Merging Markov and DCT Features for Multi-Class JPEG Steganalysis

Tomáš Pevný[a], Jessica Fridrich[b]

[a]Department of Computer Science, Binghamton University, State University of New York
[b]Department of Electrical and Computer Engineering, Binghamton University,State University of New York *

ABSTRACT

Blind steganalysis based on classifying feature vectors derived from images is becoming increasingly more powerful. For steganalysis of JPEG images, features derived directly in the embedding domain from DCT coefficients appear to achieve the best performance (e.g., the DCT features[10] and Markov features[21]). The goal of this paper is to construct a new multi-class JPEG steganalyzer with markedly improved performance. We do so first by extending the 23 DCT feature set,[10] then applying calibration to the Markov features described in[21] and reducing their dimension. The resulting feature sets are merged, producing a 274-dimensional feature vector. The new feature set is then used to construct a Support Vector Machine multi-classifier capable of assigning stego images to six popular steganographic algorithms—F5,[22] OutGuess,[18] Model Based Steganography without ,[19] and with[20] deblocking, JP Hide&Seek,[1] and Steghide.[14] Comparing to our previous work on multi-classification,[11, 12] the new feature set provides significantly more reliable results.

1. INTRODUCTION

Steganography is the art of undetectable communication in which messages are embedded in innocuous looking objects, such as digital images. In the process of embedding, the original (cover) object is slightly modified to embed the data. The modified cover object is called the stego object. The embedding process usually depends on a secret stego key shared between both communicating parties. The main requirement of steganographic systems is statistical undetectability of the hidden data given the knowledge of the embedding mechanism and the source of cover objects but not the stego key (so called Kerckhoffs' principle).

Steganographic security was formalized by Cachin [7] who introduced the concept of ϵ-security. Let $\mathcal{X}$ be the set of all possible cover objects. A steganographic scheme is a pair of mappings $Emb_{\mathbf{k}} : \mathcal{X} \rightarrow \mathcal{X}$ and $Ext_{\mathbf{k}} : \mathcal{X} \rightarrow M$ both parametrized by a secret key $\mathbf{k}$ such that $Ext_{\mathbf{k}}(Emb_{\mathbf{k}}(\mathbf{x}, \mathbf{m})) = \mathbf{m}$ for all $\mathbf{x} \in X$, $\mathbf{m} \in M$, and $\mathbf{k} \in K$, where M and K are spaces of all communicable messages $\mathbf{m}$ and secret keys $\mathbf{k}$, respectively. Assuming X can be endowed with a probability distribution function f_C , the "natural" distribution of covers, the stego objects will be distributed according to pdf f_S. The Kullback-Leibler distance $D(f_C||f_S)$ is taken as the measure of statistical detectability. If $D(f_C||f_S) < \epsilon$, we say that the steganographic scheme is ϵ-secure.

Because the dimensionality of X is too large, in practice, the objects of X are represented using a simplified model. One possibility is to project each object $\mathbf{x} \in X$ onto a space of a much smaller dimension formed by "features" that, in some sense, capture everything important about $\mathbf{x}$. One could then attempt to map out the distributions of features f_C and f_S from a large database of cover and stego objects. For steganalysis, machine learning techniques are used to train a classifier capable of distinguishing cover and stego feature sets in the feature space.[4, 5, 9, 10] Such steganalytic methods are called blind. Their biggest advantage is that there is no need to construct specific targeted steganalytic methods whenever a new method appears. Blind methods can also classify objects to known steganographic schemes (so called multi-class steganalysis) providing valuable feedback to forensic examiners towards the goal of extracting the secret message.

The idea to use a trained classifier to detect data hiding was first introduced in a paper by Avcibas et al.,[4] where image quality metrics were proposed as features and the method was tested on several robust

[*] Jessica Fridrich: E-mail: fridrich@binghamton.edu, Telephone: +1 607 777 6177, Fax: +1 607 777 4464

watermarking algorithms as well as least significant bit embedding (LSB) in the spatial domain. Avcibas et al.[3, 5] later proposed a different set of features based on binary similarity measures between the LSB plane and the second LSB plane capitalizing on the fact that most steganographic schemes use the LSB of image elements as the information-carrying entity. Farid[8, 9, 16] constructed the features from higher-order moments of distribution of coefficients obtained using quadrature mirror filters and the coefficient prediction errors from several high-frequency sub-bands. Other authors have investigated the problem of blind steganalysis using trained classifiers.[13, 17, 23]

For best results, the features for steganalysis should react sensitively to embedding changes but be otherwise insensitive to image content. Virtually all steganographic methods for the most common image format—JPEG work by manipulating the quantized DCT coefficients. Since the embedding changes are lumped in the DCT domain, constructing the features in the same domain will likely lead to a more sensitive feature set. The first feature set targeted to JPEG images that also employed the concept of calibration was proposed by Fridrich.[10] The calibration is a procedure through which one can estimate the features of the cover image from the stego image. In this paper, we will call this feature set "DCT features." Recently, Shi et al.[21] proposed another feature set for JPEG images based on Markov models of DCT plane. This feature set will be called "Markov features." Previous comparisons of performance of blind steganalyzers[11] based on different feature sets indicated that feature sets targeted to JPEG images have remarkably better performance than general purpose feature sets.

The contribution of this paper is three fold. First, we compare the performance of classifiers employing DCT features[10] and Markov features.[21] Second, by analyzing both classifiers, we propose a new merged feature set, whose detection accuracy is remarkably better than the detection accuracy of both of its predecessors. Third, we use the proposed feature set to construct a general blind multi-classifier for single-compressed JPEG images with a wide range of quality factors. We report its performance by classifying images to 6 known JPEG steganographic techniques (F5, OutGuess, Steghide, JP Hide&Seek, Model based steganography with and without deblocking) for 34 JPEG quality factors.

The paper is organized as follows. In the next section, we briefly review the construction of DCT and Markov features, and describe the new merged feature set. In Section 3, we give the implementation details of the SVMs used in this paper and describe the training and testing methodology. In the same section, we compare the performance of all three features sets on binary and multi-classification problems. In Section 4, we construct a seven-class multi-classifier for detecting steganographic algorithms for single-compressed JPEG images embedded with six popular JPEG steganographic algorithms and various quality factors. The paper is concluded in Section 5.

2. FEATURES

In this section, we describe the new feature set for steganalysis of JPEG images. We start with the description of the original and extended DCT feature set and a short review of the recently proposed Markov features.[21] Then, we present the new Merged feature set created as a combination of the extended DCT and calibrated Markov feature sets.

All features in the merged set will be calibrated. Calibration is a process used to estimate macroscopic properties of the cover image from the stego image. We quickly review the inner workings here, since it forms an essential part of the feature calculation. More detailed description of calibration can be found in.[10–12] During calibration, the stego JPEG image J_1 is decompressed to the spatial domain, cropped by a few pixels in both directions, and compressed again with the same quantization matrix as the stego image J_1. The newly obtained JPEG image J_2 has most macroscopic features similar to the original cover image. This is because the cropped image is visually similar to the original image. Moreover, the cropping brings the 8×8 DCT grid "out of sync" with the previous compression, which effectively suppresses the influence of the previous JPEG compression and the embedding changes. The calibrated feature is obtained as the difference between the features calculated for J_1 and J_2. This calibrated feature will be less sensitive to the image content and more sensitive to embedding changes.

2.1. Extended DCT feature set

The original DCT features (originally published in[10]) were constructed by use of 23 functionals $\mathbf{F}$ that produce a scalar, vector, or a matrix when applied to the stego image. Each functional $\mathbf{F}$ is evaluated for the stego image J_1 and its calibrated version J_2. The calibrated feature f is obtained as the difference $\mathbf{F}(J_1) - \mathbf{F}(J_2)$, if $\mathbf{F}$ is a scalar, or as an L_1 norm $\|\mathbf{F}(J_1) - \mathbf{F}(J_2)\|_{L_1}$ if $\mathbf{F}$ is a vector or a matrix. The functionals $\mathbf{F}$ are defined as follows.

Let the luminance of a stego JPEG file be represented with a DCT coefficient array $d_{ij}(k)$, $i, j = 1, \ldots, 8$, $k = 1, \ldots, n_B$, where $d_{ij}(k)$ denotes the (i, j)-th quantized DCT coefficient in the k-th block (there are total of n_B blocks).

The first functional is the histogram $\mathbf{H}$ of all $64 \times n_B$ luminance DCT coefficients

$$\mathbf{H} = (H_L, \ldots, H_R), \tag{1}$$

where $L = \min_{i,j,k} d_{ij}(k)$, $R = \max_{i,j,k} d_{ij}(k)$.

The next 5 functionals are the histograms

$$\mathbf{h}^{ij} = (h^{ij}_L, \ldots, h^{ij}_R), \tag{2}$$

of coefficients of 5 individual DCT modes $(i, j) \in \{(1, 2), (2, 1), (3, 1), (2, 2), (1, 3)\} \triangleq \mathcal{L}$.

The next 11 functionals are dual histograms represented with 8×8 matrices $\mathbf{g}^d_{ij}$, $i, j = 1, \ldots, 8$, $d = -5, \ldots, 5$

$$\mathbf{g}^d_{ij} = \sum_{k=1}^{n_B} \delta(d, d_{ij}(k)), \tag{3}$$

where $\delta(x, y) = 1$ if $x = y$ and 0 otherwise.

The next 6 functionals capture inter-block dependency among DCT coefficients. The first functional is the variation V

$$V = \frac{\sum\limits_{i,j=1}^{8} \sum\limits_{k=1}^{|\mathbf{I}_r|-1} |d_{ij}(\mathbf{I}_r(k)) - d_{ij}(\mathbf{I}_r(k+1))| + \sum\limits_{i,j=1}^{8} \sum\limits_{k=1}^{|\mathbf{I}_c|-1} |d_{ij}(\mathbf{I}_c(k)) - d_{ij}(\mathbf{I}_c(k+1))|}{|\mathbf{I}_r| + |\mathbf{I}_c|}, \tag{4}$$

where $\mathbf{I}_r$ and $\mathbf{I}_c$ denote the vectors of block indices $1, \ldots, n_B$ while scanning the image by rows and by columns, respectively.

Two next two blockiness functionals are scalars calculated from the decompressed JPEG image representing an integral measure of inter-block dependency over all DCT modes over the whole image:

$$B_\alpha = \frac{\sum\limits_{i=1}^{\lfloor (M-1)/8 \rfloor} \sum\limits_{j=1}^{N} |\mathbf{c}_{8i,j} - \mathbf{c}_{8i+1,j}|^\alpha + \sum\limits_{j=1}^{\lfloor (N-1)/8 \rfloor} \sum\limits_{i=1}^{M} |\mathbf{c}_{i,8j} - \mathbf{c}_{i,8j+1}|^\alpha}{N \lfloor (M-1)/8 \rfloor + M \lfloor (N-1)/8 \rfloor}. \tag{5}$$

In (5), M and N are image height and width in pixels and $\mathbf{c}_{i,j}$ are grayscale values of the decompressed JPEG image, $\alpha = 1, 2$.

The remaining three functionals are calculated from the co-occurrence matrix of neighboring DCT coefficients

$$\begin{aligned}
N_{00} =& \mathbf{C}_{0,0}(J_1) - \mathbf{C}_{0,0}(J_2) \\
N_{01} =& \mathbf{C}_{0,1}(J_1) - \mathbf{C}_{0,1}(J_2) + \mathbf{C}_{1,0}(J_1) - \mathbf{C}_{1,0}(J_2) + \mathbf{C}_{-1,0}(J_1) - \mathbf{C}_{-1,0}(J_2) + \mathbf{C}_{0,-1}(J_1) - \mathbf{C}_{0,-1}(J_2) \qquad (6) \\
N_{11} =& \mathbf{C}_{1,1}(J_1) - \mathbf{C}_{1,1}(J_2) + \mathbf{C}_{1,-1}(J_1) - \mathbf{C}_{1,-1}(J_2) + \mathbf{C}_{-1,1}(J_1) - \mathbf{C}_{-1,1}(J_2) + \mathbf{C}_{-1,-1}(J_1) - \mathbf{C}_{-1,-1}(J_2),
\end{aligned}$$

Functional	Dimensionality
Global histogram $\mathbf{H}_l$	11
5 AC histograms $\mathbf{h}_l^{ij}$	5×11
11 Dual histograms $\mathbf{g}_{ij}^d$	11×9
Variation V	1
2 Blockiness B_α	2
Co-occurrence matrix $\mathbf{C}_{st}$	25

Table 1. Extended DCT feature set with 193 features.

where

$$\mathbf{C}_{st} = \frac{\sum_{i,j=1}^{8}\sum_{k=1}^{|\mathbf{I}_r|-1} \delta\left(s, d_{ij}(\mathbf{I}_r(k))\right)\delta\left(t, d_{ij}(\mathbf{I}_r(k+1))\right) + \sum_{i,j=1}^{8}\sum_{k=1}^{|\mathbf{I}_c|-1} \delta\left(s, d_{ij}(\mathbf{I}_c(k))\right)\delta\left(t, d_{ij}(\mathbf{I}_c(k+1))\right)}{|\mathbf{I}_r| + |\mathbf{I}_c|}. \tag{7}$$

The original motivation for using the L_1 norm to form the DCT features is the reduction of their dimensionality. It is apparent, however, that by using the L_1 norm, some information potentially useful for steganalysis is lost. By replacing the L_1 norm with a higher-dimensional alternative, we will preserve more information and obtain better classification results at the expense of increased dimensionality. Replacing the L_1 norm directly with the difference, however, is not feasible because the feature set dimensionality would substantially increase and there would be too many features holding little information (e.g., histogram bins for large values of DCT coefficients). This would eventually negatively affect the performance and increase the complexity of the classifier. In order to alleviate the information loss due to using the L_1 norm and to keep the dimensionality of features "reasonable," we replaced the L_1 norm by the following differences.

For the global histogram functional $\mathbf{H}$ and for 5 histograms of individual DCT modes $\mathbf{h}^{ij}, (i,j) \in \mathcal{L}$, we take the differences of elements in the range $[-5,+5]$. Thus, the histogram features are

$$\mathbf{H}_l(J_1) - \mathbf{H}_l(J_2),\ l \in \{-5,\ldots,5\},$$

$$\mathbf{h}_l^{ij}(J_1) - \mathbf{h}_l^{ij}(J_2),\ l \in \{-5,\ldots,5\}.$$

For the dual histogram functionals $\mathbf{g}^d, d \in \{-5,\ldots,+5\}$, we take the difference of the 9 lowest AC modes

$$\mathbf{g}_{ij}^d(J_1) - \mathbf{g}_{ij}^d(J_2),\ (i,j) \in \{(2,1),(3,1),(4,1),(1,2),(2,2),(3,2),(1,3),(2,3),(1,4)\}.$$

For the co-occurrence matrix functionals, we use the central elements in the range $[-2,+2]\times[-2,+2]$. This yields 25 features

$$\mathbf{C}_{st}(J_1) - \mathbf{C}_{st}(J_2),\ (s,t) \in [-2,+2]\times[-2,+2].$$

The rationale behind restricting the range of the differences between functionals to a small interval around zero is that DCT coefficients follow a generalized Gaussian distribution centered around zero. Thus, the central part of the functionals holds the most useful information for steganalysis.

After we replace the L_1 norm by the proposed differences, the dimensionality of the feature set (further referred to as the *extended DCT feature set*) becomes 193 (see Table 1).

2.2. Original, calibrated, and reduced Markov features

The Markov feature set as proposed in[21] models the differences between absolute values of neighboring DCT coefficients as a Markov process. The feature calculation starts by forming the matrix $F(u,v)$ of absolute values of DCT coefficients in the image. The DCT coefficients in $F(u,v)$ are arranged in the same way as pixels in the image by replacing each 8×8 block of pixels with the corresponding block of DCT coefficients. Next, four difference arrays are calculated along four directions: horizontal, vertical, diagonal, and minor diagonal (further denoted as $F_h(u,v)$, $F_v(u,v)$, $F_d(u,v)$, and $F_m(u,v)$ respectively)

$$\begin{aligned}
F_h(u,v) &= F(u,v) - F(u+1,v),\\
F_v(u,v) &= F(u,v) - F(u,v+1),\\
F_d(u,v) &= F(u,v) - F(u+1,v+1),\\
F_m(u,v) &= F(u+1,v) - F(u,v+1).
\end{aligned}$$

From these difference arrays, four transition probability matrices $\mathbf{M}_h, \mathbf{M}_v, \mathbf{M}_d, \mathbf{M}_m$ are constructed as

$$\begin{aligned}
\mathbf{M}_h(i,j) &= \frac{\sum_{u=1}^{S_u-2}\sum_{v=1}^{S_v}\delta(F_h(u,v)=i, F_h(u+1,v)=j)}{\sum_{u=1}^{S_u-1}\sum_{v=1}^{S_v}\delta(F_h(u,v)=i)},\\
\mathbf{M}_v(i,j) &= \frac{\sum_{u=1}^{S_u}\sum_{v=1}^{S_v-2}\delta(F_v(u,v)=i, F_v(u,v+1)=j)}{\sum_{u=1}^{S_u}\sum_{v=1}^{S_v-1}\delta(F_v(u,v)=i)},\\
\mathbf{M}_d(i,j) &= \frac{\sum_{u=1}^{S_u-2}\sum_{v=1}^{S_v-2}\delta(F_d(u,v)=i, F_d(u+1,v+1)=j)}{\sum_{u=1}^{S_u-1}\sum_{v=1}^{S_v-1}\delta(F_d(u,v)=i)},\\
\mathbf{M}_m(i,j) &= \frac{\sum_{u=1}^{S_u-2}\sum_{v=1}^{S_v-2}\delta(F_m(u+1,v)=i, F_m(u,v+1)=j)}{\sum_{u=1}^{S_u-1}\sum_{v=1}^{S_v-1}\delta(F_m(u,v)=i)},
\end{aligned}$$

where S_u and S_v denote the dimensions of the image and $\delta = 1$ if and only if its argument(s) are satisfied. Since the range of differences between absolute values of neighboring DCT coefficients could be quite large, if the matrices $\mathbf{M}_h, \mathbf{M}_v, \mathbf{M}_d, \mathbf{M}_m$ were taken directly as features, the dimensionality of the feature set would be too large. Thus, the authors proposed to only use the central $[-4, +4]$ portion of the matrices with the caveat that the values in the difference arrays $F_h(u,v)$, $F_v(u,v)$, $F_d(u,v)$, and $F_m(u,v)$ larger than 4 were set to 4 and values smaller than -4 were set to -4 prior to calculating $\mathbf{M}_h, \mathbf{M}_v, \mathbf{M}_d, \mathbf{M}_m$. Thus, all four matrices have the same dimensions 9×9 and the number of features is $4 \times 81 = 324$.

The Markov features as proposed in[21] were uncalibrated. Because calibration is known to improve features' sensitivity to embedding while reducing image-to-image variations, we incorporated the calibration into the process of calculating the features. As expected, this significantly improved the performance of Markov features. Let $\mathbf{M}$ denote the transition probability matrix in a specific direction. The calibrated Markov features are formed by differences $\mathbf{M}^{(c)} = \mathbf{M}(J_1) - \mathbf{M}(J_2)$, where J_1 is the stego image and J_2 its calibrated version. The dimension of the calibrated Markov feature set, $\mathbf{M}_h^{(c)}, \mathbf{M}_v^{(c)}, \mathbf{M}_d^{(c)}, \mathbf{M}_m^{(c)}$, remains the same as its original version.

2.3. Merged feature set

Even though different DCT modes in one 8×8 block are orthogonal, neighboring DCT coefficients may still exhibit mutual correlations. Markov features capture this residual *intra-block* dependency among DCT coefficients of similar spatial frequencies within the same 8×8 block. Because the extended DCT features model *inter-block* dependencies between DCT coefficients, it makes sense to merge them. Another incentive for merging is our observation (see sections 3.1 and 3.2) that both feature sets complement each other in performance. For example, the extended DCT feature set is better in detecting JpHide&Seek, while the calibrated Markov feature set is better in detecting F5.

A direct combination of both feature sets would produce a 517-dimensional feature vector. To reduce the resulting dimensionality, we used the average $\overline{\mathbf{M}} = (\mathbf{M}_h^{(c)} + \mathbf{M}_v^{(c)} + \mathbf{M}_d^{(c)} + \mathbf{M}_m^{(c)})/4$ of all four calibrated matrices, instead. This feature vector has dimensionality 81. We observed that the averaged features

$\overline{\mathbf{M}}$ produced very similar performance as their full version $\mathbf{M}_h^{(c)}, \mathbf{M}_v^{(c)}, \mathbf{M}_d^{(c)}, \mathbf{M}_m^{(c)}$. After merging the 193 extended DCT features with the 81 averaged calibrated Markov features, the dimension of the resulting merged feature set became $193 + 81 = 274$.

3. COMPARISON OF FEATURES

3.1. Binary classifiers

In this section, we compare the performance of tree sets of binary classifiers employing the original 23 DCT features, the original 324 Markov features (without calibration), and 274 Merged features. We do this on single-compressed JPEG images with quality factor 75 embedded by one of the following algorithm: F5, JP Hide&Seek, Model Based Steganography without deblocking (MB1), Model Based Steganography with deblocking (MB2), OutGuess, and Steghide. We chose quality factor 75, because it is the default quality factor in OutGuess. For each feature set and embedding algorithm, we constructed a binary classifier detecting cover and stego images, which yields the total of $3 \times 6 = 18$ binary classifiers.

For classification, we used soft-margin support vector machines (C-SVM) with Gaussian kernel.[6] The training parameters of the C-SVMs were determined by grid-search performed on the following multiplicative grid

$$(C, \gamma) \in \{(2^i, 2^j) | i \in \mathcal{Z}, j \in \mathcal{Z}\}.$$

To overcome the problem that this grid is unbounded, we exploit the fact that for most practical problems, the error surface of SVMs estimated using cross-validation is convex. The grid-search for a particular SVM started by evaluating all grid points common to all trained SVMs. After that, we checked if the best point (determined by the smallest cross-validation error) was at the boundary of the grid. If so, we enlarged the grid for this machine in the direction perpendicular to the boundary the best point laid on. We kept doing this until the best point ended up within the explored grid (not on the boundary). This simple algorithm ensured that the distance between the best point and the optimal point was small (within the size of the grid) under the convexity assumption.

The training set for every classifier contained 3400 examples of cover images and 3400 examples of stego images embedded with a random bitstream. With the exception of MB2, examples of three message lengths 100%, 50%, and 25% of embedding capacity of a given algorithm were equally included in the training set. For MB2, we only embedded messages of one length equivalent to 30% of the embedding capacity of MB1 to minimize cases when the deblocking algorithm fails. For JP Hide&Seek, in compliance with the directions provided by its author, we calculated the embedding capacity as 10% of the JPEG file size.

The testing images were prepared in the same way (the same embedding algorithm and relative message length) as the training images, but from a disjoint set of 2500 raw images. The testing set contained images with completely different scenes, taken by different cameras, and by different photographers. As mentioned earlier, all images in the testing and training sets were single-compressed JPEGs with quality factor 75.

Table 2 shows the performance of all 18 binary classifiers. Note that the performance of the 23 DCT and original Markov features is complementary. The DCT features are better in detecting JP Hide&Seek and OutGuess, while the Markov features can better detect Steghide. The comparison on Model based steganography with and without deblocking, and F5 algorithms is less clear, since the DCT features have a lower false positive rate. Also note that the original Markov features are almost unable to detect short messages embedded using JP Hide&Seek.

Table 2 also shows us that the new Merged feature set outperforms both its predecessors. Its false positive rate is below 0.5% on all algorithms, while the detection accuracy is higher than 99% except for JP Hide&Seek (92.01%) and F5 (98.36%) with 25% message length.

3.2. Multi-classifier

The task of multi-classification is more difficult than the binary classification presented in the previous section. In this section, we compare performance of multi-classifiers employing the original 23 DCT features, the original Markov features, and the new Merged feature set. This comparison better demonstrates the advantages and weaknesses of a particular feature set. Multi-classifiers were trained to classify into 7 classes: cover, F5, OutGuess, JP Hide&Seek, MB1, MB2, and Steghide.

	Message	Detection accuracy		
cover vs.	length	DCT	Markov	Merged
F5	100%	99.49%	99.80%	**99.92%**
	50%	98.80%	99.20%	**99.84%**
	25%	84.54%	86.94%	**98.36%**
	cover	**99.80%**	91.53%	99.64%
JP Hide&Seek	100%	**99.88%**	98.08%	99.52%
	50%	98.56%	84.38%	**99.60%**
	25%	86.46%	27.16%	**92.01%**
	cover	99.32%	97.00%	**99.56%**
MB1	100%	99.64%	**99.96%**	**99.96%**
	50%	98.92%	**99.96%**	99.92%
	25%	86.94%	99.72%	**99.72%**
	cover	97.72%	97.20%	**99.88%**
MB2	30%	92.29%	99.92%	**100.00%**
	cover	98.92%	98.48%	**99.92%**
OutGuess	100%	99.92%	99.92%	**100.00%**
	50%	99.64%	99.68%	**99.96%**
	25%	98.36%	97.84%	**99.48%**
	cover	99.48%	98.04%	**99.76%**
Steghide	100%	99.84%	99.96%	**100.00%**
	50%	99.48%	99.92%	**99.92%**
	25%	90.93%	98.88%	**99.32%**
	cover	97.40%	98.00%	**99.92%**

Table 2. Comparison of detection accuracy of binary classifiers employing 23 DCT, original Markov, and new Merged features. All classifiers were trained and tested on single-compressed JPEG images with quality factor 75. The reported results were calculated for images from the testing set only.

To classify images into $n = 7$ classes, we chose the "max-wins" method which employs $\binom{n}{2}$ binary SVM classifiers for every pair of classes. During classification, the feature vector is presented to all binary classifiers and the histogram of their answers is formed. The class corresponding to the highest peak in the histogram is selected as the target class. According to,[15] the "max-wins" is one of the best current approaches to multiple class problems for practitioners.

Thus, each multi-classifier consists of $\binom{7}{2} = 21$ binary classifiers. The training and testing sets were prepared in exactly the same way, as in Section 3.1. Also, the grid-searches used to find the training parameters of individual binary classifiers were performed in the same fashion.

Tables 3–5 show the confusion matrices of all three classifiers. We again observe the complementary performance of 23 DCT features and the Markov features. The Markov features perform poorly in detecting JP Hide&Seek, F5, and OutGuess, while their detection accuracy of Model Based Steganography is very good.

The multi-classifier employing the new Merged feature set (Table 5) significantly outperformed the other two. Its false positive rate (cover image classified as stego) is 0.84%. The detection of steganographic algorithms on images with longer messages (messages at least 50% long) is highly accurate with the error rate less than 3%. As can be expected, with decreasing message length, the detection accuracy decreases but stays above 90%.

4. MULTI-CLASSIFIER FOR SINGLE-COMPRESSED IMAGES

In the previous sections, we showed that the new Merged feature set enables markedly better blind steganalysis and classification of JPEG images. In this section, we use this feature set to construct a multi-classifier for single-compressed JPEG images for a broad range of 34 quality factors from the set

	Classified as						
Embedding algorithm	Cover	F5	JP Hide&Seek	MB1	MB2	OutGuess	Steghide
F5 100%	0.32%	97.40%	1.04%	0.60%	0.00%	0.12%	0.52%
JP Hide&Seek 100%	0.00%	0.52%	98.32%	0.56%	0.00%	0.12%	0.48%
MB1 100%	0.08%	0.16%	0.72%	94.44%	0.32%	1.56%	2.72%
OutGuess 100%	0.00%	0.04%	0.52%	0.08%	0.04%	99.08%	0.24%
Steghide 100%	0.04%	0.04%	1.68%	2.96%	0.24%	1.52%	93.53%
F5 50%	0.96%	91.65%	0.92%	4.12%	0.28%	0.76%	1.32%
JP Hide&Seek 50%	0.32%	0.88%	90.46%	5.23%	0.04%	0.40%	2.68%
MB1 50%	0.80%	0.52%	0.16%	87.57%	2.20%	1.92%	6.83%
OutGuess 50%	0.08%	0.16%	0.20%	0.48%	0.08%	98.64%	0.36%
Steghide 50%	0.28%	0.44%	0.16%	3.99%	3.47%	2.84%	88.82%
MB2 30%	6.75%	0.40%	0.36%	1.76%	88.46%	0.56%	1.72%
F5 25%	10.99%	63.60%	1.04%	16.98%	2.56%	0.68%	4.16%
JP Hide&Seek 25%	6.15%	1.28%	74.96%	12.74%	0.92%	0.24%	3.71%
MB1 25%	11.02%	1.68%	0.56%	69.17%	6.63%	1.12%	9.82%
OutGuess 25%	1.32%	0.76%	0.24%	2.80%	3.23%	89.14%	2.52%
Steghide 25%	7.07%	1.36%	0.24%	12.42%	11.14%	1.96%	65.81%
Cover	96.45%	0.12%	0.20%	1.44%	0.40%	0.08%	1.32%

Table 3. Confusion matrix of the multi-classifier employing the original DCT feature set (23 features).

	Classified as						
Embedding algorithm	Cover	F5	JP Hide&Seek	MB1	MB2	OutGuess	Steghide
F5 100%	0.16%	98.08%	0.08%	0.92%	0.00%	0.48%	0.28%
JP Hide&Seek 100%	1.32%	2.84%	95.41%	0.08%	0.00%	0.32%	0.04%
MB1 100%	0.00%	0.08%	0.04%	98.48%	0.24%	0.80%	0.36%
OutGuess 100%	0.00%	0.08%	0.04%	0.72%	0.16%	98.04%	0.96%
Steghide 100%	0.00%	0.12%	0.04%	0.76%	0.12%	2.40%	96.57%
F5 50%	2.04%	95.29%	0.32%	1.12%	0.04%	0.68%	0.52%
JP Hide&Seek 50%	12.50%	4.51%	81.71%	0.40%	0.04%	0.44%	0.40%
MB1 50%	0.00%	0.44%	0.04%	97.28%	0.68%	0.76%	0.80%
OutGuess 50%	0.12%	0.68%	0.08%	0.68%	0.16%	95.17%	3.12%
Steghide 50%	0.04%	0.76%	0.00%	1.56%	0.40%	6.47%	90.77%
MB2 30%	0.04%	0.32%	0.08%	1.92%	96.96%	0.12%	0.56%
F5 25%	16.26%	80.42%	0.76%	1.16%	0.12%	0.76%	0.52%
JP Hide&Seek 25%	68.17%	5.15%	25.16%	0.40%	0.12%	0.48%	0.52%
MB1 25%	0.16%	1.40%	0.04%	88.74%	3.00%	2.20%	4.47%
OutGuess 25%	0.84%	2.56%	0.36%	2.00%	0.40%	84.90%	8.95%
Steghide 25%	0.60%	1.72%	0.24%	5.43%	1.36%	14.06%	76.60%
Cover	91.61%	5.51%	1.40%	0.48%	0.12%	0.44%	0.44%

Table 4. Confusion matrix of the multi-classifier employing the original Markov feature set (324 features).

$$Q_{34} \quad = \quad \{63, \ldots, 94, 96, 98\}.$$

This multi-classifier can be constructed in two fundamentally different ways. We can either add the quality factor as an additional feature or we can prepare a dedicated multi-classifier for each quality factor from the set Q_{34}. Because of the following reasons, we opted for the latter design.

1. Since the statistics of DCT coefficients varies greatly with the quality factor, the influence of the additional feature (the quality factor), whose purpose is to "shift" the classification hyperplane in the

	Classified as						
Embedding algorithm	Cover	F5	JP Hide&Seek	MB1	MB2	OutGuess	Steghide
F5 100%	0.00%	99.52%	0.04%	0.08%	0.04%	0.08%	0.24%
JP Hide&Seek 100%	0.32%	0.00%	99.64%	0.00%	0.00%	0.04%	0.00%
MB1 100%	0.00%	0.00%	0.04%	98.76%	0.44%	0.04%	0.72%
OutGuess 100%	0.00%	0.04%	0.04%	0.08%	0.00%	99.64%	0.20%
Steghide 100%	0.00%	0.00%	0.04%	0.12%	0.08%	0.44%	99.32%
F5 50%	0.16%	99.36%	0.00%	0.00%	0.04%	0.24%	0.20%
JP Hide&Seek 50%	0.28%	0.04%	99.60%	0.00%	0.00%	0.08%	0.00%
MB1 50%	0.00%	0.00%	0.04%	97.04%	1.36%	0.08%	1.48%
OutGuess 50%	0.04%	0.08%	0.00%	0.20%	0.12%	99.28%	0.28%
Steghide 50%	0.04%	0.00%	0.00%	0.36%	0.12%	0.76%	98.72%
MB2 30%	0.00%	0.04%	0.04%	1.08%	98.48%	0.00%	0.36%
F5 25%	1.84%	97.12%	0.20%	0.00%	0.16%	0.36%	0.32%
JP Hide&Seek 25%	8.23%	0.32%	91.45%	0.00%	0.00%	0.00%	0.00%
MB1 25%	0.12%	0.12%	0.04%	90.10%	1.92%	0.36%	7.35%
OutGuess 25%	0.52%	0.28%	0.04%	0.20%	0.08%	98.08%	0.80%
Steghide 25%	0.60%	0.04%	0.00%	0.76%	0.20%	1.44%	96.96%
Cover	99.16%	0.24%	0.44%	0.00%	0.08%	0.08%	0.00%

Table 5. Confusion matrix of the multi-classifier employing the new Merged feature set (274 features).

feature space, might fade out among the other 274 features. Consequently, the features for images with different quality factors might get mixed up, which will confuse the detector. The collection of dedicated multi-classifiers will perform better because this mixing is prevented by design.

2. The complexity of training of binary $C-$SVMs is $O(n_{\mathrm{im}}^3)$, where n_{im} is the number of training examples. Thus, training the collection of multi-classifiers is faster, which allows us to use more examples for training. For the same number of examples for training, the ratio between the training complexity for one classifier and for separate 34 classifiers is proportional to the square of the number of quality factors $34^2 = 1156$.

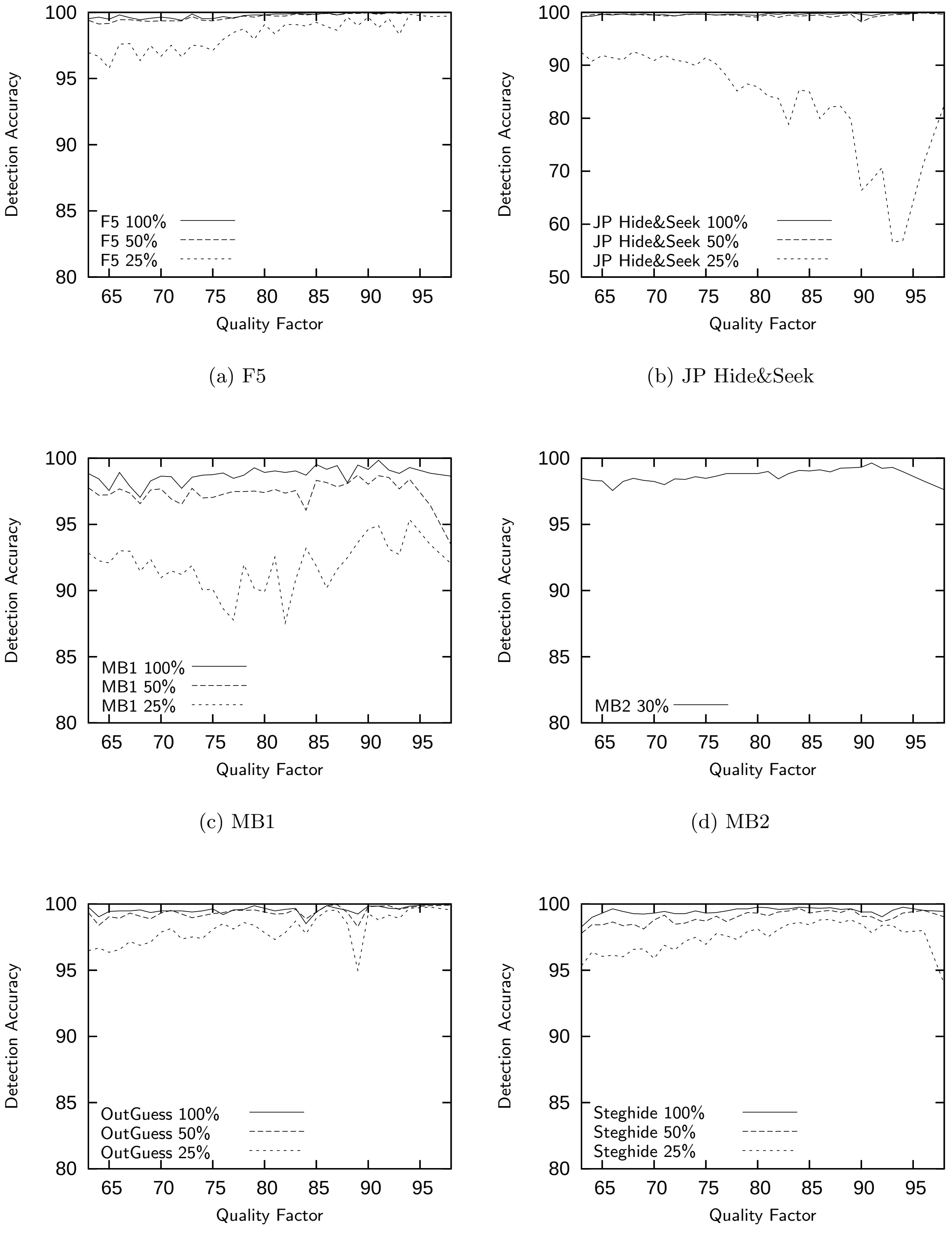
For each quality factor, the training and testing sets as well as the multi-classifiers were prepared in exactly the same way as in Section 3.2. We had to modify the implementation of OutGuess ver. 0.2[2] to produce JPEGs with quality factor lower than 75, since the original version was only able to produce JPEG images with quality factor 75 or higher.

The total number of images used for training was $34 \times 17 \times 3400 = 1,965,200$ and for testing $24 \times 17 \times 3400 = 1,445,000$.

4.1. Discussion of results

One of the benefits of the analysis reported in the previous section is mutual comparison of statistical detectability of the steganographic algorithms. Which algorithm offers the best security? This comparison cannot be done directly, however, because we embedded a fixed percentage of embedding capacity for each algorithm and these capacities vary significantly across algorithms. Figure 3 shows the absolute embedding capacity (in bits per non-zero DCT coefficient) for each steganographic algorithm averaged over 6000 images. We can see that F5, JP Hide&Seek, and MB1 are high-capacity algorithms when compared to OutGuess, Steghide, or MB2. Interpreting the detection results of Figure 1 while taking into account the absolute embedding capacity of each algorithm, we can conclude that OutGuess is by far the most detectable algorithm, while MB1 is the least detectable.

Also, with the exception of JP Hide&Seek, the detection is slightly more reliable for larger quality factors than for lower quality factors. We do not know the reason for the dip in detectability of 25% messages for JP Hide&Seek around the quality factor 90.



(a) F5

(b) JP Hide&Seek

(c) MB1

(d) MB2

(e) OutGuess

(f) Steghide

Figure 1. Detection accuracy in percents for the multi-classifiers trained for each JPEG quality factor for all 6 tested steganographic methods. The false positive rate is shown in Figure 2.

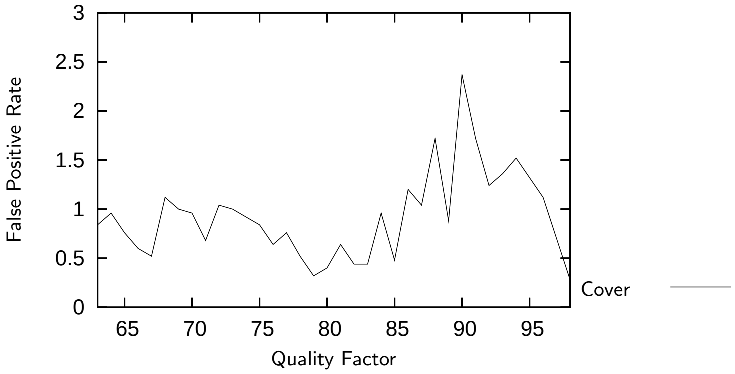


Figure 2. False positive rate in percents for the multi-classifiers trained for each JPEG quality factor.

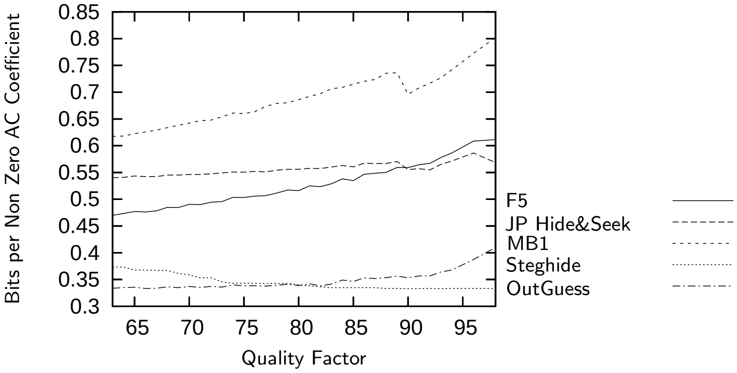


Figure 3. Capacities of five popular steganographic algorithms as a function of JPEG quality factor averaged over 6000 images. We consider the capacity of MB2 as 30% of capacity of MB1.

Note that the detection accuracy on MB2 embedded images containing 30% messages (30% capacity of MB1) is better than the detection accuracy on MB1 images containing 50% messages. To better see the difference in detectability, we plot the detection rate for both algorithms in Figure 4. This means that the steganalyzer is more successful in detecting shorter messages embedded by MB2 than in detecting longer messages in images embedded by MB1. Thus, based on this steganalysis engine, MB1 is less detectable than its more advanced version, MB2. MB2 introduces more embedding changes into the JPEG file with the goal to preserve a selected higher-order inter-block statistics, the blockiness. As a result, however, it disturbs other statistics and eventually becomes more detectable. In other words, MB2 has lower embedding efficiency[†] than MB1. This finding is consistent with what was recently reported in[21] and is in contrast with older experiments using the original 23 dimensional DCT feature set and a simple linear classifier.[10]

5. CONCLUSIONS

In this paper, we present and test a new set of features for steganalysis of JPEG images with a wide range of quality factors. The feature set was obtained by merging and modifying two previously proposed feature sets with complementary performance (the 23 DCT feature set[10] that captures inter-block dependencies among DCT coefficients and Markov features[21] which capture intra-block dependencies). In particular, we expanded the DCT features by replacing the L_1 norm in their calibration by differences and we added calibration to Markov features and reduced their dimensionality by a factor of 4. According to our experiments on multi-classification of single-compressed JPEG images, the new merged feature set provides significantly better results than previous art.

[†] The average number of bits embedded per one embedding change.

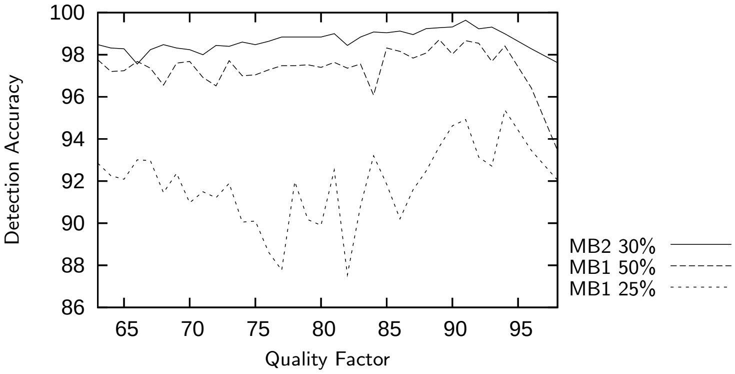


Figure 4. Detection accuracy of the multi-classifier described in Section 3.2 on images embedded by MB1 with 50% message length and MB2 embedded with 30% message length.

We have determined that the more advanced version of Model Based Steganography with deblocking is more detectable than the version without deblocking. This indicates that embedding efficiency is a more influential attribute for steganographic security than was previously thought. This finding also puts a new perspective on the design principle that strives to preserve selected statistics by introducing more embedding changes (e.g., the mechanism of embedding in OutGuess and in MB2).

Right now, images that underwent double compression will be with high probability misclassified by our steganalyzer. This is because double JPEG compression drastically changes the statistics of DCT coefficients. We intend to extend our work to correctly handle double compressed JPEG images by first analyzing each image for signs of double compression and estimating the previous quality factor. This will be a pre-processing step applied before blind steganalysis. Double-compressed images will then be handled separately through a different SVM multi-classifier that will only classify to algorithms capable of producing double compressed images (F5 and OutGuess) and to the cover class. Single compressed images will be sent to the classifier constructed in Section 3.2.

REFERENCES

1. JP Hide&Seek. `http://linux01.gwdg.de/~alatham/stego.html`.
2. Outguess ver. 0.2. `http:\\www.outguess.org`.
3. I. Avcibas, M. Kharrazi, N. Memon, and B. Sankur. Image steganalysis with binary similarity measures. *EURASIP Journal on Applied Signal Processing*, 17:2749–2757, 2005.
4. I. Avcibas, N. Memon, and B. Sankur. Steganalysis using image quality metrics. In E. Delp and P. W. Wong, editors, *Proceeedings of SPIE Electronic Imaging, Security and Watermarking of Multimedia Contents III*, volume 4314, pages 523–531, 2001.
5. I. Avcibas, B. Sankur, and N. Memon. Image steganalysis with binary similarity measures. In *Proceedings of International Conference on Image Processing*, volume 3, pages 645–648, 2002.
6. Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
7. 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, 1998.
8. H. Farid and S. Lyu. Steganalysis using higher-order image statistics. *IEEE Transactions on Information Forensics and Security*, 1(1):111–119, 2006.
9. 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, 2002.

10. 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, 2005.
11. J. Fridrich and T. Pevný. Towards multi–class blind steganalyzer for JPEG images. In M. Barni, I. Cox, T. Kalker, and H. J. Kim, editors, *4th International Data Hiding Workshop*, volume 3710 of *Lecture Notes in Computer Science*, pages 39–53, 2005.
12. J. Fridrich and T. Pevný. Multi–class blind steganalysis for JPEG images. In E. Delp and P. W. Wong, editors, *Proceedings of SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, 2006.
13. J.J. Harmsen and W.A. Pearlman. Steganalysis of additive noise modelable information hiding. In *Proceedings of SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents V*, pages 131–142, Santa Clara, CA, 2003.
14. S. Hetzl and P. Mutzel. A graph-theoretic approach to steganography. In J. Dittmann et al., editor, *Communications and Multimedia Security. 9th IFIP TC-6 TC-11 International Conference*, volume 3677 of *Lecture Notes in Computer Science*, pages 119–128, 2005.
15. C. Hsu and C. Lin. A comparison of methods for multi-class support vector machines. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001. http://citeseer.ist.psu.edu/hsu01comparison.html.
16. S. Lyu and H. Farid. Steganalysis using color wavelet statistics and one-class support vector machines. In E. Delp and P. W. Wong, editors, *Proceedings of SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306, pages 35–45, 2004.
17. P. Moulin and Y. Wang. Statistical modeling and steganalysis of DFT-based image steganography. In E. Delp and P. W. Wong, editors, *Proceedings of SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 607202–1–607202–11, 2006.
18. N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, 2001.
19. P. Sallee. Model based steganography. In Kalker, I.J. Cox, and Yong Man Ro, editors, *International Workshop on Digital Watermarking*, volume 2939 of *Lecture Notes in Computer Science*, pages 154–167, 2004.
20. Phil Sallee. Model-based methods for steganography and steganalysis. *Int. J. Image Graphics*, 5(1):167–190, 2005.
21. Y. Q. Shi, C. Chen, and W. Chen. A Markov process based approach to effective attacking JPEG steganography. In *Proceedings of the 8-th Information Hiding Workshop*, 2006.
22. 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, 2001.
23. G. Xuan, Y.Q. Shi, J. Gao, D. Zou, C. Yang, Z. Zhang, P. Chai, C. Chen, and W. Chen. Steganalysis based on multiple features formed by statistical moments of wavelet characteristic function. In M. Barni, editor, *Information Hiding. 7th International Workshop*, volume 3727 of *Lecture Notes in Computer Science*, pages 262–277, 2005.