On Completeness of Feature Spaces in Blind Steganalysis

Jan Kodovský
SUNY Binghamton
Department of ECE
Binghamton, NY 13902-6000
jan@kodovsky.com

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

ABSTRACT

Blind steganalyzers can be used for many diverse applications in steganography that go well beyond a mere detection of stego content. A blind steganalyzer can also be used for constructing targeted attacks or as an oracle for designing steganographic methods. The feature space itself provides a low-dimensional model of covers useful for benchmarking. These applications require the feature space to be complete in the sense that the features fully characterize the space of covers. Incomplete feature sets may skew benchmarking scores and lead to poor steganalysis. As a simple test of completeness, we propose a general approach for constructing steganographic methods that approximately preserve the whole feature vector and thus become practically undetectable by any steganalyzer that uses the same feature set. We demonstrate the plausibility of this approach, which we call the Feature Correction Method (FCM) by constructing the FCM for a 274-dimensional feature set from a state-of-the-art blind steganalyzer for JPEG images.

Categories and Subject Descriptors

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

General Terms

Security, Algorithms, Theory

Keywords

Blind steganalysis, completeness, FCM, steganography

1. INTRODUCTION

It is widely believed that there exist two classes of steganalysis attacks—targeted and blind. Targeted attacks use features constructed using knowledge of specific details about the embedding mechanism, while blind schemes use a conglomerate of features that can potentially detect *arbitrary* steganographic method. Even though this distinction appears unambiguous and natural, the boundary between these two classes is quite blurry and the overlap is larger than what it seems at the first sight. We wish to emphasize at this point that targeted steganalysis does not have to be quantitative [10, 17] (able to estimate the number of embedding changes) and both approaches can use feature extraction and machine learning even though these two design elements are more typically associated with blind schemes.

The main distinction between the two steganalysis classes appears to be in the *scope* of the feature set. In targeted schemes, the feature set is not meant to be a complete descriptor of covers. Its purpose is to merely detect specific embedding changes. In fact, targeted schemes often use a single feature. On the other hand, in blind schemes the features' role is much more ambitious—they play the role of a low-dimensional cover *model* that is more manageable to work with than the original space of covers. However, unless the features completely characterize every dependency among individual elements of the cover, in theory we will always be able to construct a steganographic method by using the "gaps" that exist in the model. Note that the selection of the classifier in blind steganalysis is of secondary in importance to the selection of features.

A good low-dimensional model of covers can be used for a multitude of diverse tasks in steganography that go well beyond simply detecting the presence of secret messages:

- Detecting steganography
- Design of steganographic schemes
- Benchmarking
- Targeted attacks

We now briefly discuss each of these applications and point out issues some of which can be linked to the completeness of the feature space investigated in this paper. The first application, which is what blind steganalysis is typically associated with, is a detection problem

$$\begin{aligned} H_0 &: \text{ object contains secret message} \\ H_1 &: \text{ object does not contain secret message.} \end{aligned}$$

This difficult composite hypothesis testing scenario is usually approached using machine learning methods with the hope that if the classifier is presented with sufficiently many examples of cover and stego objects embedded using many different stego schemes, it will be able to generalize to previously unseen stego methods [2, 3, 6, 22, 23, 39, 1, 36, 32, 28,

MM&Sec'08, September 22–23, 2008, Oxford, United Kingdom.

14]. An alternative approach is to only characterize the set of covers and label all covers that do not fall within this set as stego (one-class detectors [23]). Both approaches require the feature set to be complete—to exhaustively describe covers.

A good blind steganalyzer can (and should) be used as an oracle for design of steganographic schemes. An exemplary case of steganography design guided by blind steganalyzers was recently provided by the authors of the YASS algorithm [33, 31]. Other examples include [14, 13]. The fact that a new algorithm is undetectable using existing blind schemes, unfortunately, does not mean that it is statistically undetectable, unless the feature set is complete.

The feature space can be used as a simplified model of covers for benchmarking steganographic schemes by calculating the KL divergence (or some other statistics) between the features of covers and stego objects from a fixed large database. For this application, it is very important that there should not exist a steganographic method with reasonable embedding capacity that preserves the feature vector. Again, we require the features to be complete.

By training a blind scheme on a set of stego images produced by a specific embedding algorithm, S, we can obviously construct a targeted detector for the following hypothesis testing problem

$$\begin{aligned} \mathrm{H}_0 &: \text{ object does not contain secret message embedded using method } S \\ \mathrm{H}_1 &: \text{ object contains secret message embedded using method } S. \end{aligned}$$

In this case, dimensionality reduction methods [24] could be applied to decrease the dimensionality of the feature space and provide a simpler and perhaps more accurate targeted detector. Moreover, it may be possible to use the quantitative response of the detector to derive an estimate of the number of embedding changes, converting thus the targeted attack into a quantitative one. While feature dimensionality reduction is desirable for this specific application, it may prove fatal when used for the other applications mentioned above. This is because a general steganography detector should be able to detect novelty (previously unseen steganographic methods) for which a redundant feature may suddenly become crucial. In fact, non-trivially redundant (characterizing) features describe covers and are important in steganalysis (see Section 2). In Section 3, we formalize the concept of a complete feature set and give examples of methods for finding characterizing features. The importance of completeness for benchmarking steganography is discussed in Section 3.1.

It is, in general, very hard to establish completeness of a feature space for complex covers, such as digital images, due to their richness and high dimensionality. Although in this paper we make no attempt to provide approaches verifying completeness, in Section 4 we propose a new approach to steganography using which one can probe the completeness of a feature set by constructing schemes that approximately preserve the feature vector. We call this method the Feature Correction Method (FCM) and implement it for the Merged feature set for JPEG images [28]. Experiments on a test database of images indicate that the FCM is, indeed, undetectable using the Merged feature set. At the same time, we point out that while the FCM is undetectable using the feature set on which it was built, the FCM may be detectable using a different set of features. Augmenting the feature set with these new features, one can again construct the FCM for the augmented set and thus gradually approach completeness, at least in some practical manner. The paper is summarized in Section 5, where we discuss possible avenues for future research on this topic.

2. ON IMPORTANCE OF REDUNDANT FEATURES

We argue that removing certain redundant features from steganography detectors can be detrimental. In fact, redundant features can be quite useful for blind steganalysis. Imagine the class of single-compressed cover images in the JPEG format with a fixed quality factor and the feature vector $\mathbf{u}$ formed by three quantities

$$\mathbf{u} = (h(0), h(-1), h(1)),$$

where $h(i)$, $i = 0, -1, 1$ are histogram values of quantized DCT coefficients equal to i (normalized so that $\sum h(i) = 1$). For natural images, the histogram of DCT coefficients is approximately symmetric and thus we have $h(-1) \approx h(1)$. Therefore, when analyzing the space of all covers, dimensionality reduction methods would conclude that either $h(-1)$ or $h(1)$ be removed since these features are highly redundant. We may even take into consideration some steganographic schemes, such as F5 [37], Model based steganography [30], Steghide [16], OutGuess [29], and reach the same conclusion—the two features are highly redundant and do not help us distinguish between cover and stego images. Thus, we remove $h(-1)$, obtaining the reduced feature set

$$\mathbf{v} = (h(0), h(1)).$$

Of course, any steganography detector that uses $\mathbf{v}$ as the feature space, will not detect Jsteg[1] because Jsteg does not embed into 0's and 1's and thus preserves their counts. On the other hand, the feature set $\mathbf{u}$, which contains the seemingly useless redundant feature, will be successful in detecting Jsteg. The redundancy here is non-trivial in the sense that $h(-1) \approx h(1)$ only holds for natural covers rather than the whole set of possible covers. In fact, non-trivial redundancies among features can be quite useful for steganalysis because they provide information about covers. We now attempt to formalize these observations.

3. CHARACTERISTIC FEATURES AND COMPLETE FEATURE SETS

The set of all theoretically possible covers will be denoted $\mathcal{C}$. For example, for covers in the form of $N \times N$ 8-bit grayscale digital images, $\mathcal{C} = \{0, 1, \ldots, 255\}^{N \times N}$. As in Cachin's definition of steganographic security, we assume that there exists a source of covers represented with a random variable $\mathtt{c}$ on $\mathcal{C}$ with probability density function (pdf) P_c. The support of P_c is significantly smaller than $\mathcal{C}$ because most images have extremely low probability of being selected as covers (they are not generated by the source). Thus, at least in theory we can define the set of natural images, $\mathcal{C}_0$, as[2]

$$\mathcal{C} \supset \mathcal{C}_0 = \{c \in \mathcal{C} | P_c(c) > 0\}.$$

[1] `http://zooid.org/~paul/crypto/jsteg/`

[2] Or, perhaps, more realistically, $\mathcal{C}_0 = \{c \in \mathcal{C} | P_c(c) > \delta\}$ for some small $\delta > 0$.

A feature is any mapping $x : \mathcal{C} \to \mathbb{R}$. We call x *characterizing* if

$$c \in \mathcal{C}_0 \Rightarrow x(c) = 0.$$

The feature set $(x_1, x_2, \dots x_n)$ is *complete* if every x_i is characterizing and

$$\{x_i(c) = 0,\ \forall i\} \Rightarrow c \in C_0.$$

Thus, a complete feature set completely determines the set of natural images

$$\mathcal{C}_0 = \{c \in \mathcal{C} | x_i(c) = 0,\ \forall i\}.$$

The feature selection in blind steganalysis is all about the quest to obtain a complete feature set. This task, however, appears to be a very hard problem for complex covers, such as digital images. We can only hope to approach completeness in some approximate manner. First of all, requiring exact equality $x(c) = 0$ for all $c \in \mathcal{C}_0$ would not be reasonable. In practice, we always have an approximate equality, among other reasons also because natural images contain stochastic components (noise). What we strive for in practice is $E[x(\mathsf{c})] = 0$ and the variance $Var[x(\mathsf{c})]$ to be as small as possible. There exist many features that are approximately characterizing, such as those originating from quantitative steganalysis [10, 17]. Good examples are estimators of the number of LSB changes, e.g., [5, 17, 9].

Non-trivial dependencies among features (pixels or DCT coefficients) are especially useful for constructing approximately characterizing features. In the simple example from Section 2, an approximately characterizing feature for JPEG images would be the difference $x(c) = h(1) - h(-1)$ or in general $h(i) - h(-i)$, $i = 1, 2, ...$

Another method to construct such features is to use known statistical properties of natural images. For instance, it is well established that the histograms $h_{kl}(i)$ of DCT coefficients for a fixed spatial frequency (k, l), $k, l = 0, \dots, 7$, follow the generalized Gaussian distribution. Thus, a potentially useful approximately characterizing feature would be the error between the histogram and its parametric Generalized Gaussian fit

$$x_{kl}(c) = \sum_i h_{kl}(i) - g(i; \hat{\mu}, \hat{\alpha}, \hat{\beta}),$$

where

$$g(x; \mu, \alpha, \beta) = \frac{\alpha}{2\beta\Gamma(1/\beta)} e^{-\left(\frac{|x-\mu|}{\beta}\right)^{\alpha}}$$

is the generalized Gaussian pdf and $\hat{\mu}, \hat{\alpha}, \hat{\beta}$ are the mean, shape, and width parameters estimated from $h_{kl}(i)$.

A general approach to obtain approximately characterizing features is called calibration [8, 18]. Many features that appear useful for detection of steganographic embedding are not characterizing because $E[x(\mathsf{c})] \neq 0$ or their variance is too large. Consequently, they are less effective for steganalysis, e.g., the center of gravity of the histogram characteristic function [15, 18]. If we can, however, estimate the cover image from the stego image, we can make any feature approximately characterizing by taking the difference

$$x_{\text{cal}}(c) = x(c) - x(\hat{c}), \tag{1}$$

where $\hat{c}$ stands for the estimated cover. Because for cover images, $c \approx \hat{c}$, the calibrated feature x_{cal} is approximately characterizing.

Most features for blind steganalysis were intuitively designed as being sensitive to embedding and then calibrated to make them less sensitive to image content (characterizing) and responsive to certain type of stego embedding, which we now interpret as the result of the quest for completeness. We note that calibration has been shown, indeed, to improve features' ability to detect stego content [28].

3.1 Completeness for benchmarking

Completeness is an especially crucial property for benchmarking steganographic schemes. Steganographic scheme is a mapping $S : \mathcal{C} \times \mathcal{M} \times \mathcal{K} \to \mathcal{C}$ that assigns a stego object, $s \in \mathcal{C}$, to each triple (c, M, K), where $M \in \mathcal{M}$ is a secret message selected from the set of communicable messages, $\mathcal{M}$, and $K \in \mathcal{K}$ is the steganographic secret key. Assuming the covers are selected with pdf P_c and embedded with a message and secret key both randomly (uniformly) chosen from their corresponding sets, the set of all stego images is again a random variable, s, on $\mathcal{C}$, with pdf P_s. The measure of steganographic statistical detectability is the Kullback–Leibler divergence [4]

$$D(P_c||P_s) = \sum_{c \in \mathcal{C}} P_c(c) \log \frac{P_c(c)}{P_s(c)}. \tag{2}$$

Benchmarking steganographic schemes would be easy if we could compute the KL divergence (2). This is, unfortunately, rarely possible due to the large dimensionality of $\mathcal{C}$. In practice, this issue is usually approached from two directions. In the first one, a simplified analytical model is accepted for $\mathcal{C}$ that enables computing P_s as well as (2) analytically or numerically. The requirement to be able to carry out the necessary calculations severely limits the complexity of the model. In practice, this means that even though some methods can be shown to be secure in Cachin's sense [30, 35, 34, 25, 16, 29, 7], they may still be easily detectable using a better model [38, 28, 32]. An alternative approach is to select a low-dimensional model of $\mathcal{C}$ and estimate the KL divergence or some other two-sample measure from a large database of cover and stego images.

Good models for benchmarking should be spanned by a complete feature set. Of course, verifying completeness is quite nontrivial. Denoting

$$\mathcal{C}_i = \{c \in \mathcal{C} | x_i(c) = 0\},$$

for a complete feature set

$$\bigcap_{i=1}^{n} \mathcal{C}_i = \mathcal{C}_0.$$

For a feature set built in an ad hoc manner, we will likely have

$$\bigcap_{i=1}^{n} \mathcal{C}_i \supset \mathcal{C}_0 \tag{3}$$

because the features will not form a complete statistical description of natural covers. This means that the benchmark built from such a feature set will give too optimistic results about a scheme's undetectability. Moreover, there may exist schemes that produce stego objects outside of $\mathcal{C}_0$ but are benchmarked as undetectable.

4. TOWARDS COMPLETENESS – THE FEATURE-CORRECTION METHOD

The discussions in the preceding sections point to the following two practical problems. First, we would like to know if a given feature set is complete and, second, if it is not, how to augment it so that it becomes complete, or at least "more complete." In this paper, we study the easier first problem. It is clear that if we can construct a steganographic scheme that is undetectable in a given feature space, the feature space is not complete. We now describe a general methodology for constructing such a scheme by requiring that it approximately preserves the *entire* feature vector while still providing reasonable capacity. This strategy will be called the Feature Correction Method (FCM).

In the past, various authors proposed schemes that preserve selected cover statistics [30, 35, 34, 25, 16, 29, 7] (e.g., histogram of DCT coefficients) or their models [30]. The steganalysis-aware steganography [26] was designed to be undetectable using the wavelet features [6]. It uses iterative projections on convex sets to find a slight modification of the cover image that communicates the required message and stays within the convex set of cover images. It remains to be seen how well this approach scales with the number of features that must be preserved and if it can be applied when the set of cover images is not convex.

Instead of describing the FCM in a general setting, we explain the main idea on a specific example. Taking $\mathcal{C}$ to be the set of JPEG images with a fixed quality factor of 75, we selected the Merged feature set as described in [28]. The reason for this choice is that this set combined with an SVM learning engine leads to some of the most accurate steganalysis of JPEG images based on various independent comparisons reported in [27, 28, 19, 33]. The 274-dimensional feature vector consists of 193 extended DCT features combined with 81 Markov features. The DCT features capture inter-block dependencies among DCT coefficients and some dependencies in the spatial domain (blockiness), while the Markov features are designed to measure intra-block dependencies. All features are calibrated in an attempt to make them approximately characterizing.

The proposed feature correction method is based on the principle of statistical restoration as introduced by Provos [29] and others [34]. First, we outline the main strategy of the method and then explain in more detail the individual components.

In contrast with methods that utilize side information at the sender (e.g., the uncompressed cover image), such as MMx [20] or perturbed quantization [11], the FCM embeds a secret message into a JPEG cover image. The set of all DCT coefficients from the cover JPEG image, $\mathcal{D}$, is divided using a secret stego key into two disjoint subsets $\mathcal{D}_e \cup \mathcal{D}_c = \mathcal{D}$ with cardinalities $|\mathcal{D}_e|$ and $|\mathcal{D}_c|$. In the first (embedding) phase, the payload is embedded in non-zero coefficients from $\mathcal{D}_e$ using wet paper codes (WPC) with improved embedding efficiency [12] (the dry coefficients are the non-zero DCTs). The use of WPCs eliminates the need to take any special precautions about shrinkage (situation when a non-zero DCT coefficient in the cover is modified to 0 during embedding). WPCs with improved embedding efficiency further reduce the number of embedding changes. Whenever the parity (LSB) of the DCT coefficient is to be changed, we increase and decrease the value by 1 and select the change that perturbs the feature vector the least. This way, during embedding we are already making sure that the feature vector is modified as little as possible.

In the second (correction) phase, which starts after embedding the entire payload in $\mathcal{D}_e$, additional modifications are made in the unused part of the image, $\mathcal{D}_c$, to bring the feature vector closer to its original position. Each non-zero coefficient in $\mathcal{D}_c$ is visited and the feature vector is computed after modifying the coefficient by $-2, -1, 1, 2$. The modification that brings the feature vector the closest to the original cover feature vector is then realized. Changes by more than 2 are not allowed as they would introduce visible (and thus detectable) distortion. Note that in the worst case when all four modifications increase the distance, the coefficient is not modified at all.

To complete the description of the embedding algorithm, we need to supply the metric for measuring the distance between features and the method for splitting the set of all DCT coefficients into the subsets $\mathcal{D}_e$ and $\mathcal{D}_c$. First, the features are scaled to have unit variance on the set of cover images. The distance between the scaled features is the usual Euclidean norm

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\sigma_i^2}, \tag{4}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_n)$ are the feature vectors of two images and σ_i^2 is the variance of the i-th feature estimated from a large database of cover images. Ideally, before computing the Euclidean distance, the features should also be made uncorrelated (independent) using principal component analysis (PCA) or independent component analysis (ICA). Without transforming the data, we take the risk that a small change in the feature vector can still be easily detectable if the changes to individual features violate dependencies that occur among cover images. When using the transformed data, however, it is of paramount importance that robust versions of PCA/ICA are used otherwise outliers could create new dependencies among the transformed features. We intend to investigate robust versions of PCA/ICA for applications in FCM in our future work. In this paper, we do not transform the features and use the norm (4) directly. As will be demonstrated in Section (4.2), even this simple version of the FCM is quite effective in resisting blind steganalysis.

The problem of splitting the set of DCT coefficients into the two subsets $\mathcal{D}_e$ and $\mathcal{D}_c$ is discussed in Section 4.2.

4.1 Differential feature computation

When making a modification to a DCT coefficient during both the embedding and correction phases, we need to recompute the whole feature vector. Assuming the feature vector computation requires $O(N)$ operations, where N is the number of DCT coefficients in the image, the FCM would require $O(N \times N_0) = O(N^2)$ operations (N_0 is the number of non-zero DCT coefficients). This complexity would make large scale testing of the FCM infeasible. Fortunately, recomputing the feature vector after changing a single DCT coefficient can be carried out much more efficiently.

Consider, for example, the first-order (global histogram) features. After modifying a single DCT coefficient from d to, say, $d+1$, there is no need to recompute the histogram of DCT coefficients, h. Instead, we can simply update it using the following rule

$$\begin{aligned} h(d) &\leftarrow h(d) - 1 \\ h(d+1) &\leftarrow h(d+1) + 1. \end{aligned}$$

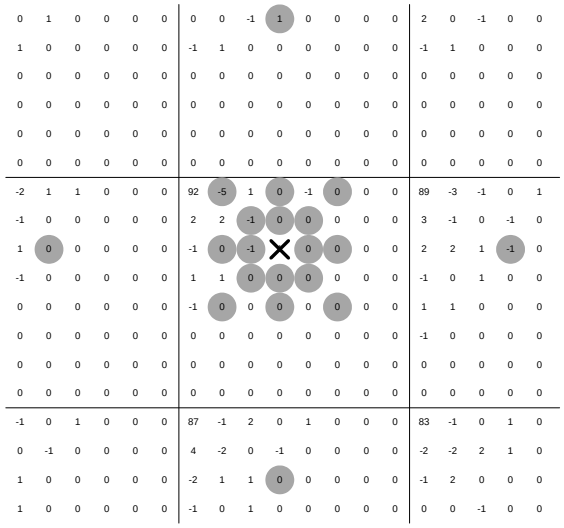

Figure 1: Portion of DCT plane with the coefficient being modified marked with a cross. The update rules for DCT domain based features require knowledge of all the highlighted coefficients.

In other words, after computing the feature vector from the cover image, its updated versions can be obtained using this differential feature computation strategy. Similar update rules can be obtained for histograms of individual DCT coefficients and for the dual histograms.

Updating the features based on higher-order statistics, such as co-occurrence matrices, variation, or Markov features, is slightly more complex. The update rules need to consider the local neighborhood of the coefficient in the DCT plane. The cross in Figure 1 marks the DCT coefficient being modified. The gray circles are the coefficients whose values enter the update rules for all 272 features that are computed from quantized DCT coefficients (all features with the exception of two blockiness features computed in the spatial domain). Note that the number of such coefficients is constant and independent of N. For example, the four coefficients from the four adjacent 8×8 blocks need to be considered to update the variation feature and the co-occurrence matrices. The circles inside the same block are needed to update the 81 Markov features.

The two blockiness features are calculated in the spatial domain as sums of discontinuities between 8×8 blocks. Thus, to update these features, we need to decompress the block that contains the modified coefficient. This requires implementing local inverse DCT transform. The amount of computations is, again, constant and independent of the number of DCT coefficients in the image N.

Because the features are calibrated to make them approximately characterizing, the differential feature computation is in reality more complicated. During calibration, the JPEG image is decompressed to the spatial domain, cropped by 4×4 pixels, and recompressed with the same quantization matrix, obtaining thus an approximation to the cover image. The final feature vector is the difference between the feature vector calculated from the JPEG image under investigation and the estimated cover image (1). The change in one DCT coefficient in the image will thus influence four

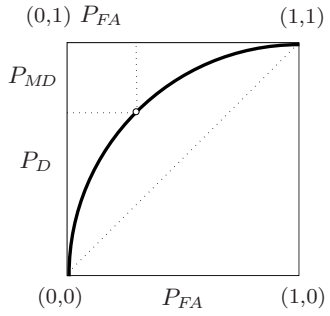

Figure 2: Detection error $P_E = \min \frac{1}{2} \cdot (P_{FA} + P_{MD})$ used for evaluating the statistical detectability of the FCM.

blocks in the cropped and recompressed image. To update the feature vector of the cropped/recompressed image, we need to update *all* DCT coefficients in these four 8×8 DCT blocks, considering also all their dependent coefficients from neighboring blocks as shown in Figure 1. Fortunately, the amount of coefficients that enter the update rules is still constant independent of N. Thus, the complexity of the differential feature computation is $O(N)$ instead of $O(N^2)$ if the feature vector was always recomputed as a whole, which is a significant savings.

4.2 Experimental results

In this section, we demonstrate that the FCM is indeed undetectable using a classifier based on the same feature set. Statistical detectability is evaluated experimentally using a blind steganalyzer implemented as a soft-margin support vector machine (SVM) with Gaussian kernel. More details about the classifier are in the original publication [28]. We used a database of 6000 JPEG images with quality factor 75 with $69,753$ nonzero DCT coefficients on average. The database was divided into $3,500$ training images and $2,500$ testing images. Thus, after creating stego images from all of them, we obtained total of $7,000$ images for training and $5,000$ images for testing. As in [33, 13, 21], the statistical detectability was measured using the minimal probability P_E of misclassification for equal prior probabilities of covers and stego images

$$P_E = \frac{P_{FA} + P_{MD}}{2}, \tag{5}$$

where P_{FA} is the probability of false alarms and P_{MD} is the probability of missed detections (see Figure 2).

Figure 3 shows the percentage of how much the embedding distortion (4) was reduced during the correction phase as a function of the relative size of $\mathcal{D}_c$. The results are averaged over all 6000 images in the database. For each tested size of $\mathcal{D}_c$, the SVM was trained separately. The relative message length was fixed to 0.10 bpac (measured with respect to the whole image and not just $\mathcal{D}_e$).

We note that by reserving only about 2% of DCT coefficients for corrections, the embedding distortion could be reduced by almost 70% from its value at the end of the embedding phase. By further enlarging $\mathcal{D}_c$, the distortion could be further reduced, but the efficiency of the corrections quickly decreases. This is because the number of DCT coef-

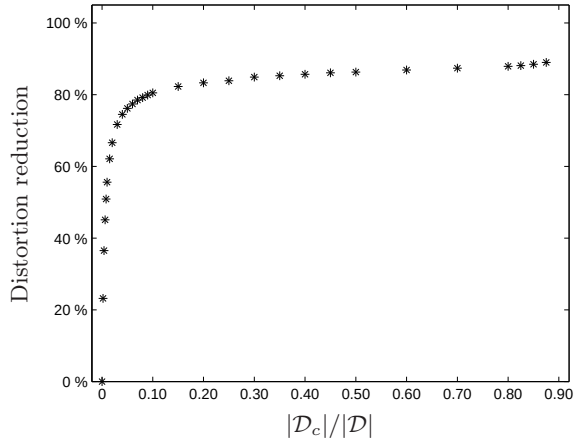

Figure 3: Feature space distortion reduction achieved during the correction phase as a function of $|\mathcal{D}_c|/|\mathcal{D}|$ averaged over 6000 images. The relative payload was fixed to 0.10 bpac.

ficients in $\mathcal{D}_c$ that are being skipped increases (their changes do not reduce the distortion norm). Furthermore, larger $\mathcal{D}_c$ implies smaller $\mathcal{D}_e$, which prevents application of more efficient codes during the embedding stage. This results in more embedding changes in $\mathcal{D}_e$ and thus a higher starting distortion for the correction phase.

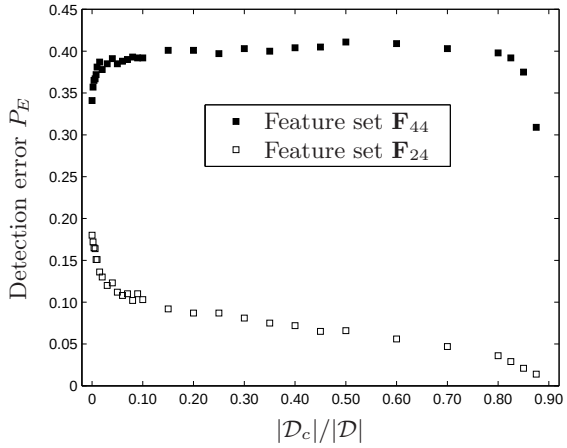

Figure 4: Error P_E for SVM steganalyzers constructed from the original feature set $\mathbf{F}_{44}$ and from the modified feature set $\mathbf{F}_{24}$ with slightly different cropping in calibration as functions of the size of $\mathcal{D}_c$ (for payload 0.10 bpac).

Figure 4 (feature set $\mathbf{F}_{44}$) shows the detection error P_E in the same experiment (for definition of symbols $\mathbf{F}_{24}$ and $\mathbf{F}_{44}$, see Section 4.3). Accepting the philosophy that it is better to make as few embedding changes as possible, we conclude that leaving about 10% for corrections is the overall best strategy for splitting $\mathcal{D}$ into $\mathcal{D}_e$ and $\mathcal{D}_c$ because the error P_E is approximately constant for $|\mathcal{D}_c|/|\mathcal{D}| \gtrapprox 0.10$.

In Figure 5, we illustrate the number of individual correction types during the correction phase still within the same experiment. We plotted the average absolute number of corrections in which the DCT coefficient was changed towards zero or away from zero (its absolute value was decreased or increased, respectively). Note that the changes towards zero lead to better distortion reduction more often than changes away from zero. This is compatible with the finding reported in [21]. Furthermore, there are more changes by 1 than by 2, which is intuitively to be expected.

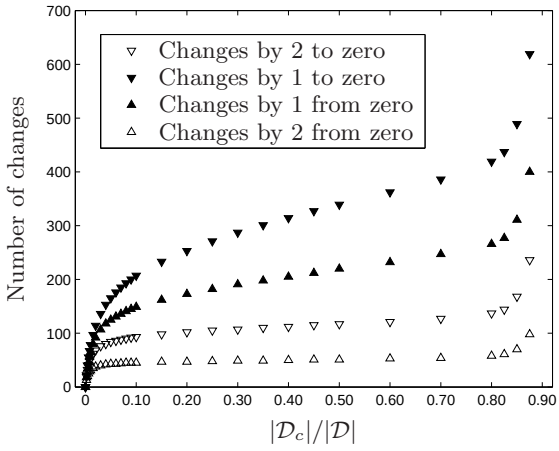

Figure 5: Absolute number of individual correction types during the correction phase for different size of $\mathcal{D}_c$ (for payload 0.10 bpac).

Figure 6 compares the performance of the FCM with the nsF5 [13] and MMx [20]. In summary, we can say that the FCM is practically undetectable using the SVM steganalyzer trained on the same features as those used for the FCM (the detection error P_E is more than 30% for payload 0.20 bpac and over 40% for shorter messages). Consequently, we conclude that the 274-dimensional feature space is not complete.

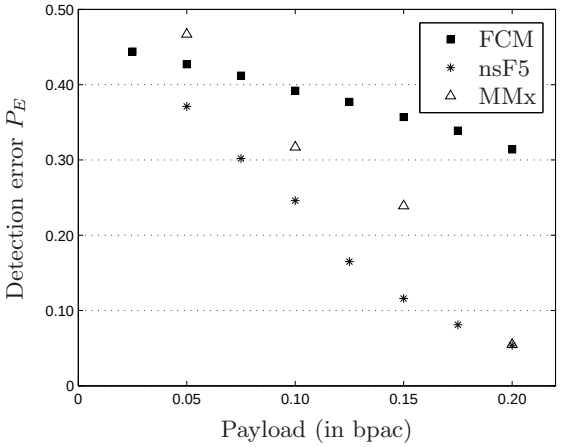

Figure 6: Steganalyzer error P_E for the FCM compared with nsF5 and MMx (for $\mathcal{D}_c = 0.10$).

4.3 Towards completeness

Even though the FCM evades detection using steganalyzers that use the same feature set, it does not necessarily mean that the FCM is a good steganographic scheme as it may be detectable using a different feature set. To prove this point, we steganalyzed the FCM using the same feature set with a slightly modified calibration process. Instead of cropping by 4 pixels in each direction, we cropped by 2 pixels in the vertical direction and by 4 pixels in the horizontal direction. Different cropping leads to a different estimate of the cover image and, consequently, to different characterizing features. We denote the original 274-dimensional feature space with 4×4 cropping by $\mathbf{F}_{44}$ and its modified version by $\mathbf{F}_{24}$. Even though the new features from $\mathbf{F}_{24}$ are strongly correlated with features from $\mathbf{F}_{44}$, because the FCM is highly targeted to $\mathbf{F}_{44}$, the steganalysis detector based on $\mathbf{F}_{24}$ is significantly more successful in detecting FCM (see Figure 7). This means that features from $\mathbf{F}_{24}$ capture different attributes of cover images that are not completely covered by features from $\mathbf{F}_{44}$. In fact, the modifications made during the correction phase of the FCM only introduce additional distortion which increases the detectability using the feature set $\mathbf{F}_{24}$ (Figure 4). In other words, the set $\mathcal{C}_{44} = \bigcap_i \mathcal{C}_i$ computed from the feature space $\mathbf{F}_{44}$ (see equation 3), and its $\mathbf{F}_{24}$ counterpart, $\mathcal{C}_{24}$, have smaller intersection than one might expect. While it is true that $\mathcal{C}_{24} \cap \mathcal{C}_{44} \supset \mathcal{C}_0$, when performing the FCM, we are not bringing the feature vector back to $\mathcal{C}_0$ but to $\mathcal{C}_{44}$.

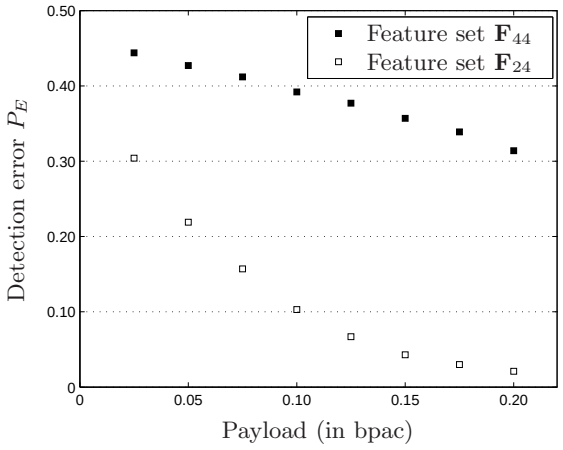

Figure 7: Error P_E for SVM steganalyzers constructed from the original feature set $\mathbf{F}_{44}$ and from the modified feature set $\mathbf{F}_{24}$ with slightly different cropping in calibration as a function of the payload (for $\mathcal{D}_c = 0.10$).

5. CONCLUSIONS

In this paper, we formalize the concept of characterizing features and complete feature sets for applications in steganography, steganalysis, and benchmarking. As a tool for testing whether or not the feature set is complete, we propose a steganographic scheme that approximately preserves the feature vector. We call this scheme the Feature Correction Method (FCM) and demonstrate its feasibility by constructing the FCM for a 274-dimensional feature set from a state-of-the-art blind steganalyzer for JPEG images. The resulting steganographic scheme was statistically undetectable using the same feature set at payloads exceeding 0.1 bits per non zero AC DCT coefficient.

The FCM method is a general concept that can be applied to other feature sets. The fact that the FCM was successful when applied to a feature set that forms the basis of a powerful steganalyzer stresses the need to employ alternative steganalysis tools (alternative feature sets) to obtain more reliable steganalysis results in practice. Indeed, we demonstrated that the FCM could be reliably detected using a steganalyzer built from a slightly modified feature set.

The issue of complete statistical description of natural images will likely remain unresolved in the near future. It is quite possible that the true dimensionality of a complete feature space is proportional to the number of pixels in the image. On the other hand, its effective dimensionality may be substantially lower if we realize that natural images consist of segments of quite structured content superimposed with a small noise component.

6. ACKNOWLEDGMENTS

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under the research grant number FA8750-04-1-0112 and 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 AFRL, AFOSR, or the U.S. Government.

7. REFERENCES

[1] I. Avcibas, M. Kharrazi, N. D. Memon, and B. Sankur. Image steganalysis with binary similarity measures. *EURASIP Journal on Applied Signal Processing*, 17:2749–2757, 2005.

[2] 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, Steganography, and Watermarking of Multimedia Contents III*, volume 4314, pages 523–531, San Jose, CA, January 22–25, 2001.

[3] I. Avcibas, N. D. Memon, and B. Sankur. Image steganalysis with binary similarity measures. In *Proceedings IEEE, International Conference on Image Processing, ICIP 2002*, volume 3, pages 645–648, Rochester, NY, September 22–25, 2002.

[4] C. Cachin. An information-theoretic model for steganography. In D. Aucsmith, editor, *Information Hiding, 2nd International Workshop*, volume 1525 of *Lecture Notes in Computer Science*, pages 306–318, Portland, OR, April 14–17, 1998. Springer-Verlag, New York.

[5] S. Dumitrescu, X. Wu, and Z. Wang. Detection of LSB steganography via sample pair analysis. In F. A. P. Petitcolas, editor, *Information Hiding, 5th International Workshop*, volume 2578 of *Lecture Notes in Computer Science*, pages 355–372, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.

[6] 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.

[7] E. Franz. Steganography preserving statistical properties. In F. A. P. Petitcolas, editor, *Information Hiding, 5th International Workshop*, volume 2578 of *Lecture Notes in Computer Science*, pages 278–294, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.

[8] J. Fridrich. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of *Lecture Notes in Computer Science*, pages 67–81, Toronto, Canada, May 23–25, 2004. Springer-Verlag, New York.

[9] J. Fridrich and M. Goljan. On estimation of secret message length in LSB steganography in spatial domain. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306, pages 23–34, San Jose, CA, January 19–22, 2004.

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

[11] J. Fridrich, M. Goljan, and D. Soukal. Perturbed quantization steganography. *ACM Multimedia System Journal*, 11(2):98–107, 2005.

[12] J. Fridrich, M. Goljan, and D. Soukal. Wet paper codes with improved embedding efficiency. *IEEE Transactions on Information Forensics and Security*, 1(1):102–110, 2006.

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

[14] 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.

[15] J. J. Harmsen and W. A. Pearlman. Steganalysis of additive noise modelable information hiding. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents V*, volume 5020, pages 131–142, Santa Clara, CA, January 21–24, 2003.

[16] S. Hetzl and P. Mutzel. A graph–theoretic approach to steganography. In J. Dittmann, S. Katzenbeisser, and A. Uhl, editors, *Communications and Multimedia Security, 9th IFIP TC-6 TC-11 International Conference, CMS 2005*, volume 3677 of *Lecture Notes in Computer Science*, pages 119–128, Salzburg, Austria, September 19–21, 2005.

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

[18] A. D. Ker. Steganalysis of LSB matching in grayscale images. *IEEE Signal Processing Letters*, 12(6):441–444, June 2005.

[19] M. Kharrazi, H. T. Sencar, and N. D. Memon. Benchmarking steganographic and steganalytic techniques. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 252–263, San Jose, CA, January 16–20, 2005.

[20] Y. Kim, Z. Duric, and D. Richards. Modified matrix encoding technique for minimal distortion 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 314–327, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.

[21] J. Kodovský and J. Fridrich. Influence of embedding strategies on security of steganographic methods in the JPEG domain. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, pages 2 1–2 13, San Jose, CA, January 27–31, 2008.

[22] S. Lyu and H. Farid. Steganalysis using color wavelet statistics and one-class support vector machines. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306, pages 35–45, San Jose, CA, January 19–22, 2004.

[23] S. Lyu and H. Farid. Steganalysis using higher-order image statistics. *IEEE Transactions on Information Forensics and Security*, 1(1):111–119, 2006.

[24] Y. Miche, B. Roue, A. Lendasse, and P. Bas. A feature selection methodology for steganalysis. In B. Günsel, A. K. Jain, A. M. Tekalp, and B. Sankur, editors, *Multimedia Content Representation, Classification and Security, InternationalWorkshop*, volume 4105 of *Lecture Notes in Computer Science*, pages 49–56, Istanbul, Turkey, September 11–13, 2006. Springer-Verlag.

[25] H. Noda, M. Niimi, and E. Kawaguchi. Application of QIM with dead zone for histogram preserving JPEG steganography. In *Proceedings IEEE, International Conference on Image Processing, ICIP 2005*, pages II – 1082–5, Genova, Italy, September 11–14, 2005.

[26] A. Orsdemir, H. O. Altun, G. Sharma, and M. F. Bocko. Steganalysis-aware steganography: Statistical indistinguishability despite high distortion. 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 15 1–15 9, San Jose, CA, January 27–31, 2008.

[27] T. Pevný and J. Fridrich. Towards multi-class blind steganalyzer for JPEG images. In M. Barni, I. J. Cox, T. Kalker, and H. J. Kim, editors, *International Workshop on Digital Watermarking*, volume 3710 of *Lecture Notes in Computer Science*, Siena, Italy, September 15–17, 2005. Springer-Verlag, Berlin.

[28] T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 3 1–3 14, San Jose, CA, January 29 – February 1, 2007.

[29] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323–335, Proceedings of the ACM Symposium on Applied Computing, August 13–17, 2001.

[30] P. Sallee. Model-based methods for steganography and steganalysis. *International Journal of Image Graphics*, 5(1):167–190, 2005.

[31] A. Sarkar, K. Solanki, and B. S. Manjunath. Further study on YASS: Steganography based on randomized embedding to resist blind steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, pages 16–31, San Jose, CA, January 27–31, 2008.

[32] 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.

[33] K. Solanki, A. Sarkar, and B. S. Manjunath. YASS: Yet another steganographic scheme that resists blind steganalysis. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, Lecture Notes in Computer Science, pages 16–31, Saint Malo, France, June 11–13, 2007. Springer-Verlag, New York.

[34] K. Solanki, K. Sullivan, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. Provably secure steganography: Achieving zero K-L divergence using statistical restoration. In *Proceedings IEEE, International Conference on Image Processing, ICIP 2006*, pages 125–128, Atlanta, GA, October 8–11, 2006.

[35] R. Tzschoppe, R. Bäuml, J. B. Huber, and A. Kaup. Steganographic system based on higher-order statistics. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents V*, volume 5020, pages 156–166, Santa Clara, CA, January 21–24, 2003.

[36] Y. Wang and P. Moulin. Statistical modelling and steganalysis of DFT-based image steganography. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 2 1–2 11, San Jose, CA, January 16–19, 2006.

[37] A. Westfeld. High capacity despite better steganalysis (F5 – a steganographic algorithm). In I. S. Moskowitz, editor, *Information Hiding, 4th International Workshop*, volume 2137 of *Lecture Notes in Computer Science*, pages 289–302, Pittsburgh, PA, April 25–27, 2001. Springer-Verlag, New York.

[38] A. Westfeld and R. Böhme. Exploiting preserved statistics for steganalysis. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of *Lecture Notes in Computer Science*, pages 82–96, Toronto, Canada, May 23–25, 2004. Springer-Verlag, Berlin.

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