0, \dots, 255\}^{n_1 \times n_2}$ for an 8-bit grayscale cover (stego) image with $n_1 \times n_2$ pixels. A linear predictor will be described by a kernel $\mathbf{K} = (K_{ij}) \in \mathbb{R}^{k_1 \times k_2}$, while the image of predicted pixel values is $\hat{\mathbf{x}} = \mathbf{K} \star \mathbf{x}$, where the star denotes convolution. All kernels will be normalized to $\sum_{ij} K_{ij} = -1$ so that the noise residual $\mathbf{r} = \mathbf{x} - \hat{\mathbf{x}}$ is the result of a high-pass filter applied to $\mathbf{x}$.

2. METHODOLOGY

In this section, we outline the methodology for optimizing the predictor parameters and evaluating their performance that will be used in all experiments in this paper.

Let us assume that we have available a total of N cover images from a given source and a corresponding set of N stego images. Prior to optimization, we randomly split all cover-stego image pairs into two disjoint sets, $\mathcal{O}$ and $\mathcal{E}$, with $|\mathcal{O}| = N^{\text{opt}}$ and $|\mathcal{E}| = N - N^{\text{opt}}$ pairs used solely for predictor optimization and evaluation,

*The problem of optimizing the predictor for universal steganalysis will not be investigated in this paper also due to the fact that it is unclear how to correctly formulate this difficult problem.

†The terms "predictor" and "kernel" are used in this paper interchangeably.

respectively. The performance of each optimized predictor will be evaluated by reporting the minimum detection error under equal priors

$$P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{1}{2} \left(P_{\mathrm{FA}} + P_{\mathrm{D}}(P_{\mathrm{FA}}) \right) \tag{1}$$

averaged over ten random splits of $\mathcal{E}$ into $|\mathcal{E}| - 1000$ images used for training and 1000 images for testing. Next, we describe in detail the predictor design, which proceeds on $\mathcal{O}$.

2.1 Kernel parametrization

Each prediction kernel is parametrized before optimization. For example,

$$\mathbf{K} = \begin{pmatrix} 0 & 0 & 0 & 0 & b & c & d \\ 0 & 0 & a & 0 & a & 0 & 0 \\ d & c & b & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2}$$

contains four parameters a, b, c, d but the optimization is carried over parameters different from a (the so-called free parameters). Since we always normalize the kernel so that the sum of all elements is -1, the parameter a can be computed from the rest. In (2), there are three free parameters, $\theta^{\mathrm{free}} = (b, c, d)$, while a can be computed from the normalization, and 13 elements of $\mathbf{K}$ are set to zero and will not participate in the optimization. Note that this particular kernel is forced to be symmetrical about the center element.

2.2 Feature vector

Given a noise residual $\mathbf{r} = (r_{ij})$ obtained using a predictor, we will form the steganalysis features as the joint probability distributions of neighboring residual samples. Based on the arguments outlined in Ref. 13, we use four-dimensional co-occurrence matrices formed by groups of four horizontally and vertically adjacent residual samples after they were quantized and truncated to a finite range:

$$r_{ij} \leftarrow \mathrm{round}\left(\mathrm{trunc}_T \left(\frac{r_{ij}}{q} \right) \right), \tag{3}$$

where $\mathrm{trunc}_T(x) = x$ for $x \in [-T, T]$ and $\mathrm{trunc}_T(x) = T\mathrm{sign}(x)$ otherwise, and $q > 0$ is a quantization step. The co-occurrence matrix, $\mathbf{C} = (C_{\mathbf{d}})$, $\mathbf{d} = (d_1, \ldots, d_4) \in \{-T, \ldots, T\}^4$ with $T = 2$, is the sum

$$\mathbf{C}_{\mathbf{d}} = \mathbf{C}^{(\mathrm{h})}_{\mathbf{d}} + \mathbf{C}^{(\mathrm{v})}_{\mathbf{d}}, \tag{4}$$

where

$$\mathbf{C}^{(\mathrm{h})}_{\mathbf{d}} = \{(i,j) | r_{ij} = d_1, r_{ij+1} = d_2, r_{ij+2} = d_3, r_{ij+3} = d_4\}, \tag{5}$$

and $\mathbf{C}^{(\mathrm{v})}_{\mathbf{d}}$ is the vertical co-occurrence matrix defined analogically. We note that the vertical co-occurrence matrix, however, is formed from a residual computed using a transpose of the kernel, $\mathbf{K}^{\mathrm{T}}$.

As in Ref. 13, the dimensionality of the co-occurrence matrix (4) will be reduced by leveraging symmetries of natural images. This will make the features more compact and better populated and will also increase the performance-to-dimensionality ratio. The symmetrization uses the sign-symmetry[‡] as well as the directional symmetry of images by applying the following two rules for all $\mathbf{d} = (d_1, \ldots, d_4)$:

$$\mathbf{C}_{\mathbf{d}} \leftarrow \mathbf{C}_{\mathbf{d}} + \mathbf{C}_{-\mathbf{d}}, \tag{6}$$

$$\mathbf{C}_{\mathbf{d}} \leftarrow \mathbf{C}_{\mathbf{d}} + \mathbf{C}_{\overleftarrow{\mathbf{d}}}, \tag{7}$$

where $\overleftarrow{\mathbf{d}} = (d_4, d_3, d_2, d_1)$. After eliminating duplicates from $\mathbf{C}$ (which had originally $(2T+1)^4 = 625$ elements), only 169 unique elements remain.

[‡]Sign-symmetry means that taking a negative of an image does not change its statistical properties.

2.3 Objective function

For the optimization, we need an objective function that would measure the detection performance. Since the optimization may involve a large number of evaluations, it is important that the objective function be fast. It is equally important that it be sufficiently smooth to avoid being trapped in local minima. Our first choice was to use the `L2R_L2LOSS` criterion, which is the margin of a linear support vector machine as proposed in Ref. 9. The authors reported that as few as 80 pairs of cover and stego images were enough to make the margin well-behaved for multi-parameter optimization. However, using the margin turned out problematic in our case because changing the predictor changes both the distribution of stego and cover features. Since the margin is a geometric quantity, one needs to normalize the distribution of cover features, which is rather difficult for a multivariate distribution.

Consequently, we decided to use as the objective function the total detection error under equal priors (1) averaged over ten splits of $\mathcal{O}$ into random and eqyally sized training and testing sets. To decrease the complexity of evaluating the objective function, we used the ensemble classifier[22] with automatic setting for the number of base learners and the subspace dimensionality. This way, the most time consuming part of evaluating the objective function was computing the feature vector and not the training, whose complexity was rather negligible.

2.4 Optimization method

The predictor will be optimized w.r.t. its free kernel parameters as well as the quantization step q. We will denote the set of all parameters as $\{\theta^{\text{free}}, q\}$. For the optimization, we used the gradient-free Nelder–Mead (N-M) algorithm [25, Chapter 9.5] implemented by Borggaard.[4] One vertex of the initial simplex, $\mathbf{v}^{(0)} = \{\theta^{\text{ini}}, 1.5\}$, was always computed as the kernel with its free parameters set to θ^{ini}, the predictor optimal in the least-square sense estimated from 50 randomly chosen cover images from the training part of $\mathcal{O}$. The remaining vertices of the initial simplex were obtained by stretching each parameter by δ, $\mathbf{v}^{(j)} = \{\ldots, \mathbf{v}^{(0)}_{j-1}, \mathbf{v}^{(0)}_{j}(1+\delta), \mathbf{v}^{(0)}_{j+1}, \ldots,\}$, $j = 1, \ldots, |\theta^{\text{ini}}| + 1$. Thus, a larger initial simplex can be obtained by increasing δ. In our experiments, we set $\delta = 0.3$.

The iterations stop when the difference between the minimal and maximal value on the simplex is below a certain tolerance $\epsilon = 10^{-6}$ or when the total number of iterations reaches 300.

Since the complexity of evaluating the objective function is linear in N^{opt}, to speed up the optimization, N^{opt} should be as small as possible. There is, however, a trade-off between speed and the smoothness of the objective function. Low values of N^{opt} would lead to a non-smooth objective function, which would increase the chances of getting trapped in local minima of the detection error P_{E}, requiring either a restart of the N-M algorithm or too many iterations to converge. According to our experience, it was in the end more efficient to use a higher value of $N^{\text{opt}} = 2000$. Figure 1 shows the detection error (1) when optimizing a 3×3 rotationally symmetrical kernel with one free kernel parameter, $\theta^{\text{free}} = \{b\}$:

$$\mathbf{K} = \begin{pmatrix} b & a & b \\ a & 0 & a \\ b & a & b \end{pmatrix} \tag{8}$$

for $N^{\text{opt}} = 500$ (left) and $N^{\text{opt}} = 2000$ (right). In this particular case, the source was the BOSSbase database ver. 0.92[10] with 9,074 cover images of size 512×512. It can be clearly seen that the surface of the right plot of the objective function is smoother.

3. OPTIMIZING W.R.T. SOURCE AND STEGO METHOD

Our initial set of experiments aims at optimizing a simple predictor operating on the local 3×3 neighborhood with structure shown in (8). Our goal is to investigate how the optimal kernel parameter and the quantization step depend on the type of the cover source, the stego method, and even the steganography payload.

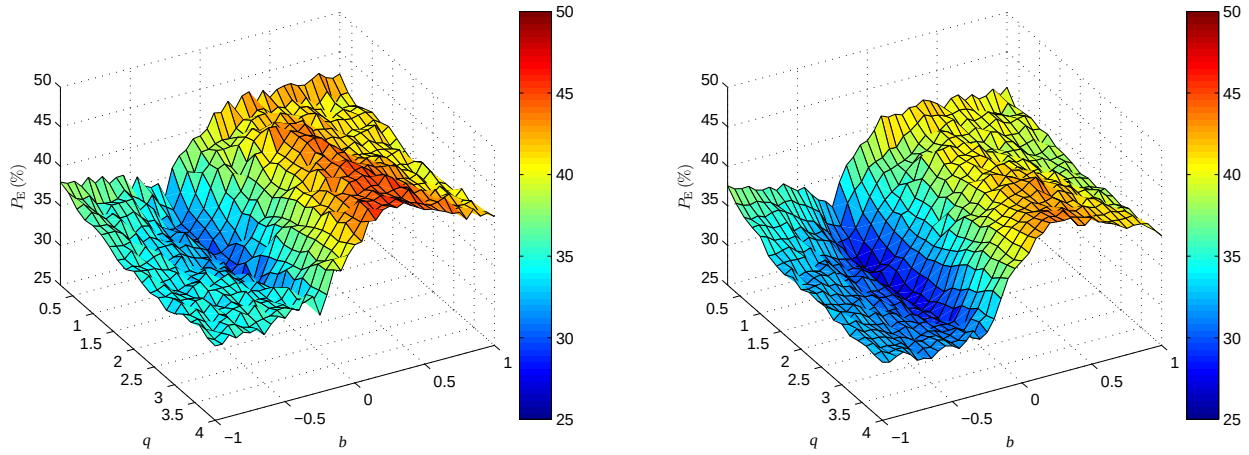

Figure 1. Detection error P_{E} as a function of one free kernel parameter (for kernel (8)) and the quantization step q for two sizes of the set $\mathcal{O}$: $N^{\mathrm{opt}} = 500$ left and $N^{\mathrm{opt}} = 2000$ right.

3.1 Cover sources

The experiments will be done on four different cover sources all containing grayscale 512×512 images. The first three are the BOSSbase ver. 0.92, NRCS512, and LEICA512, which contain raw, uncompressed images. The fourth database was obtained by JPEG compressing BOSSbase database with quality factor 80.[§] BOSSbase contains images originally taken with seven different digital cameras in the raw format and then converted to grayscale and resized so that the smaller side was 512 pixels long and center-cropped to 512×512. The NRCS512 database was derived from the NRCS database of $3,322$ raw scans of negatives coming from the USDA Natural Resources Conservation Service (`http://photogallery.nrcs.usda.gov`). Two 512×512 images were obtained by cropping the central 512×1024 part of each NRCS image, splitting it in two, and converting each image to grayscale. Thus, the NRCS512 image set contained a total of $2 \times 3322 = 6644$ images. All images from the third database, LEICA512, were taken by Leica M9 in their native resolution of 18 megapixels in the raw DNG format. They were then converted to grayscale and central-cropped to the size of 512×512 pixels. All conversion to grayscale and resizing was carried out using the script 'convert' available from the BOSS web site.[10] The JPEG compression was done in Matlab R2011b using the command 'imwrite'.

3.2 Steganographic methods

Three steganographic algorithms with contrasting embedding mechanisms will be considered. The first is the simple non-adaptive ± 1 embedding (also called LSB Matching), which we abbreviate here as PM1. We assume that the algorithm is implemented with ternary matrix embedding that is optimally coded to minimize the number of embedding modifications. In particular, the relative payload α bpp (bits per pixel) can be embedded with change rate $H_3^{-1}(\alpha)$, where $H_3^{-1}(x)$ is the inverse of the ternary entropy function $H_3(x) = -x \log_2 x - (1-x) \log_2 (1-x) + x$. (For more details, see, e.g., Chapter 8 in Ref. 11.) The second algorithm is HUGO,[28] which was designed to minimize embedding distortion in a high-dimensional feature space computed from differences of four neighboring pixels. We used the embedding simulator available from the BOSS website[10] with $\sigma = 1$ and $\gamma = 1$, for the parameters of the distortion function, and the switch –T 255, which means that the distortion function was computed with threshold $T = 255$ instead of the default value $T = 90$ used in the BOSS challenge.[10] We did it to remove a weakness of HUGO with $T = 90$ that makes the algorithm vulnerable to first-order attacks due to an artifact present in the histogram of pixel differences.[23] The third, Edge-Adaptive (EA) algorithm, due to Luo et al.[24] confines the embedding changes to pixel pairs whose difference in absolute value is as large as possible (e.g., around edges). Both HUGO and the EA algorithm are adaptive and place the embedding changes to those parts of the image that are hard to model, which distinguishes them from the non-adaptive ± 1 embedding.

[§]The images were always decompressed to the spatial domain before embedding.

Alg.	Pld.	Kernel	BOSSbase $(a,b),\ q$	BOSSbase P_E	NRCS512 $(a,b),\ q$	NRCS512 P_E	LEICA512 $(a,b),\ q$	LEICA512 P_E
HUGO	0.1	KB	(0.50, -0.25), 1.00	43.90	(0.50, -0.25), 2.00	48.62	(0.50, -0.25), 1.75	38.13
		LSE	(0.45, -0.20), 2.00	44.31	(0.51, -0.26), 1.75	48.90	(0.48, -0.23), 1.50	38.43
		Opt	(0.49, -0.24), 2.00	43.78	(0.60, -0.35), 1.69	48.86	(0.57, -0.32), 1.52	36.54
	0.4	KB	(0.50, -0.25), 1.00	26.37	(0.50, -0.25), 1.00	43.95	(0.50, -0.25), 1.75	13.58
		LSE	(0.45, -0.20), 1.50	27.65	(0.51, -0.26), 2.00	43.91	(0.48, -0.23), 1.50	13.35
		Opt	(0.51, -0.26), 1.58	26.49	(0.37, -0.12), 2.37	43.50	(0.38, -0.13), 1.98	12.07
EA	0.1	KB	(0.50, -0.25), 2.00	37.85	(0.50, -0.25), 2.00	47.66	(0.50, -0.25), 2.00	24.77
		LSE	(0.45, -0.20), 2.00	35.64	(0.51, -0.26), 1.75	47.66	(0.48, -0.23), 2.00	23.94
		Opt	(0.46, -0.21), 1.91	35.42	(0.67, -0.42), 1.84	47.36	(0.37, -0.12), 2.34	17.96
	0.4	KB	(0.50, -0.25), 1.75	17.93	(0.50, -0.25), 1.00	39.56	(0.50, -0.25), 1.75	4.62
		LSE	(0.45, -0.20), 1.75	16.00	(0.51, -0.26), 1.50	39.48	(0.48, -0.23), 2.00	4.30
		Opt	(0.26, -0.01), 1.92	13.74	(0.39, -0.14), 1.58	37.06	(0.40, -0.15), 2.09	3.52
PM1	0.1	KB	(0.50, -0.25), 1.00	31.05	(0.50, -0.25), 1.00	47.82	(0.50, -0.25), 1.00	36.89
		LSE	(0.45, -0.20), 1.00	32.56	(0.51, -0.26), 1.50	48.54	(0.48, -0.23), 1.50	38.19
		Opt	(0.55, -0.30), 0.58	31.42	(0.67, -0.42), 0.72	47.41	(0.56, -0.31), 0.93	37.11
	0.4	KB	(0.50, -0.25), 1.00	12.50	(0.50, -0.25), 1.00	40.52	(0.50, -0.25), 1.00	10.49
		LSE	(0.45, -0.20), 1.00	13.66	(0.51, -0.26), 1.00	41.99	(0.48, -0.23), 1.50	11.09
		Opt	(0.52, -0.27), 1.03	12.48	(0.73, -0.48), 0.55	39.70	(0.32, -0.07), 1.27	8.28

Table 1. Optimized kernel parameters, a, b, and the quantization step q, together with the average testing error $P_{\rm E}$ for three stego methods, two payloads, and three databases of raw images.

3.3 Experiments on raw images

The results of experiments on raw images are shown in Table 1. The optimization algorithm was run as described in Section 3 with $N^{\rm opt} = 2000$ cover and stego images for optimizing the predictor. The remaining images from each image source were all used for testing. To investigate the effect of the message length, we repeated the experiments for two payload sizes – 0.1 and 0.4 bpp. The tables show the values of the kernel parameters a and b in (8) as well as the quantization step q. The rows with 'KB' show the average testing error (1) on $\mathcal{E}$ with the Ker–Böhme kernel,

$$\mathrm{KB} = \begin{pmatrix} -0.25 & 0.5 & -0.25 \\ 0.5 & 0 & 0.5 \\ -0.25 & 0.5 & -0.25 \end{pmatrix}, \tag{9}$$

derived in Ref. 3 when optimized over q, 'LSE' is the least-square kernel fit to covers again optimized over q, and 'Opt' denotes optimization over both the kernel and q. Shaded cells highlight interesting cases.

Note that the optimized kernel for BOSSbase is almost always rather close to the LSE kernel (the kernel that minimized the square prediction error on covers), which also coincides with the KB kernel (9). The improvement for HUGO and ± 1 embedding is thus solely due to optimizing over q rather than the kernel. The biggest improvement is observed for the EA algorithm for which the optimal kernel parameter is very different from the KB or LSE kernels – the corner parameter is almost zero, making the predictor support constrained to the four-pixel "cross" surrounding the central pixel. This can be explained by inspecting the embedding mechanism that hides message bits only in horizontal/vertical pixel pairs whose absolute difference is above a threshold determined beforehand by the payload size and the statistics of differences of each cover image. Another interesting observation is that the optimal quantization step for both adaptive methods is high, while it is small for the non-adaptive PM1. This is understandable since the adaptive methods embed in those regions of the image where the residual is large. Quantizing with a larger q moves some of the residual samples from its marginal back to the interior of the co-occurrence. In contrast, since the changes for PM1 embedding are not concentrated in edges or textures, there is no need to quantize strongly. In fact, the optimal q for small payload was $q = 0.58$. Overall, the improvement in detection error over using the KB and LSE predictors can be as large as $3 - 6\%$ in some cases.

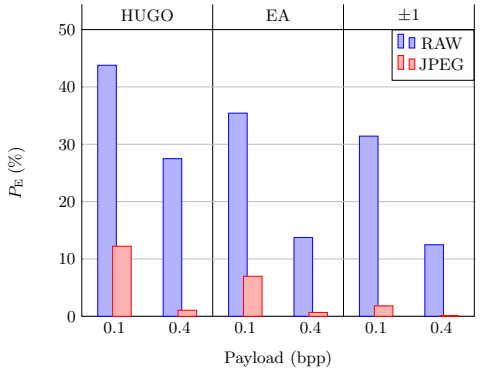

Figure 2. Average testing detection error P_E for RAW images and decompressed JPEGs (QF 80) from BOSSbase for three algorithms and two payloads.

For the NRCS512 database, the optimal predictors vary wildly across the two payloads (including the quantization step). Since the detection in this source is overall very unreliable due to the extreme noisiness of the scans, the optimized kernels most likely have no particular significance as they are likely affected too much by the employed machine learning and the noisiness of the objective function.

As expected, the LEICA512 source is the easiest for steganalysis due to strong correlations among neighboring pixels. In shaded cells, the optimized kernel is very different from the KB and LSE kernels and the improvement can be very significant (e.g., for EA at 0.1 bpp where the detection error improved by almost 8%). It is rather interesting to contrast the optimal kernel with Eq. (6.17) from Ref. 3 stating that the parameters of the LSE kernel, which is there recommended for weighted-stego steganalysis, should satisfy $-a/b = 1/(2\rho)$, where ρ is the correlation among neighboring pixels. Since this correlation is much stronger in LEICA512 than in BOSSbase or NRCS512, one would expect the ratio $-a/b$ to decrease and not increase. We interpret this as yet another example that optimizing the predictor for a binary detection problem is different than for source modeling. It is entirely possible, though, that our conclusions are due to the entire detection framework we use and if the residual was differently utilized, we may have ended up with a different optimal kernel.

Finally, as expected, HUGO is the least detectable algorithm out of the three, while EA is surprisingly less secure than the simple PM1 embedding in NRCS512 and LEICA512 for both the small as well as the large payload. The detection accuracy strongly depends on the cover source, which is to be expected.

3.4 Experiments on JPEG decompressed images

The experiment from the previous section was carried out in exactly the same manner on BOSSbase ver. 0.92 JPEG compressed with quality factor 80 using the 'imwrite' command in Matlab. The results, which are displayed in Figure 2, show that feature-based classifiers can detect steganography in decompressed JPEGs significantly more reliably than in raw, uncompressed images. For example, for PM1 embedding in BOSSbase at payload 0.1, which translates to change rate $0.0112 \approx 1/90$ with an optimal ternary coder, using optimized kernel, we obtain the detection error of 1.82%, down from 31.42% for the corresponding source of raw images. We see two reasons for this shockingly better accuracy. First, decompressed JPEGs are smoother due to the low-pass effect of lossy compression. Second, realize that the original BOSSbase is a mixture of seven different sources, which increases the spread of the features and complicates the decision boundary. JPEG compression, on the other hand, homogenizes the cover source and decreases the spread of cover features, enabling thus much more reliable detection. Because the accuracy of detection of PM1 embedding is expected to be basically the same as for the LSB embedding (since our features are "parity-unaware"), our feature-based detector likely outperforms in this particular source the best structural LSB detectors published in the literature that do not use JPEG compatibility, which, according to the best knowledge of the authors, never happened before. The results can be

Optimization on $\mathcal{O}$			Evaluation on $\mathcal{E}$		
Structure	$P_{\mathrm{E}}^{\mathrm{ini}} \to P_{\mathrm{E}}^{\mathrm{opt}}$	Optimized predictor	$P_{\mathrm{E}}^{\mathrm{indiv}}$	$P_{\mathrm{E}}^{\mathrm{merged}}$	Dim
$(\ a \quad 0 \quad a\ )$	$29 \to 29$	$(\ 0.5 \quad 0 \quad 0.5\ ), 1.95$	28.76	28.76	169
$(\ a \quad 0 \quad b\ )$	$28 \to 26$	$(\ 0.048 \quad 0 \quad -0.952\ ), 0.93$	30.04	25.09	338

Table 2. Complementing the second-order linear predictor by allowing an asymmetric kernel. Kernel orientation and co-occurrence scan are parallel.

further vastly improved by using more complex feature sets instead of the simple 169-dimensional symmetrized co-occurrence from one type of residual. Investigation of this fascinating new finding is left for a different paper as this topic is clearly outside of the scope reserved for this paper.

Besides the exciting finding above, however, the predictor optimization for this source is not significant, which is why we do not report any detailed results here. The performance improvement of optimized kernels is mostly due to the quantization step instead of the kernel. This is possibly because JPEG compression makes the cover sources more homogeneous. Interestingly, however, for decompressed JPEG covers the optimal kernel was close to the KB kernel even for the EA algorithm. A quick inspection of the influence of individual co-occurrence bins revealed that JPEG compression almost empties particular co-occurrence bins which are later filled up again by embedding. The optimization seems to leverage this artifact of covers instead of the peculiarities of the embedding operation.

4. CONDITIONAL OPTIMIZATION

The accuracy of feature-based steganalysis can be significantly improved by forming the features from multiple different predictors[14,15,17,21,22] as each predictor captures a different type of relationship among neighboring residual samples. This approach is recognized as steganalysis using rich models.[13] Merging features whose performance is correlated, albeit strong, is, however, not as effective as when one combines diverse features that are not necessarily as strong when used individually. Thus, having optimized a certain kernel structure, it makes sense to optimize the next predictor with respect to the existing one. In fact, one can imagine building the entire rich model in this manner.

We next investigate the possibility of "cascading" the predictor design by optimizing the next predictor w.r.t. an existing set of predictors. We start with some simple small-scale experiments that already reveal quite interesting facts and then scale up the approach. All experiments in this section are performed on the BOSSbase cover source and (unless mentioned otherwise) HUGO at payload 0.4 bpp as the stego source.

4.1 Complementing the 2nd-order predictor

It has been pointed out in Refs. 14,15,17 that second-order residuals obtained using the kernel $\mathbf{K} = (\ 0.5 \quad 0 \quad 0.5\ )$ are highly effective against HUGO.[¶] This predictor essentially assumes that the image is locally linear in the horizontal direction.

Running our optimization w.r.t. the quantization step only, we determined $q = 1.95$ as the one minimizing the detection error. The next question we asked was which 1×3 kernel with structure $(\ a \quad 0 \quad b\ )$ optimally supplements the second-order predictor. The optimization discovered that the best option is to use the first-order differences with quantization step $q = 0.93$, which is essentially the residual used in the SPAM feature vector!

The optimization results are displayed in Table 2. We use similar tables to report the results of other experiments in this section. The first column shows the structure of the kernels for optimization (recall that we do not optimize over a). The second column shows the value of the objective function (see its definition in Section 2.3) at the initial point $\mathbf{v}^{(0)}$ of the simplex, $P_{\mathrm{E}}^{\mathrm{ini}}$, and the final value, $P_{\mathrm{E}}^{\mathrm{opt}}$, after the optimization ends. The third column contains the final optimized predictor. The fourth and fifth columns hold the average detection

[¶]This is because HUGO preserves the joint pdf of pixel differences but not second-order differences among pixels.

Optimization on $\mathcal{O}$			Evaluation on $\mathcal{E}$		
Structure	$P_{\mathrm{E}}^{\mathrm{ini}} \to P_{\mathrm{E}}^{\mathrm{opt}}$	Optimized predictor	$P_{\mathrm{E}}^{\mathrm{indiv}}$	$P_{\mathrm{E}}^{\mathrm{merged}}$	Dim
$\begin{pmatrix} a \\ 0 \\ a \end{pmatrix}$	$26 \to 26$	$\begin{pmatrix} 0.5 \\ 0 \\ 0.5 \end{pmatrix}, 1.95$	25.45	25.45	169
$\begin{pmatrix} a \\ 0 \\ b \end{pmatrix}$	$25 \to 24$	$\begin{pmatrix} 0.205 \\ 0 \\ 0.795 \end{pmatrix}, 1.06$	31.76	23.68	338

Table 3. Optimizing the second-order linear predictor by allowing an asymmetrical kernel. Kernel orientation is perpendicular to the co-occurrence scan.

error on $\mathcal{E}$ when only the optimized kernel is used individually, $P_{\mathrm{E}}^{\mathrm{indiv}}$, and after merging the kernels from all rows above, $P_{\mathrm{E}}^{\mathrm{merged}}$. Finally, the last column shows the dimensionality after merging the features.

Notice that the individual performance of the second predictor is not very high and there certainly exist other kernels and quantization steps that would give higher individual performance. However, when considered *jointly* with the first predictor, adding these 169 features decreases the error from almost 29% to about 25%.

We repeated the same experiment with kernels oriented perpendicularly to the scan of the co-occurrence. Several interesting phenomena are apparent in Table 3 that shows the results. First, forming the co-occurrence in a perpendicular direction to the kernel orientation leads to better detection of HUGO. Our intuitive understanding of this, confirmed by many experiments,[13] is that the larger is the support of the kernel combined with the co-occurrence matrix, the better. For example, the horizontal kernel $(\ 0.5 \quad 0 \quad 0.5\ )$ combined with 4th-order horizontal co-occurrence has a support of 6 pixels, while the vertical kernel $(\ 0.5 \quad 0 \quad 0.5\ )^T$ combined with the same co-occurrence matrix has a support of 12 pixels. Second, the best kernel is no longer the first-order difference as before. Third, there is an even bigger contrast between the rather poor individual performance of the second predictor and the improvement it provides when merged with the features from the first predictor.

4.2 Cascading the 3×3 kernel (guided)

In the next experiment, we decided to cascade predictors defined on the local 3×3 neighborhood (Table 4). As in the previous section, the design is "guided" by restricting the structure of the kernel at each step. The first kernel identified by the optimization is very close to the KB kernel, which also gives the smallest square prediction error on BOSSbase, and the quantization step is $q \approx 1$. In the second and third steps, we allowed an asymmetric structure for the "central cross." The optimization found essentially a one-dimensional vertical linear predictor and its corresponding horizontal counterpart. For both predictors, the optimal quantization step was $q > 2$, indicating that the predictors are forced to "see" embedding changes in textures and around edges. Note that while the third predictor has a rather weak individual performance ($P_{\mathrm{E}}^{\mathrm{indiv}} = 32.21\%$), it complements the previous two predictors rather well, lowering the testing error by one and a half percent. However, it is obvious that lowering the error further by cascading kernels of the same type becomes increasingly harder. We hypothesize that cascading kernels with the same support is not the most efficient way of improving performance per dimensionality as such kernels cannot be by definition too diverse. Having said this, it is certainly interesting that this iterative design gave a 507-dimensional feature vector with detection error $\sim 20\%$, which is close to the performance of the winning team in the BOSS competition.[15]‖

To show the influence of the embedding algorithm on the resulting optimized kernels, we repeated this experiment using EA instead of HUGO (Table 5). It is mentioned in Section 3.3 that the optimized kernel adapts to a weakness of EA by nullifying the corner coefficients. By inspecting the corner coefficients of the second and the third cascaded kernels, it seems that this weakness is completely utilized by the first kernel. Note that the value of the objective function increases from $P_{\mathrm{E}}^{\mathrm{opt}} = 13$ in the second row to $P_{\mathrm{E}}^{\mathrm{ini}} = 14$ in the third

‖The BOSS score is not directly comparable to our experiment due to cover mismatch that plagued the detection results of participating teams.[14,17]

Optimization on $\mathcal{O}$			Evaluation on $\mathcal{E}$		
Structure	$P_{\mathrm{E}}^{\mathrm{ini}} \to P_{\mathrm{E}}^{\mathrm{opt}}$	Optimized predictor	$P_{\mathrm{E}}^{\mathrm{indiv}}$	$P_{\mathrm{E}}^{\mathrm{merged}}$	Dim
$\begin{pmatrix} b & a & b \\ a & 0 & a \\ b & a & b \end{pmatrix}$	$28 \to 27$	$\begin{pmatrix} -0.259 & 0.509 & -0.259 \\ 0.509 & -1 & 0.509 \\ -0.259 & 0.509 & -0.259 \end{pmatrix}$, 1.58	26.49	26.49	169
$\begin{pmatrix} c & b & c \\ a & 0 & a \\ c & b & c \end{pmatrix}$	$26 \to 22$	$\begin{pmatrix} -0.034 & 0.503 & -0.034 \\ 0.064 & -1 & 0.064 \\ -0.034 & 0.503 & -0.034 \end{pmatrix}$, 2.23	27.22	21.77	338
$\begin{pmatrix} c & b & c \\ a & 0 & a \\ c & b & c \end{pmatrix}$	$22 \to 20$	$\begin{pmatrix} -0.044 & -0.092 & -0.044 \\ 0.682 & -1 & 0.682 \\ -0.044 & -0.09 & -0.044 \end{pmatrix}$, 2.03	32.21	20.25	507

Table 4. Cascading predictors on the local 3×3 neighborhood by guiding the process with predefined kernel symmetries for HUGO.

row. This is caused by random selection of subspaces and bootstraps in the ensemble classifier[22] together with a relatively small size of the set $\mathcal{O}$ and the final rounding to integers.

4.3 Cascading the vertical 5×1 kernel (unguided)

In the last experiment of this section, we investigate an unguided design for a 5×1 kernel that is perpendicular to the co-occurrence scan. By unguided, we mean that the kernel structure at each step was fixed to $(\ a \quad b \quad 0 \quad c \quad d\ )^{\mathrm{T}}$ and thus the optimization was carried over four parameters – three free kernel parameters and the quantization step. The results of the first four steps are shown in Table 6, where for compactness we display the optimal kernels graphically. The kernels seem to form a "basis" of sorts as they try to complement each other. By merging the cascaded features, the detection error is gradually decreasing but eventually exhibits signs of saturation when this process continues (not shown in the table). This is most likely because cascading the same predictor structure does not allow for enough diversity to further lower the error.

Using the conditional optimization for the entire design of a rich model, however, is somewhat problematic when approached the way described in this paper. We observed that when the optimization is run over five or more parameters, the optimal parameter vector becomes frequently trapped in local minima and does not find a better solution (even after restarting from a different initial condition) even when better solutions are known to exist. Moreover, for higher dimensionality of the parameter vector the objective function seems to contain numerous shallow regions where the search randomly wanders around without converging. This is undoubtedly tied to the particular form of the objective function. Our attempts to start with a larger kernel structure, such as a general unconstrained 5×5 kernel, and iterating the optimization did not provide meaningful or particularly good results.

This problem forced us to restrict the structure of the next predictor, in which case the optimization seems to produce interesting interpretable results. However, this approach towards building the rich model would mean heavy involvement of the user, which is undesirable.

5. SUMMARY

Pixel predictors are commonly employed when constructing steganalysis features from noise residuals as co-occurrences of adjacent residual samples. The predictor plays an important role – it is known that combining features computed from residuals obtained using a diverse set of predictors markedly improves detection performance. In this paper, we introduce a method for optimizing the predictor parameters to improve detection performance for a fixed source and stego method within a specific detection framework. The predictor parameters are kernel elements of a linear filter and a quantization step using which the residual is quantized.

On four different cover sources, three spatial-domain steganographic methods, and two payloads, we show the effectiveness of the proposed approach. Among other findings, we observed that the optimal predictor may

Optimization on $\mathcal{O}$			Evaluation on $\mathcal{E}$		
Structure	$P_{\mathrm{E}}^{\mathrm{ini}} \to P_{\mathrm{E}}^{\mathrm{opt}}$	Optimized predictor	$P_{\mathrm{E}}^{\mathrm{indiv}}$	$P_{\mathrm{E}}^{\mathrm{merged}}$	Dim
$\begin{pmatrix} b & a & b \\ a & 0 & a \\ b & a & b \end{pmatrix}$	$18 \to 15$	$\begin{pmatrix} -0.015 & 0.265 & -0.015 \\ 0.265 & -1 & 0.265 \\ -0.015 & 0.265 & -0.015 \end{pmatrix}$, 1.92	14.06	14.06	169
$\begin{pmatrix} c & b & c \\ a & 0 & a \\ c & b & c \end{pmatrix}$	$14 \to 13$	$\begin{pmatrix} -0.267 & 0.428 & -0.267 \\ 0.606 & -1 & 0.606 \\ -0.267 & 0.428 & -0.267 \end{pmatrix}$, 1.50	17.82	12.58	338
$\begin{pmatrix} c & b & c \\ a & 0 & a \\ c & b & c \end{pmatrix}$	$14 \to 13$	$\begin{pmatrix} -0.189 & 0.510 & -0.189 \\ 0.368 & -1 & 0.368 \\ -0.189 & 0.510 & -0.189 \end{pmatrix}$, 1.81	14.93	11.91	507

Table 5. Cascading predictors on the local 3×3 neighborhood by guiding the process with predefined kernel symmetries for EA.

Optimization on $\mathcal{O}$			Evaluation on $\mathcal{E}$		
Structure	$P_{\mathrm{E}}^{\mathrm{ini}} \to P_{\mathrm{E}}^{\mathrm{opt}}$	Optimized predictor	$P_{\mathrm{E}}^{\mathrm{indiv}}$	$P_{\mathrm{E}}^{\mathrm{merged}}$	Dim
$(\ a \quad b \quad 0 \quad c \quad d\ )^{\mathrm{T}}$	$25 \to 23$	 1.71	24.36	24.36	169
$(\ a \quad b \quad 0 \quad c \quad d\ )^{\mathrm{T}}$	$23 \to 22$	 1.69	24.67	23.12	338
$(\ a \quad b \quad 0 \quad c \quad d\ )^{\mathrm{T}}$	$23 \to 21$	 2.48	26.71	21.82	507
$(\ a \quad b \quad 0 \quad c \quad d\ )^{\mathrm{T}}$	$22 \to 21$	 2.31	35.74	20.13	676

Table 6. Cascading predictors on the local 5×1 neighborhood.

strongly depend on the embedding mechanism as well as the cover source. The improvement in detection error ranges from rather small to quite substantial, depending on the source and stego method.

The proposed framework is also applicable to the case when the predictor is optimized w.r.t. a set of existing predictors, which allows "cascading" the predictors to maximize the performance–dimensionality ratio.

According to our experience, the method as proposed in this paper is limited to optimizing over a rather small parameter vector (e.g., up to dimension of five), which is most likely due to the character of the objective function. Search for better behaved objective functions that may remove this limitation is considered as part of the future effort.

The predictor optimization may also be of lesser importance when applied to a rich model consisting of feature sets from potentially hundreds of predictors as the individual feature sets may compensate as a whole for deficiencies of others. However, when the goal is to select a small subset of features with an overall good performance, the optimized predictors are expected to play an important role.

One interesting finding not related to the topic of this paper, which is predictor optimization, is that feature-based steganalysis can be very effective for sources consisting of decompressed JPEG images. It appears that there is potential to outperform structural detectors in such sources by a rather large margin, for example, by making the features parity aware. This topic will be pursued as part of future research.

6. 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. The authors would like to thank Jan Kodovský for many useful discussions.

REFERENCES

1. I. Avcibas, N. D. Memon, and B. Sankur. Steganalysis using image quality metrics. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents III*, volume 4314, pages 523–531, San Jose, CA, January 22–25, 2001.
2. 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.
3. R. Böhme. *Improved Statistical Steganalysis Using Models of Heterogeneous Cover Signals*. PhD thesis, Faculty of Computer Science, Technische Universität Dresden, Germany, 2008.
4. J. Borggaard, 2009, Software available at `http://people.sc.fsu.edu/~jburkardt/m_src/nelder_mead/nelder_mead.html`.
5. V. Chonev and A. D. Ker. Feature restoration and distortion metrics. In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XIII*, volume 7880, pages 0G01–0G14, San Francisco, CA, January 23–26, 2011.
6. R. Cogranne, C. Zitzmann, L. Fillatre, F. Retraint, I. Nikiforov, and P. Cornu. A cover image model for reliable steganalysis. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Workshop*, pages 178–192, 2011.
7. H. Farid and L. Siwei. Detecting hidden messages using higher-order statistics and support vector machines. In F. A. P. Petitcolas, editor, *Information Hiding, 5th International Workshop*, volume 2578 of Lecture Notes in Computer Science, pages 340–354, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.
8. T. Filler and J. Fridrich. Gibbs construction in steganography. *IEEE Transactions on Information Forensics and Security*, 5(4):705–720, 2010.
9. T. Filler and J. Fridrich. Design of adaptive steganographic schemes for digital images. In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XIII*, volume 7880, pages OF 1–14, San Francisco, CA, January 23–26, 2011.
10. T. Filler, T. Pevný, and P. Bas. BOSS (Break Our Steganography System). `http://boss.gipsa-lab.grenoble-inp.fr`, July 2010.
11. J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, 2009.
12. J. Fridrich, M. Goljan, and D. Hogea. New methodology for breaking steganographic techniques for JPEGs. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents V*, volume 5020, pages 143–155, Santa Clara, CA, January 21–24, 2003.
13. J. Fridrich and J. Kodovský. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 2011. Under review.
14. J. Fridrich, J. Kodovský, M. Goljan, and V. Holub. Breaking HUGO – the process discovery. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Workshop*, Lecture Notes in Computer Science, pages 85–101, Prague, Czech Republic, May 18–20, 2011.
15. J. Fridrich, J. Kodovský, M. Goljan, and V. Holub. Steganalysis of content-adaptive steganography in spatial domain. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Workshop*, Lecture Notes in Computer Science, pages 102–117, Prague, Czech Republic, May 18–20, 2011.
16. M. Goljan, J. Fridrich, and T. Holotyak. New blind steganalysis and its implications. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 1–13, San Jose, CA, January 16–19, 2006.
17. G. Gül and F. Kurugollu. A new methodology in steganalysis : Breaking highly undetactable steganograpy (HUGO). In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Workshop*, Lecture Notes in Computer Science, pages 71–84, Prague, Czech Republic, May 18–20, 2011.
18. T. S. Holotyak, J. Fridrich, and D. Soukal. Stochastic approach to secret message length estimation in $\pm k$ embedding 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 673–684, San Jose, CA, January 16–20, 2005.

19. 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.
20. J. Kodovský and J. Fridrich. On completeness of feature spaces in blind steganalysis. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 123–132, Oxford, UK, September 22–23, 2008.
21. J. Kodovský and J. Fridrich. Steganalysis in high dimensions: Fusing classifiers built on random subspaces. In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XIII*, volume 7880, pages OL 1–13, San Francisco, CA, January 23–26, 2011.
22. J. Kodovský, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 2011. To appear.
23. J. Kodovský, J. Fridrich, and V. Holub. On dangers of overtraining steganography to incomplete cover model. In J. Dittmann, S. Craver, and C. Heitzenrater, editors, *Proceedings of the 13th ACM Multimedia & Security Workshop*, pages 69–76, Niagara Falls, NY, September 29–30, 2011.
24. W. Luo, F. Huang, and J. Huang. Edge adaptive image steganography based on LSB matching revisited. *IEEE Transactions on Information Forensics and Security*, 5(2):201–214, June 2010.
25. J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2nd edition edition, 2006.
26. T. Pevný, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 75–84, Princeton, NJ, September 7–8, 2009.
27. T. Pevný, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, 5(2):215–224, June 2010.
28. T. Pevný, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. In R. Böhme and R. Safavi-Naini, editors, *Information Hiding, 12th International Workshop*, volume 6387 of Lecture Notes in Computer Science, pages 161–177, Calgary, Canada, June 28–30, 2010. Springer-Verlag, New York.
29. G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction (Stochastic Modelling and Applied Probability)*. Springer-Verlag, Berlin Heidelberg, 2nd edition, 2003.
30. D. Zo, Y. Q. Shi, W. Su, and G. Xuan. Steganalysis based on Markov model of thresholded prediction-error image. In *Proceedings IEEE, International Conference on Multimedia and Expo*, pages 1365–1368, Toronto, Canada, July 9–12, 2006.