Calibration Revisited

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

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

ABSTRACT

Calibration was first introduced in 2002 as a new concept to attack the F5 algorithm [3]. Since then, it became an essential part of many feature-based blind and targeted steganalyzers in JPEG as well as spatial domain. The purpose of this paper is to shed more light on how, why, and when calibration works. In particular, this paper challenges the thesis that the purpose of calibration is to estimate cover image features from the stego image. We classify calibration according to its internal mechanism into several canonical examples, including the case when calibration hurts the detection performance. All examples are demonstrated on specific steganographic schemes and steganalysis features. Furthermore, we propose a modified calibration procedure that improves practical steganalysis.

Categories and Subject Descriptors

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

General Terms

Algorithms, Security, Theory

Keywords

Calibration, Steganalysis, Features, YASS

1. INTRODUCTION

The concept of calibration was used for the first time in [3], where the authors introduced it as a method to estimate the cover image histogram from the stego image when attacking the steganographic algorithm F5 [24]. The same concept was later used in [4] to successfully attack OutGuess [19]. Calibration was also shown to improve detection accuracy of feature-based blind steganalysis [17] because it provides the steganalyst with a reference image from which the baseline value of features can be obtained with the net benefit of decreasing features' image-to-image variations.

The first blind detector that incorporated calibration used 23 calibrated features [1]. Here, the term calibrated feature was the L_1 norm of the difference between a specific (non-calibrated) functional calculated from the stego image and the same functional obtained from a reference image. An extended version of this feature vector that appeared in [17] replaced the L_1 norm with individual differences. The 274-dimensional feature vector, which we abbreviate in this paper as PEV-274 was obtained by considering several different models for DCT coefficients and using the sample statistics of the models as features.

Although calibration was originally introduced for the JPEG domain, there were attempts to use this powerful concept in the spatial domain as well [9, 10]. In fact, the image obtained using the predictor in WS steganalysis [2] can also be considered as a reference image even though it was not formulated within the framework of calibration.

Despite the fact that calibration has been shown to improve steganalysis, the authors are not aware of any study that would investigate its limitations and explain its inner workings on a deeper level. Moreover, there seem to exist some fallacies as to how calibration works. Going back to the original paper [3], calibration was credited with increasing the features' sensitivity to embedding while decreasing their image-to-image variations. While this is in principle correct, this beneficial effect of calibration does not have to be solely due to the fact that the reference image provides an estimate of cover image features. Indeed, when the payload is small, the best estimate of the cover image features are the features derived from the stego image itself. This is quite strikingly apparent in WS steganalysis [2] where the predictor values are on average much further from the cover than stego. What is more important here is that the embedding changes are on average *erased* from the reference image while still providing an image that is close to the cover.

The main contribution of this paper is in classifying different forms of calibration, clarifying their inner workings, and confirming our claims experimentally. Furthermore, we use the newly gained insight and propose a modified calibration procedure that improves practical steganalysis.

The paper is organized as follows. First, in Section 2 we describe the setup of our experiments, including the image database, the machine learning tool, and the performance criterion used to evaluate the accuracy of steganalysis. In Section 3 after explaining the process of calibration as introduced in [17], we motivate our research by presenting a

MM&Sec'09, September 7–8, 2009, Princeton, New Jersey, USA.
Copyright 2009 ACM 978-1-60558-492-8/09/07 ...$10.00.

Feature	Dimensionality
Global histogram $\mathbf{H}_l$	11
Five AC histograms $\mathbf{h}_l^{ij}$	5×11
11 dual histograms $\mathbf{g}_{ij}^d$	11×9
Variation V	1
Two types of blockiness $\mathbf{B}_\alpha$	2
Co-occurrence matrix $\mathbf{C}_{ij}$	5×5
Markov features $\mathbf{M}_{ij}$	9×9

Table 1: List of all features from the PEV-274 feature set.

rather surprising comparison of steganalytic results when we do and do not involve calibration. Continuing in Section 4, we first introduce all necessary notation and then classify calibration according to its internal mechanism into several canonical examples. Section 5 contains experimental confirmation of the proposed classification for several steganographic schemes and a fixed steganalysis feature set. Equipped with a new insight, Section 6 outlines a generalized view of calibration that improves steganalysis in practice, a claim which is demonstrated on selected state-of-the-art steganographic schemes. Section 7 concludes the paper and summarizes our contributions.

2. SETUP OF EXPERIMENTS

We wish to motivate our study and verify our claims using selected experiments with existing steganographic schemes on real-life imagery and with a specific blind steganalyzer. Therefore, before presenting the technical arguments, we describe our experimental setup, including the image database, the feature set, the machine learning tool, and the quantity for evaluating the performance of steganalysis.

2.1 Image Database

As reported in [13], the properties of images in the database used for testing (e.g., their size, JPEG quality factor, or the average number of nonzero DCT coefficients) should accompany every experimental steganalysis results since these factors may influence the results substantially. Our controlled image database consists of 6500 photographs in native resolution coming from several different camera sources (more than 20 different camera models spanning five camera brands). All images were acquired in a raw format, converted to 8-bit grayscale, and resized using bilinear interpolation so that the smaller side of the image was 512 pixels (aspect ratio preserved). For algorithms that embed in JPEG images, all images in the database were compressed with the JPEG quality factor 75. The average number of nonzero AC DCT coefficients in each image was $65,887$.

2.2 PEV-274 Feature Set

All our experiments involve the PEV-274 feature set [17] consisting of 193 DCT based features and 81 Markov features [21]. Table 1 lists the individual feature types and symbols using which we refer to them in this paper. We adopted the same notation as in the original publication [17], where a more detailed description of the feature set can be found. This feature set was selected because it is very popular and because it provides reliable steganalysis results. This set has been also previously used as an oracle for design of steganographic schemes [15, 22, 20], for performance comparisons [14, 13], and benchmarking [18].

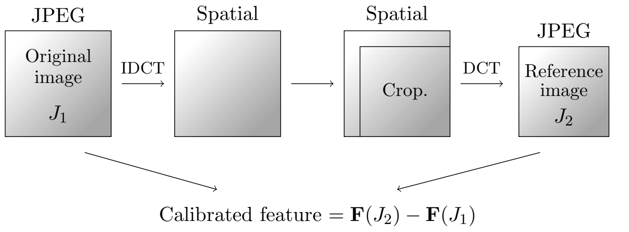

Figure 1: The process of calibration as incorporated in the PEV-274 feature set.

2.3 Performance Evaluation

All steganalyzers were implemented as binary classifiers realized using a soft-margin support vector machine with a Gaussian kernel. The hyperparameters were were optimized by a gradient method [7] on the training set. The image database was randomly divided into two halves; one half was used for traning and the other for testing.

The steganalyzer performance is evaluated using the minimal probability of misclassification, P_E, for equal prior probabilities of covers and stego images

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

where P_{FA} is the probability of false alarms, P_{MD} is the probability of missed detections and the minimum is taken over the whole ROC curve. This measure was previously used in [22, 14, 11].

2.4 Steganographic Schemes

We use the following state-of-the-art steganographic schemes: an improved version of the F5 algorithm [24] called nsF5 [14], the MME3 algorithm [12], Jsteg [23], JP Hide & Seek, Steghide [6], and YASS [22, 20]. The nsF5 algorithm is currently the most secure algorithm for JPEG images that does not utilize side-information at the embedder. In nsF5, the problem of shrinkage is eliminated by incorporating wet paper codes (WPC) with improved embedding efficiency [5]. The codes improve the security of F5 because the same payload can be embedded using fewer embedding changes [14]. The MME3 algorithm provides the best security at payloads smaller than 0.1 bits per nonzero AC DCT coefficient (bpac).[1] It utilizes side-information at the embedder in the form of the uncompressed image. Jsteg is historically the first steganographic algorithm for JPEG and is easily detectable. JP Hide&Seek (JPHS) is a more complicated modification of Jsteg, while Steghide preserves the global histogram of the cover JPEG image.

The algorithm YASS works completely differently from all the other algorithms because it does not embed information in the domain of quantized DCT coefficients. Instead, it embeds data robustly in an alternative domain. We use eight different configurations for YASS, including both the

[1] These claims are taken from [14].

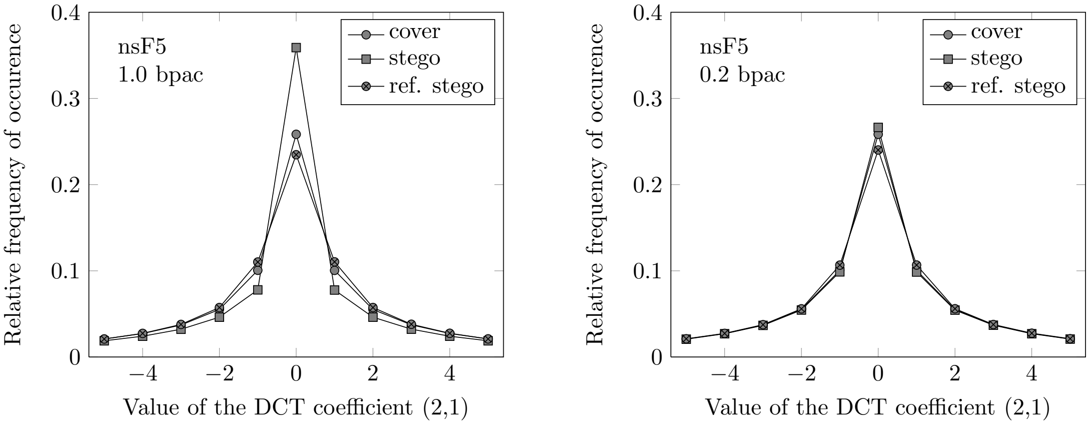

Figure 2: The effect of nsF5 embedding on the histogram of the DCT mode (2,1) for payloads 1.0 bpac (left) and 0.2 bpac (right). The graph was obtained as an average over all 6500 images in our database.

original version of the algorithm published in [22] as well as its modifications [20]. The configurations are described in the appendix.

3. MOTIVATION

In this section, we first review the process of calibration as used in the PEV-274 feature set. Then, we reproduce and extend the experiment presented in [3] to show that the effect of calibration is likely different from the one claimed in this publication. To further motivate our study, we present the results of steganalysis of YASS when the calibration is *turned off*.

3.1 Calibration in Steganalysis of F5

Calibration starts with a JPEG image J_1 under investigation, decompresses it into the spatial domain using inverse DCT, crops by four pixels in both directions, and recompresses the cropped image using the quantization matrix of J_1. As a result, a different JPEG image, J_2, is obtained.[2] In this paper, we refer to J_2 as the *reference image*. Given a feature $\mathbf{F}$, which is a mapping that assigns a feature vector to each image, its calibrated version is the difference $\mathbf{F}(J_2) - \mathbf{F}(J_1)$.[3] Figure 1 shows a pictorial explanation of this process.

In [3], the authors include the following heuristic explanation why calibration works:

> “Cropping the image produces an image that is perceptually similar to the original and therefore its DCT coefficients should have approximately the same statistical properties as the DCT coefficients of the cover image. Furthermore, the spatial shift by four pixels ensures that the 8×8 grid of recompression *does not see* the previous JPEG compression and thus the obtained DCT coefficients are not influenced by previous quantization (and possible embedding) in the DCT domain. Therefore, the statistics of the reference image (its feature vector $\mathbf{F}$) can be seen as an approximation of the cover image statistics.”

This claim was demonstrated on the histogram of coefficients from an individual DCT mode after full F5 embedding with 1.0 bpac. We reproduced this experiment using nsF5. Figure 2 (left) shows the resulting histograms.

When a very large payload is embedded, the reference image histogram may, indeed, be closer to the cover image histogram. However, the situation is quite different for smaller payloads. Figure 2 (right) shows the impact of nsF5 embedding on the same histogram for a payload of 0.2 bpac, which corresponds to the change rate 0.04.[4] Even though this payload is still rather large when compared with the embedding capacity of nsF5, we can see that the histogram of the reference image no longer approximates the cover image histogram. In fact, the stego image histogram is a better approximation. Quantifying this observation using the L_2 norm between histograms, for payload of 1.0 bpac, the reference image histogram is on average (over all images in our database) 3.3 times closer to the cover image histogram than the stego image histogram. On the other hand, for a smaller payload of 0.2 bpac, the stego image histogram is 2.9 times closer to the cover image histogram than the reference image histogram.

We would like to point out that even though the reference image does not really approximate the cover image (or its statistics), calibration may still improve steganalysis, which goes against the intuitive explanation of calibration as provided in [3]. To demonstrate this, we performed steganalysis of nsF5 for relative payload 0.2 bpac with the following 11-dimensional global histogram of DCT coefficients as the feature $\mathbf{F} = (\mathbf{H}_{-5}, \dots, \mathbf{H}_5)$. A non-calibrated feature vector leads to minimal combined error rate $P_E = 0.46$,

[2] Note that cropping by 4 pixels in the spatial domain is by far not the only way how to perform calibration of JPEG images. As suggested in [1], very similar results are indeed obtained by applying a slight amount of rotation or resizing since such operations also desynchronize the original 8×8 grid, erasing thus the impact of embedding in the DCT domain. Image J_2 can have slightly different dimensions from the original J_1, but this does not affect further feature extraction procedure because the features are normalized.

[3] Note that $\mathbf{F}$ was called a functional in [3, 17].

[4] Recall that nsF5 uses WPC with improved embedding efficiency.

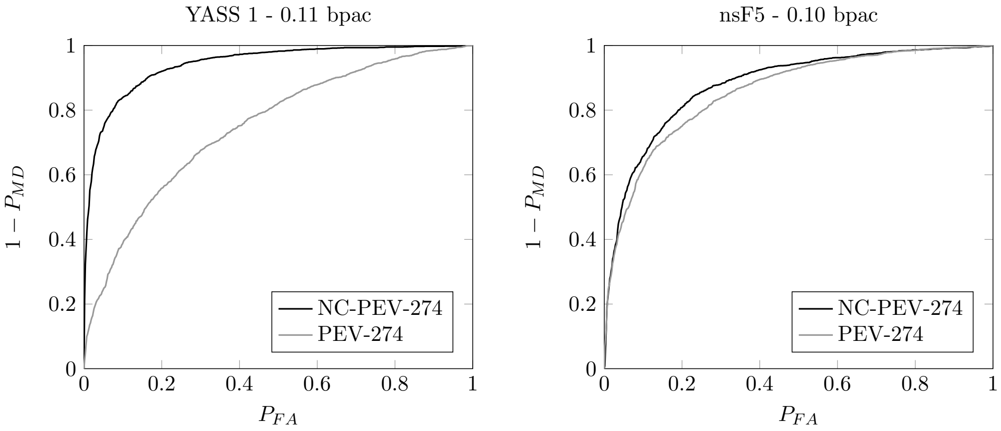

Figure 3: ROC curves for YASS 1 (left) and nsF5 (right) using the PEV-274 feature set and its non-calibrated version NCPEV-274. The setting of YASS 1 is in the appendix.

which is basically random guessing. Note that by a non-calibrated version we mean $\mathbf{F}(J_1)$ instead of the difference $\mathbf{F}(J_2) - \mathbf{F}(J_1)$. Using a calibrated version of $\mathbf{F}$, the error is reduced to $P_E = 0.28$.

3.2 Steganalysis of YASS

The steganographic algorithm YASS was developed with the intention to disable the positive effect of calibration in blind steganalysis. The steganalysis results independently reported in [20, 13, 16] indicate that the feature set PEV-274 indeed, cannot detect YASS reliably. However, when a non-calibrated version of the PEV-274 feature set is used (we denote it here as NCPEV-274), YASS becomes significantly more detectable (see Figure 3 (left)). For comparison, Figure 3 (right) shows the ROC curves for nsF5 with a similar payload. We remark here that non-calibrated feature sets other than PEV-274 were also shown to detect YASS relatively reliably [16].

The experiments in this section provoke numerous important questions. How exactly does calibration affect statistical detectability of steganographic algorithms and why does it fail for YASS? Generally, under what conditions does calibration help and when does it make steganalysis worse? What is the real purpose of calibration if it is not to approximate the cover image? It appears that in certain cases omitting calibration may improve steganalysis, which suggests a potential improvement by calibrating only selected features.

4. ANALYSIS OF CALIBRATION

The rather surprising properties of calibration presented in the previous section motivated us to further analyze possible impacts of calibration on steganalysis. In this section, we discuss several scenarios for calibration and illustrate them with examples from literature. At the end of the section, we introduce a framework that will enable us to quantify our insight.

4.1 Notation

The space of all possible images will be denoted by $\mathcal{X}$. Since the dimension of $\mathcal{X}$ is usually very large for the steganalyst to operate with (e.g., $\mathcal{X} = \{0, \ldots, 255\}^{M \times N}$ for 8-bit grayscale images of dimensions up to $M \times N$), some lower-dimensional projection is usually applied, which can be achieved by representing an image with a feature vector. Denoting the feature space by $\mathcal{F}$, the corresponding feature map is $\mathbf{F} : \mathcal{X} \to \mathcal{F}$. Typically, $\mathcal{F} = \mathbb{R}^n$ with $n \approx 10^2 - 10^3$. Note that the reliability of steganalysis is highly dependent on the mapping $\mathbf{F}$. In general, $\mathbf{F}$ should be sensitive to the embedding changes and ideally no information should be lost by projecting $\mathcal{X}$ to $\mathcal{F}$, as far as the distinguishability between cover and stego classes is concerned. The problem of proper feature selection is not of our concern in this paper.

Cover images are typically modeled as a random variable on $\mathcal{X}$. The process of embedding a secret message is realized by an embedding mapping, $\text{Emb} : \mathcal{X} \to \mathcal{X}$, that may be parametrized by a stego key or a change rate. We use lower case symbols c and s to denote the cover and the corresponding stego image, $s = \text{Emb}(\text{c})$. The goal of steganalysis is to distinguish between the distributions of $\mathbf{F}(c)$ and $\mathbf{F}(s)$.

The central concept in calibration is the *reference transform* $r : \mathcal{X} \to \mathcal{X}$, which maps the image $x \in \mathcal{X}$ to its reference image $r(x) \in \mathcal{X}$. One example of such a mapping r is the spatial-domain cropping followed by compression shown in Figure 1. In steganalysis of ±1 embedding [8, 9], the reference mapping was realized by resizing by a factor of two. The prediction filter in WS steganalysis [2] can also be interpreted as a reference transformation. We denote the feature vector of the reference image as $\mathbf{F}_r = \mathbf{F} \circ r$, where $\circ$ stands for the composition of mappings. We refer to $\mathbf{F}_r$ as the *reference feature*. The *calibrated feature* is defined simply as the difference between the feature vectors extracted from the image and its reference version

$$\mathbf{F}_{cal}(x) \triangleq \mathbf{F}_r(x) - \mathbf{F}(x) \quad \forall x \in \mathcal{X}. \qquad (1)$$

Figure 4 clarifies the introduced notation.

4.2 Examples

In this section, we present a series of canonical examples of how the reference feature mapping $\mathbf{F}_r$ might look like and how it influences the distinguishability between the classes of cover and stego features. Our goal is to determine the properties that $\mathbf{F}_r$ should possess to improve steganalysis. Note that according to the definition of $\mathbf{F}_r$, it is fully de-

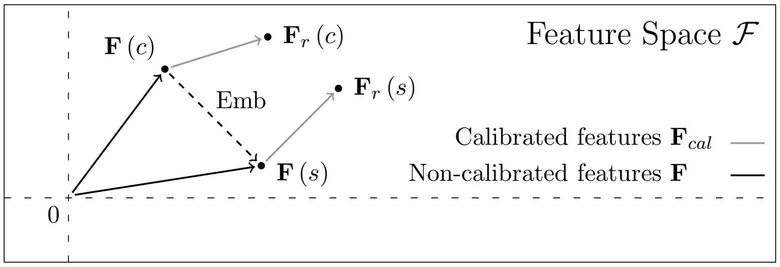
Feature Space $\mathcal{F}$
$\mathbf{F}(c)$
$\mathbf{F}_r(c)$
$\mathbf{F}_r(s)$
Emb
$\mathbf{F}(s)$
Calibrated features $\mathbf{F}_{cal}$
Non-calibrated features $\mathbf{F}$
0

Figure 4: A diagram showing the cover- and stego-image features $\mathbf{F}(c)$, $\mathbf{F}(s)$, with their corresponding reference features $\mathbf{F}_r(c)$, $\mathbf{F}_r(s)$.

termined by the feature mapping $\mathbf{F}$ and the reference transform r. (A possible generalized point of view is discussed in Section 6.) Follow Figure 5 for a schematic illustration of individual examples.

Example 1. Parallel Reference

In this first example, $\mathbf{F}_r(x) = \mathbf{F}(x) + \mathbf{F}^\star$, where $\mathbf{F}^\star$ is some specific feature vector. In other words, calibration can be seen as a constant feature-space shift. As a result, $\mathbf{F}_{cal}(x) = \mathbf{F}_r(x) - \mathbf{F}(x) = \mathbf{F}^\star \ \forall x \in \mathcal{X}$. Therefore, applying calibration causes a complete failure of steganalysis because the classes of cover and stego images become indistinguishable. We call this situation *parallel reference* since actions of $\mathbf{F}_r$ on cover and stego can be seen as parallel shifts.

In practice, the shift will not be the same for every image. Nevertheless, even when the shifts $\mathbf{F}^\star$ differ from image to image, following some distribution, calibration still fails. According to our experiments, most PEV-274 features are of this type when detecting the steganographic algorithm YASS (see the results in Section 5.2).

Example 2. Cover Estimate

Here, the reference transform r maps each stego image to an image $r(s) = \hat{c}$ whose feature approximates the cover image feature while $\mathbf{F}_r(c) \approx \mathbf{F}(c)$ for the cover image. Symbolically, $\mathbf{F}_r(s) = \mathbf{F}(\hat{c}) \approx \mathbf{F}_r(c) \approx \mathbf{F}(c)$. Therefore, $\mathbf{F}_{cal}(c) \approx 0$ and $\mathbf{F}_{cal}(s) \neq 0$, provided the stego-image feature differs from the cover-image feature, which is the very basic requirement for the feature mapping $\mathbf{F}$. Note that this scenario stood behind the original idea of calibration – to come up with a good cover-image estimate [3, 8, 9].

Provided that the reference cover image $\hat{c} = r(s)$ leads to an accurate cover-feature estimate, $\mathbf{F}(\hat{c}) \approx \mathbf{F}(c)$, calibration may improve steganalysis depending on how much different $\mathbf{F}(s)$ is from $\mathbf{F}(c)$. It is easy to see that if $\mathbf{F}(s)$ is close to $\mathbf{F}(c)$, the detectability might get actually worse. In the extreme case when $\mathbf{F}(c) = \mathbf{F}(s)$ (embedding preserves the feature vector), calibration of this form would not make any difference (covers would still be indistinguishable from stego images) unless $\mathbf{F}_{cal}(c)$ and $\mathbf{F}_{cal}(s)$ exhibit different statistical properties (while still $\mathbf{F}_{cal}(c) \approx \mathbf{F}_{cal}(s) \approx 0$). This situation is covered by Example 5.

Example 3. Stego Estimate

This is a complementary situation to the previous example in which r provides an estimate of the stego feature instead of the cover feature. In other words, $r(s) = \hat{s}$, such that $\mathbf{F}_r(c) \approx \mathbf{F}_r(s) = \mathbf{F}(\hat{s}) \approx \mathbf{F}(s)$. In this case, calibration would work equally well. In certain cases, a practical form of this example may be realized by repetitive embedding, when the feature value changes significantly when applied to the cover image, while it has a much smaller effect on stego images. This form of calibration may be especially useful for attacking idempotent embedding operations, such as LSB embedding. A real-life example of this scenario is the targeted attack on OutGuess [4].

Before we proceed with the next example, note that in this scenario (and in the previous scenario where r provided a cover-feature estimate) the actual value of $\mathbf{F}(s)$ is not important, provided it is far enough from $\mathbf{F}(c)$ in terms of distance in $\mathcal{F}$. Especially note that we do not require the embedding operation to shift the feature vector consistently in one direction. Provided the embedding operation indeed shifts the feature vector of the given image in the feature space consistently in one direction, we consider the next situation.

Example 4. Eraser

Here, the reference image does not provide estimates of cover- or stego-image features. Instead, we require

1. $\mathbf{F}_r(c) \approx \mathbf{F}_r(s) \triangleq \mathbf{F}_w$, the reference cover- and stego-image features should be close to each other, ideally identical.

2. $\mathbf{F}_w$ should be *close enough* to both $\mathbf{F}(c)$ and $\mathbf{F}(s)$.

Requirement 1 means that r has to be robust w.r.t. embedding changes. Alternatively, we will say that r *erases embedding changes* (hence the name for this scenario). Requirement 2 ensures that the calibration is non-trivial in the following sense. Suppose r trivially maps all images to one specific image $y \in \mathcal{X}$. Consequently, $\mathbf{F}_r(x) = \mathbf{F}(y) = const.$, $\forall x \in \mathcal{X}$. Even though the first requirement is ideally satisfied, the calibrated features $\mathbf{F}_{cal}$ defined by (1) are just shifted (and negative) versions of the original features $\mathbf{F}$ and calibration has no effect on the distinguishability between c and s. Therefore, $\mathbf{F}_w$ should be close to both $\mathbf{F}(c)$ and $\mathbf{F}(s)$. Furthermore, the closer we are with $\mathbf{F}_w$ to the original cover- and stego-image features, the smaller the variance of $\mathbf{F}_w$ is and the better detection we can expect.

We stress that in this case the requirement of $\mathbf{F}(s)$ being different enough from $\mathbf{F}(c)$ is not sufficient. In order to make calibration work here, we require the embedding shift $\mathbf{F}(c) \rightarrow \mathbf{F}(s)$ to be consistent in terms of direction.

Example 5. Divergent Reference

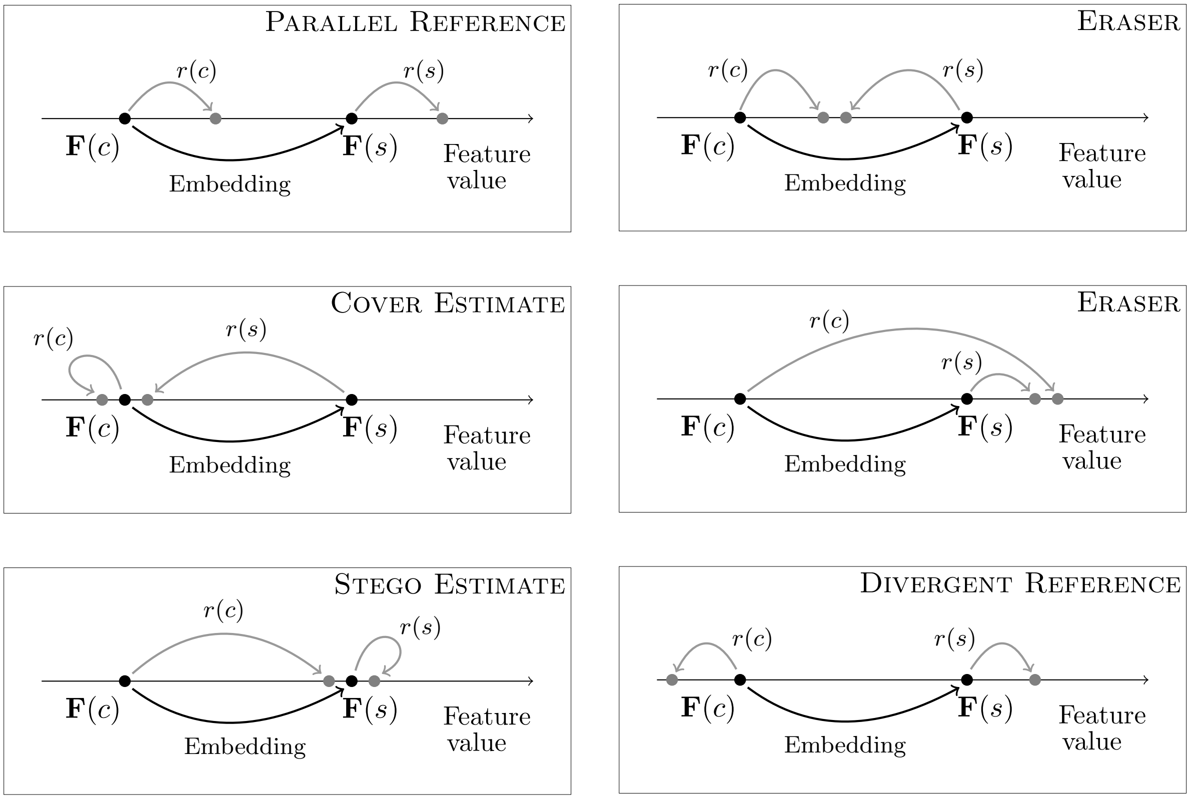

Figure 5: Illustration of individual examples from Section 4.2.

By divergent reference, we mean the situation when $\mathbf{F}_r(c) = \mathbf{F}(c) - \mathbf{F}_1,\ \mathbf{F}_r(s) = \mathbf{F}(s) - \mathbf{F}_2,\ \mathbf{F}_1 \neq \mathbf{F}_2$. In other words, the action of the reference mapping can be interpreted as a feature space shift of the original feature vector $\mathbf{F}(x)$ to a different direction depending on whether the input x is a cover or a stego image. Therefore, the resulting calibrated feature will be $\mathbf{F}_1$ for the cover image and $\mathbf{F}_2$ for the stego image (implying perfect detectability). Remarkably, in this example calibration will work even when the steganography preserves the feature vector, $\mathbf{F}(c) = \mathbf{F}(s)$, because the input to $\mathbf{F}_r(x)$ is the whole image $x \in \mathcal{X}$ and not just the feature vector!

In practice, it is not possible to achieve exactly the same shift for every cover and stego image as in this case $\mathbf{F}_{cal}$ would basically serve as a detector itself, returning the label $\mathbf{F}_1$ for cover and $\mathbf{F}_2$ for stego. Modeling $\mathbf{F}_1$ and $\mathbf{F}_2$ as random variables in $\mathcal{F}$, provided the distributions of $\mathbf{F}_1$ and $\mathbf{F}_2$ differ, it might be still beneficial to calibrate. A good example of this scenario is the situation when we use the histogram bins of zeros and ones to attack Jsteg. Because Jsteg preserves the counts of zeros and ones, $\mathbf{F}(c) = \mathbf{F}(s)$, the features themselves are useless for steganalysis of Jsteg. However, their calibrated versions improve the distinguishability between cover and stego features because the reference mapping r reacts differently to cover and stego images (see Cases 5.1 and 5.2 in Section 5.2).

4.3 Framework for Calibration

The five canonical examples presented in the previous subsection illustrate different principles how calibration may work and when it is useful and why. We now explain a framework within which the examples can be formulated and quantified. Modeling cover images as a random variable c on $\mathcal{X}$, the cover feature vector $\mathbf{F}(\mathsf{c})$ is a random variable on $\mathcal{F}$ whose central tendency and spread will be described using robust statistics, median $\mathbf{m}_c$ and Median Absolute Deviation (MAD) M_c:

$$\mathbf{m}_c = \text{median}\left[\mathbf{F}(\mathsf{c})\right], \quad (2)$$

$$M_c = \text{median}\left[\|\mathbf{F}(\mathsf{c}) - \mathbf{m}_c\|\right], \quad (3)$$

both calculated over all cover images in our database. Note that M_c is a scalar quantity while $\mathbf{m}_c$ is generally a vector because the median is applied to each coordinate of $\mathbf{F}(\mathsf{c})$. The symbol $\|\cdot\|$ denotes the L_2 norm. The steganographic embedding, $s = \text{Emb}(c)$, is modeled as a shift $\mathbf{F}(c) \to \mathbf{F}(s)$ in the feature space represented as the difference $\mathbf{F}(s) - \mathbf{F}(c)$, which we again consider as a vector random variable with median $\mathbf{m}_e$ and MAD M_e. This time, the random variable is taken over covers, stego keys, and messages, all distributed uniformly on their corresponding spaces. Note that even if the embedding shift $\mathbf{F}(c) \to \mathbf{F}(s)$ is truly random, or even if there is no shift at all, it can still be described by $\mathbf{m}_e$ and M_e, and calibration might still work (see Example 5 in Section 4.2, divergent reference).

Next, we consider the process of cover-image calibration as another feature space shift, $\mathbf{F}(c) \to \mathbf{F}_r(c)$, with the difference $\mathbf{F}_r(c) - \mathbf{F}(c)$ with median $\mathbf{m}_{rc}$ and MAD M_{rc}. Similarly, we use $\mathbf{m}_{rs}$ and M_{rs} as statistical descriptors of the stego-calibration shift $\mathbf{F}(s) \to \mathbf{F}_r(s)$. Here, we need to keep in mind that $\mathbf{F}_r = \mathbf{F} \circ r$ and that its domain is, in fact, the original space $\mathcal{X}$. We can think of the image x as a side-information for the feature space transform $\mathbf{F}(x) \rightarrow \mathbf{F}_r(x)$.

Finally, in some situations it might be useful to view $\mathbf{F}_r(s)$ with respect to $\mathbf{F}_r(c)$, as the shift $\mathbf{F}_r(c) \rightarrow \mathbf{F}_r(s)$ with median $\mathbf{m}_q$ and MAD M_q. This, indeed, makes sense because the reference features of cover and stego images are often required to be close to each other (with the exception of Example 5). Since a one dimensional sketch would be less informative, we illustrate the introduced concepts in two dimensions in Figure 6.

The benefit of this framework is that it laid out entirely in the feature space $\mathcal{F}$. In Section 5, we experimentally justify the usefulness of this framework for modeling the effects of calibration on steganalysis of the steganographic schemes from Section 2.4.

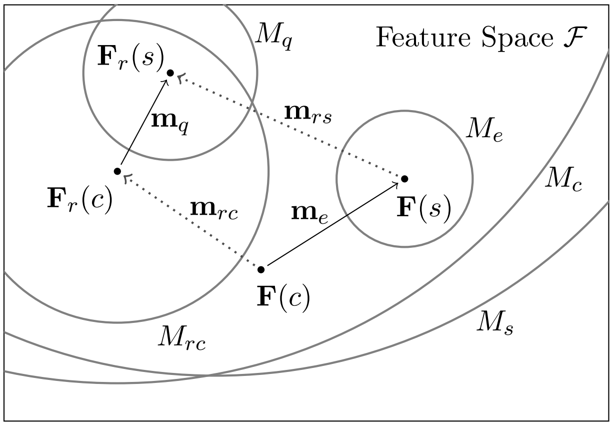

Figure 6: Two-dimensional illustration of the feature-space model as introduced in Section 4.3.

5. EXPERIMENTAL PART

In this section, we study the effects of calibration on individual features from the PEV-274 feature set. We do so for images from the database described in Section 2.1 and steganographic schemes listed in Section 2.4. For each steganographic method and relative payload, we obtained 6500 cover images and the same number of stego images. For both cover and stego images, their corresponding reference images were created using the spatial-domain cropping as explained in Figure 1. The non-calibrated NCPEV-274 features were extracted from all cover and stego images and their corresponding reference versions. All features were scaled so that cover-image features exhibited unit variance. Finally, the sample values of the median and MAD, $m_e, M_e, m_q, M_q, m_{rc}, M_{rc}, m_{rs}$, and M_{rs}, were computed separately for every feature (see Figure 6). Here, we use non-boldface symbols because the medians will be always computed for scalar quantities.

5.1 On the Importance of Baseline Value

In Figure 7, we plot the histogram of the median embedding shift m_e and M_e over all features and steganographic methods for payloads 0.10 and 0.20 bpac.[5] Because all features were scaled to have unit variance on the cover features, we can see that the size of the embedding shift is almost always very small. This confirms that image-to-image variations of features are large compared with the embedding distortion. Therefore, a single (non-calibrated) feature does not have much distinguishing power, which underlies the need for a baseline value provided by the reference transform r.

5.2 Framework Validation

In this section, we demonstrate on specific cases that the canonical examples explained in Section 4.2, indeed, occur within the PEV-274 feature set. Table 2 lists the sample values of central tendency and spread for the quantities introduced in Section 4.3 for several carefully selected combinations of steganographic methods, payloads, and features. Every case listed in Table 2 was given a unique index (the last column) that will be used for referencing. For better readability, the most relevant values for each case are highlighted.

The situation when the distributions of shifts $\mathbf{F}(c) \rightarrow \mathbf{F}_r(c)$ and $\mathbf{F}(s) \rightarrow \mathbf{F}_r(s)$ are very similar, and therefore calibration hurts performance, was called Parallel Reference in Section 4.2. We remind the reader that the shifts are described by m_{rc}, M_{rc} and m_{rs}, M_{rs}, respectively. Cases 1.1–1.6 show examples of Parallel Reference features because $m_{rc} \approx m_{rs}$ and $M_{rc} \approx M_{rs}$. Cases 1.1–1.3 correspond to the YASS algorithm, which exhibited the most frequent occurrence of this effect in our tests. Parallel Reference, however, may occur for some features for other algorithms as well (Cases 1.4–1.6).

The Cover Estimate Example 2 from Section 4.2 describes the situation when the reference transform improves steganalysis by making calibrated features of cover images approximately zero and calibrated features of stego images nonzero. This example is characterized by $m_e \approx -m_{rs}$ and $M_e \approx M_{rs}$ with small values of m_{rc} and M_{rc}. This situation can be nicely observed for embedding-sensitive features and large payloads, where the reference transform r indeed provides an estimate of the of cover feature. Cases 2.1–2.3 in Table 2 correspond to features that significantly change with embedding (histogram bins for nsF5 and Jsteg and co-occurrence $\mathbf{C}_{11}$ for nsF5). The histogram of the DCT mode $(2, 1)$ (Case 2.4) also falls into this category as Steghide preserves only the global histogram but not necessarily the histograms of individual DCT modes.

With decreasing payload size, Cover Estimate is less likely to occur because the embedding distortion becomes smaller while the properties of the reference mapping remain unchanged.

Within the PEV-274 feature set, we did not observe any cases of Example 3, the Stego Image Estimation, for any steganographic scheme. A real-life example of this scenario is the attack on OutGuess [4].

We now proceed to Example 4 called Eraser. In Table 2, we demonstrate this by Cases 4.1–4.3. The characterizing property of this scenario is that the reference features are close to each other when compared with the size of a consistent embedding shift. In other words, the median and MAD of the shift $\mathbf{F}_r(c) \rightarrow \mathbf{F}_r(s)$ should be small, $m_q \approx 0$, $M_q \approx 0$, despite the rather large relative values of distortions caused by the reference mapping (large values of

[5] For YASS, the payload size cannot be easily controlled. The average payload values for this case are listed in Table 4 in the appendix.

Algorithm	Payload	Feature	m_e	M_e	m_q	M_q	m_{rc}	M_{rc}	m_{rs}	M_{rs}	Example.Case
YASS 3	0.187	V	+0.0147	0.0048	+0.0149	0.0049	+**0.0109**	**0.0103**	+**0.0110**	**0.0103**	1.1
YASS 2	0.051	$\mathbf{B}_2$	+0.0121	0.0064	+0.0116	0.0067	−**0.0053**	**0.0272**	−**0.0054**	**0.0270**	1.2
YASS 1	0.110	$\mathbf{M}_{-4-4}$	+0.0007	0.0139	+0.0010	0.0160	+**0.0083**	**0.0448**	+**0.0083**	**0.0447**	1.3
nsF5	0.200	$\mathbf{g}_{41}^{-1}$	+0.0150	0.0169	+0.0122	0.0337	+**0.0432**	**0.1039**	+**0.0430**	**0.1042**	1.4
MME3	0.100	$\mathbf{M}_{0-3}$	−0.0013	0.0106	−0.0004	0.0161	+**0.0142**	**0.0310**	+**0.0145**	**0.0310**	1.5
JPHS	0.100	$\mathbf{h}_{-2}^{13}$	+0.0000	0.0000	+0.0000	0.0000	+**0.0590**	**0.1437**	+**0.0591**	**0.1437**	1.6
nsF5	1.000	$\mathbf{H}_0$	+**0.5349**	0.1449	+0.1702	0.0731	−**0.0188**	0.0198	−**0.3806**	0.0823	2.1
nsF5	1.000	$\mathbf{C}_{11}$	−**1.4643**	0.3327	−0.1632	0.1029	+**0.0642**	0.0722	+**1.3728**	0.2582	2.2
Jsteg	0.200	$\mathbf{H}_{-2}$	+**0.5280**	0.1424	+0.0598	0.0234	+**0.0147**	0.0284	−**0.4504**	0.1275	2.3
Steghide	0.200	$\mathbf{h}_2^{21}$	−**0.2265**	0.1080	+0.0303	0.0808	+**0.0070**	0.1540	+**0.2518**	0.1780	2.4
nsF5	0.200	$\mathbf{h}_0^{12}$	+0.0434	0.0103	−**0.0019**	**0.0090**	−0.0707	0.0341	−0.1183	0.0381	4.1
MME3	0.100	$\mathbf{M}_{00}$	+0.0173	0.0032	+**0.0018**	**0.0035**	−0.0195	0.0234	−0.0350	0.0234	4.2
MME3	0.100	$\mathbf{H}_0$	+0.0154	0.0022	+**0.0019**	**0.0028**	−0.0188	0.0198	−0.0322	0.0199	4.3
JPHS	0.100	$\mathbf{h}_{-1}^{12}$	+0.0000	0.0000	+0.4369	0.3988	+**0.1078**	**0.1336**	+**0.5634**	**0.4876**	5.1
Jsteg	0.200	$\mathbf{H}_1$	+0.0000	0.0000	+0.1133	0.0228	+**0.0242**	**0.0391**	+**0.1313**	**0.0473**	5.2
Steghide	0.200	$\mathbf{C}_{-11}$	−0.0052	0.0107	+0.1215	0.0298	+**0.0587**	**0.0462**	+**0.1832**	**0.0609**	5.3
nsF5	0.200	$\mathbf{M}_{-13}$	+0.0012	0.0264	−0.0243	0.0487	+**0.0145**	**0.1101**	−**0.0102**	**0.1067**	5.4
YASS 4	0.118	$\mathbf{M}_{31}$	−0.2860	0.2650	−0.1749	0.2851	+**0.1176**	**0.3262**	+**0.2415**	**0.3256**	5.5

Table 2: Experimental verification of calibration examples from Section 4.3. For selected combinations of the embedding method, payload, and NCPEV-274 feature (notation taken from Table 1), we computed the sample statistics $m_e, M_e, m_q, M_q, m_{rc}, M_{rc}, m_{rs}$, and M_{rs}. For better readability, values most relevant to individual cases are highlighted.

m_{rc}, M_{rc} and m_{rs}, M_{rs}). Note that for small payloads, the histogram bin of a steganographic scheme that disturbs first-order statistics may become an Eraser rather than Cover Estimation (Case 4.1).

By far the most frequent situation was the Divergent Reference illustrated by the last set of Cases 5.1–5.5. Here, as opposed to the Parallel Reference (Cases 1.1–1.6), the reference statistics m_{rc}, M_{rc} and m_{rs}, M_{rs} should simply be different. The more different they are, the larger the benefit of calibration. Cases 5.1 and 5.2 demonstrate the intriguing situation when the feature value itself is preserved during embedding (and therefore useless for steganalysis), while its calibrated form has a good distinguishing power due to the fact that the reference transform r reacts differently to cover and stego images. Cases 5.3–5.5 were included to illustrate Divergent Reference on various steganographic methods and non-preserved features.

Note that all cases, with the exception of Parallel Reference, can be interpreted as special cases of Divergent Reference. Since both Cover Estimate and Eraser need the existence of the embedding shift $\mathbf{F}(c) \to \mathbf{F}(s)$, the property of the reference transform, $\mathbf{F}_r(c) \approx \mathbf{F}_r(s)$, basically implies that the shifts $\mathbf{F}(c) \to \mathbf{F}_r(c)$ and $\mathbf{F}(s) \to \mathbf{F}_r(s)$ must be different. This is what we request from calibration – the two shifts must be as different as possible in order to easily distinguish between cover and stego features.

To summarize our observations, we showed that calibration does not have to (and in general it does not) approximate the cover-image feature to improve steganalysis. At the same time, we showed examples when calibration is harmful (Cases 1.1–1.5). Moreover, all five calibration scenarios described in Section 4.2 do occur in real life.

An important fact that needs to be stressed is that we only studied each feature individually while ignoring the dependences among individual features. Therefore, we have to be careful about the interpretation in terms of the *global behavior* within the PEV-274 feature set. The individual features may be useful even without calibration, i.e., without their baseline value provided by the reference mapping r, when we utilize dependences among them. This topic is the subject of the next section.

6. AN IMPROVED CALIBRATION

The experiments described so far demonstrate that several different mechanisms are responsible for the positive effect of calibration. At the same time, calibration may have a catastrophically negative effect on steganalysis when Parallel Reference occurs. To prevent such failures, in this section we propose a modified calibration procedure and demonstrate that it improves steganalysis in practice.

For convenience of explanation, let us assume that $\mathbf{F}$ is a one-dimensional feature. Given an image $x \in \mathcal{X}$, we extract its feature $\mathbf{F}(x)$ and the reference feature $\mathbf{F}_r(x)$. Applying a linear transformation to the two-dimensional vector $(\mathbf{F}_r(x), \mathbf{F}(x))$, we obtain the vector $\hat{\mathbf{F}}(x) = (\mathbf{F}_r(x) - \mathbf{F}(x), \mathbf{F}_r(x) + \mathbf{F}(x)) = (\mathbf{F}_{cal}(x), \mathbf{F}_r(x) + \mathbf{F}(x))$. Since the first component of $\hat{\mathbf{F}}$ is the calibrated feature $\mathbf{F}_{cal}$, we can think of calibration as a one-dimensional projection of $\hat{\mathbf{F}}$ to its first component. This projection, however, may remove potentially useful information that might help steganalysis.

Therefore, we propose to calibrate by taking the Cartesian product of the feature and its reference form, rather than their difference.[6] Note that now, even when the reference feature value $\mathbf{F}_r$ is useless as far as the distinguishability between cover and stego classes is concerned (Parallel Reference), the performance of the steganalyzer will not be compromised.[7] In this case, $\mathbf{F}_r$ will be simply a non-influential feature that is likely to be removed if any feature reduction procedure is applied.

We subjected this modified process of calibration to a large-scale test. For each embedding method and payload, we constructed a separate steganalyzer using the non-calibra-

[6] Our implementation of the feature extractor can be downloaded from http://dde.binghamton.edu/ccmerged/.

[7] Care needs to be taken to avoid problems when applying machine learning algorithms due to increased feature dimensionality.

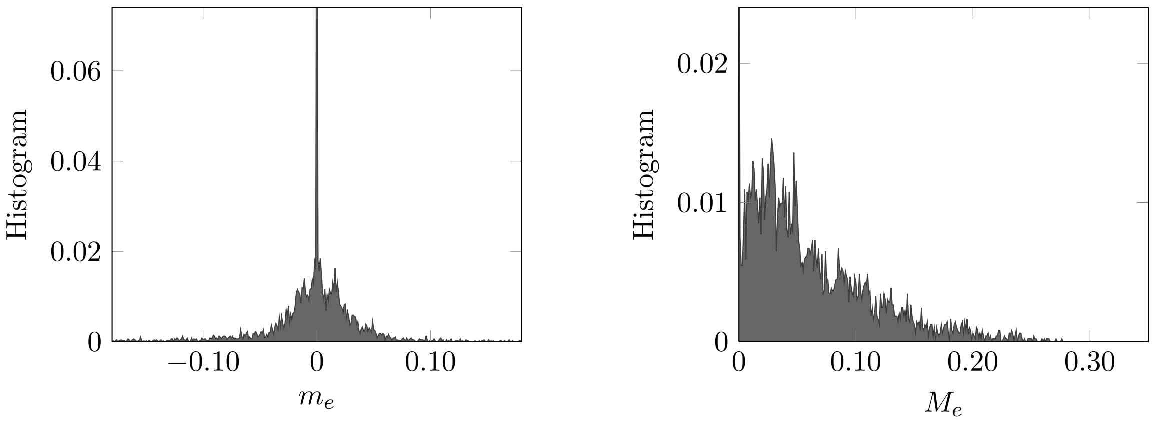

Figure 7: Normalized histogram of the median embedding shift m_e (left) and MAD M_e (right) over all steganographic methods from Section 2.4 and payloads 0.10 and 0.20 bpac.

ted feature set NCPEV-274 the set PEV-274 calibrated by subtracting $\mathbf{F}_r$, and by taking the Cartesian product (the MPEV-274 set). Since our goal was to compare the performance of individual feature sets rather than embedding methods, we chose different payloads for different methods, depending on the detectability. (We wanted payloads for which the methods would be neither too easy nor too difficult to detect.) For the steganographic methods nsF5, MME3, and JPHS, we chose the payloads 0.05,0.10,0.15, and 0.20 bpac. For more detectable algorithms, Jsteg and Steghide, we chose smaller payloads, 0.02,0.03,0.04, and 0.05 bpac. For YASS, since the payload cannot be easily controlled, we show averages over all images in our database. The results are summarized in Table 3.

The error rates P_E were obtained for five different divisions of the database into a training and a testing set. In Table 3, we report the average values. In most cases, all five values were within one percent of the mean, (see Figure 8), which indicates that the results are statistically reliable. The MPEV-274 feature set always produces the best steganalysis. Note that calibration by spatial domain cropping involved in PEV-274 makes steganalysis of YASS remarkably worse for all settings. This is not surprising because YASS was created with the intention to disable calibration. On the other hand, calibration by Cartesian product (MPEV-274) slightly improves the detectability compared with non-calibrated features. This means that there are features for which even for YASS the reference mapping r improves steganalysis.

A careful inspection of Table 3 reveals that except for JP Hide & Seek, where the calibrated features PEV-274 perform significantly better than the non-calibrated features NCPEV-274 and YASS, where the calibrated features PEV-274 failed, the feature sets PEV-274 and NCPEV-274 have a very similar performance. This is rather surprising because calibration was thought to improve steganalysis.

We provide the following heuristic explanation for this phenomenon. The key observation is that the individual features involved in NCPEV-274 exhibit strong dependences.[8] If we put two correlated features next to each other, they serve mutually as "reference" values in the same sense as if we put $\mathbf{F}$ and $\mathbf{F}_r$ next to each other as in our modified calibration procedure. Consequently, the steganalysis performance of two dependent features may be remarkably better than if we used those features individually (as we did in our experiments in Section 5). Taking this to an extreme, we can say that not only pairs of features but also the individual

			P_E	
Algorithm	bpac	NCPEV	PEV	MPEV
nsF5	0.05	0.361	0.360	**0.331**
	0.10	0.202	0.218	**0.177**
	0.15	0.100	0.094	**0.077**
	0.20	0.048	0.040	**0.036**
Jsteg	0.02	0.097	0.132	**0.083**
	0.03	0.042	0.051	**0.032**
	0.04	0.022	0.021	**0.018**
	0.05	0.015	0.013	**0.010**
Steghide	0.02	0.114	0.127	**0.083**
	0.03	0.055	0.056	**0.043**
	0.04	0.031	0.031	**0.024**
	0.05	0.021	0.015	**0.011**
MME3	0.05	0.309	0.310	**0.277**
	0.10	0.187	0.207	**0.165**
	0.15	0.130	0.149	**0.107**
	0.20	0.023	0.017	**0.012**
JPHS	0.05	0.306	0.100	**0.094**
	0.10	0.160	0.066	**0.054**
	0.15	0.076	0.034	**0.022**
	0.20	0.039	0.014	**0.006**
YASS 1	0.110	0.133	0.317	**0.113**
YASS 2	0.051	0.179	0.347	**0.164**
YASS 3	0.187	0.102	0.121	**0.082**
YASS 4	0.118	0.120	0.303	**0.109**
YASS 5	0.159	0.075	0.241	**0.064**
YASS 6	0.032	0.269	0.342	**0.258**
YASS 7	0.078	0.244	0.298	**0.225**
YASS 8	0.138	0.211	0.251	**0.180**

Table 3: Steganalysis of selected algorithms when using differently calibrated feature sets.

[8] Consider, for example, the symmetry of global and local DCT histograms.

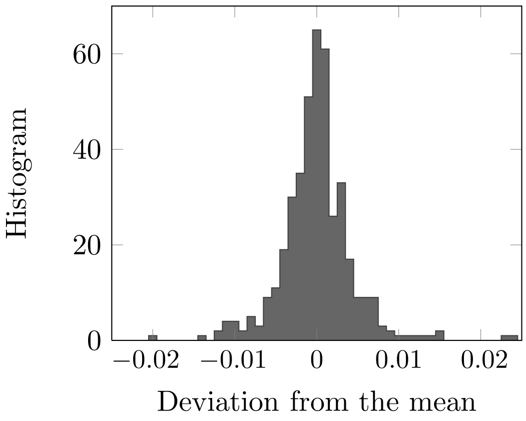

Figure 8: Deviations of individual values of P_E from the means over all five runs. The histogram is taken over the steganographic algorithms and payloads reported in Table 3.

elements of the entire feature vector mutually "calibrate" each other. Indeed, if all the features were independent, the performance of NCPEV-274 would be probably very poor.

7. CONCLUSION

In the past, calibration has been proposed as a process in which a steganalytic feature is supplied with a baseline (reference) value to improve the feature's ability to distinguish between cover and stego features. However, even though calibration is generally recognized as beneficial, there are cases when it may have a catastrophically negative effect on the reliability of steganalysis or when it may have very little or no effect. Furthermore, it seems that the benefit of calibration may be reduced by dependences among individual features when the number of features is large.

In this paper, we argue that the established thesis that calibration provides an estimate of cover image features is not necessarily correct. In fact, we recognize five different archetypes of calibration based on its mechanism through which it provides a given feature with its reference value. Our view is supported by experiments on real steganographic schemes and with a feature set that is widely used for steganalysis of JPEG images. Furthermore, our newly acquired insight enabled us to propose a modified approach to calibration in which the reference feature value is adopted as an additional feature instead of subtracted from the original feature value. Calibration performed in this way removes the problem of catastrophic failures for some steganographic schemes and it improves steganalysis across a wide range of steganographic schemes and payloads.

8. ACKNOWLEDGMENTS

The work on this paper was supported by Air Force Office of Scientific Research under the research grant number FA9550-08-1-0084 and FA9550-09-1-0147. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of AFOSR or the U.S. Government.

9. REFERENCES

[1] 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.

[2] 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.

[3] J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG images: Breaking the F5 algorithm. In *Information Hiding, 5th International Workshop*, volume 2578 of *Lecture Notes in Computer Science*, pages 310–323, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.

[4] 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.

[5] 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.

[6] 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.

[7] S. S. Keerthi, V. Sindhwani, and O. Chapelle. An efficient method for gradient-based adaptation of hyperparameters in svm models. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 673–680. MIT Press, Cambridge, MA, 2007.

[8] A. D. Ker. Improved detection of LSB steganography in grayscale images. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of *Lecture Notes in Computer Science*, pages 97–115, Toronto, Canada, May 23–25, 2004. Springer-Verlag, Berlin.

[9] A. D. Ker. Resampling and the detection of LSB matching in color bitmaps. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 1–15, San Jose, CA, January 16–20, 2005.

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

[11] A. D. Ker, T. Pevný, J. Kodovský, and J. Fridrich. The Square Root Law of steganographic capacity. In

A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 107–116, Oxford, UK, September 22–23, 2008.

[12] 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.

[13] 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*, volume 6819, pages 2 1–2 13, San Jose, CA, January 27–31, 2008.

[14] J. Kodovský, J. Fridrich, and T. Pevný. 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.

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

[16] Bin Li, Yun Q. Shi, and Jiwu Huang. Steganalysis of yass. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 139–148, Oxford, UK, September 22–23, 2008.

[17] 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.

[18] T. Pevný and J. Fridrich. Benchmarking for steganography. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of *Lecture Notes in Computer Science*, Santa Barbara, CA, June 19–21, 2008. Springer-Verlag, New York.

[19] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323–335, Washington, DC, August 13–17, 2001.

[20] 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*, volume 6819, pages 16–31, San Jose, CA, January 27–31, 2008.

[21] 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.

[22] 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*, volume 4567 of *Lecture Notes in Computer Science*, pages 16–31, Saint Malo, France, June 11–13, 2007. Springer-Verlag, New York.

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

[24] 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.

APPENDIX

In this appendix, we provide details of all eight YASS settings used in our experiments (see Table 4). Following the same notation as in the original publications, QF_h is the hiding quality factor(s) and B is the big block size. Settings 1, 4, 5, and 7 incorporate a mixture-based modification of YASS embedding with several different values QF_h based on block variances (the decision boundaries are in the column "DBs"). Settings 3 and 8 use attack-aware iterative embedding (column *rep*). The payload values in Table 4 are averages over all images in our database in terms of bpac. In all experiments, the advertising quality factor was fixed at $QF_a = 75$ and the input images were in the raw (uncompressed) format. With these choices, YASS appears to be the least detectable [13].

Notation	QF_h	DBs	B	*rep*	Payload
YASS 1	65,70,75	3,7	9	0	0.110
YASS 2	75	-	9	0	0.051
YASS 3	75	-	9	1	0.187
YASS 4	65,70,75	2,5	9	0	0.118
YASS 5	50,55,60,65,70	3,7,12,17	9	0	0.159
YASS 6	75	-	10	0	0.031
YASS 7	65,70,75	3,7	10	0	0.078
YASS 8	75	-	10	1	0.138

Table 4: Settings for YASS as tested in the paper.