Steganalysis of Content-Adaptive Steganography in Spatial Domain

Jessica Fridrich, Jan Kodovský, Vojtěch Holub, and Miroslav Goljan

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

Abstract. Content-adaptive steganography constrains its embedding changes to those parts of covers that are difficult to model, such as textured or noisy regions. When combined with advanced coding techniques, adaptive steganographic methods can embed rather large payloads with low statistical detectability at least when measured using feature-based steganalyzers trained on a given cover source. The recently proposed steganographic algorithm HUGO is an example of this approach. The goal of this paper is to subject this newly proposed algorithm to analysis, identify features capable of detecting payload embedded using such schemes and obtain a better picture regarding the benefit of adaptive steganography with public selection channels. This work describes the technical details of our attack on HUGO as part of the BOSS challenge.

1 Introduction

Steganalysis is a signal detection problem – the task is to discover the presence of secretly embedded messages in objects, such as digital images or audio files. Since the dimensionality of digital media is typically very large, the detection is always preceded by dimensionality reduction – the objects are represented using a feature vector of a lower dimensionality. Steganalyzers are built in the feature space by training a classifier on a large database of cover and stego objects.

The main goal of this paper is to improve detection of adaptive steganography that makes embedding changes in hard-to-model regions of covers. A recent example of this type of steganography is HUGO [14]. Although this algorithm was designed for images in raster formats, the ideas can be applied to other domains and other media types. What distinguishes HUGO from other algorithms is that it approximately preserves a very high-dimensional feature vector and thus takes into consideration a large number of complex dependencies among neighboring pixels. With the help of advanced syndrome-coding techniques, HUGO embedding was reported undetectable using state-of-the-art steganalyzers even at rather large payloads [14].

It appears that as steganographers turn to feature spaces of very high dimension, steganalysts need to do the same to capture more subtle relationships among individual pixels. This brings about two major problems – how to form good high-dimensional feature sets and how to train classifiers in high dimensions

with a limited number of training examples. To detect content-adaptive embedding, we need better models of local content, which could be achieved simply by adding more features. However, the dimensionality should be increased with care and one needs to make sure the features are *diverse* and *well populated* even in complex/textured regions. We propose to form the features as co-occurrences of image noise residuals obtained from higher-order local models of images.

The second problem presents a formidable challenge because training classifiers in high-dimensions requires a large number of examples to properly generalize to unknown images. However, it is not always easy or even possible for the Warden to obtain a sufficiently large number of examples from a given cover source. Additionally, training Support Vector Machines (SVMs) on a large number of examples in high-dimensional spaces can quickly become computationally prohibitive. To address these issues, we propose *ensemble classifiers* obtained by fusing decisions of base learners trained on random subspaces of the feature space. This machine learning approach is scalable and achieves accuracy comparable to SVMs. Its low complexity and scalability is especially convenient for rapid design and development – an attribute we view as vital for construction of practical steganalyzers as well as for winning steganography competitions.

The HUGO algorithm is described in [14] and a brief description also appears in [1] in this volume. In the next section, we introduce HOLMES – a strategy for constructing a large number of diverse features capable of detecting embedding changes in more complex parts of images. The ensemble classifier is detailed in Section 3, while all experiments are described in Section 4. We experimentally establish HUGO's detectability, compare its security with its non-adaptive ± 1 version, and contrast the performance of HOLMES to previous art. The paper is summarized in Section 5, where we also discuss the implications of our attack on design of future steganographic schemes.

Everywhere in this article, boldface symbols are used for vectors and capital-case boldface symbols for matrices or higher-dimensional arrays. The symbols $\mathbf{X} = (x_{ij}) \in \mathcal{X} = \{0, \dots, 255\}^{n_1 \times n_2}$ and $\mathbf{Y} = (y_{ij}) \in \mathcal{X}$ will always represent pixel values of 8-bit grayscale cover and stego images with $n = n_1 n_2$ pixels.

2 The HOLMES feature set

Spatially-adaptive steganography makes embedding changes in those regions of the cover image that are hard to model, which makes the detection more difficult. On the other hand, the public selection channel could also be a weakness because the Warden can estimate the probability with which each pixel is modified. The authors of this paper were unable to utilize this information to improve their attack.

HUGO approximately preserves the joint statistic of differences between up to four neighboring pixels in four different directions. Thus, a better model is needed that can "see" farther than four pixels. We achieve this by working with higher-order noise residuals obtained by modeling the local content using polynomials.

2.1 Residuals

A popular way to design steganalysis methods is to extract the features not directly from the stego image $\mathbf{Y}$ but from a signal with a more favorable SNR – the image noise residual $\mathbf{R} = (r_{ij})$:

$$r_{ij} = y_{ij} - \text{Pred}(\mathcal{N}(\mathbf{Y}, i, j)), \tag{1}$$

where $\text{Pred}(\mathcal{N}(\mathbf{Y}, i, j))$ is an estimate of the cover image pixel x_{ij} from its neighborhood $\mathcal{N}(\mathbf{Y}, i, j)$.

A tempting option is to implement Pred(.) as a denoising filter. In fact, some previously proposed steganalysis features were designed exactly in this manner. In WAM [7], the predictor is the Wiener filter applied to wavelet coefficients. In [4], a shift-invariant linear predictor was used for an entire subband in a decomposition obtained using quadrature mirror filters. The problem with using denoising filters and linear filters, however, is that they place substantial weight on the central pixel being denoised / predicted. Consequently, the predicted value is generally a *biased* estimate of the cover pixel and the stego signal becomes suppressed in the residual (1). What is really needed for steganalysis is an unbiased estimate of the central pixel obtained from the neighboring pixels, *excluding* the pixel being estimated. The recently proposed SPAM feature set [13], as well as the earlier work [2, 15], use the value of the neighboring pixel as the prediction:

$$\text{Pred}(\mathcal{N}(\mathbf{Y}, i, j)) = y_{i,j+1}. \tag{2}$$

While the noise residual $\mathbf{R}$ is confined to a narrower dynamic range when compared to $\mathbf{Y}$, it remains high-dimensional and cannot be used directly as a feature in machine learning. To reduce its dimensionality, features are usually constructed as some integral quantities. Considering the noise residual as a Markov chain, one can take its sample transition probability matrix [2, 13, 15] or the sample joint probability matrix (the co-occurrence matrix) as a feature. To capture higher-order dependencies among pixels, higher-order co-occurrence matrices are usually formed. However, the number of elements in 2D and 3D matrices rapidly increases and the bins become sparsely populated, making them less useful for steganalysis. This problem is usually resolved by marginalization before forming the co-occurrences – the residual is truncated, $r_{ij} \leftarrow \text{trunc}_T(r_{ij})$, where $\text{trunc}_T(x) = x$ when $x \in [-T, -T+1, \ldots, T]$, and $\text{trunc}_T(x) = T\text{sign}(x)$ otherwise. The truncation, however, introduces an undesirable information loss. Consider a locally linear part of an image, such as sky with a gradient of blue. The differences between neighboring pixels may be quite large due to the color gradient and thus end up being truncated despite the fact that this portion of an image is well modellable. Similar situation may occur around edges. Even though the content around the edge pixels may be quite complex, the values of pixels that follow the edge appear predictable using polynomial models (see Fig. 1).

These considerations motivated us to propose Higher-Order Local Model Estimators of Steganographic changes (HOLMES). Instead of the simplistic estimator (2), we compute the residuals using a family of local linear estimators.

Fig. 1. Close-up of a horizontal edge. Note that the grayscales in the horizontal direction are quite smooth and thus can be well approximated using polynomial models.

The residuals in Table 1 are intentionally shown in their integer versions to avoid the need for rounding.

Residual type	s	Horizontal residual $\mathbf{R}^{\mathrm{h}} = (r^{\mathrm{h}}_{ij})$
First order	2	$y_{i,j+1} - y_{ij}$
Second order	3	$y_{i,j-1} - 2y_{ij} + y_{i,j+1}$
Third order	4	$y_{i,j-1} - 3y_{ij} + 3y_{i,j+1} - y_{i,j+2}$
Fourth order	5	$-y_{i,j-2} + 4y_{i,j-1} - 6y_{ij} + 4y_{i,j+1} - y_{i,j+2}$
Fifth order	6	$-y_{i,j-2} + 5y_{i,j-1} - 10y_{ij} + 10y_{i,j+1} - 5y_{i,j+2} + y_{i,j+3}$
Sixth order	7	$y_{i,j-3} - 6y_{i,j-2} + 15y_{i,j-1} - 20y_{ij} + 15y_{i,j+1} - 6y_{i,j+2} + y_{i,j+3}$

Table 1. Horizontal residuals from higher-order local models and their span s.

For example, the third and fourth order residuals can be derived from a locally quadratic model spanning three and four neighbors of the central pixel, respectively. They can also be interpreted as higher-order differences among neighboring pixels or discrete derivatives. The set of pixels involved in computing the residual is called a *clique* and its cardinality will be called *span* and always denoted s.

The residuals listed in Table 1 are all computed over horizontal cliques. The reader will readily supply the corresponding formulas for the vertical, diagonal, and minor-diagonal directions, $\mathbf{R}^{\mathrm{v}}$, $\mathbf{R}^{\mathrm{d}}$, $\mathbf{R}^{\mathrm{m}}$. There are numerous other possibilities how to define the residuals, each providing a different type of information. One particular case that turned out to be quite effective for attacking HUGO are the so-called MINMAX residuals:

$$r^{\mathrm{MIN}}_{ij} = \min\{r^{\mathrm{h}}_{ij}, r^{\mathrm{v}}_{ij}, r^{\mathrm{d}}_{ij}, r^{\mathrm{m}}_{ij}\}, \qquad r^{\mathrm{MAX}}_{ij} = \max\{r^{\mathrm{h}}_{ij}, r^{\mathrm{v}}_{ij}, r^{\mathrm{d}}_{ij}, r^{\mathrm{m}}_{ij}\}. \tag{3}$$

For pixel ij close to an edge, one of the MINMAX residuals will be large (in the direction perpendicular to the edge), while the other will likely be computed along the edge. Features built from these MINMAX residuals thus better adapt to textures and improve detection of adaptive embedding.

Of course, one can think of a myriad of other local predictors, such as the non-directional Ker–Böhme kernel [9] defined on 3×3 cliques:

$$\begin{aligned} r_{ij}^{\mathrm{KB}} &= 2y_{i-1,j} + 2y_{i+1,j} + 2y_{i,j-1} + 2y_{i,j+1} \\ &\quad - y_{i-1,j-1} - y_{i-1,j+1} - y_{i+1,j-1} - y_{i+1,j+1} - 4y_{ij} \end{aligned} \tag{4}$$

or directional kernels designed to model local image content around an edge (the model for a diagonal edge is shown in (5)) defined on cliques of span 6:

$$r_{ij}^{\mathrm{EDGE}} = 2y_{i-1,j} + 2y_{i,j+1} - y_{i-1,j-1} - y_{i-1,j+1} - y_{i+1,j+1} - y_{ij}. \tag{5}$$

Higher-order models better adjust to the local content and thus produce residuals with a more favorable SNR. Moreover, involving a clique of neighboring pixels in the linear combination "averages out" the embedding changes from the predicted value and thus further improves the prediction. According to our experience, even residuals of order as high as 5 or 6 provide useful information for steganalysis.

The reader will immediately notice that the higher-order predictors from Table 1 will have a larger dynamic range, which calls for a larger threshold T for their marginalization. To prevent rapid growth of feature dimensionality, the authors introduced quantized versions of the residuals:

$$Q_q(r_{ij}) = \mathrm{floor}\left(\frac{r_{ij}}{q}\right), \tag{6}$$

where q is a quantization step and $\mathrm{floor}(x)$ is the largest integer smaller than or equal to x. For small T, such as $T = 3$ or 4, the best detection is obtained by quantizing r_{ij} with the coefficient at the predicted pixel (see Section 4.1). In other words, for residuals of span 3–7, one should choose $q = 2, 3, 6, 10, 20$ (see Table 1).

The second-order quantized residual with $q = 2$ can be interpreted in another manner. Consider decreasing the dynamic range of the image by 50% by removing the LSB of each grayscale. The dynamic range of the resulting image is twice smaller and we also lost approximately 50% of all embedding changes – those that were LSB flips. However, the remaining changes are easier to detect due to the decreased dynamic range of the transformed image.

2.2 Features

Our features will be co-occurrence matrices formed from neighboring residual samples. To keep the notation compact, we introduce several different types of co-occurrence operators that can be applied to any two-dimensional array (residual) to produce a co-occurrence matrix or dimensionality $(2T+1)^m$, where m is the order of the co-occurrence. For example, the horizontal co-occurrence matrix of order m is

$$C^{\mathrm{h}}_{d_1 \ldots d_m}(\mathbf{R}) = \Pr(r_{ij} = d_1 \wedge \ldots \wedge r_{i,j+m-1} = d_m), \quad d_1, \ldots, d_m \in [-T, \ldots, T]. \tag{7}$$

The operators $C^{\mathrm{v}}_{d_1\ldots d_m}$, $C^{\mathrm{d}}_{d_1\ldots d_m}$, and $C^{\mathrm{m}}_{d_1\ldots d_m}$ for the vertical (v), diagonal (d), and minor diagonal (m) directions are defined analogically. Note that forming the co-occurrence matrices makes sense even when r_{ij} is non-stationary. In fact, for natural images r_{ij} is a mixture – residuals in smooth regions fill out the neighborhood of $(d_1, \ldots, d_m) = (0, \ldots, 0)$, while residuals around vertical edges will concentrate at the boundary of the matrix. Thus, different textures will likely occupy different parts of the co-occurrence matrix.

We will also make use of the fourth-order co-occurrence from residuals forming 2×2 squares:

$$C^{\mathrm{s}}_{d_1\ldots d_4}(\mathbf{R}) = \Pr(r_{ij} = d_1 \wedge r_{i+1,j} = d_2 \wedge r_{i,j+1} = d_3 \wedge r_{i+1,j+1} = d_4). \quad (8)$$

There are many possibilities how to combine the residual and the co-occurrence operator to obtain features. And all combinations capture different relationships among pixels and are thus potentially useful for steganalysis. Certain combinations, however, provide little information. Since HUGO approximately preserves the joint probability distributions of differences between four neighboring pixels along all four directions, the matrices whose elements are computed from neighboring residuals whose union of cliques spans more than four pixels are more effective for steganalysis of HUGO. Thus, we require $s + m > 5$, where s is the span of the residual and m the co-occurrence order. For example, when working with first-order residuals ($s = 2$), we recommend to take co-occurrences of at least the fourth order, while for second-order residuals ($s = 3$) the third order may be sufficient.

Another pair of parameters that needs to be adjusted jointly is T and m. With larger m, one should correspondingly decrease T otherwise the co-occurrence matrix becomes too sparse and its elements become too noisy to provide useful detection statistic. It is worth mentioning that the marginals in the co-occurrence matrix may be as important (or even more important than) the inside of the matrix. According to our experience, even co-occurrences with $T = 1$ and $m \in \{5, 6\}$ still provide quite useful information for detection.

Based on a large number of experiments, we identified several combinations of residuals and co-occurrences that provided the best results. They are listed in Table 2. Each row corresponds to a feature type (a combination of a residual and a co-occurrence operator). All feature types between highlighted lines of the table are to be combined with all parameter sets in the second column. When a parameter is a set, e.g., $(3, \{1, 2\}, 3, 4)$), it means that the features are computed with both $(3, 1, 3, 4)$ and $(3, 2, 3, 4)$.

The first four feature types in the table are computed from the MINMAX residuals. The matrices for the horizontal and vertical directions (and diagonal and minor diagonal directions) are added together to decrease dimensionality and provide a more stable statistic. The following two feature types can be thought of as sums of joint distributions of consecutive residuals modeled as Markov chains in each direction (they are similar in spirit to the SPAM feature set [13]), while the next one is computed from the Ker–Böhme residual (4).

This list should be taken as an example rather than a hard recommendation. The reader will easily come up with other forms of residuals and co-occurrence

Feature	Parameters (s, q, m, T)
$C^{\mathrm{h}}(\mathbf{R}^{\mathrm{MIN}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{MIN}})$	$(3, \{1, 2\}, 3, 4)$,$(3, \{1, 2\}, 4, 2)$
$C^{\mathrm{d}}(\mathbf{R}^{\mathrm{MIN}}) + C^{\mathrm{m}}(\mathbf{R}^{\mathrm{MIN}})$	$(4, \{2, 3\}, 3, 4)$, $(4, \{2, 3\}, 4, 2)$
$C^{\mathrm{h}}(\mathbf{R}^{\mathrm{MAX}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{MAX}})$	$(5, \{2, 3, 6\}, 3, 4)$,$(5, \{2, 3, 6\}, 4, 2)$
$C^{\mathrm{d}}(\mathbf{R}^{\mathrm{MAX}}) + C^{\mathrm{m}}(\mathbf{R}^{\mathrm{MAX}})$	$(6, \{5, 10\}, 3, 4)$,$(7, \{10, 20\}, 3, 4)$
$C^{\mathrm{h}}(\mathbf{R}^{\mathrm{h}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{v}})$	$(2, \{1, 2\}, 4, 2)$,$(3, 2, 5, 1)$,$(3, 2, 6, 1)$
$C^{\mathrm{d}}(\mathbf{R}^{\mathrm{d}}) + C^{\mathrm{m}}(\mathbf{R}^{\mathrm{m}})$	
$C^{\mathrm{h}}(\mathbf{R}^{\mathrm{KB}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{KB}})$	$(9, \{1, 2, 4\}, 3, 3)$
$C^{\mathrm{s}}(\mathbf{R}^{\mathrm{MIN}})$,$C^{\mathrm{s}}(\mathbf{R}^{\mathrm{MAX}})$	$(3, 2, 4, 2)$

Table 2. Features formed by co-occurrence matrices and their parameters.

operators that may also lead to accurate detection of embedding. The steganalyst should select the individual sets so that they are diverse and complement each other as highly correlated features are undesirable. In practice, the size of the final feature set will be limited by the ability of the steganalyst to train a high-dimensional feature vector. If the dimensionality needs to be reduced, one can apply feature selection techniques or marginalize the set in some other way, for example by forming linear combinations of individual features.

The direction we adopted in this paper is to avoid hand design as much as possible and, instead, leave this job to the machine learning algorithm. We form a large feature set preferably consisting of a union of many *diverse* feature sets. Rather than mindlessly increasing the threshold T, we keep the threshold small and add more diverse feature sets by combining different types of residuals and co-occurrence operators. The emphasis here is on diversity and the ability of the features to "calibrate themselves" – to provide useful baseline information about each other [10]. For example, it makes sense to pair the parameter set $(s, q, m, T) = (3, 1, 3, 4)$ with $(3, 2, 3, 4)$ as the former provides more detailed information around the origin ($d_1 = d_2 = d_3 = 0$) while the same feature computed from the quantized residual "can see" twice as far before marginalizing the residuals.

Overall, our strategy for attacking HUGO is to assemble the feature set by merging multiple diverse subsets and let each subset contribute to the overall detection. In the next section, we supply the missing piece – a scalable machine-learning tool that can handle high-dimensional features and a large number of training examples with low complexity and good performance.

3 Ensemble classifier

High feature dimensionality may negatively influence the complexity of training and classification as well as the ability of a classifier to generalize to previously unseen examples from the same source. Overcoming these problems becomes difficult especially when the class distinguishability is small and the number of examples from the cover source limited. Today, the machine learning tool of choice by steganalysts are kernelized SVMs, which are quite resistant to the curse of dimensionality. However, their complexity does not scale well and one

can rather quickly run into memory and processing bottlenecks. The complexity is smaller for efficient implementations of linear SVMs but can become too large as well if one desires to use linear SVMs as a development tool when many ideas need to be tested in a short period of time.

To lower the complexity, we decided to use ensemble classifiers based on fusing decisions of weak base learners trained on random subsets of the feature space. In order to make the supervised ensemble strategy work, the individual base learners have to be sufficiently diverse in the sense that they should make different errors on unseen data. The diversity is often more important than the accuracy of the individual classifiers, provided their performance is better than random guessing. From this point of view, overtrained base learners are not a big issue. In fact, ensemble classification is often applied to relatively weak and unstable classifiers since these yield higher diversity. It was shown that even fully overtrained base learners, when combined through a classification ensemble, may produce accuracy comparable to state-of-the-art techniques [3].

What makes ensemble classifiers especially attractive is that they scale well with dimensionality and the number of training examples and, according to our experience, their performance is comparable to that of Gaussian SVMs. Detailed description of ensemble classifiers, their analysis, and relationship to previous art appears in [11]. Here, we only provide a brief description. Starting with the full feature set of dimensionality d, the steganalyst first randomly selects $d_{\text{red}} \ll d$ features and trains a classifier (base learner) on them. The classifier is a mapping $F : \mathbb{R}^d \to \{0, 1\}$, where 0 stands for cover and 1 for stego.[1] This process is repeated L times, each time with a different random subset. As a result, L base learners, $F_1, \ldots, F_L$, are obtained. Given a feature $\mathbf{b} \in \mathbb{R}^d$ from the testing set, the final decision is obtained by fusing the decisions of all L individual base learners:

$$F_{\text{ens}}(\mathbf{b}) = \Im(F_1(\mathbf{b}), \ldots, F_L(\mathbf{b})) \in \{0, 1\}, \tag{9}$$

where $\Im$ is some fusion rule.

Note that all classifiers in the algorithm are trained on feature spaces of a fixed dimension d_{red} that can be chosen to be significantly smaller than the full dimensionality d. Our base learners were the low-complexity Fisher Linear Discriminants (FLDs) and we used a simple voting for the fusion rule

$$\Im(F_1(\mathbf{b}), \ldots, F_L(\mathbf{b})) = \begin{cases} 1 & \text{when } \sum_{i=1}^{L} F_i(\mathbf{b}) > L/2 \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The voting could be replaced by other aggregation rules. For example, when the decision boundary is a hyperplane, one can use the sum of projections on the normal vector of each classifier or the sum of likelihoods of each projection after fitting models to the projections of cover and stego images. Because in our experiments all three fusion strategies gave essentially identical results, we recommend using voting due to its simplicity. The individual classifiers should be adjusted to meet a desired performance criterion. In this paper, the decision

[1] F is really a map from $\mathbb{R}^{d_{\text{red}}} \to \{0, 1\}$ as each learner works with a subset of features.

threshold was always set to produce minimum overall average classification error $P_{\rm E} = \min_{P_{\rm FA}}(P_{\rm FA} + P_{\rm MD}(P_{\rm FA}))/2$ on the training data, which is the quantity that we also use to report the accuracy of detection in this paper.

4 Experiments

The main bulk of our experiments was carried out on BOSSbase 0.92 [5, 1] containing 9,074 grayscale images originally acquired by seven digital cameras in the RAW format (CR2 or DNG) and subsequently processed by resizing and cropping to the size of 512×512 pixels. All tests were done by randomly dividing the BOSSbase into a training set of 8,074 images and a testing set of 1000 images. This split was repeated and the median value of $P_{\rm E}$ and its Mean Absolute Deviation (MAD) are what we report in graphs and tables. We remark that the selection of random feature subsets in our ensemble classifier was also different in each run.

4.1 Initial tests

In our first set of experiments, we test the performance of selected individual feature sets listed in Table 3 to show the influence of the parameters (s, q, m, T) on the detection performance. The first set (MARKOV) is a direct equivalent of the second-order SPAM [13] with two differences – the first-order differences were replaced with second-order differences and the transitional probability matrix with the joint matrix (co-occurrence). It is rather interesting that by changing a *single line* of code SPAM turns into a significantly more powerful feature set – $P_{\rm E}$ has dropped from 42% [14] to 28.6%.[2] The second row of the table informs us that the detection is even better with the MINMAX residual, while the its quantized version shaves another 1% from $P_{\rm E}$. The next two rows are mergers of five sets of total dimensionality 7,290 and 6,250 for co-occurrence matrices of order $m = 3$ and 4 with $T = 4$ and $T = 2$, respectively. Adding features steadily leads to better performance.

The feature sets in the last two rows were quantized with q equal to the coefficient at x_{ij} in the higher-order residual (inspect Table 1) as this choice of q gave us the best performance. This is confirmed in Table 4 with the MINMAX residual with $s = 5$ (fourth-order residual) by showing $P_{\rm E}$ as a function of q while fixing all other parameters and variables ($L = 31$, $d_{\rm red} = 1000$).

According to our experiments on BOSSbase, adding more features generally leads to better detection. However, adding uninformative or dependent features will obviously decrease the detection accuracy. Clever marginalizations may also improve detection while keeping the dimensionality low. For example, we *added* all five co-occurrence matrices of third order listed in row 4 in Table 3 to form one 1458-dimensional vector. Then, we did the same with the features from row

[2] This comparison is not really fair as the results were obtained on two different databases – BOWS2 vs. BOSSbase – while the latter appears somewhat easier to steganalyze.

Feature set	(s, q, m, T)	d	P_{E}	Best	Worst	L	d_{red}
MARKOV	$(3, 1, 3, 4)$	1458	28.6±0.9	25.5	31.0	31	1000
MINMAX	$(3, 1, 3, 4)$	1458	27.3±0.8	25.1	31.3	31	1000
MINMAX	$(3, 2, 3, 4)$	1458	26.2±1.2	23.2	28.4	31	1000
MINMAX	$(\{3, 4, 5, 6, 7\}, c, 3, 4)$	7290	20.0±0.8	17.8	22.6	81	1600
MINMAX	$(\{3, 4, 5, 6, 7\}, c, 4, 2)$	6250	20.9±0.4	19.0	23.5	81	1600

Table 3. Performance of individual feature sets on BOSSbase 0.92. The acronyms MARKOV and MINMAX stand for co-occurrences $C^{\text{h}}(\mathbf{R}^{\text{h}}) + C^{\text{v}}(\mathbf{R}^{\text{v}})$, $C^{\text{d}}(\mathbf{R}^{\text{d}}) + C^{\text{m}}(\mathbf{R}^{\text{m}})$, and $C^{\text{h}}(\mathbf{R}^{\text{MIN}}) + C^{\text{v}}(\mathbf{R}^{\text{MIN}})$, $C^{\text{h}}(\mathbf{R}^{\text{MAX}}) + C^{\text{v}}(\mathbf{R}^{\text{MAX}})$, respectively. The quantization step in the last two sets was set to the coefficient at x_{ij} in the higher-order residual ($c = 2, 3, 6, 10, 20$ for $s = 3, 4, 5, 6, 7$).

5 to form a 1250-dimensional vector. Putting these two matrices together gave us a $1458 + 1250 = 2708$-dimensional vector with $P_{\text{E}} = 22\%$ under the same testing conditions (with $L = 81$ and $d_{\text{red}} = 1600$). Obviously, adding feature sets is by no means the optimal operation and we prefer to leave the marginalization to an automated procedure instead of hand-tweaking. For experiments in the next section, we prepared a feature set by merging various combinations of residuals and co-occurrence matrices (the set is described in the Appendix).

q	2	4	6	8	10	12
P_{E}	30.50	26.75	26.05	26.75	27.70	28.20

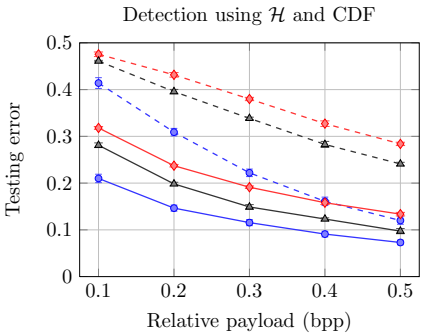
Table 4. Detection error P_{E} for the MINMAX feature set with parameters $(5, q, 3, 4)$ as a function of $q \in \{2, 4, 6, 8, 10, 12\}$. The best performance is achieved when q is equal to 6 – the coefficient at x_{ij} in the higher-order residual.

4.2 Performance on BOSSbase

The purpose of experiments in this section is three-fold: to evaluate the detectability of HUGO, compare the HOLMES features and our ensemble classifier with the current state of the art – the CDF set [12], and to compare HUGO with non-adaptive ±1 embedding. Unless stated otherwise, all detectors were implemented using ensemble classifiers with FLDs as described in Section 3. We used a 33,963-dimensional feature set $\mathcal{H}$ implemented with $L = 81$ and $d_{\text{red}} = 2800$ (see the Appendix). The CDF classifier used $L = 51$ and $d_{\text{red}} = 500$. The values of d_{red} were determined by hand based on our experience.

All results are displayed in the self-explanatory Fig. 2. The CDF set has higher detection accuracy when implemented using a Gaussian SVM (G-SVM) instead of our ensemble classifier. However, unlike G-SVM, the ensemble classifier is capable of handling the high-dimensional HOLMES features which resulted in a consistently lower detection error P_{E} than the error for the CDF trained with a G-SVM. HUGO is confirmed to be more secure than non-adaptive ±1 embedding but the difference is less pronounced than what was reported in [14].

It is also interesting to compare the increase in detection accuracy for both algorithms and feature sets. While the improvement for HUGO is about 5–12%, the detectability of ±1 embedding improved only by 2–7%.



α		HUGO			±1 embedding		
bpp	$\mathcal{H}$	CDF	G-SVM	$\mathcal{H}$	CDF	G-SVM
0.1	41.4	47.6	46.1	21.0	31.8	28.1
0.2	30.9	43.2	39.6	14.7	23.7	19.8
0.3	22.2	38.0	33.9	11.6	19.1	14.9
0.4	16.1	32.7	28.3	9.1	15.8	12.3
0.5	12.0	28.4	24.1	7.3	13.4	9.7

Fig. 2. Detection error P_{E} for HUGO and ±1 embedding for five relative payloads for the CDF and HOLMES classifiers. The error bars are MAD over 100 database splits 8074/1000. The CDF set was implemented with both our ensemble classifier and as a G-SVM (only 10 splits 8074/1000 were performed using G-SVM due to computational complexity).

Since BOSSbase images were resized to quite a small size, the correlations among neighboring pixels weaken significantly in textured regions, such as grass, sand, or foliage. Visual inspection confirmed that such textures start resembling random noise on the pixel level, which makes their steganalysis very difficult if possible at all since HUGO avoids regions where the content can be accurately modeled. To identify the type of images on which our classifier makes consistently correct and wrong decisions, we carried out the following experiment. Using the same setup with the HOLMES feature set $\mathcal{H}$, we repeated the random 8,074/1000 split of BOSSbase 1000 times (with $L = 81$ and $d_{\text{red}} = 2400$) and counted how many times a given cover image was classified as stego and vice versa. Each image $i \in \{1, \dots, 9074\}$ appeared in the testing set N_i times, where N_i is a binomial r.v. with mean 110 and standard deviation 9.9. Fig. 3 shows the probability $p_i = \delta_i / N_i$ of correctly detecting cover image i as cover (cover i was correctly classified δ_i times). In the figure, the BOSSbase is ordered by cameras. First, note that the detection heavily depends on the camera model. While cover images from Pentax can be classified with average accuracy of about 95%, images from Canon Rebel are significantly harder to classify (66%). This difference is most likely a combined effect of varying depth of field across both cameras (which is influenced by the lens), in-camera processing (some cameras denoise their images), the resizing script, and the environment in which the images were taken. All this forms the cover source and gives it unique properties that have a *major* effect on statistical detectability of embedding changes.

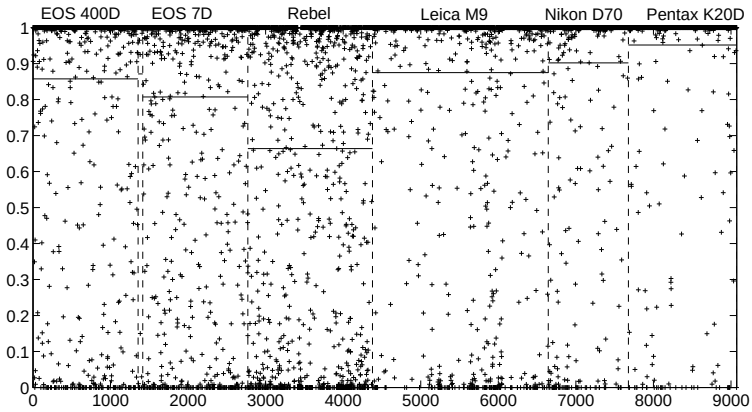


Fig. 3. Probability with which each cover image $i \in \{1, \ldots, 9074\}$ from BOSSbase was correctly classified as cover over 1000 random splits (8074/1000). The images are sorted by cameras. The average detection for each camera is displayed with a horizontal line.

Second, notice that some cover images are persistently classified as stego (FAs) – the steganalyzer errs on them with probability 1. In fact, we identified 743 cover images that were *always* detected as stego and 674 stego images always detected as cover (MDs). Most of these images were highly textured and/or with a large contrast, and many contained complex content, such as shots in a forest with many fine branches. The high dimensionality of the feature set and the relatively low number of training examples mean that some images will be located in sparsely populated regions of the feature space. The classifier generalizes to them but, due to lack of similar features in their neighborhood, the decision boundary is not likely to be well placed. As a result, some images are consistently misclassified.

Also, 6627 cover images were always correctly detected as cover and 6647 stego images were always detected as stego. The intersection of these two sets contains 4836 BEST images that were always detected correctly both in their cover and stego forms. These easiest-to-classify images did not contain many edges or textures, some were out-of-focus shots or shots with low depth of field and images with a small dynamic range of pixel values. Table 5 displays the average grayscale, average number of pixels saturated at 255, and average texture defined as $t = c \cdot \sum_{ij} |x_{ij} - x_{i,j-1}|$, with c being a scaling constant. Overall, images with a high number of saturated pixels and bright / textured images are harder to classify. Lower average grayscale is connected to a lower dynamic range, which indeed will make detection of embedding changes easier. The effect of saturated pixels, however, is more mysterious.

Images	Avg. gray	Avg. saturation	Texture
BEST	74.1	2046	1.73
FAs	101.3	4415	4.66
MDs	102.0	5952	3.95

Table 5. Average grayscale, number of pixels saturated at 255, and texture for BEST, FAs, and MDs from BOSSbase.

4.3 Performance on BOSSrank

This section briefly discusses our performance on the BOSSrank set used for the BOSS competition [1]. It consists of 847 images taken by Leica M9 and 153 images from Panasonic Lumix DMC-FZ50. Total of 518 images were covers, while the remaining 482 were stego images embedded with relative payload 0.4 bpp.

The best score we achieved on BOSSrank was $1 - P_{\text{E}} = 80.3\%$ or $P_{\text{E}} = 19.6\%$.[3] It was obtained for a submission generated from a 25,993-dimensional feature set trained on $34,719$ images[4] with $L = 31$ and $d_{\text{red}} = 2400$. More details about this feature set and our experience with BOSS appear in our other paper in this volume [6]. The drop in performance w.r.t. our results on BOSSbase is caused by the cover-source mismatch and the lack of robustness of our ensemble classifier.[5] While our detector was trained on BOSSbase, BOSSrank images are coming from a different source. The Panasonic Lumix images are not in BOSSbase at all and they were taken in JPEG instead of the RAW format. While the Leica M9 is in BOSSbase, it forms only about 25% of the database (2267 images). The cover source mismatch is a serious issue for practical steganography as it lowers the detection accuracy and complicates controlling the error rates of practical detectors. The cover-source mismatch is also the reason why our detector that used the higher-dimensional set $\mathcal{H}$ performed worse on BOSSrank even though we observed the opposite for BOSSbase.

5 Conclusion

Modern steganographic algorithms, such as HUGO, hide messages by approximately preserving a high-dimensional representation of covers that captures many complex dependencies among individual cover elements. The embedding is thus naturally adaptive and confines the modifications to hard-to-model regions of covers. This is the reason why steganalyzers that work in feature spaces of low dimension do not detect this type of embedding well. A possible way to improve the detection is to work with high-dimensional features as well. The two key open problems are the formation of such feature spaces and machine learning whose complexity scales favorably with dimension.

[3] Our error on Leica was 17.7% and 30.0% on Panasonic.

[4] All training images were obtained from RAW images using the same BOSS script.

[5] Other classifiers, including linear SVMs, Gaussian SVMs, and the FLD were equally susceptible to the cover-source mismatch.

In particular, it is not sufficient to blindly increase the feature dimensionality for example by increasing the order of co-occurrence matrices or their range (threshold). This way, we would be adding sparsely-populated (noisy) features with low detectability. In this paper, we propose a methodology called HOLMES for forming a diverse high-dimensional feature vector. It consists of two steps – computing several types of higher-order residuals and then forming co-occurrence matrices from their neighboring values in a standard fashion. The residuals should be computed in the embedding domain and using pixel predictors that only depend on the neighboring pixels but not the central pixel being predicted. We also discovered that good residuals for content-adaptive steganalysis may be obtained using non-linear processing as minimal and maximal values of residuals computed from several different directions – the MINMAX residual. The emphasis should be on high *diversity* of the features rather than dimensionality so that combining features improves detection.

Having formed a high-dimensional feature vector, we coin the use of ensemble classifiers obtained by fusing decisions of simple detectors implemented using the Fisher linear discriminant. They were a crucial element in our participation in BOSS as their low complexity, simplicity, and speed enabled rapid development and optimization of the feature set to maximize the performance.

To summarize our attack, we were unable to use the fact that for HUGO the probability of embedding changes at individual pixels can be approximately estimated. It does not appear that giving the Warden probabilistic information about the selection channel is a weakness. Another lesson learned is that, as the level of sophistication of steganographic schemes increases, steganalysis needs to use high-dimensional feature sets and scalable machine learning.

Our attack on HUGO also reveals quite useful information about steganography design. While the authors of HUGO did strive to preserve a high-dimensional feature vector, they scaled the dimensionality simply by increasing the threshold T. Most features in this high-dimensional feature vector are, however, quite uninformative and trying to preserve them eventually weakens the algorithm. Instead, the dimensionality needs to be increased by adding more diverse features. We expect the future versions of HUGO working with more diverse feature spaces, such as the set $\mathcal{H}$, to be significantly more secure to attacks.

6 Acknowledgements

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. The authors would like to thank to the Soukal's Family, Tice Lerner, Jim Pittaresi, Thomas Gloe, Peggy Goldsberry,

and Jonathan Cohen for providing their images for research purposes and to the BOSS Team for setting up the BOSS competition.

Appendix – the final feature set

Feature type	(s, q, m, T)	Dimensionality
MINMAX	$(3,1,3,4)$, $(3,2,3,3)$, $(4,\{2,3\},3,3)$ $(5,\{2,6\},3,3)$, $(6,10,3,3)$, $(7,20,3,3)$	$1458 + 7 \times 686 + 10 \times 162$
MARKOV	$(3,\{1,2\},4,1)$, $(4,\{2,3\},4,1)$, $(5,\{2,6\},4,1)$ $(6,\{5,10\},4,1)$, $(7,\{10,20\},4,1)$	$1458 + 7 \times 686 + 10 \times 162$
MINMAX	$(3,2,5,1)$	2×243
MINMAX	$(2,\{1,2\},4,2)$	2×1250
KB	$(9,\{1,2,4\},3,4)$	3×729
SQUARE	$(3,2,4,1)$	2×162
CALI	$(3,2,3,4)$, $(4,2,3,4)$	2×1458
EDGE	$(6,\{1,2,4\},3,4)$	3×1458
MINMAX	$(\{3,4,5,6,7\},c,3,4)$ summed	$1458 + 1250$
MARKOV	$(\{3,4,5,6,7\},c,4,2)$ summed	$1458 + 1250$

Table 6. The final HOLMES feature set $\mathcal{H}$ of dimensionality 33,963.

All feature types in a block between two highlighted lines are to be combined with all parameter sets. The KB set was formed by $C^{\mathrm{h}}(\mathbf{R}^{\mathrm{KB}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{KB}})$, where $\mathbf{R}^{\mathrm{KB}}$ is the residual (4). The SQUARE set is obtained from the MINMAX residual with co-occurrence operator (8). In the CALI set, prior to computing the features from the MINMAX residual, the image was convolved with an averaging 2×2 kernel [1 1; 1 1] in an attempt to calibrate the features as in [8]. The residual for EDGE was formed using (5) as the minimum and maximum values along edges in four different directions (residual $\mathbf{R}^{\mathrm{EDGEMIN}}$ and $\mathbf{R}^{\mathrm{EDGEMAX}}$) and then applying $C^{\mathrm{h}}(\mathbf{R}^{\mathrm{EDGEMIN}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{EDGEMIN}})$, $C^{\mathrm{h}}(\mathbf{R}^{\mathrm{EDGEMAX}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{EDGEMAX}})$. The last four sets were obtained as sums of all five sets whose parameters appear in the second column.

References

1. P. Bas, T. Filler, and T. Pevný. Break our steganographic system – the ins and outs of organizing BOSS. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Workshop*, Lecture Notes in Computer Science, Prague, Czech Republic, May 18–20, 2011.
2. C. Chen and Y.Q. Shi. JPEG image steganalysis utilizing both intrablock and interblock correlations. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 3029–3032, May 2008.
3. A. Cutler and G. Zhao. PERT - perfect random tree ensembles. *Computing Science and Statistics*, 33:490–497, 2001.

4. 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.
5. T. Filler, T. Pevný, and P. Bas. BOSS. http://boss.gipsa-lab.grenoble-inp.fr/BOSSRank/, July 2010.
6. 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, Prague, Czech Republic, May 18–20, 2011.
7. 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.
8. A. D. Ker. Steganalysis of LSB matching in grayscale images. *IEEE Signal Processing Letters*, 12(6):441–444, June 2005.
9. 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.
10. 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.
11. 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, Watermarking, Security, and Forensics of Multimedia XIII*, volume 7880, pages OL 1–13, San Francisco, CA, January 23–26, 2011.
12. J. Kodovský, T. Pevný, and J. Fridrich. Modern steganalysis can detect YASS. In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XII*, volume 7541, pages 02–01–02–11, San Jose, CA, January 17–21, 2010.
13. 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.
14. T. Pevný, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. In P. W. L. Fong, R. Böhme, and Rei Safavi-Naini, editors, *Information Hiding, 12th International Workshop*, Lecture Notes in Computer Science, pages 161–177, Calgary, Canada, June 28–30, 2010.
15. Y. Q. Shi, C. Chen, and W. Chen. A Markov process based approach to effective attacking JPEG steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Science, pages 249–264, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.