Steganalysis of DCT-Embedding Based Adaptive Steganography and YASS

Qingzhong Liu
Department of Computer Science
Sam Houston State University
Huntsville, TX 77341, U.S.A.
liu@shsu.edu

ABSTRACT

Recently well-designed adaptive steganographic systems, including ±1 embedding in the DCT domain with optimized costs to achieve the minimal-distortion [8], have posed serious challenges to steganalyzers. Additionally, although the steganalysis of Yet Another Steganographic Scheme (YASS) was actively conducted, the detection of the YASS steganograms by a large B-block parameter has not been well explored.

In this paper, we aim to detect the state-of-the-art adaptive steganographic system in DCT-embedding and to improve the steganalysis of YASS. To detect DCT-embedding based adaptive steganography, we design the features of differential neighboring joint density on the absolute array of DCT coefficients between the original JPEG images and the calibrated versions. To discriminate YASS steganograms from covers, the candidate blocks that are possibly used for embedding and the non-candidate block neighbors that are impossibly used for information hiding are identified first. The difference of the neighboring joint density between candidate blocks and the non-candidate neighbors is obtained. Support Vector Machine (SVM) and logistic regression classifiers are employed for classification.

Experimental results show that our approach is very promising when detecting DCT-embedding based adaptive steganography. Compared to the steganalysis based on CC-PEV feature set, our method greatly improves the detection accuracy; the advantage is especially noticeable in the detection of the steganograms with low relative payload. In steganalysis of YASS, our approach is superior to a previous well-known steganalysis algorithm; our method remarkably improves the detection accuracy especially in the detection of the YASS steganograms that are produced with a large B-block size, which was not well addressed before.

Categories and Subject Descriptors

I 4.9 **[Image Processing and Compute r Vision**]: Applications; K.6.m **[Miscellaneous**]: Insurance and Security.

General Terms

Algorithms and Security

Keywords

Steganography, steganalysis, adaptive steganography, YASS, DCT, JPEG, neighboring joint density, logistic regression, SVM, CC-PEV.

MM&Sec'11, September 29–30, 2011, Buffalo, New York, USA.

.

1. INTRODUCTION

Steganography aims to enable covert communication by embedding data into digital files and making the hidden message invisible. The potential of exploiting steganography for covert dissemination is great: for example, a recent espionage issue reveals that steganography has been used by governmental intelligent agency [1-3]. For several purposes, it is a heightened need to realize effective countermeasures for steganography.

In digital images, to this date, a few popular steganographic systems such as LSB embedding, LSB matching [35, 41], spread spectrum steganography [34], Outguess [38], F5 [45**]**, model-based steganography [40], steghide [15], BCH syndrome code based less detectable JPEG steganography [39], and recently well-designed highly undetectable steganography (HUGO) [37], have been successfully steganalyzed [6, 10-14, 19, 20, 22, 24-26, 28, 33, 36, 42]. Although these remarkable advances have been achieved, recently well-designed steganographic systems, such as Gibbs construction-based steganography [7], Syndrome-Trellis Codes based steganography [9], have posed new challenges for steganalysis. Filler and Fridrich recently presented a practical framework of adaptive steganographic systems [8] by optimizing the parameters of additive distortion functions and minimizing the distortion for ±1 embedding in the DCT domain, which greatly improves the prior art of hiding data in wide-spread JPEG images. The experimental results shown in [8] demonstrated DCT-embedding based adaptive system is undetectable, attacked by a previous state-of-the-art steganalysis method, while the relative payload ratio is less than 0.15 bpac.

YASS was designed to be a secure JPEG steganographic algorithm with randomized embedding [43]. However, the locations of the embedding host blocks are not randomized enough. The embedding in YASS also introduces extra zero DCT coefficients into the embedding host blocks, and hence leaves a clue to be exposed Li et al. presented a simple and efficient detection method by comparing the frequency of zero coefficients of the embedding host blocks and the neighboring blocks in DCT domain [23]. The detection performance is very promising while the parameter of the big block (B-block) size is small (e.g., the size is set to 9 and 10). However, the detection performance apparently deteriorates while the parameter of B-block size increases [23]. Recently, Kodovsky et al. designed 1234 features to detect YASS and tested twelve different configurations of YASS. In these twelve configurations, the parameter of B-block size is not larger than 11 [21]. However, the detection

performance on the YASS steganograms with large parameter of B-block size (12, 13, 14, and 15) was missing [21].

Aiming to detect the state-of-the-art DCT-embedding based adaptive steganography [8], we improve a previous JPEG steganalysis method [28]. We extract the neighboring joint density on the absolute array of DCT coefficients from the JPEG image under examination, and we design a calibrated algorithm to extract the reference features; the differential features between the original neighboring joint density and the reference are calculated. Support Vector Machines (SVM) [44] and logistic regression [16] classifiers are applied for classification. Experimental results demonstrate promising detection performance of our approach.

To improve the detection performance in steganalysis of YASS, we first analyze the advantage and weakness of a previous YASS detection art presented in [23]. As indicated by Li et al. [23], in YASS embedding, the selection of embedding host block is not random enough and the embedding modifies the statistics of embedding host blocks in DCT domain. However, their detection algorithm does not search all possible candidate blocks that are possibly used for embedding. Our study also finds that the YASS embedding not only increases the zero coefficients of the host blocks, but it also modifies the neighboring joint density of the DCT coefficients. Therefore, we design a new algorithm to improve the detection performance.

The remainder of the paper is organized as follows: Section 2 briefly introduces DCT-embedding based adaptive steganography, YASS and a detection art, and our previous JPEG steganalysis method. Section 3 presents our detection method for steganalysis of DCT-embedding based adaptive steganography, and section 4 describes our approach to steganalysis of YASS. Section 5 shows experiments and analysis. Conclusions are made in section 6.

2. BACKGROUND

2.1 DCT-embedding based Adaptive Steganography

Most steganographic systems aim to minimize the distortion of original cover. In [8], a practical framework for optimizing the parameters of additive distortion functions to minimize statistical detectability was presented by defining a rich parametric model. To realize DCT-embedding based adaptive steganography, an inter/intra-block cost model was given, as well as the performance of embedding algorithms based on the inter/intra-block cost model when optimized using the L2-regularized L2-loss (L2R_L2LOSS) criterion, attacked by a previous state-of the art steganalysis method, which was experimentally validated as being completely undetectable at low relative payload [8]. In what follows, we briefly introduce this practical framework for DCT-embedding based adaptive steganography.

Minimal-distortion steganography can be implemented by minimizing the following cost function

$$D(\mathrm{x},\mathrm{y}) = \sum_{i=1}^{n} \rho_i\left(\mathrm{x}, y_i\right) \tag{1}$$

where $\rho_i\left(\mathrm{x}, y_i\right) \in \Re$ is the cost changing the i^{th} cover pixel x_i to y_i.

To design DCT-embedding based adaptive steganography, an inter/intra-block cost model has been defined by Filler and Fridrich [8]. Let $\theta = \left(\theta_{ir}, \theta_{ia}\right) \in \Re^{(2\Delta+1)+1} \times \Re^{(2\Delta+1)+1}$ be the model parameters describing the cost of disturbing inter- and intra-block dependencies with $\theta_{ir} = (\theta_{ir,-\Delta}, \ldots, \theta_{ir,\Delta}, \theta_{ir,\bullet})$ and $\theta_{ia} = (\theta_{ia,-\Delta}, \ldots, \theta_{ia,\Delta}, \theta_{ia,\bullet})$. The cost of changing any AC DCT coefficients x_{ij} to $y \in \mathrm{I}_{ij}$ $\mathrm{I}_{ij} = \left\{x_{ij}-1, x_{ij}, x_{ij}+1\right\} \bigcap \mathrm{I}$ is given by

$$\rho_{ij}(\mathrm{x},y) = \Theta(y) = \begin{cases} 0 & \text{if } y = x_{ij} \\ \infty & \text{if } y \notin \mathrm{I}_{ij} \\ \sum_{z\in \mathrm{N}_{ia}} \theta^2{}_{ia,x_{ij}-z} + \sum_{z\in \mathrm{N}_{ir}} \theta^2{}_{ir,x_{ij}-z} & \text{otherwise} \end{cases} \tag{2}$$

Where, N_{ia} and N_{ir} are intra- and inter-block neighborhoods. Based on the inter/intra-block cost model, while the embedding algorithms are optimized by using the multi-layered Syndrome-Trellis Codes [9] to minimize the L2R_L2LOSS criterion [8], with SVM and CC-PEV feature set [20], and Cross-Domain Feature set [21, 8], respectively, the experiments show that proposed DCT-embedding based adaptive steganography has greatly improved the state of DCT-embedding based steganography. More technical details may be referred to [8].

2.2 YASS and A Previous Detection Algorithm

The original YASS algorithm, presented in [43], includes the following steps:

1) Repeat-Accumulate error correction code is used to encode the payload;
2) The cover image is divided into big blocks of T×T (T =9, 10, …, 15), denoted by B-block;
3) In each B-block, an 8×8 block is randomly selected for payload embedding;
4) The embedding includes the following operations:
   a) Selected 8×8 block is transformed using a two-dimensional DCT;
   b) The DCT coefficients are divided by a quantization table, corresponding to the hiding quality factor $\mathrm{QF_h}$;
   c) By using QIM, binary hidden bits are embedded into the 19 low-frequency AC DCT coefficients whose values are non-zeros;
   d) The modified 8×8 block is transformed back to spatial domain;
5) The modified image is encoded in JPEG format with the advertising quality factor QF_a.

Although YASS embedding is not confined to the 8×8 block of the final JPEG compression in above step 5), the location of embedding block in B-block is not random enough. Meanwhile, the QIM-based embedding introduces additional zero DCT coefficients in the modified 8×8 block, and hence, Li et al. designed the following algorithm to break YASS [23].

Li et al. feature extraction algorithm for YASS detection [23]

Transform a JPEG image under examination to spatial domain, denoted by I_1;

For T = 9 to 15
  For s = 1 to T
    (a) Divide I_s into non-overlapping consecutive $T\times T$ B-blocks;
    (b) Collect 8×8 blocks from the upper left of all B-blocks and perform 2D DCT;

(c) Quantize the DCT coefficients by using QF_a;
(d) Compute the probability of zero rounded re-quantized DCT coefficients in candidate embedding bands and denote it by $Z_T(s)$;
(e) Crop the first s columns and the first s rows of I_1 to generate a new image I_{s+1} for the next inner-loop;
End

Compute the values of $\frac{1}{T-7}\sum_{i=1}^{T-7} Z_T(i)$ and $\frac{1}{7}\sum_{j=T-6}^{T} Z_T(j)$ as features.

End

As shown by the above algorithm, the features are extracted from the candidate blocks along the diagonal direction of B-blocks, not from all possible 8×8 candidate blocks in B-blocks. In a $T\times T$ B-block, there are (T-7) ×(T-7) block candidates for embedding. Unfortunately, the above algorithm only selects the (T-7) blocks along diagonal direction, not all (T-7) ×(T-7) candidate blocks. As a result, the chance of the candidates along diagonal direction only hits 1/(T-7). While the value of T is large, the hit ratio is pretty low. For instance, T=15, the hit ratio is only 1/8 = 0.125. The experimental results in the reference [23] also demonstrate that the detection accuracy is not so good with a large T value.

2.3 Neighboring Joint Density based JPEG Steganalysis

We have shown that information-hiding in DCT domain generally modifies the neighboring joint density [26, 28]. Accordingly, a JPEG-based steganalysis method based on neighboring joint density was proposed, which has been validated to outperform Markov-process based steganalysis that was originally presented in [6, 42]. Here we briefly introduce our prior detection method.

Our previous study shows that certain manipulations such as JPEG-based double compression, information hiding, and resampling, modify the neighboring joint density and leave a clue to reveal the operations [26-29]. In general, neighboring joint density of DCT coefficients is symmetric about the origin. We designed the neighboring joint density features on the absolute array of DCT coefficients, described as follows.

2.3.1 Neighboring Joint Density on Intra-block

Let F denote the quantized DCT coefficient array consisting of M×N blocks F_{ij} (i = 1, 2, …, M; j = 1, 2, …, N). The intra-block neighboring joint density matrix on horizontal direction $absNJ_{1h}$ and the matrix on vertical direction $absNJ_{1v}$ are given by:

$$absNJ_{1h}(x,y)=\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{m=1}^{8}\sum_{n=1}^{7}\delta(|c_{ijmn}|=x,|c_{ijm(n+1)}|=y)}{56MN} \quad (3)$$

$$absNJ_{1v}(x,y)=\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{m=1}^{7}\sum_{n=1}^{8}\delta(|c_{ijmn}|=x,|c_{ij(m+1)n}|=y)}{56MN} \quad (4)$$

Where c_{ijmn} is the DCT coefficient located at the m^{th} row and the n^{th} column in the block F_{ij}; δ = 1 if its arguments are satisfied, otherwise δ = 0; x and y are integers. For computational efficiency, we define $absNJ_1$ as the neighboring joint density features on intra-block, calculated as follows:

$$absNJ_1(x,y)=\{absNJ_{1h}(x,y)+absNJ_{1v}(x,y)\}/2 \quad (5)$$

In our prior detection, the values of x and y are in the range [0, 5], and $absNJ_1$ consists of 36 features.

2.3.2 Neighboring Joint Density on Inter-block

The inter-block neighboring joint density matrix on horizontal direction $absNJ_{2h}$ and the matrix on vertical direction $absNJ_{2v}$ are constructed as follows:

$$absNJ_{2h}(x,y)=\frac{\sum_{m=1}^{8}\sum_{n=1}^{8}\sum_{i=1}^{M}\sum_{j=1}^{N-1}\delta(|c_{ijmn}|=x,|c_{i(j+1)mn}|=y)}{64M(N-1)} \quad (6)$$

$$absNJ_{2v}(x,y)=\frac{\sum_{m=1}^{8}\sum_{n=1}^{8}\sum_{i=1}^{M-1}\sum_{j=1}^{N}\delta(|c_{ijmn}|=x,|c_{(i+1)jmn}|=y)}{64(M-1)N} \quad (7)$$

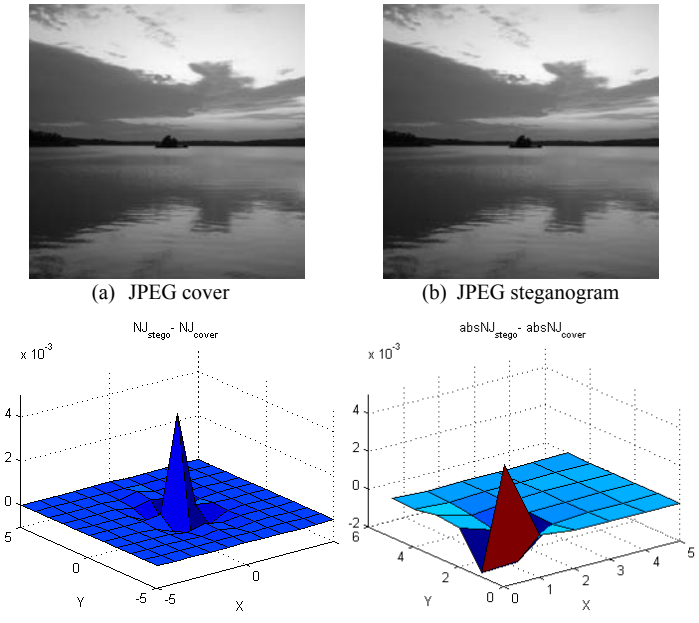
We define $absNJ_2$ as the neighboring joint density features on inter-block, calculated as follows:

$$absNJ_2(x,y)=\{absNJ_{2h}(x,y)+absNJ_{2v}(x,y)\}/2 \quad (8)$$

Similarly, the values of x and y are in [0, 5] and $absNJ_2$ has 36 features.

3. CALIBRATED NEIGHBORING JOINT DENSITY APPROACH TO STEGANALYSIS OF JPEG-BASED ADAPTIVE STEGANOGRAPHY

Although DCT-embedding based adaptive steganography aims to minimize the distortion cost through Syndrome-Trellis Codes, we find that it does modify the neighboring joint density features proposed in reference [28], shown by Figure 1.



(a) JPEG cover (b) JPEG steganogram

(c) Difference of NJ density (d) Difference of absNJ density

Figure 1. An example to demonstrate the modification of neighboring joint density features by DCT-embedding based adaptive steganography.

Figure 1(a) and (b) show a JPEG cover and the JPEG steganogram produced by using DCT-embedding based adaptive hiding algorithm [8] with the hiding ratio 0.4 bits per non-zero-AC (bpac). The cover image is downloaded from [4]. The adaptive hiding tool is available at [5]. Figure 1(c) shows the difference of the intra-block based neighboring joint density extracted from (a) and (b). Figure 1(d) shows the difference of the neighboring joint density on the absolute array of DCT coefficients, defined by equation (5). Although the modification is small, the information-hiding does modify the neighboring joint density.

Considering that the joint density varies across different digital images, to reflect the modification of the density caused by the embedding, based on a self-calibration approach that was presented in [11], we design a calibrated neighboring joint density, described as follows:

1. The neighboring joint density features $\text{absNJ}_1(x,y)$ and $\text{absNJ}_2(x,y)$, defined by equations (5) and (8), are extracted from a JPEG image under examination;
2. The testing JPEG image is decoded in spatial domain, and cropped by i rows and j columns ($0 \le i < 7$, $0 \le j < 7$, and $i+j > 0$). The cropped image is encoded in JPEG format with the same quantization matrix, and the joint density features, denoted by $\text{absNJ}_1^{c}{}_{i,j}(x,y)$ and $\text{absNJ}_2^{c}{}_{i,j}(x,y)$, are extracted from the cropped JPEG images, here
$$(i,j) \in \{(0,1),(0,2),...,(1,0),(1,1),...,(7,7)\};$$
3. The mean values of absNJ_1^{c} and absNJ_2^{c} are calculated by
$$\overline{absNJ}_1^{\,c}(x,y) = \frac{1}{63}\sum_{(i,j)} absNJ_1^{c}{}_{i,j}(x,y) \quad (9)$$
$$\overline{absNJ}_2^{\,c}(x,y) = \frac{1}{63}\sum_{(i,j)} absNJ_2^{c}{}_{i,j}(x,y) \quad (10)$$
4. The differential joint density features are given by
$$absNJ_1^{D}(x,y) = absNJ_1(x,y) - \overline{absNJ}_1^{\,c}(x,y) \quad (11)$$
$$absNJ_2^{D}(x,y) = absNJ_2(x,y) - \overline{absNJ}_2^{\,c}(x,y) \quad (12)$$
5. The differential ratio features are obtained by
$$R_{absNJ_1}(x,y) = absNJ_1^{D}(x,y) \Big/ \overline{absNJ}_1^{\,c}(x,y) \quad (13)$$
$$R_{absNJ_2}(x,y) = absNJ_2^{D}(x,y) \Big/ \overline{absNJ}_2^{\,c}(x,y) \quad (14)$$

The ratio features are defined in equations (13) and (14), denoted by *diff-absNJ-ratio*, and the features defined by equations (9) to (12), denoted by *ref-diff-absNJ*, are used to detect DCT-embedding based adaptive steganography. In our study to detect adaptive steganography, the integer parameters x and y are set from 0 to 5, producing 36 features in (13) and 36 features in (14), so *diff-absNJ-ratio* contains 72 features, and *ref-diff-absNJ* contains 144 features.

4. NEIGHBORING JOINT DESNITY BASED YASS-DETECTION ALGORITHM

By searching all possible 8×8 candidate blocks in B-blocks, we extract the neighboring joint density of the DCT coefficients from all candidate blocks and the 8×8 block neighbors that impossibly belong to the candidate set for information hiding, and calculate the difference of the joint density values of the candidates and the non-candidate neighbors. Our algorithm of feature design to detect YASS steganogram is described as follows:

1. Decode an input JPEG image under scrutiny to spatial domain, and divide it into non-overlapping consecutive $T \times T$ B-blocks (T = 9, 10, …, 15);
2. In each $T \times T$ B-block, search all 8×8 blocks possibly used for information hiding, total $(T-7)^2$ candidate blocks. The set of all candidate blocks of the image under detection is denoted by C. For each candidate block C(i) (i=1,2, …, cn), subtract 128 from each pixel value, then apply two-dimensional DCT transform, quantize the DCT coefficients by using the quantization matrix corresponding to QF_a and obtain the absolute DCT coefficient array. The neighboring joint density features, defined by equation (5), are extracted from the absolute DCT coefficient array, denoted by $absNJ(i; x,y)$.
3. From all adjacent 8×8 blocks to the candidate block C(i) in the horizontal or vertical direction but without any overlapping to C(i), denoted by N(i), we identify the adjacent 8×8 blocks that do not belong to C, the set of candidate blocks for YASS embedding. The non-candidate block neighbors are denoted by NC(i). The neighboring joint density defined by equation (5) are extracted from these non-candidate neighboring blocks, and the average neighboring joint density is denoted by $avg\_NC\_absNJ(i; x, y)$, the difference of the joint density from the candidate block C(i) and the average neighboring joint density is given by
$$diff\_absNJ(i; x, y) = absNJ(i; x, y) - avg\_NC\_absNJ(i; x, y) \quad (15)$$
4. The total number of candidate blocks is cn. The mean values of the differential joint density, which are the features for YASS steganalysis in our algorithm, are given by the following
$$diff\_absNJ(x,y) = \sum_i diff\_absNJ(i; x, y) \Big/ cn \quad (16)$$

It should be noted that in a $T \times T$ B-block, which is not on the boundary of the image under examination, if an 8×8 block candidate is located

(a) inside of the B-block, it has four non-candidate neighbors, shown by Figure 2(a);
(b) on one of the four boundary borders of the B-block but not on any corner, it has three non-candidate neighbors, shown by Figure 2(b);
(c) on one of the four corners of the B-block, it has two non-candidate neighbors, shown by Figure 2(c);

Figures 2 (a), (b), and (c) illustrate the above scenarios. The square in dash stands for a B-block, a complete block in the B-block represents a candidate block for possible hiding and the non-candidate block neighbors are across the square.

In our YASS detection, the values of x and y are set in [0, 2] and *diff_absNJ* contains 9 features, corresponding to each value of T. We expect that the *diff_absNJ* features extracted from covers are approximately zero-valued, but the values of the features from YASS steganograms are not constrained to zeros. Figure 3 validates our conjecture and implies the effectiveness of our proposed features.

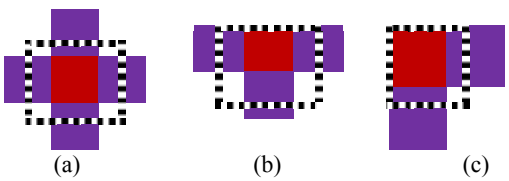
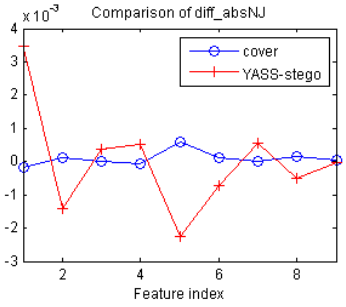


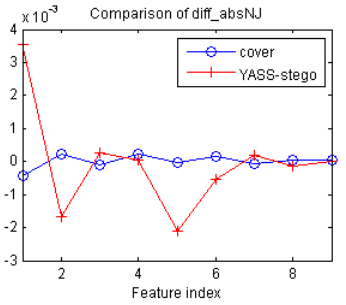
Figure 2. A candidate block is located in a B-block (dashed) and the non-candidate neighbors are across two B-blocks.

(a) YASS steganogram (T=9)



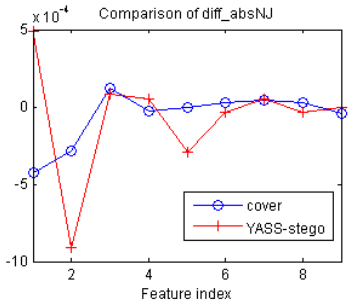
(b) diff_absNJ features of the cover and the left YASS stego-image (T = 9)

(c) YASS steganogram (T=10)



(d) diff_absNJ features of the cover and the left YASS stego-image (T = 10)

(e) YASS steganogram (T=15)



(f) diff_absNJ features of the cover and the left YASS stego-image (T = 15)

Figure 3. Modification of the diff-absNJ features by YASS embedding ($QF_h = QF_a = 75$) with B-block size T=9, 10, and 15, wherein the feature indices from 1, 2, …, to 9 correspond to the (x, y) pairs in equation (16) from (0, 0), (0, 1), …, to (2,2).

5. EXPERIMENTS

5.1 Steganalysis of DCT-Embedding based Adaptive Steganography

5.1.1 Setup

1000 BOSSRank cover images downloaded from [4] are first converted into JPEG images with the quality factor "75". The JPEG-based adaptive steganograms are produced by using the DCT-embedding based hiding tool [5], and the parameter of hiding bits per non-zero-AC (bpac) is set from 0.1 to 0.4 with the step of 0.05 bpac. We extract 72-dimensional ratio features, defined by (13) and (14), or *diff-absNJ-ratio*, and 144-dimensional features, or *ref-diff-absNJ*, from the JPEG covers and the adaptive steganograms. To compare our feature sets and a recently well-designed feature set, CC-PEV [20, 36], we also extract the 548-dimensional CC-PEV features from the covers and steganograms. A logistic regression classifier [16, 32] and Support Vector Machines (SVM) [42], are used for the detection. In each experiment, 50% samples are randomly selected for training, and the other 50% samples are used for testing. In each experiment, the testing results can be divided into True Negative (TN), False Negative (FN), False Positive (FP), and True Positive (TP). Without losing a generality, we measure the detection accuracy by 0.5*TN/(TN+FP)+0.5*TP/(TP+FN). To compare the detection performance, two hundred experiments are operated for each feature set at each hiding ratio by using each classifier, and the mean detection accuracy over 200 experiments is obtained. In the application of SVM, we particularly adopt two popular SVM algorithms, LibSVM [46] and SVM_light [17], and we compare the detection performance of these two SVM implementation algorithms with linear, polynomial, and radial basis function (RBF) kernels. On average, in our experiments, a linear LibSVM hits the highest detection accuracy.

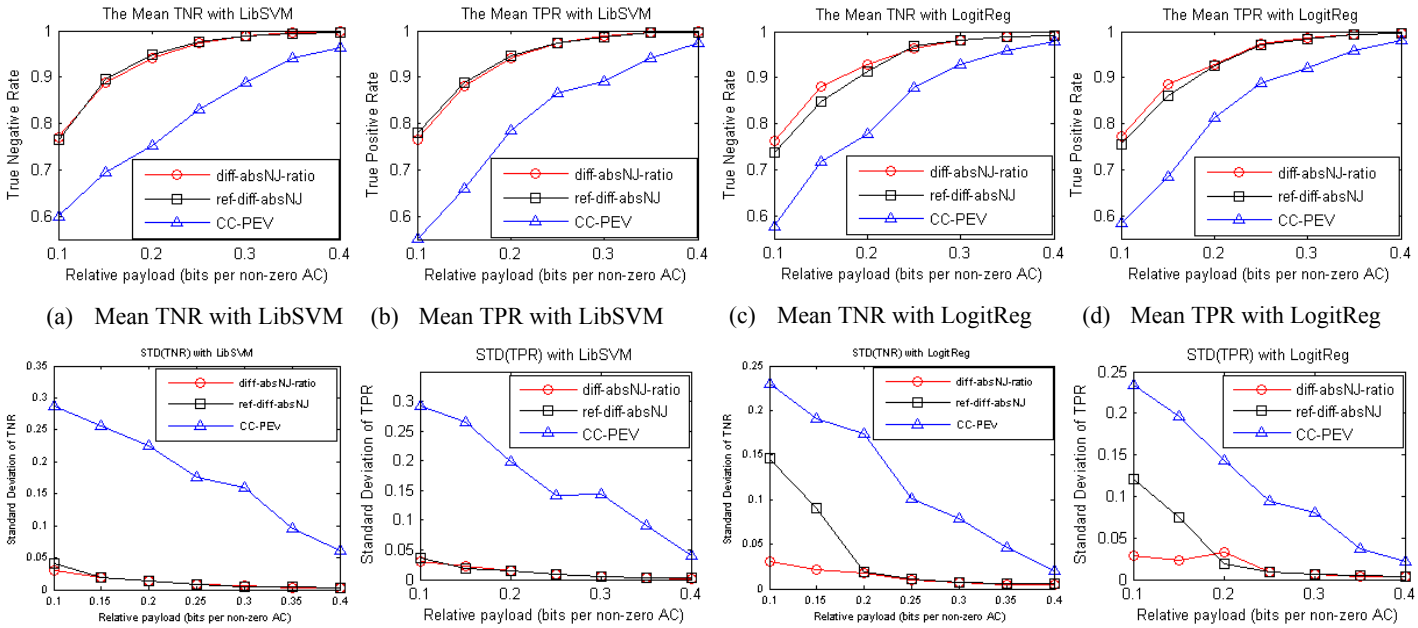
5.1.2 Experimental Results

Table 1 lists the mean values of detection accuracy on testing feature sets over two hundred experiments by using the 72-dimensional *diff-absNJ-ratio* feature set, 144-dimensional *ref-diff-absNJ* feature set, and 548-dimensional *CC-PEV* feature set with linear LibSVM and logistic regression (denoted by LogitReg) classifier. The experimental results show that the *diff-absNJ-ratio* and *ref-diff-absNJ* feature sets outperform CC-PEV feature set regarding detection accuracy. Especially at the relative payload parameter of 0.1 bpac and 0.15 bpac, *diff-absNJ-ratio* and *ref-diff-absNJ* feature sets improve the detection accuracy by about 15~20%, either using SVM or using logistic regression classifier.

Table 1. Average detection accuracy (%) over 200 experiments at different hiding ratios (measured by relative payload, bpac), by applying SVM and logistic regression classifier to 548-dim *CC-PEV*, 72-dim *diff-absNJ-ratio*, and 144-dim *ref-diff-absNJ*.

bpac	CC-PEV		diff-absNJ-ratio		ref-diff-absNJ	
	SVM	LogitReg	SVM	LogitReg	SVM	LogitReg
0.1	57.7	58.0	76.8	76.7	**77.2**	74.6
0.15	67.7	70.0	88.5	88.3	**89.3**	85.5
0.2	76.9	79.6	94.2	92.8	**94.8**	91.9
0.25	84.8	88.3	97.4	96.9	**97.5**	97.0
0.3	88.9	92.5	**98.8**	98.3	98.7	98.3
0.35	94.2	96.0	**99.6**	99.2	99.5	99.1
0.4	96.9	98.0	**99.8**	99.4	99.7	99.3

Additionally, the mean and the standard deviation (STD) values of true negative rate (TNR) and true positive rate (TPR) over 200 experiments are given by Figure 4. While we consider the mean detection accuracy with the standard deviation together, in detection of the steganograms at low relative payload, either using SVM or logistic regression classifier, the detection accuracy with CC-PEV is not impressive, but the corresponding standard deviation are pretty high, which means that the detection performance by using CC-PEV is very unstable across different experiments. It also claims the undetectablility of DCT-embedding based adaptive steganography against CC-PEV feature set. In comparison to CC-PEV, our feature sets demonstrate the superiority, either in terms of detection accuracy or the detection stability across different experiments.



(a) Mean TNR with LibSVM (b) Mean TPR with LibSVM (c) Mean TNR with LogitReg (d) Mean TPR with LogitReg

(e) Std(TNR) with LibSVM (f) Std(TPR) with LibSVM (g) Std(TNR) with LogitReg (h) Std(TPR) with LogitReg

Figure 4. The mean and standard deviation of true negative rate (TNR) and true positive rate (TPR) by applying LibSVM and logistic regression to diff-absNJ-ratio, ref-diff-absNJ, and CC-PEV feature sets.

5.2 Steganalysis of YASS

5.2.1 Setup

Similarly, the original 1000 BOSSRank cover images downloaded from [4] are used for YASS embedding. We set $QF_h = QF_a = 75$ and $QF_h = QF_a = 50$ respectively. Accordingly, we encode the 1000 BOSSRank cover images in JPEG format at the quality factor of 75 and 50 respectively, which are used as JPEG covers. In creation of YASS steganograms, QF_h and QF_a are set to the same quantization factor in order to avoid double JPEG compression, because the YASS steganograms could be detected by using the detection method to expose double JPEG compression.

In our experiments, the embedding parameter T of B-block size is set from 9 to 15. To conduct a comparative study, we extract the *diff_absNJ* features defined in (16), and the zero-valued probability features presented by Li et al. [23]. A linear LibSVM and logistic regression classifier are used for classification (similar to the steganalysis of DCT-embedding based adaptive steganography, we compared LibSVM and SVM_light with linear, polynomial and RBF kernels and finally selected LibSVM with linear kernel in these experiments). In each experiment, 50% samples are randomly selected for training, and the other 50% samples are used for testing; 200 experiments are operated for each feature set at each B-block size by using each learning classifier.

5.2.2 Experimental Results

The mean value and standard deviation of the detection accuracy on testing feature sets over 200 experiments are listed in Tables 2 and 3. The detection accuracy on testing set is calculated by the half of the sum of true positive rate and true negative rate, or 0.5*TP/(TP+FN)+0.5*TN/(TN+FN).

Experimental results show that the detection method presented in [23] delivers good performance in detection of the YASS steganograms that are produced with small B-block size (e.g., T = 9, the detection accuracy is over 99%). However, the detection performance apparently deteriorates while the parameter of B-block size increases (e.g., T = 15, either using SVM or logistic regression classifier, the detection accuracy is less than 70%). As a comparison, our method performs well either in small or large parameter of B-block size, using SVM or logistic regression classifier. While T=9, 10, and 11 in the case $QF_h = QF_a = 75$, and T=9, 10, 11, 12, and 13, in the case $QF_h = QF_a = 50$, all detection accuracy values obtained by our approach are over 99%; while T= 15, our approach hits the detection accuracy over 90% for $QF_h = QF_a = 75$, and 95% for $QF_h = QF_a = 50$, either using SVM or logistic regression classifier.

The standard deviation values by using the *diff_absNJ* feature set are smaller than the values obtained by compared method, which implies that our method is more stable than the compared method. In summary, the experimental results shown by tables 2 and 3 indicate that our method is more effective and reliable compared to the prior detection art based on zero-valued DCT density feature set.

Table 2. The average detection accuracy ± standard deviation (%) over 200 experiments with *diff_absNJ* and zero-valued DCT density feature set [23], using LibSVM and LogitReg for the detection of YASS steganograms ($QF_h = QF_a = 75$)

T	Diff_absNJ		Zero-valued DCT density	
	SVM	LogitReg	SVM	LogitReg
9	**99.9±0.1**	**99.9±0.1**	99.8±0.3	**99.9**±0.6
10	**99.8±0.1**	**99.8±0.1**	99.0±0.4	99.1±0.5
11	99.0±**0.3**	**99.2**±0.4	93.6±2.3	97.5±0.6
12	98.2±0.4	**98.4±0.3**	74.3±3.9	94.3±0.7
13	96.7±0.6	**97.0±0.4**	61.7±2.1	86.4±1.0
14	94.7±**0.6**	**95.1±0.6**	53.2±4.0	76.8±1.0
15	90.8±0.8	**91.0±0.7**	48.2±1.5	69.8±1.2

Table 3. The average detection accuracy ± standard deviation (%) over 200 experiments with *diff_absNJ* and zero-valued DCT density feature set [23], using LibSVM and LogitReg for the detection of YASS steganograms ($QF_h = QF_a = 50$)

T	Diff_absNJ		Zero-valued DCT density	
	SVM	LogitReg	SVM	LogitReg
9	99.7±**0.2**	99.6±0.3	99.8±0.3	**99.9**±0.5
10	99.8±0.2	**99.9±0.1**	99.3±0.4	99.0±0.7
11	99.7±**0.2**	**99.8±0.2**	99.3±0.5	97.6±0.5
12	**99.5±0.2**	99.3±0.3	92.6±1.3	94.3±0.9
13	99.1±**0.3**	**99.3**±0.4	88.2±3.4	86.3±0.9
14	**97.9±0.4**	**97.9±0.4**	73.0±3.9	76.9±1.0
15	**95.0**±0.8	**95.0±0.5**	62.2±2.4	69.6±1.1

5.3 Discussion

In steganalysis of DCT-embedding based adaptive steganography, to obtain the calibrated neighboring joint density features, the JPEG image under scrutiny is cropped 63 times individually with the shifting from (0, 1) to (7,7), the neighboring joint density features are extracted from these 63 cropped versions, and the mean values of the features are used as calibrated features. Compared to the calibration that only takes once-cropping (e.g., only shifting by 4 rows and 4 columns), the computation cost is relatively high. However, the calibrated neighboring joint density obtained by 63-cropping is generally closer to the neighboring joint density of original cover, shown by Figure 5, wherein the mean absolute values of the difference of the neighboring joint density between 1000 covers and the calibrated versions are given. Figure 6 plots the mean values of the relative difference on the 1000 covers and the calibrated versions. Relative difference is calculated by $\left|absNJ^c(x,y) - absNJ(x,y)\right| / absNJ(x,y)$ wherein $absNJ(x,y)$ and $absNJ^c(x,y)$ stand for the neighboring joint density from un-calibrated image and from the calibrated version respectively. Because DCT-embedding based adaptive steganography has been well designed to remain original statistical property through Syndrome-Trellis Codes and minimize the distortion cost, the differences of the features from a cover and from the steganogram are very small; in such case, if the calibrated features are closer to those from original cover, it is better to improve the detection accuracy. Our experiments show that if we only take once-cropping (cropped by 4 rows and 4 columns) to obtain calibrated features, compared to the detection by 63-times-cropping, the detection accuracy decreases by about 6% in detecting the steganograms at relative payload 0.1 bpac.

It is worth noting that 63-times-cropping is not only useful to produce calibrated features, but also very effective to expose misaligned-cropping and recompression-based forgery in JPEG images. We have designed shift-recompression-based forgery detection, which is very promising to reveal the relevant forgery manipulations in JPEG images [31].

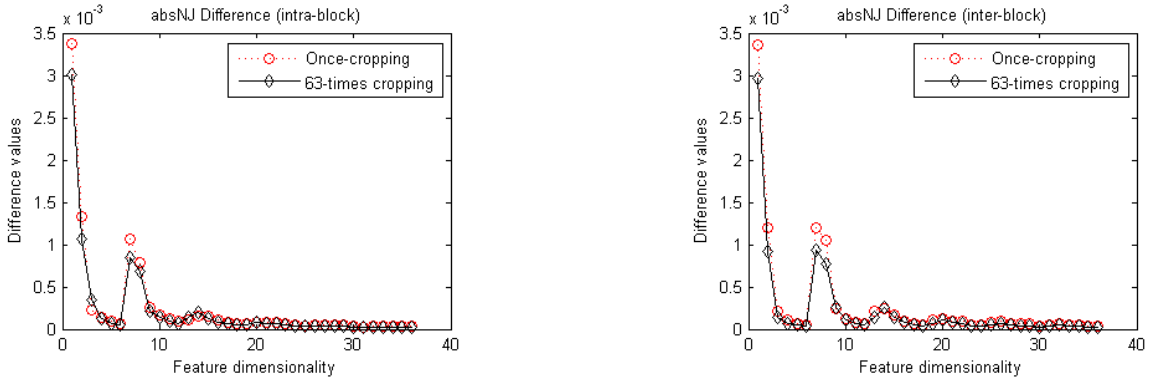


Figure 5. A comparison of the difference of neighboring joint density between once-cropping and 63-times-cropping.

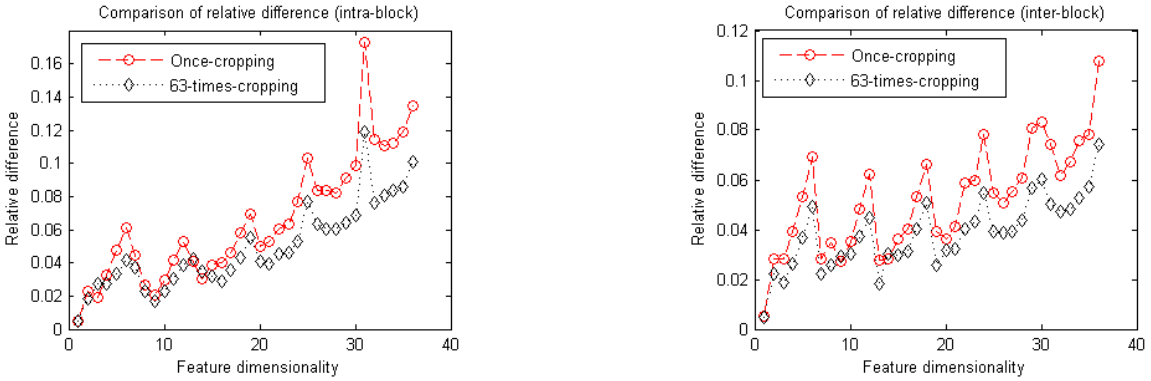


Figure 6. A comparison of the relative difference of neighboring joint density between once-cropping and 63-times-cropping.

In steganalysis of YASS, although Li et al indicated that the weakness of the YASS steganographic system, their detection algorithm does not search all candidate host blocks that are possibly used for information hiding, and the detection performance is not so well while the B-block size is large. By searching all possible candidate blocks and comparing the neighboring joint density of these candidate blocks and the non-candidate neighboring blocks, we have greatly improved the detection performance. We should mention that in YASS embedding, if the embedding positions of binary hidden bits are not limited into the 19 low-frequency AC DCT coefficients (e.g., the AC DCT coefficients are randomly selected for embedding), our approach is still effective for the detection, because our feature extraction is not limited to the position of 19 low-frequency AC coefficients.

In the original YASS embedding algorithm, the upper-left of the first B-block is overlapped with the upper-left of the first 8×8 block. If the first B-block randomly is misplaced from the upper-left point of the first 8×8 block, we can search all possibility of mismatching. There are 64 combinations including the original exact matching, accordingly we can retrieve the *diff_absNJ* features in each mismatching, in order to detect such polymorphism of YASS steganographic system.

In addition to SVM and logistic regression classifier [32], other learning classifiers, such as evolving neuro-fuzzy inference system [18] and ensemble classifier have been applied to steganalysis [12-14, 22, 24]. In terms of both detection accuracy and computation cost, logistic regression is one good option.

The experimental results also demonstrate that the detection accuracy under low compression quality $QF_h = QF_a = 50$ is generally higher than the detection accuracy under high compression quality $QF_h = QF_a = 75$. From our standpoint, the low compression quality factor takes large quantization steps during JPEG compression to obtain quantized DCT coefficients, and hence produces a smaller magnitude of quantized DCT coefficients. The chance of the modification to these small magnitude quantized DCT coefficients by YASS QIM embedding accordingly increases, and the amount of relative modification increases. As a result, the detection accuracy on the YASS steganograms that are produced at low quality is generally higher than the results on the high quality factor.

To design undetectable steganography in JPEG images, based on the relationship between image complexity and detection performance [24, 25], and our prior study of JPEG steganalysis [28], we have designed a JPEG-based statistically invisible steganography by ±1 embedding in the large magnitude of quantized DCT coefficients in the 8×8 blocks with complicated texture, which is simple and straightforward [30]. By combining this method with the methodology of adaptive steganography [8], we surmise that the steganographic system of being highly undetectable can be designed in DCT domain.

6. CONCLUSIONS

In this paper, we propose an improved approach based on neighboring joint density to detect a well-designed adaptive steganography in DCT domain, which has greatly improved earlier DCT-embedding arts. We also propose a new approach to steganalysis of YASS, by comparing the neighboring joint density of all candidate host blocks that are possibly used for data embedding and the non-candidate neighboring blocks. Support vector machine and logistic regression classifiers are employed for classification

Experiments show that, in steganalysis of DCT-embedding based adaptive steganography, our approach has gained considerable good detection performance compared to a previous state-of-the art JPEG steganalysis; the advantage of our approach is especially noticeable when detecting the steganograms at low payload embedding. In steganalysis of YASS, our method has significantly improved a previous detection method, especially in the detection of YASS steganograms that are produced by adopting a large B-block size, which was not well addressed before.

7. ACKNOWLEDGMENTS

This project was supported by Award No. 2010-DN-BX-K223 awarded by the National Institute of Justice, Office of Justice Programs, U. S. Department of Justice. The opinions, findings, and conclusions or recommendations expressed in this publication/program/exhibition are those of the authors and do not necessarily reflect those of the Department of Justice. Part of the support for this study from a 2011 Sam Houston State University (SHSU) Research Enhancement grant is also greatly appreciated.

We are grateful to Dr. Bin Li for providing us their feature extraction code and to anonymous reviewers for their insightful comments and suggestions to improve our work. In addition to BOSSRank images [4], the hiding tool of DCT-embedding based adaptive steganography [5], the code to extract CC-PEV features [47], MATLABArsenal [48] and PRtools [49] are used in this study. We are truly grateful to these authors and providers. Special thanks go to Mrs. Sharla Miles and Mrs. Delia Gallinaro at SHSU for their proofreading.

8. REFERENCES

[1] http://www.msnbc.msn.com/id/38028696/ns/technology_and_science-science/

[2] http://www.justice.gov/opa/documents/062810complaint1.pdf

[3] http://www.justice.gov/opa/documents/062810complaint2.pdf

[4] http://www.agents.cz/boss/BOSSFinal/index.php?mode=VIEW&tmpl=materials

[5] http://dde.binghamton.edu/download/stego_design/

[6] Chen C and Shi Y (2008). JPEG image steganalysis utilizing both intrablock and interblock correlations. *Proc. 2008 IEEE International Symposium on Circuits and Systems*, pp. 3029–3032.

[7] Filler T and Fridrich J (2010). Gibbs construction in steganography. *IEEE Trans. on Info. Forensics and Security*, 5(4): 705-720.

[8] Filler T and Fridrich J (2011). Design of adaptive steganographic schemes for digital images. *Proc. SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics XIII*, San Francisco, CA, January 23-26, 2011.

[9] Filler T, Judas J and Fridrich J (2011). Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Trans. on Info. Forensics and Security*, to appear.

[10] Fridrich J, Goljan M and Hogeam D (2002). Steganalysis of JPEG images: breaking the F5 algorithm. *Proc. of 5th Information Hiding Workshop*, pp. 310-323.

[11] Fridrich J (2004). Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. *LNCS,* 3200, pp.67-81.

[12] Fridrich J, Kodovsky J, Holub V and Goljan M (2011). Breaking HUGO – the process discovery. In *Proc. 13th Information Hiding Workshop*, Prague, Czech Republic, May 18–20, 2011.

[13] Fridrich J, Kodovsky J, Holub V and Goljan M (2011). Steganalysis of content-adaptive steganography in spatial domain. In *Proc. 13th Information Hiding Workshop*, Prague, Czech Republic, May 18–20, 2011.

[14] Gul G and Kurugollu F (2011). A new methodology in steganalysis: breaking highly undetectable steganography (HUGO). In *Proc. 13th Information Hiding Workshop*, Prague, Czech Republic, May 18–20, 2011.

[15] Hetzl S and Mutzel P (2005). A graph-theoretic approach to steganography. *LNCS*, 3677, pp. 119–128.

[16] Hilbe, JM (2009). *Logistic Regression Models*. Chapman & Hall/CRC Press. ISBN 978-1-4200-7575-5.

[17] Joachims T (2000). Estimating the generalization performance of a SVM efficiently. *Proc. of the International Conference on Machine Learning,* Morgan Kaufman, 2000. *SVM_light* is available at http://svmlight.joachims.org/

[18] Kasabov NK and Song Q (2002). DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 10(2): 144-154.

[19] Ker A (2005). Improved detection of LSB steganography in grayscale images. *LNCS,* 3200, pp. 97–115.

[20] Kodovsky J and Fridrich J (2009). Calibration revisited. *Proceedings of the 11th ACM Multimedia and Security Workshop*, Princeton, NJ, September 7-8, 2009.

[21] Kodovsky J, Pevny T and Fridrich J (2010). Modern steganalysis can detect YASS, *Proc. SPIE, Electronic Imaging, Media Forensics and Security XII*, San Jose, CA, January 17–21, pp. 02-01 - 02-11, 2010.

[22] Kodovsky J and Fridrich J (2011). Steganalysis in high dimensions: fusing classifiers built on random subspaces. *Proc. SPIE* 7880, 78800L (2011); doi:10.1117/12.872279

[23] Li B, Shi Y and Huang J (2009). Steganalysis of YASS. *IEEE Trans. Information Forensics and Security,* 4(3):369-382.

[24] Liu Q, Sung AH, Chen H and Xu J (2008). Feature mining and pattern classification for steganalysis of LSB matching steganography in grayscale images. *Pattern Recognition,* 41(1): 56-66.

[25] Liu Q, Sung AH, Ribeiro BM, Wei M, Chen Z and Xu J (2008). Image complexity and feature mining for steganalysis of least significant bit matching steganography. *Information Sciences,* 178(1): 21-36.

[26] Liu Q, Sung AH and Qiao M (2009). Improved detection and evaluation for JPEG steganalysis. *Proc. 17th ACM Multimedia*, pp. 873-876.

[27] Liu Q and Sung AH (2009). A new approach for JPEG resize and image splicing detection. *Proc. ACM Multimedia Workshop on Multimedia in Forensics* 2009, pp. 43-47.

[28] Liu Q, Sung AH and Qiao M (2011). Neighboring joint density-based JPEG steganalysis. *ACM Trans. on Intelligent Systems and Technology*, 2(2): article 16.

[29] Liu Q, Sung AH and Qiao M (2011). A method to detect JPEG-based double compression. In *Proc. of 8th International Symposium on Neural Networks* (2), pages 466-476.

[30] Liu Q, Sung AH, Chen Z and Huang X (2011). A JPEG-based statistically invisible steganography. In *Proc of 3rd ACM International Conference on Internet Multimedia Computing and Service*, August 5-7, 2011, Chengdu, China.

[31] Liu Q. Detection of misaligned cropping and recompression with the same quantization matrix and relevant forgery, *ACM Workshop on Multimedia in Forensics and Intelligence* 2011. Under review.

[32] Lubenko I and Ker AD (2011). Steganalysis using logistic regression. In *Proc. SPIE 7880*, 78800K (2011); doi:10.1117/12.872245

[33] Lyu S and Farid H (2005). How realistic is photorealistic. *IEEE Trans. on Signal Processing*, 53(2): 845-850.

[34] Marvel L, Boncelet C and Retter C (1999). Spread spectrum image steganography. *IEEE Trans. Image Processing*, 8(8): 1075-1083.

[35] Mielikainen J (2006). LSB matching revisited. *IEEE Signal Processing Letters* 13(5):285-287.

[36] Pevny T and Fridrich J (2007). Merging Markov and DCT features for multi-class JPEG steganalysis. *Proc. SPIE*, Vol. 6505, 650503, 2007; DOI:10.1117/12.696774.

[37] Pevny T, Filler T and Bas P (2010). Using high-dimensional image models to perform highly undetectable steganography, *Proc. 12th Information Hiding*, Calgary, Alberta, Canada, June 28-30, 2010.

[38] Provos N (2001). Defending against statistical steganalysis. *Proc. 10th USENIX Security Symposium*, vol.10, pp. 323-335.

[39] Sachnev V, Kim HJ and Zhang R (2009). Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome code. In: *Proceedings of the 11th ACM Multimedia & Security Workshop*. Pp. 131-140.

[40] Sallee P (2004). Model based steganography. *LNCS*, 2939, pp. 154–167.

[41] Sharp T (2001). An implementation of key-based digital signal steganography. *LNCS*, 2137, pp.13–26, 2001.

[42] Shi Y, Chen C and Chen W (2007). A Markov process based approach to effective attacking JPEG steganography. *LNCS,* 4437, pp. 249-264.

[43] Solanki K, Sarkar A and Manjunath B (2007). YASS: Yet another steganographic scheme that resists blind steganalysis. *LNCS,* 4567, pp. 16-31.

[44] Vapnik V (1998), *Statistical Learning Theory*, John Wiley.

[45] Westfeld A (2001). High capacity despite better steganalysis (F5 – a steganographic algorithm). *LNCS* 2137, pp. 289–302.

[46] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[47] http://dde.binghamton.edu/download/ccmerged/

[48] http://www.informedia.cs.cmu.edu/yanrong/MATLABArsenal/MATLABArsenal.htm

[49] http://www.prtools.org