Distortion Function Designing for JPEG Steganography with Uncompressed Side-image

Fangjun Huang
School of Information Science and Technology,
Sun Yat-Sen University, GD 510006, China
huangfj@mail.sysu.edu.cn

Weiqi Luo
School of Software,
Sun Yat-Sen University, GD 510006, China
weiqi.luo@yahoo.com

Jiwu Huang
School of Information Science and Technology,
Sun Yat-Sen University, GD 510006, China
isshjw@mail.sysu.edu.cn

Yun-Qing Shi
Department of Electrical and Computer Engineering,
New Jersey Institute of Technology, NJ 07102, USA
shi@njit.edu

ABSTRACT
In this paper, we present a new framework for designing distortion functions of joint photographic experts group (JPEG) steganography with uncompressed side-image. In our framework, the discrete cosine transform (DCT) coefficients, including all direct current (DC) coefficients and alternating current (AC) coefficients, are divided into two groups: first-priority group (FPG) and second-priority group (SPG). Different strategies are established to associate the distortion values to the coefficients in FPG and SPG, respectively. In this paper, three scenarios for dividing the coefficients into FPG and SPG are exemplified, which can be utilized to form a series of new distortion functions. Experimental results demonstrate that while applying these generated distortion functions to JPEG steganography, the intrinsic statistical characteristics of the carrier image will be preserved better than the prior-art, and consequently the security performance of the corresponding JPEG steganography can be improved significantly.

Categories and Subject Descriptors
I.4 [**Image Processing and computer vision**]

Keywords
JPEG, steganography, steganalyzer, distortion function

1. INTRODUCTION
The key concept behind the security of steganographic systems is the statistical un-detectability. It may be influenced by many factors [1], such as the choice of cover object, the type of modification operation on cover elements, the number of embedding changes (related to the payload), and the distortion functions used to identify individual elements of cover that could be modified during embedding. Assume that the first three factors mentioned above are the same, designing the distortion function will be an important approach to minimizing the impact caused by embedding, and thus improve the security performance of steganography.

IH&MMSec'13, June 17–19, 2013, Montpellier, France.

To minimize the impact caused by data embedding, the sender should choose to modify those elements (pixels/coefficients) in such a way that the caused detectable distortion is as small as possible. Embedding the secret message bits under the guidance of minimizing distortion function can improve the security performance of steganography and has been known for a long time. In [2], Fridrich *et al*. presented the perturbed quantization (PQ) steganography. As a specific case, they pointed out that the sender can constrain the embedding changes to those DCT coefficients that experience the largest quantization error, i.e., the coefficients with the quantization error of $0.5\pm\varepsilon$ (ε is a small positive number). Such kind of coefficients, when rounded to the other value, may leave the smallest embedding distortion. In [3], another two adaptive versions of PQ, i.e., texture-adaptive PQ (PQt) and energy-adaptive PQ (PQe) have been presented. Through considering the local block content such as texture complexity and energy capacity, JPEG steganography with higher security performance can be obtained. In [4-6], the authors have combined quantization step with quantization error in their distortion function to improve the security performance of JPEG steganography. Besides the quantization step, Wang and Ni [7] presented a new JPEG distortion function with consideration of the block entropy, and the experimental results demonstrate that this new distortion function may lead to less detectability of steganalyzers. Recently, Huang *et al*. [8] presented another distortion function for JPEG steganography, which is called new PQ (NPQ). Three factors are considered, i.e., the quantization error, the quantization step and the magnitude of quantized DCT coefficients to be modified. Via nonlinearly combining these three different factors, the new distortion function, NPQ, can improve the security performance of JPEG steganography significantly as demonstrated in [8].

All the aforementioned distortion functions are employed to find the DCT coefficients that may result in less detectable distortion after modification. Generally, they are applied together with the utilization of matrix encoding (embedding) technology [9, 10]. For example, in [2] Fridrich *et al*. have exemplified how to implement PQ distortion function in JPEG steganography with the help of Wet paper codes [11, 12]. In [13], Kim *et al*. provided a simple and practical scheme to apply PQ distortion function with matrix encoding, which is based on modified binary Hamming codes [14]. This new matrix encoding strategy allows more than one embedding change in each coefficient block. Via a brute-force search, the modifications are made on those coefficients that may introduce minimal detectable distortion, and thus improving

the security performance of the corresponding JPEG steganography. According to the number of allowable changing bits in each coefficient block, these modified matrix encoding (MME) schemes are called MME2, MME3, *etc*. Similar approach can also be made based on BCH (Bose, Chaudhuri and Hocquenghem) codes [14] to improve the embedding efficiency of matrix encoding as described in [4, 6, 15, 16]. However, since the decoding of BCH codes is much more complicated than Hamming codes, some specific techniques need to be adopted by the sender to reduce the time complexity and storage complexity in the embedding process. In [5], Filler *et al*. provided the syndrome-trellis codes (STCs), which can be utilized for embedding while minimizing an arbitrary additive distortion function with a performance near the theoretical bound. This new methodology can directly improve the security performance of many existing steganographic schemes, allowing them to communicate larger payloads at the same embedding distortion or to decrease the distortion for a given payload.

In this paper, we present a new framework for designing distortion functions with uncompressed side-image, which can be applied to JPEG steganography using any of the above mentioned matrix encoding strategies. In our framework, the DCT coefficients, including the direct current (DC) coefficients and all the alternating current (AC) coefficients of JPEG image are divided into two groups: first-priority group (FPG) and second-priority group (SPG). Different strategies will be established to associate the distortion values to the coefficients in FPG and SPG, respectively. Generally, the coefficients that may result in less detectable distortion in the embedding process are grouped into FPG and the rest are into SPG. Note that in our framework, all DCT coefficients are utilized for matrix encoding, and we believe that any DCT coefficient can be modified in the embedding process. In an extreme case, we can change any single coefficient in a given JPEG image and the introduced distortion will not be perceived by today's most powerful JPEG steganalyzers. Thus in our framework, no coefficient is considered as un-changeable, and any DCT coefficient can be modified if needed in the embedding process. That is also the main difference between the distortion functions generated from our framework and those previously presented in the literature. In this paper, three different scenarios for dividing the coefficients into FPG and SPG are exemplified, which can be utilized to form a series of new distortion functions. Via applying these generated distortion functions in JPEG steganography with matrix encoding, the modifications will mainly be made on those coefficients that may result in less detectable distortion in the embedding process. Thus JPEG steganography with higher security performance can be obtained.

The rest of this paper is organized as follows. In Section II, the proposed new framework is introduced. Experimental results and analysis are illustrated in Section III, and the conclusion is drawn in Section IV.

2. PROPOSED FRAMEWORK

Suppose the raw, uncompressed side-image is available to the sender. The DCT coefficients that have been divided by quantization steps and not yet rounded are called un-rounded DCT coefficients, and those that have been divided by the quantization steps and rounded are called quantized DCT coefficients, respectively. To facilitate the explanation, the existing distortion functions such as PQ [2] and NPQ [8] are referred to as ordinary distortion functions, and those to be proposed below in this paper are referred to as advanced distortion functions. To make this paper self-contained, we will introduce the PQ and NPQ distortion functions firstly.

2.1 PQ and NPQ

Without loss of generality, the quantized coefficients and un-rounded DCT coefficients utilized for data hiding are represented by $C=(c_1,c_2,\ldots,c_N)$ and $C'=(c'_1,c'_2,\ldots,c'_N)$, respectively, where N represents the number of DCT coefficients in the quantized and un-rounded coefficient sequence. The relationship between $c_i(1\le i\le N)$ and $c'_i(1\le i\le N)$ is as follows.

$$c_i = round(c'_i) \tag{1}$$

where $round(x)$ is a function that rounds the element x to the nearest integer. Note that in Equation (1), c_i represents the quantized DCT coefficient that is obtained in JPEG compression without secret message embedding. Suppose that while embedding the secret message the modification needs to be made on c_i, and the coefficient after being modified is represented by s_i. The PQ distortion function is represented as follows.

$$d_{c_i}^{PQ} = \left\| c_i - c'_i \right| - \left| s_i - c'_i \right\| \tag{2}$$

where $|x|$ is a function that returns the absolute value of the corresponding element x. For any coefficient c_i, the PQ distortion value $d_{c_i}^{PQ}$ can be computed according to Equation (2). As pointed out in [2, 13], while embedding the secret message bits, the sender should select those coefficients with minimal PQ distortion values for modification.

NPQ can be regarded as an improved version of PQ considering the quantization step and the magnitude of the quantized DCT coefficients to be modified. Suppose the quantization step associated with the coefficient c_i is q_i. According to [8], the NPQ distortion function is represented as follows.

$$d_{c_i}^{NPQ} = d_{c_i}^{PQ} \times (q_i)^{\lambda_1} \Big/ (\mu + |c_i|)^{\lambda_2} \tag{3}$$

where λ_1 and λ_2 are two parameters that are used to control the impacts caused by q_i and $|c_i|$, respectively. As recommended in [8], the two control parameters λ_1 and λ_2 can be selected in the rage of (0, 1]. The parameter μ is utilized to avoid the zero divisors in Equation (3). When NPQ is only utilized to compute the distortion value corresponding to the non-zero DCT coefficients, μ is selected as 0. Otherwise, the parameter μ can be selected as a small number, e.g., the number of 1. For any coefficient c_i, the NPQ distortion value $d_{c_i}^{NPQ}$ can be computed according to Equation (3). As pointed out in [8], while embedding the secret message bits, the sender can select those coefficients

with minimal NPQ distortion values for modification to obtain JPEG steganography with high security performance.

2.2 The Proposed Distortion function

As mentioned above, in our new framework the DCT coefficients are divided into two groups: FPG and SPG. The coefficients in FPG and SPG are associated with distortion values calculated via using different strategies. Firstly, the impact caused by the modifications of coefficients in FPG and SPG are measured using some ordinary distortion functions. Secondly, those obtained distortion values associated with the coefficients in SPG are multiplied by a penalty factor, which is a big value. Thus the distortion values associated with the coefficients in FPG may be much less than that in SPG in general in our advanced distortion function. When conducting matrix encoding with some syndrome codes as in [4-8, 13, 16], several alternative solutions may be produced and those coefficients in FPG that may result in less distortion in the embedding process will take precedence for modification. Even if all the alternative solutions are restricted to those coefficients in SPG, the coefficients in SPG associated with smaller ordinary distortion values will still take precedence for modification. That is, the advanced distortion functions generated from our framework can orientate us to make as less distortion as possible in embedding the secret message bits, and thus the security performance of JPEG steganography will be improved. The proposed advanced distortion function is defined in Equation (4).

$$d_{c_i}^{ADV} = d_{c_i}^{ORD} \times (1+\rho) \tag{4}$$

In Equation (4), the $d_{c_i}^{ORD}$ represents the impact caused by modification operation on coefficient c_i, which is computed according to the ordinary (abbreviated as "*ORD*") distortion functions such as PQ, NPQ and some others. The penalty factor ρ is selected as a big value (e.g., 10^6) if the coefficient $c_i \in SPG$, otherwise it is selected as 0. According to Equation (4), for any coefficient c_i in the input image, the advanced (abbreviated as "*ADV*") distortion value $d_{c_i}^{ADV}$ can be easily computed.

Via applying the advanced distortion functions generated from our framework to JPEG steganography, no special processing needs to be made on those DCT coefficients with the values of +1 and -1 as that in [7, 8, 13]. Note that in [7, 8, 13], the distortion functions have only been applied on the non-zero AC DCT coefficients. If the coefficient with value of +1 or -1 is flipped to 0, the recipient will not be able to accurately locate the corresponding non-zero coefficients utilized for matrix encoding in the transmitting end, and the embedded secret message bits may not be extracted successfully. Thus special modification operation should be made by the sender on those coefficients with the quantized values of +1 and -1. For example, in [7, 8, 13] the coefficients with the quantized values of +1 and -1 can only be flipped to +2 and -2, respectively. Since the distortion functions generated from framework are applied on all the DCT coefficients, i.e., all the DCT coefficients are utilized for matrix encoding, no such special modification operation needs to be made while applying our advanced distortion functions to JPEG steganography. For any coefficient $c_i (1 \le i \le N)$ to be modified, the operation is conducted as follows.

$$s_i = \begin{cases} c_i + 1, & \text{if } (c_i - c'_i) \le 0 \\ c_i - 1, & \text{if } (c_i - c'_i) > 0 \end{cases} \tag{5}$$

where s_i is the coefficient after having been modified.

Furthermore, since all the DCT coefficients (including DC coefficients and numerous zero AC coefficients besides the non-zero AC coefficients) are included in our framework while applying those advanced distortion functions to JPEG steganography, the embedding efficiency (the number of bits embedded per embedding change [17]) of matrix encoding will be improved significantly. For example, if we select MME2 embedding strategy for matrix encoding, with the usage of $[2^k - 1, k] (k \ge 1)$ modified binary Hamming codes, k secret message bits can be embedded into $2^k - 1$ quantized DCT coefficients by changing at most two of them. That is, the larger the *k*, more efficiently the matrix encoding will be accomplished. According to [8, 9, 13], the parameter *k* of Hamming codes is determined by the number of secret message bits (represented by *n*) and the number of DCT coefficients (represented by *N*) utilized for matrix encoding. In general we will select the maximum *k* that qualifies the inequality $\frac{k}{2^k - 1} > \frac{n}{N}$. It is obviously that with embedding the same number of secret message bits, the algorithms utilizing more DCT coefficients for data hiding will result in a more efficient matrix encoding.

Note that in order to exchange the secret message bits successfully, both the sender and recipient should utilize the same coefficients to accomplish the matrix encoding. For example, in [7, 8, 13], the recipient needs to accurately locate the non-zero AC DCT that have been used in the embedding process to conduct the matrix encoding, otherwise the secret message bits cannot be extracted accurately. A special note of interest is that while applying those advanced distortion functions generated from our framework to JPEG steganography, the sender should first divide all the DCT coefficients into FPG and SPG. However, the sender does not need to share the dividing scenario with the recipient, since they (i.e., the sender and recipient) both use all the DCT coefficients to conduct matrix encoding. The recipient does not need to locate the DCT coefficients in FPG or SFG in the receiving end, and he/she can exchange the secret message with the sender easily via selecting the same matrix encoding strategy.

2.3 Three Different Scenarios for Dividing FPG and SPG

The statistics of DCT coefficients are complicated and they may interact with each other while being modified. Moreover the statistics of DCT coefficients may also have a close relationship with the secret message bits to be embedded, and the type of embedding operation that modifies the coefficients, *etc*. It is not easy to derive an optimal strategy for dividing the coefficients into FPG and SPG. However, a series of suboptimal scenarios can be found easily.

In the following, three scenarios for dividing the coefficients into FPG and SPG are exemplified. Scenario 1 is a simple and direct way. Its separation performance may not be as good as the following two scenarios. However, our experiments in next section will demonstrate that with appropriate selection of the ordinary distortion function, high secure performance can still be obtained. The fundamental idea of the next two scenarios is that in the texture area of a carrier image more coefficients will be divided into FPG, and in the flat area fewer coefficients will be divided into FPG. In Scenario 2, the first-priority and second-priority coefficients are classified according to the statistics of coefficients in DCT domain. In Scenario 3, the coefficients are divided into FPG and SPG with resorting to the statistics of JPEG image in spatial domain.

Scenario 1: The AC DCT coefficients are considered as first-priority coefficients, and the DC DCT coefficients are classified as the second-priority coefficients.

Scenario 2: Compute the standard deviation $D_i (1 \le i \le N)$ of the quantized AC DCT coefficients in each 8×8 block of JPEG image, where *N* represents the total number of 8×8 blocks in JPEG image. The average value of all the standard deviations is $\overline{D} = \frac{1}{N}\sum_{i=1}^{N} D_i$, and the maximum value among all the standard deviations is $D_{\max} = \max(D_1, D_2, \ldots, D_N)$. In each block, the number of AC DCT coefficients that belongs to FPG is computed as follows.

$$A_i = \begin{cases} 1, & \text{if } D_i < \frac{1}{32}\overline{D} \\ \left\lfloor \frac{1}{2} \times 64 \times (D_i / \overline{D}) \right\rfloor, & \text{if } \frac{1}{32}\overline{D} \le D_i < \overline{D} \\ \left\lfloor \frac{1}{2} \times 64 \times (1 + D_i / D_{\max}) \right\rfloor, & \text{if } \overline{D} \le D_i < D_{\max} \\ 63, & \text{if } \overline{D} = D_{\max} \end{cases} \quad (6)$$

where $A_i (1 \le i \le N)$ represents the number of AC DCT coefficients that should be divided in FPG in each 8×8 block, and $\lfloor x \rfloor$ is a function that rounds the element *x* to its nearest integer less than or equal to *x*.

In Equation (6), the number 64 represents that there are 64 DCT coefficients in each 8×8 block. As we pointed above, any DCT coefficient can be modified, and we can change any single coefficient in a given JPEG image in the embedding process. Other methods for dividing the coefficients into FPG and SPG may still work, e.g., we can change the number 32 to 31 or 30 in Equation (6), and the obtained distortion function may still result in JPEG steganography with high security performance. Here, we only try to illustrate the applicability of our framework and do not try to make a clear boundary between FPG and SPG.

Scenario 3: Compute the standard deviation $P_i (1 \le i \le N)$ of pixel values in each 8×8 block of the decompressed JPEG image, where *N* represents the total number of 8×8 blocks in JPEG image. The average value of all the standard deviations is $\overline{P} = \frac{1}{N}\sum_{i=1}^{N} P_i$, and the maximum value among all the standard deviations is $P_{\max} = \max(P_1, P_2, \ldots, P_N)$. In each block, the number of AC DCT coefficients that belongs to FPG is computed as follows.

$$B_i = \begin{cases} 1, & \text{if } P_i < \frac{1}{32}\overline{P} \\ \left\lfloor \frac{1}{2} \times 64 \times (P_i / \overline{P}) \right\rfloor, & \text{if } \frac{1}{32}\overline{P} \le P_i < \overline{P} \\ \left\lfloor \frac{1}{2} \times 64 \times (1 + P_i / P_{\max}) \right\rfloor, & \text{if } \overline{P} \le P_i < P_{\max} \\ 63, & \text{if } \overline{P} = P_{\max} \end{cases} \quad (7)$$

where $B_i (1 \le i \le N)$ represents the number of AC DCT coefficients that should be divided in FPG in each 8×8 block.

As seen, the philosophy for choosing the first-priority coefficients in Equation (7) is similar to that in Equation (6). Differently, in Scenario 3 the DCT coefficients are divided into FPG and SPG according to the statistical characteristics of JPEG image in spatial domain. Since decompressing the JPEG image is a nonlinear process (including de-quantization and rounding process [18]), the selected first-priority coefficients in Scenario 3 will be different from that in Scenario 2. In Scenario 3, the $B_i (1 \le i \le N)$ first-priority coefficients in each block are also selected according to the *zig-zag* scanning order, and the rest AC and DC coefficients are considered as second-priority coefficients.

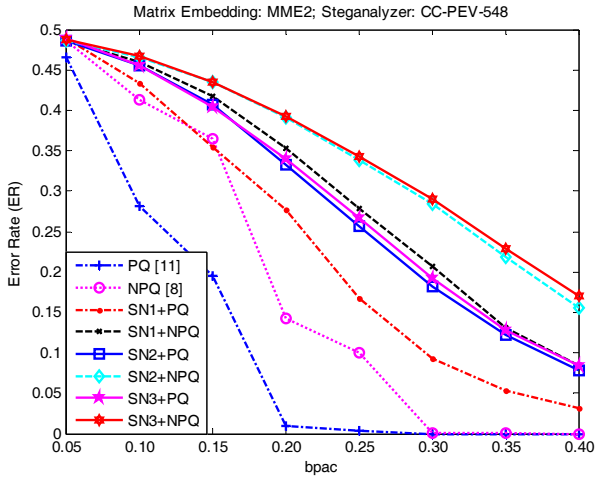
In this section, we have introduced two ordinary distortion functions and exemplified three different dividing scenarios. According to Equation (4), six different advanced distortion functions can be generated via combining those different ordinary distortion functions and dividing scenarios. Note that the framework proposed in this paper is an open system. Firstly, PQ, NPQ and any other ordinary distortion function can be used on our framework. For instance, in Equation (4) the superscript "*ORD*" stands for the ordinary distortion function. Secondly, more advanced scenarios for splitting the DCT coefficients into FPG and SPG can also be used in our framework. For example, if the more advanced scenario is found, the FPG and SPG in Equation (4) may be updated accordingly.

3. EXPERIMENTAL RESULTS

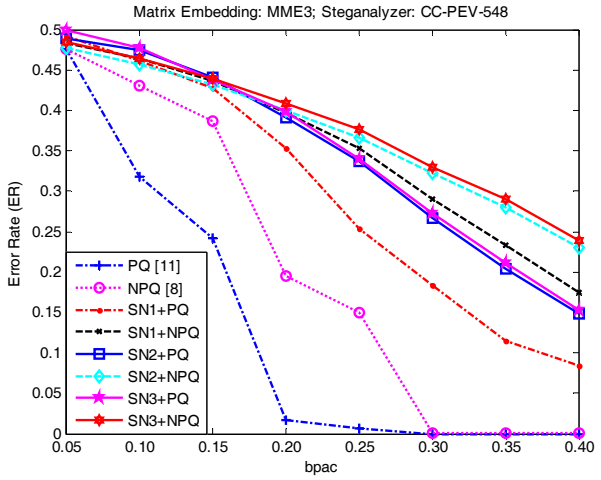
In this section, experimental results and analysis are presented to demonstrate the efficiency of our proposed framework. The test image set consists of 10000 uncompressed images which are downloaded from the BOSSBase image dataset [19]. All the images are with the size of 512×512. In the following, the uncompressed image is called input image and the JPEG compressed image without any message embedding is called cover image. The cover and stego images are created using the same JPEG encoder [20], and the quality factor is selected as 75 in all of our testing. The secret message bits are randomly

generated, and the embedding rates are represented in terms of *bpac* (bits per non-zero quantized AC DCT coefficients) values.

The efficiency of our proposed distortion functions is tested with three state-of-the-art feature-based steganalyzers, which are called CC-PEV (Cartesian-calibrated Pevný) [21], SPAM (subtractive pixel adjacency matrix) [22] and CDF (cross domain feature) [23], respectively. The 548-dimensional CC-PEV feature vector is mainly extracted from JPEG domain, which is extended to twice its size by Cartesian calibration from the 274 feature vector designed for JPEG images [24]. The 686-dimensional SPAM feature vector is extracted from spatial domain, which is the second-order Markov model of pixel differences. Through combing the CC-PEV and SPAM feature vector, we can get the 1,234-dimensional CDF feature vector. Those feature vectors or their improved versions are popularly utilized [25-28] in detecting the classical algorithms such as F5 [5] and MB1 [29], and some modern steganographic schemes [30-33]. Since the CDF feature vector is extracted in cross domain, it may have better detection performance than CC-PEV and SPAM in general.



(a)



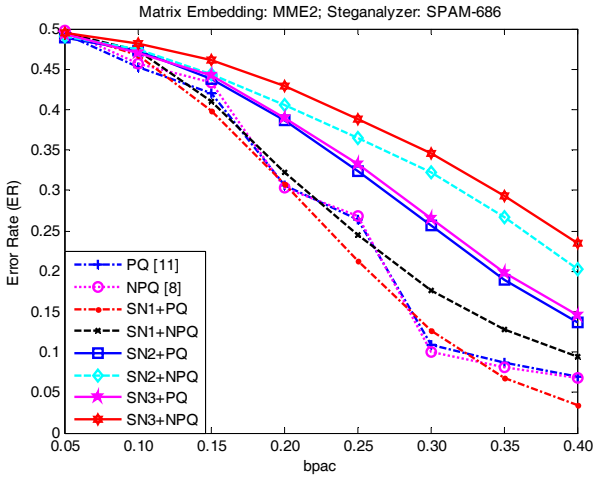
(b)

Fig. 1 The detection error rates with the steganalyzer CC-PEV-548. (a) MME2 embedding strategy. (b) MME3 embedding strategy.

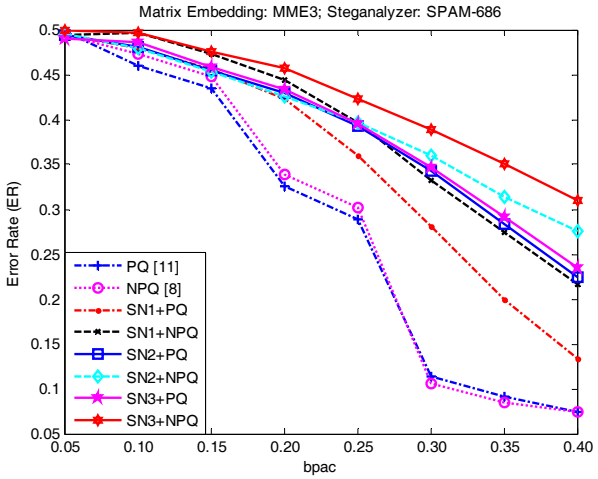
The ensemble classifier presented in [34] is employed in our testing with default parameters. It is a fully automatic framework with an efficient utilization of out-of-bag (OOB) error estimates for stopping criterion. As pointed out in [25], the proposed ensemble classifier consists of a lot of base learners independently trained on a set of cover and stego images. The decision threshold of each base learner is adjusted to minimize the total detection error under equal priors on the training set:

$$P_E = \min_{P_{FA}} \frac{1}{2}(P_{FA} + P_{MD}(P_{FA})) \tag{8}$$

where P_{FA}, P_{MD} are the probabilities of false alarms and missed detection, respectively.



(a)



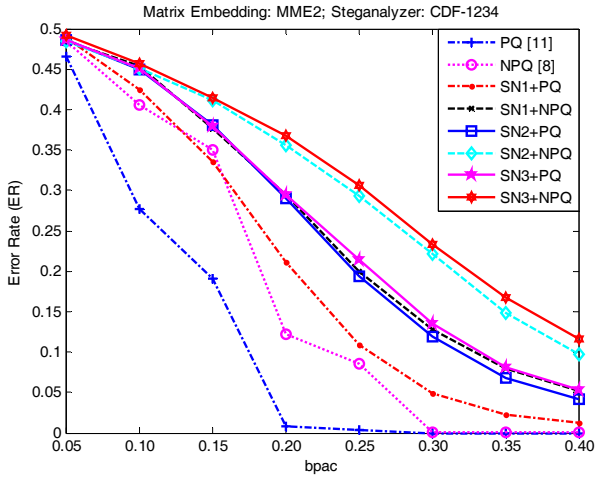
(b)

Fig. 2 The detection error rates with the steganalyzer SPAM-686. (a) MME2 embedding strategy. (b) MME3 embedding strategy.

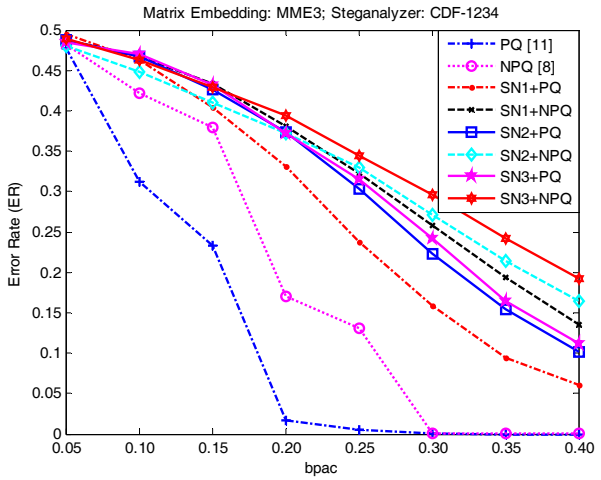
Firstly, we have applied the advanced distortion functions to JPEG steganography with MME2 and MME3 embedding strategies for a detail comparison. The PQ and NPQ are selected as the ordinary distortion functions for a demonstration. In Section 2.3, we have exemplified three scenarios for dividing the DCT coefficients into FPG and SPG. The algorithms resulted

from different dividing scenarios (abbreviated as "SN") and ordinary distortion functions are represented as "SN1+PQ", "SN1+NPQ", "SN2+PQ" "SN2+NPQ", "SN3+PQ", "SN3+NPQ", respectively. The original PQ and NPQ distortion functions are applied with MME2 and MME3 embedding strategies as in [8, 13]. Totally there are sixteen different steganographic schemes. Half of them are conducted with MME2 embedding strategy, and the other half are conducted with MME3 embedding strategy. Note that in all our testing, the two control parameters of NPQ are selected as $(\lambda_1 = 0.5, \lambda_2 = 0.2)$.

The detection error rates (ERs) corresponding to the three steganalyzers CC-PEV, SPAM, and CDF are illustrated in Figs. 1, 2 and 3, respectively. In these three figures, the horizontal axes represent the *bpac* values, and the vertical axes represent the detection error rates. For the aforementioned sixteen steganographic schemes, the embedding rates are increased from 0.05 *bpac* to 0.40 *bpac* with the step size of 0.05. The embedding strategy and the steganalyzer with the dimension size of feature vector are illustrated in the title of each figure.



(a)



(b)

Fig. 3 The detection error rates with the steganalyzer CDF-1234. (a) MME2 embedding strategy. (b) MME3 embedding strategy.

It is observed from Figs. 1, 2 and 3 that whichever dividing scenario or ordinary distortion function is selected, the security performance of the obtained JPEG steganography may be greatly improved under the guidance of those distortion functions generated from our framework. Our experimental results also demonstrate that the selection of dividing scenarios and ordinary distortion functions may have great importance in our framework. Different dividing scenarios and ordinary distortion functions may result in JPEG steganography with different security performance. As seen, with using the same ordinary distortion function, JPEG steganography resulted from Scenario 2 and Scenario 3 may have higher security performance than that from Scenario 1. With adopting the same dividing scenario, PQ and NPQ may result in JPEG steganography with different security performance too. Fortunately, via using our proposed framework, the different dividing scenarios and ordinary distortion functions can easily be combined to form efficient advanced distortion functions, even though the dividing scenarios and ordinary distortion functions are not optimal.

Secondly, in order to demonstrate the universality of our proposed framework, we have applied those generated distortion functions to JPEG steganography with STC. The Wang and Ni's method [7] has also been conducted for a comparison, which is one of the most secure JPEG steganographic schemes with using STC. The steganalyzer CDF is selected for testing, and the experimental results are shown in Fig. 4. The horizontal axes represent the *bpac* values, the vertical axes represent the detection error rates, and the distortion functions are shown on the legend. As seen, when the embedding rate is no more than 0.25 *bpac*, all the steganographic schemes resulted from our proposed and Wang and Ni's method have the similar security performance, and the obtained detection accuracy rates are near random guessing. However, with the increasing of embedding rate, the JPEG steganographic schemes resulted from our proposed distortion function may have much better security performance than that resulted from Wang and Ni's method. Via comparing Fig. 4 and Fig. 3, we can also find out that via using more efficient embedding strategy, the security performance of JPEG steganography resulted from our advanced distortion function can be improved further.

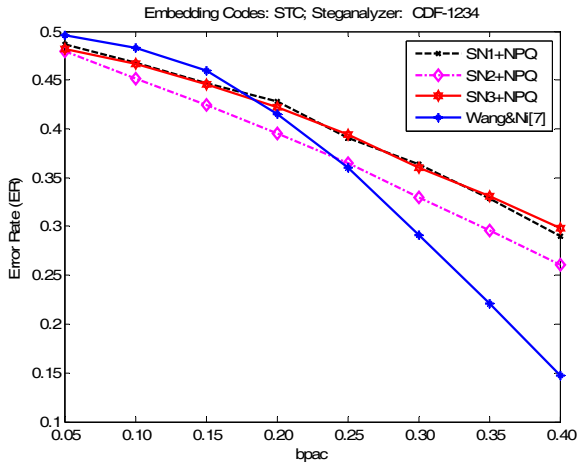


Fig. 4 The efficiency of our proposed framework with using STC.

4. CONCLUSIONS

In this paper, we have presented a new framework for designing distortion functions of JPEG steganography with uncompressed side-image, and a series of advanced distortion functions that may result in high secure JPEG steganography are exemplified. Note that our proposed framework is an open system. It will not be constrained to the aforementioned dividing scenarios and ordinary distortion functions. Other dividing scenarios and ordinary distortion functions can be adopted easily in our framework to form a series of new distortion functions.

5. ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (61173147, U1135001), the 973 Program of China (2011CB302204), the Key Projects in the National Science & Technology Pillar Program (2012BAK16B06), the Fundamental Research Funds for Central Universities (12lgpy31), and the Project Sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry ([2012]1707).

6. REFERENCES

[1] J. Fridrich, P. Lisoněk and D. Soukal, "On Steganographic embedding efficiency," in *Proc. Information Hiding Workshop* 2006, *LNCS* 4437, pp. 282-296, 2007

[2] J. Fridrich, M. Goljan and D. Soukal, "Perturbed quantization steganography with wet paper codes," in *Proc. the ACM Workshop on Multimedia & Security*, Magdeburg, Germany, September 20-21, pp. 4-15, 2004.

[3] J. Fridrich, T. Pevný and J. Kodovský, "Statistically undetectable JPEG steganography: dead ends, challenges, and opportunities," in *Proc. the ACM Workshop on Multimedia and Security*, Dallas, Texas, September 20-21, pp. 3-14, 2007.

[4] V. Sachnev, H. J. Kim, and R. Zhang, "Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding," in *Proc. the ACM Workshop on Multimedia & Security*, Princeton, New Jersey, Sep. 7-9, pp. 131–140, 2009.

[5] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using Syndrome-Trellis Codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920-935, 2010.

[6] V. Sachnev, H. J. Kim, "Modified BCH data hiding scheme for JPEG steganography," *Eurasip Journal on advances in signal processing*, vol. 2012, no. 1, pp. 89-98, 2012

[7] C. Wang, J. Ni, "An efficient JPEG steganographic scheme based on block-entropy of DCT coefficients," in *Proc. of IEEE ICASSP*, Kyoto, Japan, Mar. 25-30, pp. 1785-1788, 2012

[8] F. Huang, J. Huang, and Y. Q. Shi, "New channel selection rule for JPEG steganography," *IEEE Trans. Information Forensics and Security*, vol. 7, no. 4, pp. 1181-1191, 2012.

[9] R. Crandall, "Some notes on steganography", Posted on Steganography Mailing List, 1998. http://os.inf.tu-dresden.de/~westfeld/crandall.pdf

[10] A. Westfeld, "High capacity despite better steganalysis (F5-a steganographic algorithm)", in *Proc. Information Hiding, 4th International Workshop*, volume 2137 of *Lecture Notes in Computer Science*, pp. 289-302, 2001.

[11] J. Fridrich, M. Goljan, P. Lisoněk, and D. Soukal, "Writing on wet paper," *IEEE Trans. Signal Processing*, vol. 53, no. 10, pp. 3923-3935, 2005.

[12] J. Fridrich, M. Goljan and D. Soukal, "Wet paper codes with improved embedding efficiency," *IEEE Trans. Information Forensics and Security,* vol. 1, no. 1, pp. 102-110, 2006.

[13] Y. Kim, Z. Duric and D. Richards, "Modified matrix encoding technique for minimal distortion steganography," in *Proc. Information Hiding Workshop 2006*, *LNCS* 4437, pp. 314-327, 2007.

[14] T. K. Moon*, Error Correction Coding, Mathematical Methods and Algorithms*. Hoboken, NJ: Wiley, 2005.

[15] D. Schönfeld, A. Winkler, "Embedding with syndrome coding based on BCH codes," i*n Proc. the ACM Workshop on Multimedia & Security*, Geneva, Switzerland, Sep. 26-27, pp. 214–223, 2006

[16] R. Zhang, V. Sachnev, and H. J. Kim, "Fast BCH syndrome coding for steganography," in *Proc. Information Hiding Workshop 2009*, *LNCS* 5806, pp. 48-58, 2009.

[17] J. Fridrich, and D. Soukal, "Matrix embedding for large payloads," *IEEE Trans. Information Forensics and Security,* vol. 1, no. 3, pp. 390-395, 2006.

[18] F. Huang, J. Huang, and Y. Q. Shi, "Detecting double JPEG compression with the same quantization matrix," *IEEE Trans. Information Forensics and Security,* vol. 5, no. 4, pp. 848-856, 2010.

[19] T. Filler, T. Pevny, and P. Bas. BOSS (Break Our Steganography System). http://www.agents.cz/boss, July 2010.

[20] P. Shllee, Matlab JPEG Toolbox [Online]. Available: http://www.philsallee.com/jpegtbx/index.html

[21] J. Kodovský and J. Fridrich, "Calibration revisited," in *Proc. the ACM Multimedia & Security Workshop*, Princeton, New Jersey, Sep. 7-9, pp. 63-74, 2009.

[22] T. Pevný, P. Bas, ad J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Information Forensics and Security*, vol. 52, no. 2, pp. 215-224, 2010

[23] J. Kodovský, T. Pevný, and J. Fridrich, "Modern steganalysis can detect YASS," in *Proc. SPIE, Electronic Imaging, Security Forensics of Multimedia XII*, San Jose, California, Jan. 17–21, 2010, vol. 7541, pp. 0201–0211.

[24] T. Pevný and J. Fridrich, "Merging Markov and DCT features for multi-class JPEG steganalysis," in *Proc. SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, California, Jan. 28 - Feb. 1, 2007, vol. 6505, pp. 03.1-03.13

[25] F. Huang, J. Huang, and Y. Q. Shi, "An experimental study on the security performance of YASS," *IEEE Trans. Information Forensics and Security*, vol. 5, no. 3, pp. 374-380, 2010.

[26] J. Kodovský and J. Fridrich, "Steganalysis of JPEG images using rich models," in *Proc. SPIE, Electronic Imaging,*

Media Watermarking, Security, and Forensics of Multimedia XIV, San Francisco, CA, Jan. 23–25, 2012, vol. 8303, pp. A-1-A-13.

[27] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Trans. Information Forensics and Security,* vol. 7, no. 3, pp. 868-882, 2012.

[28] Q. Liu , A. Sung, and M. Qiao, "Neighboring joint density-based JPEG Steganalysis," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 2, pp. 1-16, 2011.

[29] P. Sallee, "Model based methods for steganography and steganalysis," *International Journal of Image Graphics*, vol. 5, no. 1, pp. 167-190, 2005.

[30] W. Luo, F. Huang, and J. Huang, "Edge adaptive image steganography based on LSB matching revisited," *IEEE Trans. Information Forensics and Security,* vol. 5, no. 2, pp. 201-214, 2010.

[31] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc*. *Information Hiding Workshop 2010*, *LNCS* 6387, pp. 161–177, 2010

[32] K. Solanki, A. Sarkar, and B. S. Manjunath, "YASS: Yet another steganographic scheme that resists blind steganalysis," in *Proc*. *Information Hiding Workshop 2007*, *LNCS 4567*, pp. 16-31, 2007.

[33] A. Sarkar, K. Solanki, and B. S. Manjunath, "Further study on YASS: Steganography based on randomized embedding to resist blind steganalysis," in *Proc. SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, 2008, vol. 6819, pp. 17.1–17.11.

[34] J. Kodovský, J. Fridrich and V. Holub, "Ensemble calssifiers for steganalysis of digital media," *IEEE Trans. Information Forensics and Security*, vol. 2, no. 7, pp. 432-444, 2012