A New Rule for Cost Reassignment in Adaptive Steganography

Wenbo Zhou, Weiming Zhang, and Nenghai Yu

***Abstract*— In steganography schemes, the distortion function is used to define modification costs on cover elements, which is distinctly vital to the security of modern adaptive steganography. There are several successful rules for reassigning the costs defined by a given distortion function, which can promote the security level of the corresponding steganographic algorithm. In this paper, we propose a novel cost reassignment rule, which is applied to not one but a batch of existing distortion functions. We find that the costs assigned on some pixels by several steganographic methods may be very different even though these methods exhibit close security levels. We call such pixels *"controversial pixel"*. Experimental results show that steganalysis features are not sensitive to controversial pixels; therefore, these pixels are suitable to carry more payloads. We name this rule the controversial pixels prior (CPP) rule. Following the rule, we propose a cost reassignment scheme. Through extensive experiments on several kinds of stego algorithms, steganalysis features, and cover databases, we demonstrate that the CPP rule can improve the security of the state-of-the-art steganographic algorithms for spatial images.**

***Index Terms*— Controversial pixels, cost reassignment, modification probability, priority, steganography, steganalysis.**

I. Introduction

STEGANOGRAPHY is a technique for covert communication, which aims to hide secret messages into ordinary digital media without drawing suspicion [1], [2], [21]. Designing steganographic algorithms for various cover sources is challenging due to the fundamental lack of accurate models. Currently, the most successful approach for designing content adaptive steganography is based on minimizing the distortion between the cover and the corresponding stego object. The distortion is obtained by assigning a cost to each modified cover element (e.g., pixel in the spatial domain image), and the messages are embedded while minimizing the total distortion which is the sum of costs of all modified elements.

The first method based on the framework of minimal-distortion is HUGO (highly undetectable stego) [3]. HUGO defines the pixel's cost by the changing amplitude of the steganalyzer's features caused by modifying the current pixel, and pixels that make the feature vectors more deviated will have higher costs. The features of steganalyzer SPAM (subtractive pixel adjacency matrix) [4] is used in HUGO. A steganalyzer's features are usually generated by exploiting correlations between the predicted residuals of neighboring pixels [4]. Because the pixels in smooth areas can be accurately predicted, the modifications in such areas will be easily detected by steganalyzers. Therefore the embedding changes of HUGO will be gathered within textured regions. However, HUGO can be detected by a steganalyzer with a higher dimension of features, such as SRM (spatial rich models) [6].

In SRM, the predicted residuals are generated in various directions and manners, so the correlations between pixels can be further exploited. Therefore, one pixel, to be in a smooth area or a textural area, should be subtly defined for steganography. If the pixel can be accurately modeled in any direction, it should be considered as a smooth point and assigned larger cost. With this insight, Hulob and Fridrich [5] proposed the algorithm WOW which assigns high costs to pixels that are more predictable by a bank of directional filters. WOW improves the security of HUGO under the detection of SRM [6]. UNIWARD (universal wavelet relative distortion) [10] generalizes the cost function of WOW to make it simpler and more suitable for embedding in an arbitrary domain, including the spatial domain and DCT domain. Hence UNIWARD has a similar performance compared to WOW in spatial domain. Li *et al.* [11] proposed the method HILL, which improves WOW by spreading the costs with a low-pass filter. In HILL [11], the local modification probabilities are evened out and thus the modifications cluster in the complex areas (summarized as cost spreading (CS) rule in Sec. III).

The above methods design cost function in an ad hoc or empirical manner. Fridrich and Kodovský [12], Sedighi *et al.* [14] proposed model-driven approaches , in which Multivariate Gaussian (MG) or Multivariate Generalized Gaussian (MVG) distribution was used to model the noise residuals of pixels by assuming them to be independent but with varying variances. The models are established by estimating the variances and then the costs are computed by minimizing the power of an optimal statistical test. In fact, small costs will be assigned to residuals modeled with large variances, which are just the highly textured regions.

Another enhanced model-driven approach MiPOD is also proposed by Sedighi *et al.* [15]. In MiPOD the interactive effects of adjacent pixels have been taken into consideration. MiPOD achieves a better security by working the power of

Manuscript received February 14, 2017; revised May 12, 2017, June 7, 2017, and June 7, 2017; accepted June 15, 2017. Date of publication June 21, 2017; date of current version July 26, 2017. This work was supported by the Natural Science Foundation of China under Grant U1636201 and Grant 61572452. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Andrew D. Ker. *(Corresponding author: Weiming Zhang.)*

The authors are with the CAS Key Laboratory of Electro-magnetic Space Information, University of Science and Technology of China, Hefei 230026, China (e-mail: zhangwm@ustc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2017.2718480

the most powerful detector instead of the KL divergence, and adopting the idea of cost spreading described in [13] by smoothing the Fisher Information of modeled pixels.

Apparently, the state-of-the-art adaptive steganographic methods consider much regarding the cost assignment which can be directly related to the modification probabilities and the embedding locations. In the meantime, steganalysis follows up. The most effective high-dimensional steganalytic features are considered to be adaptive since they regard the adaptive steganography as a kind of selection-region steganography. MaxSRM [7] forms the co-occurrence matrices considering the maximum estimated modification probability of a group of pixels as a weight coefficient, for which the steganalytic feature is inclined to extract features from the texture region. The same idea is applied in [8], in which the mean estimated modification probability of a group of pixels is used as a weight coefficient. Recently, an improving selection-channel-aware steganalysis feature was proposed by Denemark and Fridrich [9], in which the weight coefficient of co-occurrence matrices is replaced by the residuals in SRM. All of these adaptive steganalysis can achieve better performances in detecting steganography since they focus more on the textured region instead of treating textured areas and smooth ones equally.

In this case, modifying pixels only considering the texture complexity will no longer promote the security of steganography. Li *et al.* [18] and Denemark and Fridrich [19] take the interaction between adjacent pixels into consideration, and force the modifications to be clustered in the same direction (summarized as synchronizing modification direction (SMD) rule in Sec. III). This method may mislead adaptive steganalysis to estimate an incorrect modification probability for pixels, thus potentially weakening the performances of adaptive steganalysis and promoting the security of steganography.

The design of adaptive steganography is more precise in the latest works. Whether modifying in textured area, spreading costs to neighbors using the CS rule or clustering modification directions with the SMD rule, they are aimed at finding a more suitable way of embedding. However, these successful rules are only used to improve one existing distortion function for steganography. In this paper, we consider how to generate an advanced distortion function from a batch of existing ones. We propose a security enhanced rule by combining several comparable methods, following our previous work [20]. Note that some minimal-distortion steganographic methods exhibit comparable security performances in resisting detection, but they define distortion functions in completely different manners. Thus, they may assign very different costs to the same pixel. We call such pixels controversial pixels, and consider that these controversial pixels have the potential to accommodate more payloads. We can improve undetectability by giving priority of modifications to such controversial pixels. We call this novel rule the Controversial Pixels Prior (CPP) rule. Compared to our previous work, we use an simulation in Sec. VI of this paper to give a preliminary proof that the CPP rule is effective, and add discussions of the optimizing function and the metric of controversial degree. In Sec. VIII, two groups of examples, CPP(UNI_derived) and CPP(HILL_derived), are added to demonstrate that the number of basic methods for the CPP rule can be three or more, and the improvements are still significant. For further exploration, we fused the CPP rule with the CS rule and the SMD rule to achieve a further improvement.

The rest of this paper is organized as follows. After introducing the framework of minimal-distortion steganography in Sec. II, we summarize several existing rules for adaptive steganography in Sec. III. In Sec. VI, we propose our CPP rule and present a simulation to show the advantages of the CPP rule intuitively. In Sec. VII, we provide a full description of the framework of CPP-based steganographic scheme and discuss the optimizing function and the metric for controversial degree. In Sec. VIII, important parameters are determined through experiments and several groups of steganalylsis experiments have been carried out to verify the advantages of the CPP rule. We also demonstrate combining the CPP rule with other effective rules in Sec. VIII, and two other databases are used for further verification in this section. We draw conclusions in Sec. IX.

II. Preliminaries

In this paper, matrices, vectors and sets are written in boldface, and the k-ary entropy function is denoted $H_k(\pi_1, \ldots, \pi_k)$ for $\sum_{i=1}^{k} \pi_i = 1$.

The cover sequence is denoted $\mathbf{x} = (x_1, x_2, ..., x_n)$, where the signal x_i is an integer, such as the gray value of a pixel. The embedding operation on x_i is formulated by the range I_i. An embedding operation is called binary if $|I_i| = 2$ and ternary if $|I_i| = 3$ for all i. For example, the ± 1 embedding operation is ternary embedding with $I_i = \{x_i - 1, x_i, x_i + 1\}$.

In the model established in [24], the cover $\mathbf{x}$ is assumed to be fixed, so the distortion introduced by changing $\mathbf{x}$ to $\mathbf{y} = (y_1, y_2, ..., y_n)$ can be simply denoted as $D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y})$. Assume that the embedding algorithm changes $\mathbf{x}$ to $\mathbf{y} \in \mathcal{Y}$ with probability $\pi(\mathbf{y}) = P(Y = \mathbf{y})$ which is called the modification probability (MP), and thus the sender can send up to $H(\pi)$ bits of message on average with average distortion $E_\pi(D)$ such that

$$H(\pi) = -\sum_{\mathbf{y} \in \mathcal{Y}} \pi(\mathbf{y}) \log \pi(\mathbf{y}), \tag{1}$$

$$E_\pi(D) = \sum_{\mathbf{y} \in \mathcal{Y}} \pi(\mathbf{y}) D(\mathbf{y}). \tag{2}$$

For a given message length L, the sender wants to minimize the average distortion, which can be formulated as the following optimization problems:

$$\min_{\pi} \quad E_\pi(D), \tag{3}$$

$$\text{subject to } H(\pi) = L. \tag{4}$$

Following the maximum entropy principle, the optimal π has a Gibbs distribution [24]:

$$\pi_\lambda(\mathbf{y}) = \frac{1}{Z(\lambda)} \exp(-\lambda D(\mathbf{y})), \tag{5}$$

where $Z(\lambda)$ is the normalizing factor such that

$$Z(\lambda) = \sum_{\mathbf{y}\in\mathcal{Y}} \exp(-\lambda D(\mathbf{y})). \tag{6}$$

The scalar parameter $\lambda > 0$ can be determined by the payload constraint (4). In fact, as proven in [23], the entropy in (4) is monotonically decreasing in λ, thus for a given L in the feasible region, λ can be quickly determined by binary search.

In particular, if the embedding operations on x_i's are mutually independent , the distortion introduced by changing $\mathbf{x}$ to $\mathbf{y}$ can be thought to be additive, and are measured by $D(\mathbf{y}) = \sum_{i=1}^{n} \rho_i(y_i)$, where $\rho_i(y_i) \in \mathbb{R}^*$ is the cost of changing the ith cover element x_i to y_i ($y_i \in I_i, i = 1, 2 \ldots n$). In this case, the optimal π is given by

$$\pi_i(y_i) = \frac{\exp(-\lambda\rho_i(y_i))}{\sum_{y_i\in I_i} \exp(-\lambda\rho_i(y_i))}, \quad i = 1, 2, \ldots, n. \tag{7}$$

When varing $\lambda \in (0, \infty)$, we can derive a relation between $H(\pi)$ and $E_\pi(D)$, which is called the *rate-distortion bound* [23]. Practical coding methods work under this bound.

In this paper, we consider the case of ternary embedding with the range $I = \{-1, 0, +1\}$, where 0 means that the pixel values remain invariant. In general, we assume that

$$\rho_i(-1) = \rho_i(+1) \triangleq \rho_i \in [0, +\infty). \tag{8}$$

And with Eq.(8), it can be assumed that

$$\begin{cases} \pi_i(-1) = \pi_i(+1) \triangleq \tau_i \in [0, \frac{1}{3}], \\ \pi_i(0) = 1 - 2\tau_i = 1 - p_i. \end{cases} \tag{9}$$

For additive distortion, simulating optimal embedding enables us to test the security of a steganographic method, but once the distortion function is properly defined, we can replace the optimal embedding simulator with some off-the-shelf coding methods such as STCs (Syndrome-Trellis Codes) [24], which can approach the lower rate-distortion bound.

III. Previous Works on Cost Assignment for Adaptive Steganography

In previous steganographic schemes, one basic rule for cost assignment is ***Complexity Prior***. Complexity prior means that the steganographer should give priority for modification to the complex areas, in other words, unpredictable parts in an image should have high priorities. The philosophy of complexity prior is that non-periodic textures and noisy regions are difficult to model, thus making modifications in such areas often leads to trivial deviation in steganalytic feature space and to be less detectable. In fact, all of the methods detailed in [3], [5], [10]–[12], and [14] follow this rule explicitly, and define distortion functions by investigating how to reasonably define the complex degrees of pixels in the sense of resisting detection. Moreover, based on the core idea of *Complexity Prior*, several effective rules for ranking priority of pixels have been proposed by previous works. These rules for cost assignment in adaptive steganographic schemes can be summarized as follows.

IV. Cost Spreading (CS) Rule

This rule is also called clustering rule. It requires that the costs of modifying two neighboring elements should not differ greatly. In other words, an element with high modification-priority should spread its high-rank to its neighborhood, and vice versa for an element with low priority. This rule suggests that it is better to make modifications in a clustered manner rather than in a scattered one. By applying this rule, a pixel that is close to a high-rank complex region should have a higher priority than another pixel in less complex region, even though these two pixels have same costs in the definition of distortion function. Correspondingly, the embedding modifications are clustered. This rule was first successfully used in HILL [11] to improve WOW [5], and then used in [15] to improve MVG. Experiments have demonstrated that the cost spreading rule can create less deviation in a steganalytic feature space.

V. Synchronizing Modification Direction (SMD) Rule

The above-mentioned CS rule is based on the concept of minimizing the sum of costs of all changed pixels, which is called additive distortion model. In additive distortion model, the modifications of pixels are assumed to be independent. Actually, the neighboring embedding changes will interact with each other, and thus a non-additive distortion model will be more suitable for adaptive steganography intuitively. Recently, the first effective principle on how to exploit the power of non-additive distortion was found independently by Li *et al.* [18] and by Denemark and Fridrich [19]; the proposed idea is based on an assumption that the steganalytic classifier cannot distinguish an image with all pixels changed by $+1$ or -1 at the same time from an original cover image. Experiments imply that synchronizing the modification directions of neighboring pixels can significantly improve the security performance. This idea can be summarized as the rule of synchronizing modification direction, which means that neighboring pixels changed in the same directions, i.e., $+1$ or -1 at the same time, will introduce smaller costs. This rule is also called clustering modification directions (CMD) in [18].

VI. A New Rule for Ranking Priority of Pixels

The above-mentioned CS rule and SMD rule are proposed to improve a single existing distortion function, but by comparing different algorithms, we found an interesting phenomenon. Namely, that some steganographic methods have a very similar security performance while defining distortions in very different ways. Among these methods, there is a distinguishment on the costs assignment for some pixels, in other words, the costs assigned on some pixels are large in one method but small in another. We define these pixels as ***"Controversial Pixels"*** because they are assigned with very different costs in different algorithms. Even with such a discrepancy, some of these algorithms can still provide the same level of security. This phenomenon implies that modifications on such controversial pixels have little affect on features of staganalysis.

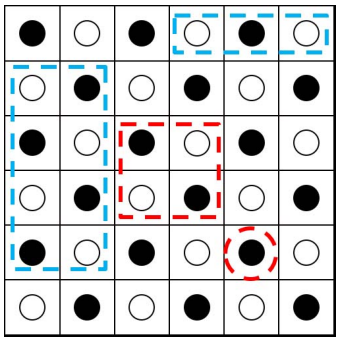


Fig. 1. Illustration of the location of controversial pixels for the CPP rule.

Based on these analyses, we propose another rule for adaptive steganography. We take advantages of the differences among comparable steganographic methods and focus on those controversial pixels. Our proposed rule is designated the controversial pixels prior (CPP) rule, and is described as follows:

A. Controversial Pixels Prior (CPP) Rule

This rule attempts to find controversial pixels by comparing several comparable steganographic methods, and the CPP rule suggests that it is better to give these controversial pixels priority of modification rather than considering the complexity of cover elements only.

Since costs represent the priority of a pixel, and the costs have a direct relationship with the modification probabilities as stated in Eq. (7), we focus on the modification probailities(abbreviated as MPs). As shown in Fig. 1, supposing that the pixels in the blue blocks are ordinary pixels, the costs of those pixels are small and equal to each other, which means they have an equally high priority of modification when just considering complexity prior. The pixels in the red blocks are some controversial ones. In current adaptive steganography, it is normal to modify pixels in the blue blocks first because of their high priorities, while the CPP rule requires us to adjust the priorities of controversial pixels in the red blocks to be higher. In this way, modifications are more likely to be clustered in the red blocks, which are occupied by controversial pixels. We conjecture that this distinctive way of modification will introduce less deviation in a steganalytic space than ordinary ways do. To justify our claim, we implement a simulation with the following steps.

(a). Take 1000 grayscale images from the BOSSBase ver.1.01 database [26] randomly, and crop a block of size 64×64 from the center of each image. The new generated image set is denoted $\{C_j\}(j \in \{1, ..., 1000\})$.

(b). Choose two adaptive steganographic methods as contrastive methods to locate controversial pixels, here we use UNIWARD [10] and WOW [5] as two basic methods because that they have exhibited a very similar security performance under the detection of SRM [6].

(c). Set the modification probabilities of controversial pixels to be $\frac{2}{3}$, which means that the payload of a single pixel can reach the maximum according to information theory. In addition, set the payload to be 0.2bpp to keep the embedded message lengths consistent in several algorithms.

(d). Modify ordinary elements of the images in $\{C_j\}$ in the order of priority after defined distortions by UNIWARD and WOW, while only modifying controversial elements after locating them using the CPP rule. The generated stego image sets are denoted as $\{SA_j\}$, $\{SB_j\}$, and $\{SC_j\}$ respectively.

(e). Calculate the 34671-D SRM steganalytic feature vector for each image. The obtained feature sets are denoted by $\{f(C_j)\}$, $\{f(SA_j)\}$, $\{f(SB_j)\}$, and $\{f(SC_j)\}$.

(f). Calculate the MMD (maximum mean discrepancy [21], which measures the distance between the feature set of cover images and that of stego images) between $\{f(C_j)\}$ and $\{f(SA_j)\}$, $\{f(SB_j)\}$, and $\{f(SC_j)\}$. Obtain the average valule of the MMD and standard deviation over 10 different independent tests on the dataset. Denote these three results as *MMD(CPP), MMD(UNI), MMD(WOW)*, and then make a comparison.

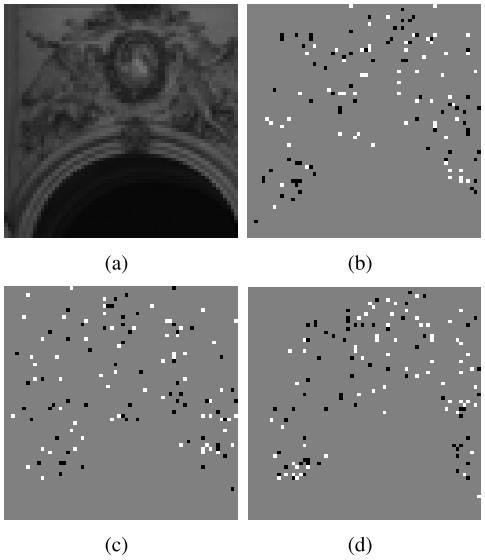


Fig. 2. Illustration of modified location in an image of size 64×64 for the CPP rule. The black, white and gray pixels represent $+1$, -1 and no change, respectively. (a) Cropped 1013.pgm of BOSSbase. (b) Modified location of CPP rule. (c) Modified location of UNIWARD. (d) Modified location of WOW.

TABLE I
SIMULATION RESULTS OF MMD, STANDARD DEVIATION AND CHANGE RATES

Payload	Embedding method	MMD	Change rate
0.2	CPP	$5.3240 \times 10^{-2} \pm 0.0051$	2.93%
	UNIWARD	$9.1198 \times 10^{-2} \pm 0.0027$	2.56%
	WOW	$8.9245 \times 10^{-2} \pm 0.0047$	3.66%

Fig. 2 provides an example of the modification locations described in Step(d) above. The statistical results of MMD and its standard deviation are given in Table I, with the rightmost row containing the total change rates.

From the statistical results, the *MMD(CPP)* value is apparently smaller than that of the other two methods. The change rate of CPP is between that of UNIWARD and WOW, even

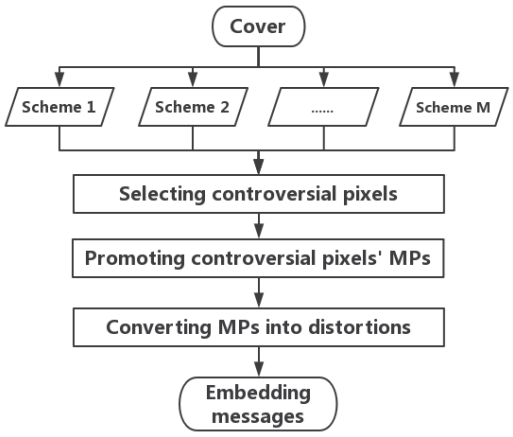


Fig. 3. Flowchart of proposed CPP based method.

though change rate has no direct relation with security. It can also be referenced data that may indicate that the CPP rule has found a better balance between the payload of a single element and the total change rate. Since MMD represents the distance in steganalytic feature space between cover set and stego set, the simulation results verify that making modifications in controversial regions is more secure than that making them in ordinary regions.

In Sec. VII, we propose a new strategy of designing distortion function by applying the CPP rule. We introduce more in Sec. VIII, in which we fuse several aforementioned rules for designing distortion functions to achieve a prominent increase in security.

VII. A Novel Strategy Based on the CPP Rule

A. Description of the CPP Based Scheme

With the new perspective on cost reassignment, we propose an enhanced steganographic method based on the CPP rule, that generates a new distortion function from several existing ones. As shown in Eq. (7), the distortion can be converted into MP, which then determines the payloads assigned on each pixel. Therefore, we fix our attention on the MPs when searching for controversial-pixels.

The framework of the proposed embedding method is depicted in Fig. 3. Suppose that we have M comparable steganographic methods whose security performances are similar. First, we compute the MPs of pixels with these M distortion functions. Second, we label the controversial-pixels according to the MPs. Third, we adjust the MPs to promote the controversial pixels' priorities with the CPP rule. Fourth, the adjusted MPs are converted into a new distortion function. Finally, the messages are embedded with STCs according to the new distortion function.

Assume that each of these M comparable steganographic schemes is defined by an additive distortion function D_k for $1 \leq k \leq M$. The cover is a spatial image consisting of N pixels $\{x_1, \ldots, x_N\}$. For the given payload γ and the distortion function D_k, we can calculate the MPs of N pixels, and denoted by $\boldsymbol{p}_k = \{p_{k,1}, p_{k,2}, \ldots p_{k,N}\}$, $1 \leq k \leq M$ with Eq. (7). Collecting all of the probabilities obtained from the M distortion functions, we obtain an $N \times M$ matrix $\mathbf{R}$.

$$\mathbf{R} = [\mathbf{r_1}, \mathbf{r_2}, ..., \mathbf{r_N}] = \begin{bmatrix} p_{1,1}, & p_{1,2}, & \cdots, & p_{1,N} \\ p_{2,1}, & p_{2,2}, & \cdots, & p_{2,N} \\ \vdots & & \cdots & \vdots \\ p_{M,1}, & p_{M,2}, & \cdots, & p_{M,N} \end{bmatrix}_{N \times M}$$

Here, the ith column $\boldsymbol{r}_i = \{p_{1,i}, p_{2,i}, p_{3,i}, ..., p_{M,i}\}^T$ consists of the MPs of pixel x_i obtained from the aforementioned M distortion functions. This MP vector records the information of priority for pixel x_i in different methods and one pixel corresponds to one vector. Since the CPP rule is aimed at finding those controversial ones, we can compute the statistical characteristics of MP vectors to judge the controversial degree.

In order to describe the dispersed degree among the elements of $\boldsymbol{r}_i$, we first calculate the mean value and variance:

$$\bar{p}_i = \frac{1}{M} \sum_{k=1}^{M} p_{k,i}, \tag{10}$$

$$v_i = \frac{1}{M} \sum_{k=1}^{M} (p_{k,i} - \bar{p}_i)^2. \tag{11}$$

Here, the variance v_i is called the "probability variance" (PV) of the pixel x_i. With v_i, the degree of controversy can be determined. It is obvious that a large v_i reflects that the changing scope of MPs is dramatic, which demonstrates that the priorities of pixel x_i are controversial in different methods, and this pixel can be given higher priority of modification in the CPP rule.

For a given payload, the total change rate of a steganographic method is correspondingly determined [13]. Thus, we should choose a certain amount number of pixels as the controversial pixels in our proposed CPP rule. Denote the $(1-\alpha)\%$ quantile of the vector of all PVs: $\boldsymbol{V} = \{v_1, v_2, ...v_N\}$ by T_α, and define those pixels with PVs larger than T_α as the controversial pixels. In other words, the pixels with the top $\alpha\%$ of large PVs are selected as controversial pixels in Fig. 3. Here, α is called controversial threshold (we will discuss the setting of α in next section). We fix our attention only on these controversial pixels. We set

$$v_i' = \begin{cases} v_i & \text{if } v_i > T_\alpha \\ 0 & \text{otherwise} \end{cases} \quad 1 \leq i \leq N. \tag{12}$$

As mentioned above, the MPs can reflect the priorities of pixels, and thus we can promote the priorities of controversial pixels by increasing their MPs using the following equation:

$$p_i' = \bar{p}_i \cdot f(v_i'), 1 \leq i \leq N, \tag{13}$$

where the function $f(*)$ is an optimizing function used to promote the MPs of pixels. There are two important attributes of the optimizing function $f(*)$:

i.) The value domain of $f(*)$ should be $[1, +\infty)$ theoretically, and $f(0) = 1$. Since we are trying to give controversial pixels higher priorities, the adjusted MP p_i' should never be smaller than the original one, $\bar{p}_i$, that the function $f(*)$ should be no smaller than 1.

ii.) The optimizing function $f(*)$ should be monotonically increasing. In the assumption of the CPP rule, controversial

pixels are those pixels which steganalysis features are not sensitive to, and the more controversial they are, the more suitable they are for embedding. Under this assumption, $f(*)$ is supposed to be monotonically increasing, and thus pixel's priority can be sufficiently promoted when its υ_i' value is sufficiently large. In other words, the more controversial the pixel, the higher its priority.

With Eq. (13), if x_i is a controversial pixel, its MP is increased by multiplying the optimizing item $f(\upsilon_i')$, and its priority of modification also increases correspondingly; otherwise, its MP remains $\bar{p}_i$ because $\upsilon_i' = 0$, and its priority also remains invariant.

Note that, for ± 1 embedding, when the MP $p_i = \frac{2}{3}$, the pixel x_i has the largest average payload, $\log_2 3$ (which is consistent with the assumption in Eq. (9)). Therefore, we limit the adjusted MPs by

$$p_i'' = \min\left\{p_i', \frac{2}{3}\right\}, 1 \le i \le N. \tag{14}$$

Actually, p_i'' is not the final MP that we used for embedding. Because that after the adjustments by Eqs. (13) and (14), the total information entropy is no more under the constraint of original payload. Therefore, we should flip the MP to distortion and then use the practical off-the-shelf coding methods STCs.

Denoting $\pi_i(+1) = \pi_i(-1) = p_i''/2$ and $\pi_i(0) = 1 - p_i''$, by Eq. (7), the corresponding distortion function $\rho_i(l)$ $(l \in I)$ satisfies

$$\pi_i(l) = \frac{\exp(-\lambda\rho_i(l))}{\sum_{t\in I}\exp(-\lambda\rho_i(t))}, l \in I; 1 \le i \le N. \tag{15}$$

To solve $\rho_i(l)$ from Eq. (15), without loss of generality, we can set $\lambda = 1$ because λ is monotonically decreasing with respect to the message length as proven in [23]. And the transformed distortion has the form

$$\rho_i(l) = \ln\frac{\pi_i(0)}{\pi_i(l)},\ l \in I, 1 \le i \le N. \tag{16}$$

We call $\rho_i(l)$ in Eq. (16) the adjusted distortion function, and it can be easily verified that the adjusted distortion satisfies Eq. (15).

Eventually, we obtain a new steganographic algorithm determined by the adjusted distortion function (16), according to which we embed messages under the payload of γ by using STCs. The details of the CPP based method is described in Algorithm 1.

Algorithm 1 CPP Based Scheme

Input: A cover image $\mathbf{x}$ with N pixels $x_1, \ldots x_n$; payload γ; M comparable distortion functions for adaptive steganography.

Output: The stego image $\mathbf{y}$.

1) Set controversial threshold α according to the target payload γ.
2) For the M distortion functions, compute the MPs of pixels, $\boldsymbol{p}_k = \{p_{k,1}, p_{k,2}, \ldots p_{k,N}\}$, $1 \le k \le M$, by Eq. (7) according to the payload γ.
3) Calculate the PV υ_i for $1 \le i \le N$ with Eq. (11), and then select the pixels with top $\alpha\%$ large PVs as controversial pixels.
4) Adjust the MPs with Eqs. (12)-(14), and then convert them into adjusted distortion functions with Eq. (16).
5) Embed messages under the payload of γ into cover image $\mathbf{x}$ with STCs according to the adjusted distortions, and finally output the stego image $\mathbf{y}$.

B. Discussions on the Optimizing Function

The key idea of the CPP rule is to promote the priorities of controversial pixels according to their degree of controversy, and we can achieve this goal using Eq. (13) (see Sec. VII-A). The optimizing function $f(*)$ is a principal element in our proposed scheme. In this subsection, we discuss the selection of $f(*)$.

The simplest consideration is a linear function. Suppose that $f_1(\upsilon_i') = k(1 + \upsilon_i')$, where $k \in [1, +\infty)$ is a scaling factor. For simplicity, let $k = 1$, and then we finally obtain

$$f_1(\upsilon_i') = 1 + \upsilon_i'. \tag{17}$$

Obviously, f_1 satisfies the two attributes of $f(*)$, and can be easily implemented.

The second choice of $f(*)$ is an exponential function. We can suppose that

$$f_2(\upsilon_i') = e^{\upsilon_i'}. \tag{18}$$

Note that the second derivative of f_2 is larger than 0, which is different from f_1, which means that the increasing speed of f_2 is faster than that of f_1.

Another choice for the function $f(*)$ is a logarithmic function, and we can suppose that

$$f_3(\upsilon_i') = \ln(e + \upsilon_i'). \tag{19}$$

The most significant difference among f_1, f_2 and f_3 concerns their second derivatives, which can be simply stated as $f_1'' = 0$, $f_2'' > 0$ and $f_3'' < 0$. The second derivative can reflect the increasing speed of a monotonically increasing function. In this discussion, the linear function f_1 obviously increases uniformly, while the increasing speed of the exponential function f_2 becomes faster and faster with increasing of υ_i', and the logarithmic function f_3 is contrary to f_2.

In this part, we only take these three types of functions as our alternative options because they are typical examples for the case of $f''(*) = 0$, $f''(*) > 0$ and $f''(*) < 0$, respectively. Moreover, these three functions all satisfy the two necessary requirements of $f(*)$. The choice of the function $f(*)$ has a direct relationship with the security performance of our CPP based scheme, thus how to determine it commands serious deliberation.

Here we perform a few tests to help to select the optimal optimizing function $f(*)$. Most of the experimental settings are same as those of the simulation described in Sec. VI. We still use UNIWARD and WOW as two basic methods for the CPP based scheme (denoted as CPP(UNI,WOW)) and the 34671-D SRM steganalytic feature is used here. The cover set consists of blocks of size 64×64 from the center of 1000 randomly selected grayscale images from BOSSBase ver.1.01, and the

TABLE II

TESTING RESULTS OF MMD, STANDARD DEVIATION AND CHANGE RATES FOR f_1, f_2 AND f_3

Payload	Function $f(*)$	MMD	Change rate
0.4	f_1	$5.9511 \times 10^{-3} \pm 4.47 \times 10^{-5}$	7.90%
	f_2	$5.3327 \times 10^{-3} \pm 4.25 \times 10^{-5}$	7.73%
	f_3	$5.6844 \times 10^{-3} \pm 7.18 \times 10^{-5}$	7.75%

TABLE III

TESTING RESULTS OF MMD, STANDARD DEVIATION AND CHANGE RATES FOR PV, R, MD AND CV

Payload	Statistics	MMD	Change rate
0.4	PV	$5.3327 \times 10^{-3} \pm 4.25 \times 10^{-5}$	7.73%
	R	$5.6579 \times 10^{-3} \pm 6.45 \times 10^{-5}$	8.20%
	MD	$5.4561 \times 10^{-3} \pm 4.95 \times 10^{-5}$	7.98%
	CV	$5.4757 \times 10^{-3} \pm 5.40 \times 10^{-5}$	7.84%

controversial threshold α is set as 10. In the tests described in the following, we use the scheme proposed in Sec. VII-A to generate three different distortion functions and stego sets with three optimizing functions f_1, f_2 and f_3. The MMDs and standard deviations of these three stego sets are obtained over the course of conducting 10 different independent tests on the dataset. The results and the average change rates are presented in Table II. From the table, we can see that the MMDs of these three functions do not differ too much from others, while f_2 outperforms f_1 and f_3.

Considering the testing results reported in Table II, we finally decide to choose f_2 as the optimizing function of our proposed CPP based scheme. Thus Eq. (13) can be rewritten as

$$p_i' = \bar{p}_i \cdot e^{\upsilon_i'}, 1 \leq i \leq N. \tag{20}$$

C. Metric for the Controversial Degree

In our proposed scheme, as described in Sec. VII-A, we use the probability variance(PV) υ_i to describe the controversial degree among several different modification probabilities. However, there are several other statistical characteristics that could be used to describe the dispersed degree of data.

In this subsection, we make a comparison among PV values and some other statistical characteristics of the data to determine a proper metric for the controversial degree.

We chose several commonly used statistical characteristics for comparison, e.g., range (denoted R), mean deviation (denoted MD), and coefficient of variation (denoted CV). For the ith element in M comparable schemes, the aforementioned values are calculated, respectively, as follows:

$$R_i = max(p_{k,i}) - min(p_{k,i}), 1 \leq k \leq M, \tag{21}$$

$$MD_i = \frac{1}{M} \sum_{k=1}^{M} |p_{k,i} - \bar{p}_i|, \tag{22}$$

$$CV_i = \frac{\sqrt{\upsilon_i}}{\bar{p}_i}. \tag{23}$$

We substitute PV with R, MD, and CV in our proposed scheme, and calculate the MMD and standard deviation between the cover and stego versions of the dataset. All of the experimental settings are completely the same as Sec. VII-B. The testing results are listed in Table III.

The MMDs of PV, R, MD and CV are extremely close to each other, while PV has the smallest MMD which corresponds to the best security performance. Thus, we will not change our decision and finally use PV as the metric for controversial degree.

VIII. EXPERIMENTS

A. Setups

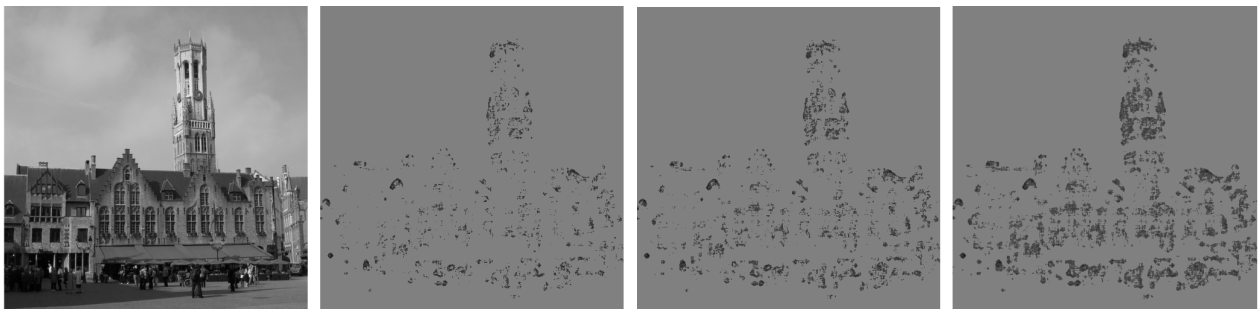
In this paper, four disjoint sets are used as the image database. In Sec. VIII-B, we first take BOWS-2-OrigEP3 [25] (simplified as BOWS-2) which containing 10,000 512×512 8-bit grayscale images as the database with which to explore the optimal setting of the controversial threshold α. We then verify using the other image set, namely the BOSSbase ver.1.01 database [26] containing 10,000 512×512 8-bit grayscale images. In Secs. VIII-D and VIII-E, all of the steganalysis experiments are based on the BOSSbase ver.1.01. In Sec. VIII-F, two other datasets, BOSSbaseC and BOSSbaseJ85 [27] are used for further verification. The security of all steganographic schemes are evaluated using a steganalyzer that is a detector trained on a given cover source and its stego version embedded with a fixed payload. The detector is first trained using the state-of-the-art 34,671-D SRM feature set [6] with the ensemble classifiers [28] for several groups of examples. For further experiments, we use the selection-channel-aware feature maxSRMd2 [9]. Performance in terms of resisting detection is evaluated by the testing error, which is computed as the mean value of the false positive rate and the false negative rate, averaged over 10 random splits of the dataset. A larger classification error rate means stronger security.

B. Determination of Controversial Threshold α

We first visualize the distribution of selected controversial pixels in Fig. 4 by varying the value of α. UNIWARD and WOW are used to define the basic distortion functions under the payload of 0.4bpp. We mark the controversial pixels in the stego version of 6.pgm of BOWS-2 by setting $\alpha = 3, 5$ and 10. Obviously, the locations of controversial pixels spread with increasing α.

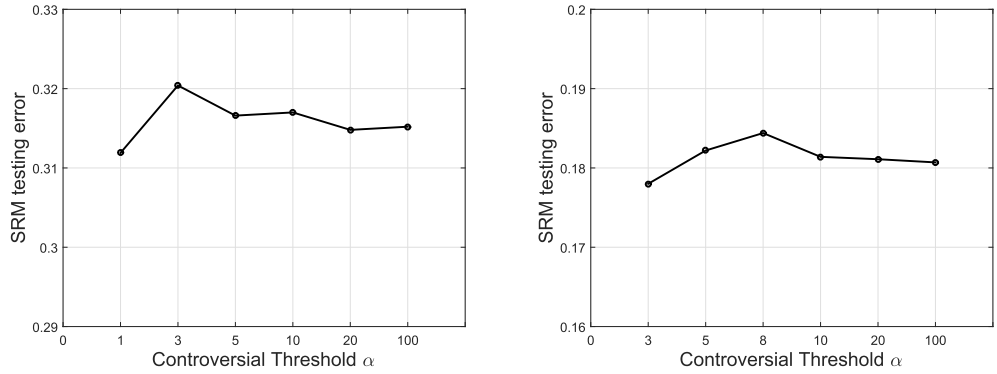
As mentioned in [13], the total change rate of pixels has a linear relation with relative payload. Considering that the message length is limited, the number of selected controversial pixels should also be adjusted corresponding to the payload.

In Fig. 5, we plot the curve of varying trends of SRM testing error with respect to α. The CPP based scheme is CPP(UNI,WOW) and the database is BOWS-2. The security performances fluctuate within a narrow range when the controversial threshold α changes both in the case of 0.2bpp and 0.4bpp. The largest testing error appears approximately at $\alpha = 3$ for 0.2bpp and at approximately $\alpha = 8$ for 0.4bpp. To determine the relation between an optimal α and payload, we conduct several steganalytic experiments under other payloads, and the chosen α values are listed



(a) (b) (c) (d)

Fig. 4. Location of the top $\alpha\%$ of large controversial pixels: (a) the sample cover image 6.pgm of BOWS-2, (b) $\alpha = 3$, (c) $\alpha = 5$, and (d) $\alpha = 10$. where the dark-gray points represent the controversial pixels and the light-gray points represent ordinary ones.



(a) (b)

Fig. 5. Varying trends of SRM testing error with respect to α under (a) 0.2 bpp and (b) 0.4 bpp for CPP(UNI,WOW); the testing set is BOWS-2.

TABLE IV
OPTIMAL α VALUES FOR CPP BASED SCHEME USING DATABASE BOWS-2

Steganalytic feature	Payload (bpp)	0.1	0.2	0.3	0.4	0.5
SRM	optimal α_1	1.5	3.7	6.0	8.3	10.6
maxSRMd2	optimal α_2	2.0	4.0	6.2	8.8	10.5

TABLE V
OPTIMAL α VALUES FOR CPP BASED SCHEME USING DATABASE BOSSBASE ver. 1.01

Steganalytic feature	Payload (bpp)	0.1	0.2	0.3	0.4	0.5
SRM	optimal α_1'	1.7	3.8	6.1	8.5	11.4
maxSRMd2	optimal α_2'	2.1	4.2	6.5	8.6	11.2

in Table IV. We also perform steganalytic experiments on CPP(HILL,MiPOD), of which the two basic methods are HILL [11] and MiPOD [15]. The steganalytic feature is maxSRMd2, since HILL and MiPOD have similar security performances under the detection of maxSRM. The optimal α values for CPP(UNI,WOW) and CPP(HILL,MiPOD) are denoted α_1 and α_2, respectively, in Table IV.

We can establish a linear regression model between α and the payload γ for α_1 and α_2 using the statistical data in Table IV:

$$\alpha_1 = 22.54\gamma - 0.75, \tag{24}$$

$$\alpha_2 = 21.60\gamma - 0.22. \tag{25}$$

Both of these two linear models have intercepts very close to zero and slopes close to 22. Thus, we finally determine the optimal value of α using the following simplified proportional relation:

$$\alpha = 22\gamma\,. \tag{26}$$

In order to prevent the optimal α overfitted to the current image set, we use BOSSbase ver.1.01 to verify that whether Eq. (26) is also appropriate for other databases. In Table V the optimal α values for CPP(UNI,WOW) and CPP(HILL,MiPOD) are denoted as α_1' and α_2', respectively.

The linear regression models for α_1' and α_2' are expressed as follows:

$$\alpha_1' = 23.38\gamma - 0.67, \tag{27}$$

$$\alpha_2' = 22.42\gamma - 0.19, \tag{28}$$

These two equations can be also simplified as Eq. (26) approximately, which verifies that our setting for computing optimal α values is reasonable. With Eq. (26), we can easily locate controversial pixels for different payloads.

C. Selection of Basic Steganographic Methods for CPP

In Sec. VII-A, we mentioned that an important condition in our proposed CPP rule is that several adaptive steganographic

TABLE VI
COMPARISONS AMONG ORIGINAL UNIWARD, BIAS_UNI, UM_UNI AND HILL, BIAS_HILL UM_HILL

Methods	Modification Probabilities			Testing errors under maxSRMd2				
	Maximum	Interquartile range	Variance	0.1	0.2	0.3	0.4	0.5
UNIWARD	0.5161	0.0964	0.0859	.3637±.0036	.2884±.0038	.2366±.0026	.1898±.0022	.1521±.0031
BIAS_UNI	0.6624	0.1050	0.1104	.3487±.0019	.2849±.0031	.2383±.0014	.1922±.0022	.1603±.0018
UM_UNI	0.5875	0.1216	0.1155	.3567±.0028	.2850±.0021	.2333±.0024	.1882±.0028	.1484±.0025
HILL	0.6062	0.1099	0.1270	.3765±.0029	.3120±.0020	.2628±.0031	.2180±.0033	.1853±.0024
BIAS_HILL	0.6656	0.1105	0.1405	.3584±.0029	.3096±.0031	.2571±.0014	.2141±.0036	.1810±.0018
UM_HILL	0.6002	0.1091	0.1196	.3775±.0028	.3143±.0021	.2685±.0024	.2216±.0028	.1874±.0025

methods exist with comparable security performances. This is the essential principle for selecting candidate algorithms for the CPP rule: basic methods have comparable security performances. In theory, any pair of steganographic methods can be used in the CPP rule as basic functions as long as they have similar security performances. Some off-the-shelf methods can be used as the basic distortion functions of CPP rule.

UNIWARD and WOW are the best examples for the CPP rule. Since they define distortion functions in really similar way, the security performances under the detection of SRM of these two methods are extremely close to each other on BOSSbase ver.1.01. This group of algorithms was used first in the following experiments.

HILL and MiPOD comprise another pair of examples in the following experiments. HILL is an adaptive algorithm which utilizes the CS rule to define costs, while MiPOD is an entirely model-driven scheme that also considers the CS rule. HILL achieves a higher level of security than MiPOD under the detection of SRM while MiPOD performs better than HILL under the selection-channel-aware steganalytic feature maxSRMd2. The pair of above-mentioned methods are not as similar as UNIWARD and WOW, but they are currently the most effective steganographic methods; thus we use them to prove the effectiveness of our CPP rule.

For further verification, we try to add the number of basic methods in the CPP rule. Recently, a new idea has been proposed in [16], in which game theory has been taken into consideration for designing distortion function. The main idea espoused in [16] is use of an existing adaptive steganographic method to define a basic distortion function, and then to define a bias function in the framework of game theory to adjust the distribution of modification probabilities. In this paper, we use UNIWARD and HILL, respectively, to define the basic distortion function and then reproduce the experiments as detailed in [16]. We find that the security of the method generated from UNIWARD is quite close to the original UNIWARD under the detection of maxSRMd2, as is that using HILL. This result indicates that the new method described in [16] can be adopted by the CPP rule as a basic method. Since the bias function is important in this method, we name the new distortion functions BIAS _UNI and BIAS_HILL.

To create another approximative basic distortion function for the CPP rule, we continue to make a few adjustments to original UNIWARD and HILL. Chen *et al.* proposed a method in [17], the main idea of which is doing preprocessing on a cover image using the technique of unsharp masking which can slightly change the textural features of cover images. After the preprocessing, we can redefine distortions on the new cover with UNIWARD or HILL, and thus obtain different distortion functions that are denoted UM_UNI and UM_HILL, respectively. We list some statistical data of modification probabilities in Table VI to show the nuances among original UNIWARD, BIAS_UNI, and UM_UNI, and among original HILL, BIAS_HILL, and UM_HILL. The sharpening parameter for UM_UNI and UM_HILL is 0.8. The security performances under maxSRMd2 are also listed, and the tested payloads range from 0.1bpp to 0.5bpp.

The three leftmost columns list statistical data for the modification probabilities of the entire cover image, including the maximum, the interquartile range, and the variance. The five rightmost columns are the testing errors and standard deviations resisting maxSRMd2 from 0.1bpp to 0.5bpp. Obviously, the three methods related to UNIWARD are quite different from each other, even though they have similar security performances. Those three methods related to HILL are in the same situation. Thus, we use these two groups of algorithms as another two examples for the CPP rule.

D. Security Performances on BOSSbase ver.1.01

In this subsection, we conduct several steganalysis experiments to verify that our proposed CPP-based scheme outperforms other methods. The first pair of examples for the CPP is based on UNIWARD and WOW, and denoted $CPP(UNI, WOW)$. Since UNIWARD and WOW are only close to each other under the detection of SRM, the steganalytic feature used to test $CPP(UNI, WOW)$ is SRM. The second group of experiments for the CPP is based on HILL and MiPOD, and denoted $CPP(HILL, MiPOD)$. As mentioned above, HILL achieves a higher level of security than MiPOD under the detection of SRM while MiPOD performs better than HILL under maxSRMd2, thus we use SRM and maxSRMd2 as steganalytic features to execute detections on $CPP(HILL, MiPOD)$.

The third group of examples can be divided into two parts. One includes three comparable methods, UNIWARD, BIAS_UNI and UM_UNI, and is denoted $CPP(UNI\_derived)$. The other includes HILL, BIAS_HILL and UM_HILL, and is denoted as $CPP(HILL\_derived)$. MaxSRMd2 is used for detection here, and both of the two parts exactly meet the basic condition of the CPP rule. The results with optimal embedding simulator and ensemble classifier are shown in Fig. 7. All numerical values of testing errors and standard deviations are listed in Table VII.

TABLE VII

NUMERICAL VALUES OF TESTING ERROR AND STANDARD DEVIATION FOR FIG. 7, FIG. 9 AND FIG. 10

Feature set	Embedding method	Testing errors from 0.1bpp to 0.5bpp				
		0.1	0.2	0.3	0.4	0.5
SRM	UNIWARD	.4020±.0023	.3242±.0029	.2609±.0016	.2073±.0033	.1637±.0028
	WOW	.4070±.0028	.3220±.0032	.2630±.0019	.2130±.0022	.1691±.0023
	$CPP(UNI, WOW)$	.4187±.0019	.3360±.0027	.2735±.0017	.2227±.0016	.1826±.0020
	HILL	.4360±.0033	.3632±.0024	.2996±.0020	.2482±.0032	.2038±.0021
	MiPOD	.4124±.0039	.3429±.0021	.2882±.0017	.2392±.0022	.1953±.0025
	$CPP(HILL, MiPOD)$	.4410±.0028	.3762±.0021	.3091±.0022	.2598±.0024	.2126±.0017
maxSRMd2	HILL	.3765±.0029	.3120±.0020	.2628±.0031	.2180±.0033	.1853±.0024
	MiPOD	.3954±.0028	.3295±.0021	.2713±.0028	.2270±.0039	.1871±.0031
	$CPP(HILL, MiPOD)$	.3999±.0023	.3384±.0019	.2832±.0013	.2384±.0031	.1995±.0020
	$CPP(UNI\_derived)$	.3759±.0022	.3047±.0029	.2524±.0016	.2052±.0021	.1727±.0034
	$CPP(HILL\_derived)$	.3883±.0027	.3282±.0022	.2791±.0027	.2323±.0012	.1957±.0024
	$CS-CPP(HILL, MiPOD)$	.4029±.0014	.3436±.0016	.2878±.0013	.2416±.0022	.2021±.0023
	$CS-CPP(HILL\_derived)$	.3933±.0028	.3326±.0026	.2830±.0016	.2381±.0021	.1999±.0024
	$SMD-CS-CPP(HILL\_derived)$	.4111±.0027	.3678±.0022	.3172±.0019	.2757±.0031	.2382±.0020
	$SMD-HILL$	.3899±.0029	.3334±.0030	.2918±.0018	.2488±.0026	.2203±.0024
	$SMD-MiPOD$	.4030±.0020	.3543±.0028	.3014±.0017	.2598±.0029	.2222±.0022

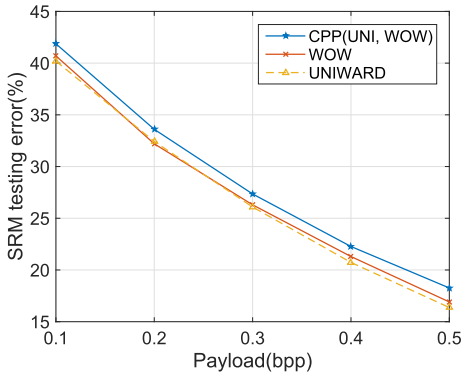


Fig. 6. Steganalytic performance of $CPP(UNI, WOW)$ under SRM.

In Fig. 6, $CPP(UNI, WOW)$ always has a higher level of security than UNIWARD and WOW for various payloads. In Fig. 7(a) and Fig. 7(b), the CPP based scheme also improves the level of security for HILL and MiPOD under both SRM and maxSRMd2. It is worth mentioning that the promotion in Fig. 7(a) for $CPP(HILL, MiPOD)$ under SRM is not as conspicuous as it is in $CPP(UNI, WOW)$, which might because the performances of HILL and MiPOD under SRM are not very close to each other, as UNIWARD and WOW are, and this divergence probably is a disadvantage in the CPP rule. The obvious promotions in Fig. 7(c) and Fig. 7(d) indicate that the CPP rule can be suitable for the case of three or more basic distortion functions. In addition, the security performance of $CPP(HILL\_derived)$ is much better than that of $CPP(UNI\_derived)$, which is due to the fact that the basic methods in $CPP(HILL\_derived)$ are related to HILL, and it outperforms UNIWARD in $CPP(UNI\_derived)$ on BOSSbase ver.1.01. The results of Fig. 7(c) and Fig. 7(d) show us that, theoretically, if we use higher-security methods as basic distortion functions, we can obtain better-performing steganographic algorithms with the CPP rule.

The statistical significance of the improved performance can be verified by hypothesis testing. The numerical results in Table VII indicate that the improvements of CPP-based methods are significant, where we use the testing errors of $CPP(HILL, MiPOD)$ and MiPOD under maxSRMd2 as examples.

TABLE VIII

TEST STATISTIC VALUES AND QUANTILES FOR HYPOTHESIS TESTING

Payload(bpp)	0.1	0.2	0.3	0.4	0.5
t	3.4627	13.8812	12.3239	13.4756	4.0804
$t_{0.025}(5)$	2.5706				

In cross-validation, testing errors are not independent because of the overlap of samples in different fold. To take account of non-independence, Dietterich defined a "5×2-fold cross-validated paired t test" [29], which defines a statistic value t that has an approximately $\boldsymbol{t}$ distribution with 5 degrees of freedom in the null hypothesis. Here, the hypotheses are:

$$H_0 : \mu_1 = \mu_2; \qquad H_1 : \mu_1 > \mu_2.$$

in which μ_1 and μ_2 are the mean values of testing errors of $CPP(HILL, MiPOD)$ and MiPOD, $\mu_1 = \mu_2$ represents that there is no significant differences between them.

We choose a significance level of $\alpha = 0.05$. The test statistic values t and quantiles $t_{0.025}(5)$ are listed in Table VIII.

In Table VIII, the test statistic t values are always larger than the corresponding quantile $t_{0.025}(5)$, which implies the promotions are significant, i.e. more than random chance. With the same method for the other groups of experiments, we have verified that the improvements of our proposed CPP-based schemes also have statistical significance.

E. Further Study of Fusing Rules

In Secs III and VI, we summarized the CS rule, the SMD rule and the CPP rule for promoting the security of adaptive steganography. These rules are independent of each other in the framework of minimal-distortion steganography. Some state-of-the-art steganographic methods combine several rules to promote steganographic security level. For example, CMD [18] combines the CS rule and the SMD rule by using HILL [11] as basic distortion function and then clusters the modification direction. In this subsection, we try to fuse our

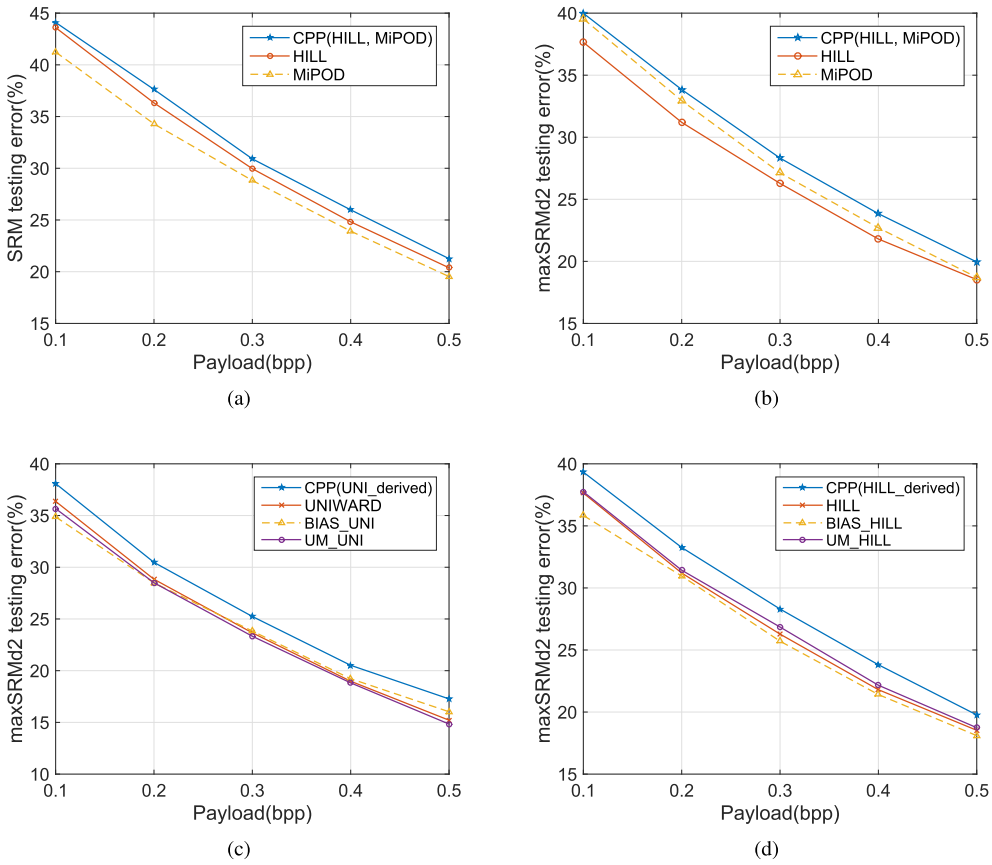


Fig. 7. Steganalytic performance of three groups of examples for CPP rule. Testing errors are obtained by steganalysis under SRM/maxSRMd2 and ensemble classifier. (a) $CPP(HILL, MiPOD)$ under SRM. (b) $CPP(HILL, MiPOD)$ under maxSRMd2. (c) $CPP(UNI\_derived)$ under maxSRMd2. (d) $CPP(HILL\_derived)$ under maxSRMd2.

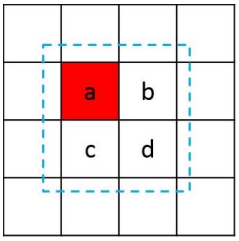


Fig. 8. Illustration of controversial pixels' block.

proposed CPP rule with CS rule and the SMD rule to obtain better performance [19].

In our proposed CPP rule, the probability variance (PV) plays an important role. To apply the CS rule, we should make a few adjustments. Instead of spreading cost to neighboring pixels, we spread the PVs of controversial pixels since PVs are strongly correlated with costs in the CPP rule.

Suppose that pixel a in Fig. 8 is a controversial one and its PV is v'_a. To smooth costs, we spread pixel a's PV to its neighbor and set

$$v''_a = v''_b = v''_c = v''_d = v'_a. \tag{29}$$

To keep the total amounts of controversial pixels invariant, we first reduce the new controversial threshold $\alpha' = \alpha/4$, and then generate new controversial pixels from the $\frac{\alpha}{4}\%$ of N pixels according to the v''_i value in Eq. (29). This step can make the controversial pixels more convergent and achieve the combination of the CS rule and CPP rule. We use $CPP(HILL, MiPOD)$ and $CPP(HILL\_derived)$ as examples, both methods are better-performing among the four CPP-based methods from the first group, $CPP(UNI, WOW)$, to the fourth group $CPP(HILL\_derived)$. We use them to test the security performance of combing the CS rule and CPP rule. The security curve and corresponding numerical values are presented in Fig. 9 and Table VII.

The SMD rule can be easily used after the CS rule and CPP rule. We just need to modify the pixels in a synchronizing direction according to the well-defined distortions by using previous rules. In this paper, we use $CPP(HILL, MiPOD)$ in Fig. 10 as an example because $CPP(HILL, MiPOD)$ has the best security performance among all four groups of examples for the CPP rule. Fig. 10(a) shows the promotion in security performance obtain by combining three rules compared to using a single CPP rule. Fig. 10(b) compares our scheme and two of the current best-performing schemes.

The promotions in security performance achieved by fusing several rules is significant, and thus we can choose different rules to generate a fusion scheme for designing distortion functions according to the security requirements.

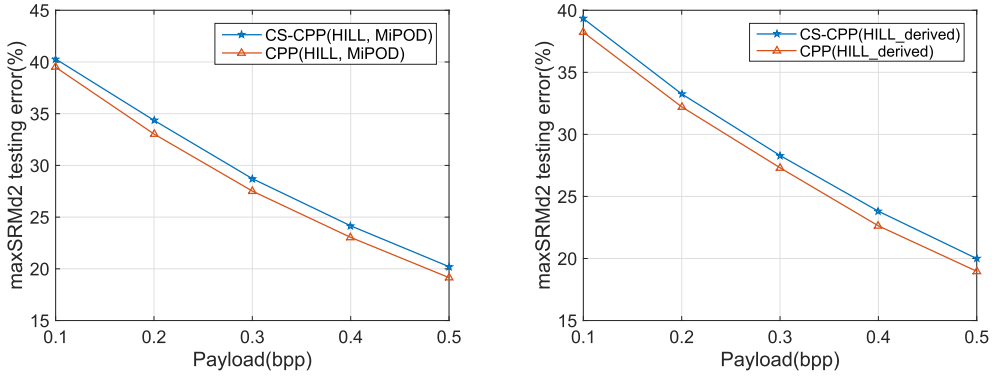


Fig. 9. Promotion of security performance after fusing the CS rule and CPP rule under the detection of maxSRM. (a) $CPP(HILL, MiPOD)$ and $CS-CPP(HILL, MiPOD)$. (b) $CPP(HILL\_derived)$ and $CS-CPP(HILL\_derived)$.

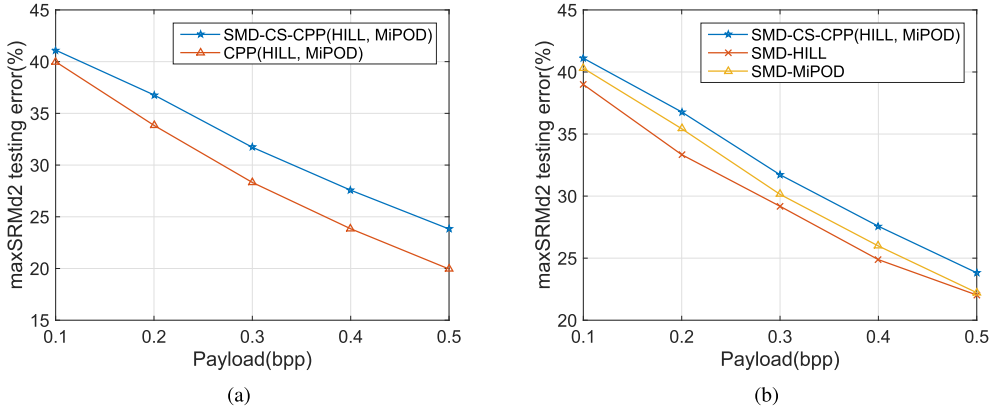


Fig. 10. Promotion of security performance after fusing the SMD rule and CPP rule under the detection of maxSRMd2. (a) $CPP(HILL, MiPOD)$ and $SMD-CS-CPP(HILL, MiPOD)$. (b) $SMD-HILL, SMD-MiPOD$ and $SMD-CS-CPP(HILL, MiPOD)$.

TABLE IX
NUMERICAL VALUES OF TESTING ERROR AND STANDARD DEVIATION ON BOSSbaseC AND BOSSbaseJ85

Database	Embedding method	Testing errors from 0.1bpp to 0.5bpp				
		0.1	0.2	0.3	0.4	0.5
BOSSbaseC	UNIWARD	.2663±.0034	.1310±.0013	.0714±.0018	.0425±.0016	.0274±.0009
	MiPOD	.2662±.0027	.1304±.0016	.0711±.0017	.0416±.0013	.0270±.0010
	HILL	.2368±.0026	.1107±.0024	.0590±.0010	.0369±.0009	.0234±.0008
	$CPP(UNI, MiPOD)$	.2864±.0042	.1427±.0012	.0828±.0021	.0477±.0008	.0314±.0008
	$CPP(HILL, MiPOD)$	.2686±.0027	.1313±.0023	.0665±.0025	.0432±.0007	.0294±.0014
BOSSbaseJ85	UNIWARD	.1401±.0017	.0766±.0014	.0472±.0013	.0308±.0010	.0201±.0012
	HILL	.1399±.0021	.0762±.0012	.0486±.0016	.0322±.0008	.0214±.0016
	MiPOD	.1217±.0016	.0674±.0014	.0431±.0013	.0296±.0013	.0191±.0011
	$CPP(UNI, HILL)$	.1640±.0022	.0911±.0016	.0572±.0027	.0381±.0009	.0249±.0009
	$CPP(HILL, MiPOD)$	.1503±.0014	.0827±.0021	.0501±.0014	.0340±.0011	.0238±.0013

F. Performances on Other Datasets

In this subsection, we execute the CPP based scheme on BOSSbaseC and BOSSbaseJ85 [27]. BOSSbaseC is obtained from BOSSbase RAW by centrally cropping its images to a size of 512×512. Images from this source are less textured but do contain acquisition noise. BOSSbaseJ85 is obtained from BOSSbase ver.1.01 by JPEG compression to quality level 85. The low-pass character of JPEG compression makes the images of BOSSbaseJ85 less textured and much less noisy. Because of the changes of textural features, the security performances for state-of-the-art steganography have changed. On BOSSbaseC, the relationship of security levels under maxSRMd2 is: UNIWARD ≈ MiPOD > HILL > WOW, where UNIWARD and MiPOD become comparable. On BOSSbaseJ85, the same relationship is: WOW > UNIWARD ≈ HILL > MiPOD, where UNIWARD and HILL comprise the comparable pair.

To implement the CPP rule, we first select candidate algorithms. As mentioned above, a principle for selecting candidate algorithms for the CPP rule is that candidate algorithms should have comparable security performances, which is also influenced by the cover database. In practice, the sender can

select candidate algorithms according to his/her cover database and design a new distortion function with the CPP rule. Note that the sender does not need to share the distortion function with the recipient owing to the advantage afforded by STC. Therefore, when applying the CPP rule, we pair UNIWARD with MiPOD on BOSSbaseC and pair HILL and UNIWARD on BOSSbaseJ85.

On the other hand, the steganalysts usually cannot have the same database as the steganographers to train classifiers which will significantly reduce the accuracy of the steganalysis due to the cover-source mismatch. However, to strictly test the security of CPP based scheme, we use the same databases as the steganographer to train classifiers. The numerical results presented in Table IX show that the CPP rule provides a promotion in steganographic security performance. Furthermore, $CPP(UNI, HILL)$ achieves a larger promotion compared to $CPP(HILL, MiPOD)$ on BOSSbaseJ85, which again verifies that using comparable basic methods for the CPP rule is important. This conclusion is also reflected in the results of $CPP(UNI, MiPOD)$ and $CPP(HILL, MiPOD)$ on BOSSbaseC.

IX. Conclusion

The most effective model for adaptive steganography is embedding messages while minimizing a carefully defined distortion function. In this paper, we summarized the previous rules for adaptive steganography and proposed the controversial pixels prior (CPP) rule to generate a new steganographic distortion function by combining pairs or groups of previous methods that have comparable performances. Experiments show that the CPP rule can improve the security of the state-of-the-art steganographic algorithms.

The CPP rule considers a combination of several existing methods instead of remaining fixed on a single method. An essential principle in selecting candidate algorithms for the CPP rule is that basic methods have comparable security performances. In addition, the CPP rule provides a novel tool for designing steganographic schemes. In previous studies, it was thought to make sense only when a new method was found that outperformed the state-of-the-art ones. Now, it also makes sense if proposed method is comparable with previous ones, because we can promote the comparable performances with the help of the CPP rule. This is the most important contribution of our work.

In the present paper, we use the CPP rule to improve steganography in spatial images with additive distortion for ± 1 embedding. In our future work, applying the CPP rule to other covers such as JPEG image, videos, and text, is an interesting direction.

Acknowledgment

The authors would like to thank DDE Laboratory of SUNY Binghamton for sharing the source code of steganography on the webpage (http://dde.binghamton.edu/download/). Specifically, they are grateful to the AE Dr. Andrew Ker for his valuable comments and helpful suggestions. They also thank the anonymous reviewers for their helpful comments.

References

[1] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[2] B. Li, J. He, J. Huang, and Y. Q. Shi, "A survey on image steganography and steganalysis," *J. Inf. Hiding Multimedia Signal Process.*, vol. 2, no. 2, pp. 142–172, 2011.

[3] T. Pevný, T. Filler, and T. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. Int. Workshop Inf. Hiding*, vol. 6387. Jun. 2010, pp. 161–177.

[4] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 215–224, Jun. 2010.

[5] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proc. IEEE Workshop Inf. Forensics Secur.*, Dec. 2012, pp. 234–239.

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

[7] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, "Selection-channel-aware rich model for steganalysis of digital images," in *Proc. 6th IEEE Int. Workshop Inf. Forensics Secur.*, Atlanta, GA, USA, Dec. 2014, pp. 48–53.

[8] W. Tang, H. Li, W. Luo, and J. Huang, "Adaptive steganalysis based on embedding probabilities of pixels," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 734–745, Apr. 2016.

[9] T. Denemark, J. Fridrich, and P. Comesaña-Alfaro, "Improving selection-channel-aware steganalysis features," in *Proc. IS&T, Electron. Imag., Media Watermarking, Secur., Forensics*, San Francisco, CA, USA, Feb. 2016, pp. 1–8.

[10] V. Holub and J. Fridrich, "Digital image steganography using universal distortion," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2013, pp. 59–68.

[11] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 4206–4210.

[12] J. Fridrich and J. Kodovský, "Multivariate Gaussian model for designing additive distortion for steganography," in *Proc. IEEE ICASSP*, Vancouver, BC, Canada, May 2013, pp. 2949–2953.

[13] B. Li, S. Tan, M. Wang, and J. Huang, "Investigation on cost assignment in spatial image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 8, pp. 1264–1277, Aug. 2014.

[14] V. Sedighi, J. Fridrich, and R. Cogranne, "Content-adaptive pentary steganography using the multivariate generalized Gaussian cover model," *Proc. SPIE*, vol. 9409, p. 94090H, Mar. 2015.

[15] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 221–234, Feb. 2016.

[16] J. Li, X. Yang, X. Liao, F. Pan, and M. Zhang, "A game-theoretic method for designing distortion function in spatial steganography," *Multimedia Tools Appl.*, vol. 76, no. 10, pp. 12417–12431, May 2017, doi: 10.1007/s11042-016-3632-7.

[17] K. Chen, W. Zhang, H. Zhou, N. Yu, and G. Feng, "Defining cost functions for adaptive steganography at the microscale," in *Proc. IEEE Workshop Inf. Forensics Secur.*, Abu Dhabi, United Arab Emirates, Dec. 2016, pp. 1–6.

[18] B. Li, M. Wang, X. Li, S. Tan, and J. Huang, "A strategy of clustering modification directions in spatial image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1905–1917, Sep. 2015.

[19] T. Denemark and J. Fridrich, "Improving steganographic security by synchronizing the selection channel," in *Proc. 3rd Workshop IH&MMSec*, Portland, OR, USA, Jun. 2015, pp. 5–14.

[20] W. Zhou, W. Zhang, and N. Yu, "Evolving distortion function by exploiting the differences among comparable adaptive steganography," in *Proc. 12th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery*, Changsha, China, Aug. 2016, pp. 1235–1244.

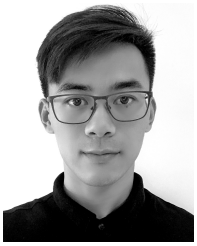
[21] T. Pevný and J. Fridrich, "Benchmarking for steganography," in *Proc. 10th Int. Workshop Inf. Hiding*, vol. 5284. May 2008, pp. 251–267.

[22] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," *Proc. SPIE Electron. Imag., Secur., Steganography Watermarking Multimedia Contents IX*, vol. 6505, pp. 0201–0215, Feb. 2007.

[23] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 705–720, Dec. 2010.

[24] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 920–935, Sep. 2011.

[25] P. Bas and T. Furon. *BOWS-2 (Break Our Watermarking System)*, accessed on Jul. 2007. [Online]. Available: http://bows2.ec-lille.fr/
[26] P. Bas, T. Filler, and T. Pevný, "'Break our steganographic system': The ins and outs of organizing BOSS," in *Proc. 13th Int. Workshop Inf. Hiding*, vol. 6958. May 2011, pp. 59–70.
[27] V. Sedighi, J. Fridrich, and R. Cogranne, "Toss that BOSSbase, Alice!" in *Proc. IS&T, Electron. Imag., Media Watermarking, Secur., Forensics*, San Francisco, CA, USA, Feb. 2016, pp. 1–9.
[28] J. Kodovský, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 432–444, Apr. 2012.
[29] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, 1998.



Wenbo Zhou received the B.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2014. He is currently pursuing the Ph.D. degree with the University of Science and Technology of China. His research interests include steganography, steganalysis, and multimedia security.

Weiming Zhang received the M.S. and Ph.D. degrees from the Zhengzhou Information Science and Technology Institute, Zhengzhou, China, in 2002 and 2005, respectively. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include multimedia security, information hiding, and privacy protection.

Nenghai Yu received the B.S. degree from the Nanjing University of Posts and Telecommunications, in 1987, the M.E. degree from Tsinghua University in 1992, and the Ph.D. degree from the University of Science and Technology of China, in 2004. He is currently a Professor with the University of Science and Technology of China. His research interests include multimedia security, multimedia information retrieval, video processing, information hiding, and security, privacy, and reliability in cloud computing.