A new cost function for spatial image steganography based on 2D-SSA and WMF.

XIE, G., REN, J., MARSHALL, S., ZHAO, H. and LI, H.

2021

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 20, 2021, accepted February 11, 2021, date of publication February 18, 2021, date of current version February 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3059690

A New Cost Function for Spatial Image Steganography Based on 2D-SSA and WMF

GUOLIANG XIE[1,2], JINCHANG REN[2,3], (Senior Member, IEEE), STEPHEN MARSHALL[1], (Senior Member, IEEE), HUIMIN ZHAO[2], AND HUIHUI LI[2]

[1]Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1XW, U.K.
[2]School of Computer Sciences, Guangdong Polytechnic Normal University, Guangzhou 510665, China
[3]National Subsea Centre, Robert Gordon University, Aberdeen AB10 7QB, U.K.

Corresponding authors: Jinchang Ren (jinchang.ren@ieee.org) and Huimin Zhao (zhaohuimin@gpnu.edu.cn)

This work was supported in part by the Dazhi Scholarship of the Guangdong Polytechnic Normal University, in part by the Key Laboratory of the Education Department of Guangdong Province under Grant 2019KSYS009, in part by the National Natural Science Foundation of China under Grant 62072122 and Grant 62006049, and in part by the University of Strathclyde Scholarship.

ABSTRACT As an essential tool for secure communications, adaptive steganography aims to communicate secret information with the least security cost. Inspired by the Ranking Priority Profile (RPP), we propose a novel two-step cost function for adaptive steganography in this paper. The RPP mainly includes three rules, i.e. Complexity-First rule, the Clustering rule and the Spreading rule, to design a cost function. We use the two-dimensional Singular Spectrum Analysis (2D-SSA) and Weighted Median Filter (WMF) in designing the two-step cost function. The 2D-SSA is employed in selecting the key components and clustering the embedding positions, which follows the Complexity-First rule and the Clustering rule. Also, we deploy the Spreading rule to smooth the resulting image produced by 2D-SSA with WMF. Extensive experiments have shown the efficacy of the proposed method, which has improved performance over four benchmarking approaches against non-shared selection channel attack. It also provides comparable performance in selection-channel-aware scenarios, where the best results are observed when the relative payload is 0.3 bpp or larger. Besides, the proposed approach is much faster than other model-based methods.

INDEX TERMS Image steganography, feature extraction, singular spectrum analysis (SSA), weighted median filtering (WMF), ranking priority profile.

I. INTRODUCTION

Steganography is a technique for embedding secret information into a cover message. It can be used for both legal and illegal purposes. For a better understanding of how this technique works, it is important to consider its different stages [1]. Steganography is mainly used in situations where a sender and a receiver need to communicate privately and security is the most important criterion.

In modern network-based communications, steganography could suffer from an active or passive attack. To ensure the security of the secret message, existing methods are to embed the secret message adaptively according to the carrier, thus it is called adaptive steganography. To generate a disguised image or stego image in image steganography, a cover image, a steganographic algorithm, some secret messages, and a digital key are required.

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

In adaptive image steganography, usually, only the cost functions should be considered by the designers, which provide the embedding cost for embedding tools, i.e. the cost of modifying pixels. With the determined embedding cost, the embedding procedure can be realized by tools such as syndrome-trellis codes [2].

In spatial image steganography, the Highly Undetectable steGO (HUGO) [3] was the first practical model to minimize additive distortion [1]. With insight taken from a steganalysis method called Subtractive Pixel Adjacency Matrix (SPAM) [4], HUGO calculates the weighted sum of differences between the feature vectors. In this way, the embedding positions focus on textural areas instead of smooth areas [5]. Next, the Wavelet Obtained Weights (WOW) was proposed, which captures the high-frequency signals in images using directional high-pass filters [6]. The WOW method is much faster than HUGO and also has better performance. The core idea of WOW was based on the assumption that large filter residuals result in high unpredictability.

Based on an improved WOW, the Spatial Universal Wavelet Relative Distortion (S-UNIWARD) method was proposed. Its cost function is defined by calculating the sum of the changes in the wavelet coefficients with respect to the cover images [7]. Meanwhile, the Multivariate Gaussian model (MG) was proposed [8], of which cost function is defined as an approximation of the Kullback-Leibler (KL) divergence between the cover and stego images [9]. MG produces better performance than HUGO when the relative payload is higher than 0.3 bit per pixel. In 2014, Li *et al.* proposed the use of a smooth filter residual to replace the weighted filter residual in determining the embedding suitability, thus creating the HILL (HIgh-pass, Low-pass, Low-pass) model [10]. With a high-pass filter to extract the high-frequency pixels and two low-pass filters to cluster the low-cost pixels, HILL provides better performance than all the methods aforementioned. In 2015, MiPOD (Minimizing the Power of Optimal Detector), based on the theory that natural images follow a joint Gaussian distribution, was proposed [11]. By using the Wiener filter to process the cover image, it divides the filtered image into multiple blocks where the variance of each block is calculated by Maximum Likelihood Estimation. Lastly, the cost can be determined by the estimated variance and it has been shown that the results of MIPOD are comparable to that of HILL.

In 2018, Hu *et al.* proposed to use Nonnegative Matrix Factorization (NMF) to design the cost function [12]. Based on the assumption that pixels in natural images are mutually dependent, the costs of a pixel and its neighbouring pixels can be determined. In 2019, Qin *et al.* improved the MG model by introducing image filter residuals [13]. The noise variance is estimated by using a neighbouring estimation, thus it is more efficient than MiPOD. Recently, a new way to explore interactions among local pixels was proposed, which is based on the Gaussian Markov Random Field model (GMRF) with four-element cross neighbourhood [14].

We followed the rules of Ranking Priority Profile proposed in [5] to design our scheme, which includes the Complexity-First rule, Spreading rule and Clustering rule. Our scheme is a generalization of existing schemes such as HUGO, WOW and S-UNIWARD incorporating the Spreading rule and the Clustering in addition to the Complexity rule. The Complexity-First rule requires that a complex area should be assigned with high priority or low cost in the embedding process. The Spreading rule requires that a pixel that is assigned with high priority should spread its importance to its neighbourhood, and vice versa. The Clustering rule states that the modifications should be clustered instead of scattered.

Our proposed two-step cost-scheme is described below. We first use the 2D-SSA [15] to automatically select the components in the cover image, following the complexity-first rule. Then, the WMF is applied to cluster the embedding positions [16]. Both the spreading rule and clustering rule are used in selecting the parameters for 2D-SSA and WMF. Comprehensive experiments are conducted to validate the efficacy of the proposed method when compared with several bench-marking approaches.

II. PRELIMINARIES AND PREVIOUS WORK

Let X and Y denote respectively an 8-bit grey cover image and its stego image. $X = (X_{ij})$, $Y = (Y_{ij}) \in \{0, \ldots, 255\}^{n_1 \times n_2}$, where i and j are the indexes of the pixel, and n_1, n_2 denote the width and length of the image. We consider the case of ternary embedding, where the possible value of stego images are restricted to $\{X_{ij}, max(X_{ij} - 1, 0), min(X_{ij} + 1, 255)\}$.

A. ADDITIVE DISTORTION IN STCs

Define β_{ij} as the change rate for the pixel X_{ij}, the maximal expected payload R that can be sent by the sender is the entropy of the introduced modification [17],

$$R(\beta) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} H(\beta_{ij}) \tag{1}$$

where $H(x) = -2x \log x - (1 - 2x) \log(1 - 2x)$ denotes the ternary entropy function [11].

As the embedding operations are assumed to be mutually independent, a distortion function $D(X, Y)$ introduced by the sender can be designed in an additive form, namely the additive distortion function [2].

$$D(X, Y) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \rho_{ij}(X_{ij}, Y_{ij}) \left| X_{ij} - Y_{ij} \right| \tag{2}$$

where $\rho_{ij} \geq 0$ denotes the cost or the security expenditure of changing the pixel value from X_{ij} to Y_{ij} [5]. With the determined embedding cost, the sender can designate the pixels for embedding with a probability β_{ij}:

$$\beta_{ij} = \frac{e^{-\lambda \rho_{ij}}}{1 + 2e^{-\lambda \rho_{ij}}} \tag{3}$$

where $\lambda > 0$ is determined from the payload constraint (3). Finally, we can use some near-optimal coding schemes, e.g. syndrome-trellis codes (STCs) [17], [18], to complete the embedding work with β_{ij}.

B. CONVENTIONAL 2D-SSA ANALYSIS

Singular Spectrum Analysis (SSA) can be used to decompose a 1-D signal into low-frequency components of the trend, oscillations, and noise [19]. Recently, 2D-SSA was found effective for smoothing images and feature extraction in hyperspectral images [15], [19]. As our datasets are limited to images, only the 2D-SSA will be studied in this paper.

In 2D-SSA, an input image X sized $n_1 \times n_2$ and a window with a dimension $B = u \times v$ are defined, where $u \in [1, n_1]$ and $v \in [1, n_2]$. A trajectory matrix $Q \in \Re^{B \times C}$ is constructed from the image X, where $C = (n_1 - u + 1)(n_2 - v + 1)$. Next, a Singular Value Decomposition (SVD) is applied to Q, which is equivalent to an eigenvalue decomposition of $Q \cdot Q^T$. As a result, the eigenvalues $(\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_B)$ and the associated eigenvectors $\Omega = (\omega_1, \omega_2, \ldots, \omega_B)$ can

be derived. The matrix Q can be written as $Q = \sum Q_\varphi$, $\varphi = [1, B]$, where each submatrix Q_φ is defined by:

$$Q_\varphi = \sqrt{\lambda_\varphi}\omega_\varphi z_\varphi^T \tag{4}$$

$$z_\varphi = \frac{Q^T \omega_\varphi}{\sqrt{\lambda_\varphi}} \tag{5}$$

We can project the elements from Q_φ to G_φ using grouping and diagonal averaging, where G_φ is the decomposed components of X. Following 2D-SSA based decomposition, we have $X = \sum G_\varphi$, which means the image X is decomposed into several matrices that represent different components. Ultimately, we can reconstruct a new $X^* = \sum G_m$ by using the designating components G_m as discussed in detail in Section III-B.

C. WEIGHTED MEDIAN FILTER (WMF)

The WMF is an extension to the classical median filter, which belongs to a broad class of non-linear filters called stack filters. The advantages of WMF are the efficiency in noise attenuation and the robustness against impulsive noise [20]. Edge preserving is essential in designing adaptive steganography, which is the key for the algorithms to focus on the complex area in the image. Also, this filter is important in applying the Spreading rule for our proposed scheme, and the detail can be found in Section IV-B2, where the suppression of impulsive noise is shown in Fig. 5.

III. THE PROPOSED NEW COST FUNCTION BASED ON 2D-SSA AND WMF

In this section, we will detail the implementation of our proposed two-step cost-assignment scheme. We will first describe our proposed steganographic framework in subsection III-A, and then the usage of the 2D-SSA will be discussed in subsection III-B, while we provide the principle of the WMF in subsection III-C. The proposed cost function is presented in subsection III-D.

A. THE PROPOSED 2DSSA-WMF BASED STEGANOGRAPHIC FRAMEWORK

As shown in Fig. 1, in steganography, the sender uses a cover image and then determines the embedding positions in this image, which is equivalent to assigning costs to pixels. Following this process, a coding method, i.e. STC, is used to embed the secret message and hence stego image is created. The enhancement to the security of steganography, introduced in this paper, is focused on the cost assignment stage.

Although most existing methods use the high-frequency part of the image to embed the secret message, such as WOW, S-UNI and HILL, the absence of a detailed analysis of these high-frequency contents may lead to poor performance against the Spatial Rich Model (SRM) attacks. The enhancement to the security of steganography, introduced in this paper, is focused on the cost assignment stage.

The 2D-SSA algorithm has two advantages, the first is that it allows us to select different high-frequency

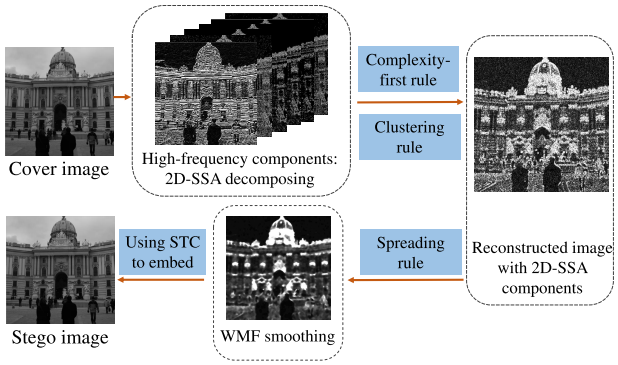


FIGURE 1. The proposed steganographic framework.

contents or complex areas, which meets the requirement of Complexity-First rule. Secondly, we found that high-frequency components were clustered if we use the least-important portions in SSA to reconstruct the image, i.e. the 8th and the 9th component in a 3×3 window. Lastly, as WMF is a kind of median filter, it can help to smooth the images or spreads the weights just as the Spreading rule requires. To this end, these tools thus comply with the rules of Ranking Priority Profile.

B. 2D-SSA BASED DECOMPOSITION OF THE COVER IMAGE

Applying 2D-SSA on a 2D-signal requires four steps, namely embedding, SVD, grouping, and diagonal averaging. Note that the embedding process in 2D-SSA is different from that in image steganography, though they share the same terminology.

1) EMBEDDING

For the image X, its matrix representation is shown as (6).

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n_2} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n_2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n_1,1} & x_{n_1,2} & \cdots & x_{n_1,n_2} \end{pmatrix} \tag{6}$$

A set of 2D-windows W_{ij} is used to calculate the trajectory matrix of X. These 2D-windows are a series of submatrices in the image X with a size $u \times v$ ($u \in [1, n_1]$,$v \in [1, n_2]$). The structure of these 2D-windows is shown in (7).

$$W_{i,j} = \begin{pmatrix} x_{i,j} & x_{i,j+1} & \cdots & x_{i,j+v-1} \\ x_{i+1,j} & x_{i+1,j+1} & \cdots & x_{i+1,j+v-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i+u-1,j} & x_{i+u-1,j+1} & \cdots & x_{i+u-1,j+v-1} \end{pmatrix} \tag{7}$$

Meanwhile, these 2D-windows can also be represented as (8), where r is within $[1, u]$:

$$W_{i,j} = \begin{pmatrix} w_{(i,j)_1} \\ w_{(i,j)_2} \\ \vdots \\ w_{(i,j)_u} \end{pmatrix}, \quad w_{(i,j)_r} = \begin{pmatrix} x_{i+r-1,j} \\ x_{i+r-1,j+1} \\ \vdots \\ x_{i+r-1,j+v-1} \end{pmatrix} \tag{8}$$

For a given pixel (i, j), the corresponding 2D-window can be rearranged into a column vector as:

$$A_{i,j} = \begin{pmatrix} w^T_{(i,j)_1} \\ w^T_{(i,j)_2} \\ \vdots \\ w^T_{(i,j)_u} \end{pmatrix} = \begin{pmatrix} x_{i,j} \\ x_{i,j+1} \\ \vdots \\ x_{i,j+v-1} \\ x_{i+1,j} \\ \vdots \\ x_{i+u-1,j+v-1} \end{pmatrix} \in \Re^{uv} \tag{9}$$

Now, the trajectory matrix Q can be derived as follows:

$$Q = \begin{pmatrix} A^T_{1,1} \\ A^T_{1,2} \\ \vdots \\ A^T_{1,n_2-v+1} \\ A^T_{2,1} \\ \vdots \\ A^T_{n_1-u+1,n_2-v+1} \end{pmatrix} \in \Re^{uv\times(n_1-u+1)(n_2-v+1)} \tag{10}$$

Note that Q is a Hankel-block-Hankel (HbH) matrix, which can be written as:

$$Q = \begin{pmatrix} H_1 & H_2 & \dots & H_{n_1-u+1} \\ H_2 & H_3 & \dots & H_{n_1-u+2} \\ \vdots & \vdots & \ddots & \vdots \\ H_u & H_{u+1} & \dots & H_{n_1} \end{pmatrix}_{u\times(n_1-u+1)} \tag{11}$$

And each submatrix H_r is a strict Hankel type matrix (12).

$$H_r = \begin{pmatrix} x_{r,1} & x_{r,1} & \dots & x_{r,n_2-v+1} \\ x_{r,2} & x_{r,3} & \dots & x_{r,n_2-v+2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{r,v} & x_{r,v+1} & \dots & x_{r,n_2} \end{pmatrix}_{v\times(n_2-v+1)} \tag{12}$$

2) SINGULAR VALUE DECOMPOSITION

Applying SVD to Q is equivalent to an eigenvalue decomposition (EVD) of $Q \cdot Q^T$. In this way, we can obtain eigenvalues $(\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_B)$ and the associated eigenvectors $\Omega = (\omega_1, \omega_2, \dots, \omega_B)$. We can rewrite Q as the sum of matrices (13). And each of these matrices can be calculated by using (4) and (5).

$$Q = Q_1 + Q_2 + \cdots + Q_B \tag{13}$$

3) GROUPING

Next, a total set of B components is divided into M disjoint sets $t_1, t_2, \cdots, t_M$ and $\sum |t_m| = B, m \in [1, M]$. Hence, the trajectory matrix Q becomes (14). A typical grouping is when $M = B$, which means each set is made of one component.

$$Q = Q_{t_1} + Q_{t_2} + \cdots + Q_{t_M} \tag{14}$$

4) DIAGONAL AVERAGING

According to [15], the matrices Q_{t_m} obtained by grouping do not necessarily have a HbH structure. Hence, a diagonal averaging process is needed, which means we need to hankelize within each block (12) and between these blocks (11). Diagonal averaging means obtaining the average in all the anti-diagonals of each Q_{t_m}.

Let $g_m = [g_{m_1}, g_{m_2}, \cdots, g_{m_{n_2}}] \in \Re^{n_2}$ denote a row of pixels projected from Q_{t_m}, then diagonal averaging can be described in (15), where $a_{r,\theta-r+1}$ refers to the elements in Q_{t_m} and K as $K = n_2 - B + 1$.

$$g_{m_\theta} = \begin{cases} \frac{1}{\theta}\sum_{r=1}^{\theta} a_{r,\theta-r+1}, & 1 \leq \theta < B \\ \frac{1}{B}\sum_{r=1}^{B} a_{r,\theta-r+1}, & B \leq \theta < K \\ \frac{1}{n_2-\theta+1}\sum_{r=\theta-K+1}^{B} a_{r,\theta-r+1}, & K \leq \theta < n_2 \end{cases} \tag{15}$$

Denote $g_{m_{i,j}}$ as the elements projected from Q_{t_m}, we can get the projected matrix G_m as:

$$G_m = \begin{pmatrix} g_{m_{1,1}} & g_{m_{1,2}} & \cdots & g_{m_{1,n_2}} \\ g_{m_{2,1}} & g_{m_{2,2}} & \cdots & g_{m_{2,n_2}} \\ \vdots & \vdots & \ddots & \vdots \\ g_{m_{n_1,1}} & g_{m_{n_1,2}} & \cdots & g_{m_{n_1,n_2}} \end{pmatrix} \tag{16}$$

Now, we can rewrite the input image X below, and each $G_m \in \Re^{n_1\times n_2}$

$$X = G_1 + G_2 + \cdots + G_M = \sum_{m=1}^{M} G_m \tag{17}$$

That means we can extract the desired components from G_m based on SVD or eigenvalues to reconstruct a new image $X^* = \sum G_m$ for our purposes.

C. WMF BASED SMOOTHING

The weighted median filter (WMF) is a type of non-linear filter that processes pixels by replacing them with their neighbouring pixels [16]. Let p denote a pixel in the image X, and $L(p)$ denote the local window of radius γ centred at p. For each pixel $q \in L(p)$, WMF associates it with a weight α_{pq} based on the affinity of the pixel p and q in the corresponding feature map f, where $f(p)$ and $f(q)$ are the features, which can be intensity, color etc. $\eta(\cdot)$ is a function that determines how p is influenced by its neighboring pixels. In this paper, we used intensity as $f(\cdot)$ and a Gaussian function $exp\{-|f(p) - f(q)|^2/(2 * \sigma^2)\}$ as $\eta(\cdot)$.

$$\alpha_{pq} = \eta(f(p), f(q)) \tag{18}$$

Let $N = (2\gamma + 1)^2$ denote the number of pixels in $L(p)$, and $X(p)$ denote the intensity of p in X. The pixels in its local window are sorted by WMF into ascending order and $X(p)$ is replaced by a new intensity $X(p^*)$. This process can be

described as (19).

$$p^* = \min \kappa, \text{ s.t.} \sum_{q=1}^{\kappa} \alpha_{pq} \geq 1/2 \sum_{q=1}^{N} \alpha_{pq} \tag{19}$$

This process is denoted as $\Gamma(\gamma, \sigma, \tau)$, where σ is the standard deviation of the Gaussian kernel. The process will repeat τ times to meet our requirements.

D. THE PROPOSED 2DSSA-WMF COST FUNCTION

The motivation of our method is that we have identified classic tools that may closely fit Li's Ranking Priority Profile [5]. The proposed new cost function is detailed as follows: we first calculate the embedding suitability matrix ζ below by using the 2D-SSA function, where s and t stand for the starting component and ending component, respectively. Next, these pixels are filtered using WMF $\Gamma(\gamma, \sigma, \tau)$, ultimately creating the cost ρ, where $\epsilon = e^{-10}$ is used to prevent infinity.

$$\zeta = \left| \sum_{m=s}^{t} G_m \right|, 1 \leq s \leq t \leq M \tag{20}$$

$$\rho = \frac{1}{\Gamma(\zeta) + \epsilon} \tag{21}$$

The pseudo-code for our algorithm is shown in Algorithm 1. After defining the cost function, we can now combine it with the STC tool and show the whole framework in Algorithm 2.

Algorithm 1 Proposed 2DSSA-WMF cost function.

Input: cover image X with size $n_1 \times n_2$; Parameters for 2D-SSA: window width and height u, v, starting component s, ending component t. Parameters for WMF: window radius γ, weight σ, iteration τ;
Output: cost ρ;
// Embedding in 2D-SSA
$n = 1$;
for $i = 1$ to $n_1 - u + 1$ **do**
  for $j = 1$ to $n_2 - v + 1$ **do**
    $T \leftarrow X(i : i + u - 1, j : j + v - 1)$;
    $Q(:, n) \leftarrow$ transpose and vectorize T;
    $n \leftarrow n + 1$;
  end for
end for
// EVD
$S \leftarrow Q * Q^T$
$[\Omega,\lambda] \leftarrow eigs(S, t)$, where $eigs()$ is the EVD function
$V \leftarrow Q^T * \Omega$
// Grouping
$\Phi \leftarrow \Omega(:, s : t) * V^T(s : t, :)$
// Diagonal averaging
$\zeta \leftarrow hankel(\Phi, u, v, s, t)$, where $hankel()$ is the Hankelization function as in (15);
// WMF filtering $\Gamma()$
Initialize Gaussian kernel histogram $\mathcal{H}$ with σ;
for $l = 1$ to τ **do**
  for $i = 1$ to n_1 **do**
    for $j = 1$ to n_2 **do**
      for $k = -\gamma$ to γ **do**
        Remove $\zeta_{i+k,j-\gamma-1}$ from $\mathcal{H}$
        Add $\zeta_{i+k,j+\gamma}$ to $\mathcal{H}$
      end for
      $\zeta_{i,j} \leftarrow \text{median}(\mathcal{H})$
    end for
  end for
end for
$\rho \leftarrow 1/(\zeta + \epsilon)$
Return ρ;

IV. EXPERIMENTS

In this section, we show the common settings for the benchmarking methods, the dataset, the steganalysis tools and the evaluation methods in Section IV-A. The parameters mentioned in Algorithm 1 are discussed in Section IV-B. Next, the performance comparison of different methods is given in Section IV-C. To test our method against a CNN model, we show the results in IV-D. Finally, running time comparisons are shown in Section IV-E.

A. EXPERIMENTAL SETTINGS

Our experiments are carried out on the BOSSbase 1.01 dataset [21], which contains 10,000 grayscale images with a size of 512 × 512 pixels each. The feature extractors used in our experiments are the Spatial Rich Model (SRM) [22] and the Threshold Local Binary Pattern (TLBP) [23]. We also used maxSRMd2 tool to test the performance when the embedding probability of each cover element, i.e. the selection-aware-channel, was shared [24]. The extracted features are trained in binary classifiers using the Fisher Linear Discriminant ensemble with the default settings [25].

The benchmarking methods that we used for comparisons are HUGO-BD [3], WOW [6], S-UNIWARD (SUNI) [7], HILL [10], and MiPOD [11]. The reason why we selected these methods is that they are widely used in the most recent image steganalysis works, i.e. [26], [27] and [28]. Note that the default settings are used in all these steganographic algorithms. The detectability is evaluated using the minimal total probability of error P_E (22), where P_{FA} and P_{MD} stand for false-alarm rate and missed-detection rate, respectively [26]. Each experiment has 5,000 cover images and 5,000 stego images, and we report the average error rate after repeating 10 times. Steganographic methods are used against steganalysis attacks, therefore, the higher the P_E the more secure the steganographic method.

$$P_E = \min(P_{FA} + P_{MD})/2 \tag{22}$$

B. PARAMETER ANALYSIS

There are several tuning parameters in our proposed method. To decide the best parameter set, we design an experiment by using a subset of 5,000 cover images randomly selected from the BOSSbase 1.01 dataset. Firstly, we created 5,000 stego

Algorithm 2 Proposed 2DSSA-WMF image steganography method.

Input: cover image X, payload π; Parameters for 2D-SSA: window width and height u, v, starting component s, ending component t. Parameters for WMF: window radius γ, weight σ, iteration τ;
Output: stego image Y;
1: Using 2D-SSA to decompose the cover image X into different components with a window sized (u, v), and reconstruct a new image X^* with the desired components (s, t)
2: Using WMF $\Gamma(\gamma, \sigma, \tau)$ to smooth the elements in X^*, and obtain the cost ρ;
3: Embedding X using STC with ρ and payload π;
4: **Return** Y;

TABLE 1. Detection error P_E for different 2D-SSA settings, with the WMF parameters set to $\gamma = 5, \sigma = 3, \tau = 2$.

2D-SSA parameters u, v, s, t	P_E
3,3,2,9	0.1328
3,3,4,9	0.1898
3,3,6,9	0.2419
3,3,7,9	0.2543
3,3,8,9	**0.2572**
3,3,9,9	0.2437
3,3,6,8	0.2410
3,3,7,8	0.2482

TABLE 2. Detection P_E for different WMF settings, with 2D-SSA parameters: $u = v = 3, s = 8, t = 9$.

window radiusγ	weight σ	iteration τ	P_E
5	3	1	0.2549
5	3	2	**0.2572**
5	3	3	0.2555
5	1	2	0.2446
5	5	2	0.2537
5	7	2	0.2522
3	1	2	0.2246
3	3	2	**0.2425**
3	5	2	0.2400
1	1	2	0.1430
1	3	2	0.1623
1	5	2	**0.1638**
7	1	2	0.2477
7	3	2	**0.2542**
7	5	2	0.2553

images by using Algorithm 2 with these cover images. Next, we used SRM as the feature extractor and Ensemble Classifier as the detector [25]. The results produced under different parameter settings are given in Tables 1 and 2 with a payload at 0.4 bpp.

1) PARAMETERS FOR 2D-SSA

There are four parameters to tune in 2D-SSA, i.e. the height, u, and the width, v, of the 2D-window for embedding, the starting component, s, and the ending component, t, for reconstruction. We first set $u = v = 3$ and compare different combinations of s and t. We select a classic image 1013.pgm from the BOSSbase 1.01 dataset as a particular example to illustrate the differences as this image contains different

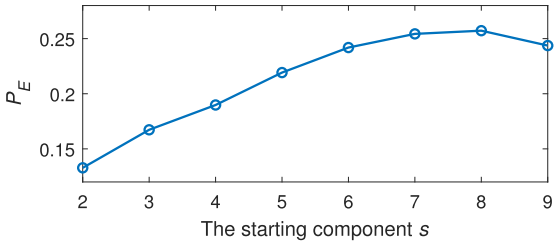


FIGURE 2. Detection error P_E for different starting component s in 2D-SSA, with $t = 9$.

kinds of edges, including, horizontal, vertical and diagonal edges. Theoretically, $s = 1$ corresponds to the low-frequency component, and that is because $\lambda_1 \gg \lambda_2 > \ldots > \lambda_B$ in SVD. We would not consider these low-frequency areas in the images, as embedding in these areas is highly detectable.

Besides, not all the high-frequency areas are useful. As shown in Fig. 3, the image on the top contains all the high-frequency areas produced by $u = v = 3$, i.e. $s = 2$ and $t = 9$, while the bottom one contains only the last two high-frequency components, i.e. $s = 8$ and $t = 9$. Those images were created by the 'im2bw' function in MATLAB. The top image actually contains all the detailed information in the cover image, which includes edges for straight lines and curves. We progressively increase the starting component of s to check how many components would provide the best performance. The results are reported in Table 1 and Fig. 2. The last two rows of Table 1 indicate that the ninth component is a key component and the sixth component may provide countereffects to the performance.

As seen in Table 1, the setting with $s = 8$ and $t = 9$ achieves the best result while the setting $s = 2, t = 9$ is the worst. This can be explained using the illustrated embedding positions in Fig. 4, where the top image is the cover image X and the other two images show the differences after embedding, i.e. |Y-X|. As shown in the middle image, both the horizontal and vertical lines were used for embedding, which is easily captured by SRM. However, in the bottom image with $s = 8$ and $t = 9$, those horizontal and vertical lines were de-emphasized and the embedding areas were clustered (red-rectangle areas), which provides improved security performance.

When only the last component ($s = t = 9$) is used, the security performance drops due to one important high-frequency component being omitted. We also tested different combinations of two non-continues components, i.e. 7th and 9th, none of these provide better performance. Note that a larger window size in 2D-SSA is not recommended as no improvement is found, though it takes 20% more time to process each image when the window size is increased from 3×3 to 5×5.

2) PARAMETERS FOR WMF

The WMF has three parameters, namely, the window radius γ, the weight σ and the number of iterations τ. We fix the parameters of 2D-SSA and vary the parameters of WMF,

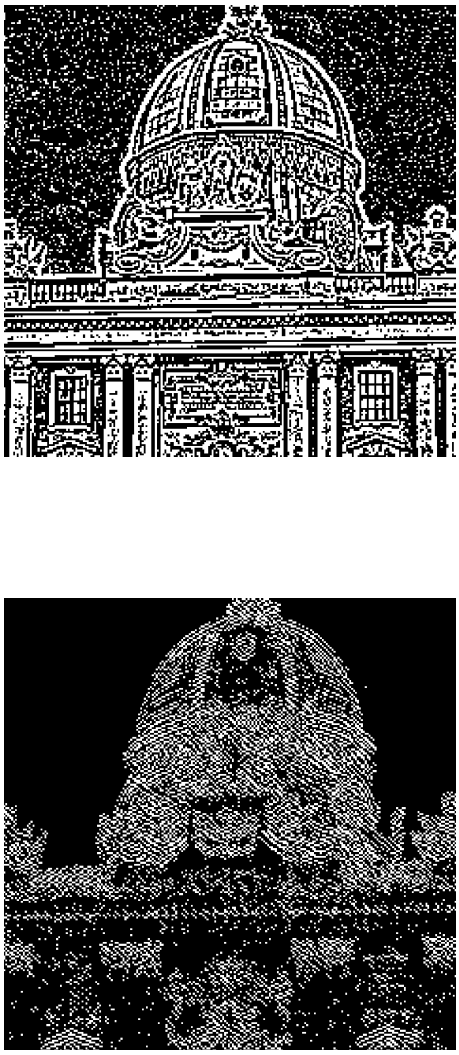


FIGURE 3. Picking all high-frequency components $s = 2, t = 9$ (top) and part of them $s = 8, t = 9$ (bottom) by 2D-SSA ($u = v = 3$).

and the results are shown in Table 2. There are four sections in Table 2, and each section corresponds to a different value of γ. The window radius γ controls how many pixels are considered when smoothing the image. With a fixed γ, we can see the impacts from the iteration τ and the weight σ.

The first six rows show the results with $\gamma = 5$. As seen in rows 1 to 3, the iteration τ did make a difference, where the performance is slightly increased when $\tau \geq 2$. To explain this, we show the cover image and the two stego images in Fig. 5, where the middle one shows the embedding signal in the low-frequency areas. These areas should be the high-cost regions to embed, as pixel values do not change dramatically. This can also be explained using the Spreading rule [5], as the cost of pixels in these areas is high, and thus the pixels inside the red-rectangles should be assigned with a high cost. However, when the iteration is 2 or more, this phenomenon disappears as the cost would be weighted by $\Gamma(\cdot)$ again. No further improvement is found when the iteration is larger

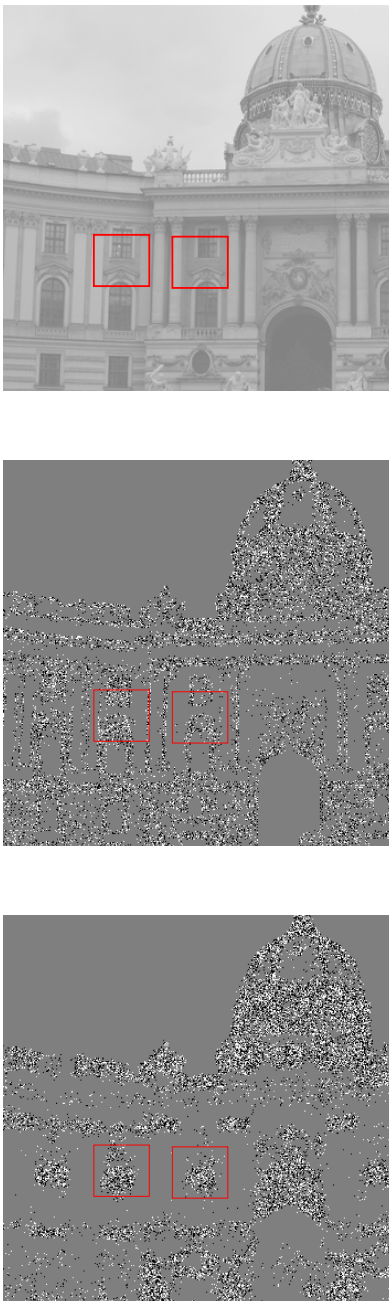
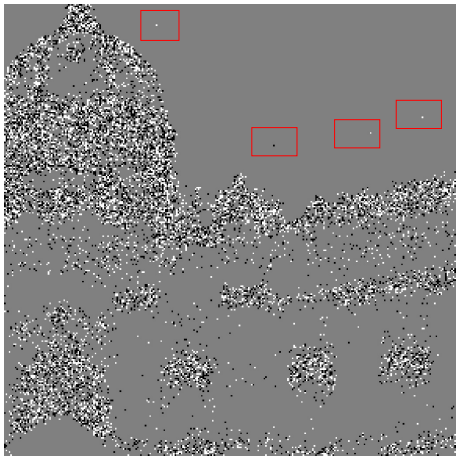


FIGURE 4. Embedding with different numbers of high-frequency components ($\gamma = 5, \sigma = 3, \tau = 2$): with all high-frequency components $s = 2, t = 9$ (middle) and with part of them $s = 8, t = 9$ (bottom, clustered in the red-rectangle areas).

than 2. Lastly, increasing the weight σ would not improve the result for $\gamma = 5$, but a pattern similar to Fig. 5 (middle) is found when $\sigma < 1$.

For the next three rows, we show the results of $\gamma = 3, \tau = 2$. The iteration τ is set to 2 to prevent a similar pattern in Fig. 5. We can see the results are slightly improved when the weight σ is increased to 3. From the tenth row to the twelfth row, we see the worst results in this table. However, when the γ is increased to 7, no improvement can be found when compared to $\gamma = 5$. That is because with the window



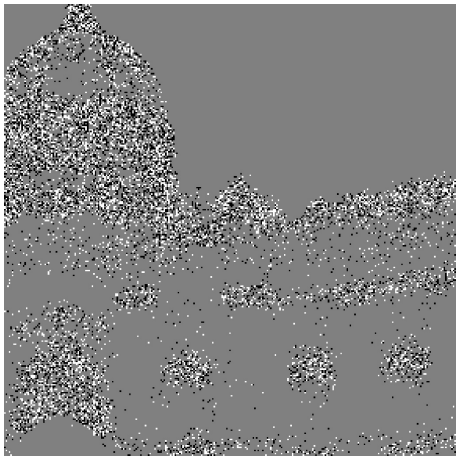


FIGURE 5. Using WMF to remove impulse noise while preserving edges ($\gamma = 5, \sigma = 3$): cover image (top), stego image with $\tau = 1$ (mid) and $\tau = 2$ (bottom).

radius γ increasing, the image will be less clustered. We show this effect in Fig. 6, from which we can see the embedding positions spreading to the low-frequency area caused by a large γ of 15.

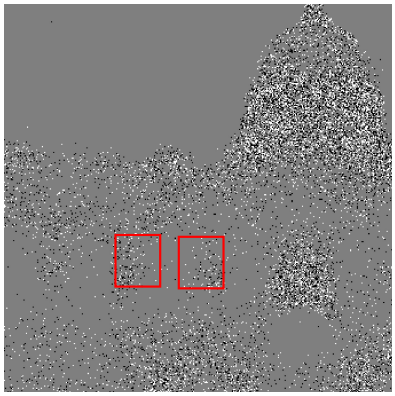


FIGURE 6. The embedding pixels are much scattered with a larger γ of WMF ($\gamma = 15, \sigma = 3$ and $\tau = 2$).

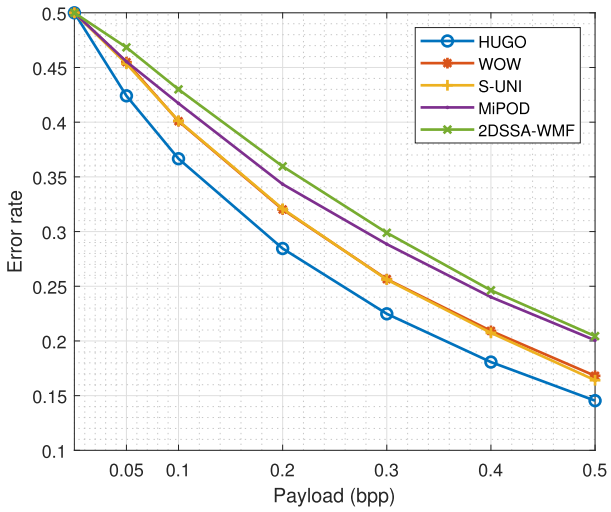


FIGURE 7. Steganalytic performance using SRM on the BOSSbase dataset.

C. COMPARING WITH OTHER BENCHMARKING METHODS

According to Section IV-B, we have selected the optimum parameter set, i.e. $u = v = 3, s = 8, t = 9; \gamma = 5; \sigma = 3$ and $\tau = 2$. With this setting, we carried out experiments on the whole BOSSbase 1.01 dataset and the results are shown in Table 3 to Table 5.

Table 3 shows the mean error rates of different steganographic algorithms and the standard deviations against SRM-based steganalysis. As can be seen, the proposed method always produces the best results in terms of detectability under different payloads, which indicates its effectiveness in defending the SRM attack. Besides, at an extremely low payload, such as 0.05 bpp, our method achieves a much better result than all other benchmarking approaches. At payloads of 0.1 to 0.3, our method provides much better performance than HUGO, WOW and S-UNI. At relatively higher payloads, i.e. 0.4 to 0.5 bpp, our method is slightly better than MIPOD.

To test the detectability against a recent steganalysis tool, TLBP, we carried out the experiments and report the results in Table 4. The advance in TLBP is that it uses the classic

TABLE 3. Detectability of different steganographic methods under various payloads against SRM with ensemble classifier.

Steganography	0.05	0.1	0.2	0.3	0.4	0.5
HUGO	0.4241 ± 0.0025	0.3666 ± 0.0036	0.2845 ± 0.0022	0.2248 ± 0.0021	0.1807 ± 0.0025	0.1454 ± 0.0013
WOW	0.4549 ± 0.0026	0.4009 ± 0.0022	0.3202 ± 0.0036	0.2565 ± 0.0014	0.2092 ± 0.0014	0.1681 ± 0.0014
S-UNI	0.4534 ± 0.0031	0.4013 ± 0.0032	0.3208 ± 0.0022	0.2561 ± 0.0015	0.2074 ± 0.0026	0.1641 ± 0.0027
MiPOD	0.4554 ± 0.0021	0.4172 ± 0.0041	0.3433 ± 0.0021	0.2885 ± 0.0026	0.2401 ± 0.0021	0.2006 ± 0.0025
2DSSA-WMF	**0.4684 ± 0.0017**	**0.4300 ± 0.0030**	**0.3596 ± 0.0030**	**0.2989 ± 0.0020**	**0.2463 ± 0.0021**	**0.2044 ± 0.0024**

TABLE 4. Detectability of different steganographic methods under various payloads against TLBP with ensemble classifier.

Steganography	0.05	0.1	0.2	0.3	0.4	0.5
HUGO	0.4326 ± 0.0025	0.3768 ± 0.0017	0.3050 ± 0.0026	0.2455 ± 0.0015	0.2014 ± 0.0021	0.1625 ± 0.0035
WOW	0.4476 ± 0.0022	0.3957 ± 0.0033	0.3202 ± 0.0040	0.2591 ± 0.0039	0.2081 ± 0.0033	0.1692 ± 0.0014
S-UNI	0.4565 ± 0.0026	0.4079 ± 0.0017	0.3239 ± 0.0020	0.2497 ± 0.0030	0.1913 ± 0.0024	0.1461 ± 0.0019
MiPOD	0.4547 ± 0.0026	0.4091 ± 0.0023	0.3384 ± 0.0025	0.2787 ± 0.0020	0.2288 ± 0.0025	0.1881 ± 0.0035
2DSSA-WMF	**0.4619 ± 0.0020**	**0.4166 ± 0.0019**	**0.3465 ± 0.0027**	**0.2881 ± 0.0034**	**0.2371 ± 0.0022**	**0.1952 ± 0.0014**

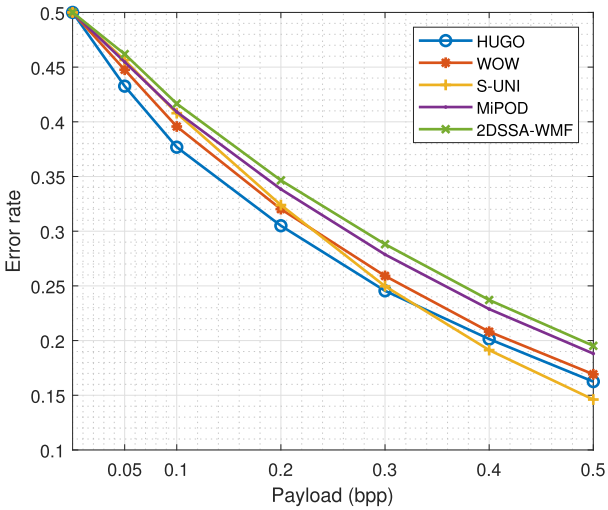


FIGURE 8. Steganalytic performance using TLBP on the BOSSbase dataset.

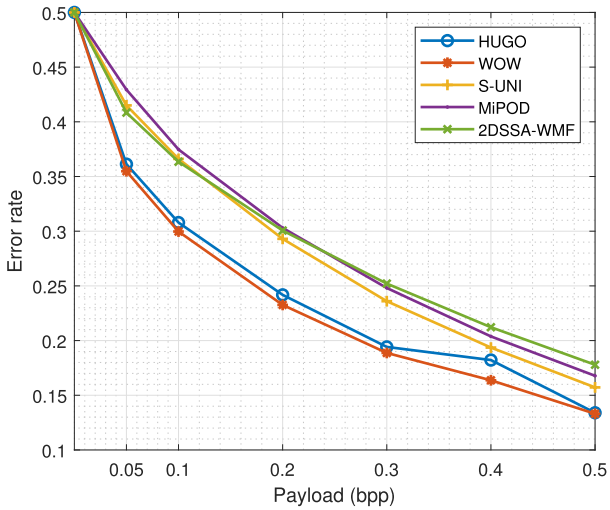


FIGURE 9. Steganalytic performance using maxSRMd2 on the BOSSbase dataset.

Local Binary Pattern method to boost the classification performance. From Fig. 8, we noticed that S-UNI was the least secure method when the payload was 0.5 bpp, under the detection of TLBP. However, the proposed method again achieves the best results under different payloads in Fig. 8.

In the last two experiments, we show the security performance of different methods in the "non-shared selection channel" scenarios. We also tested the security performance in the case where the embedding probability of each cover element is shared, i.e. the selection-channel-aware scenario. We used the maxSRMd2 [24] and the ensemble classifier for the experiments and the results are given in Fig. 9 and Table 5. From Fig. 9, we noticed that our method achieves the best results when the payload is 0.3 bpp or larger. For the payload 0.2 bpp, the result is close to that of MIPOD. When the payload is lower than 0.2 bpp, our method has a performance similar to that of S-UNI.

D. PERFORMANCE AGAINST CONVOLUTIONAL NERUAL NETWORK

In addition to the conventional steganalysis attacks, we also used a well-known CNN model, Xu-Net [29], to attack our steganographic model. In this experiment, randomly selected 4,000 pairs of images were used for training, 1,000 pairs were used for validating and the remaining 5,000 pairs were used for testing. For each steganographic method, the network was trained and tested on the specific payload-dataset only, and no transfer learning was used. We show the error rates in Table 6.

From Table 6, we can see that our method achieves the best result under the payload 0.4 bpp, which is about 3% better than the other methods. For payload 0.1 bpp, although HILL, S-UNI and 2DSSA-WMF provide similar performance, our method is the second-best.

E. COMPARISON OF COMPUTATION TIME

We further compare in Table 7 the computation time of our approach and those benchmarked methods in terms of the running time in seconds. We compared the model-based methods and convolution-based methods. The model-based methods include HUGO-BD [3], MG [8], MiPOD [11] and our proposed method. The convolution-based methods include SUNI [7] and HILL [10]. The experiments were carried out on a Personal Computer with a 4.2 GHz 8 cores

TABLE 5. Detectability of different steganographic methods under various payloads against maxSRMd2 with ensemble classifier.

Steganography	0.05	0.1	0.2	0.3	0.4	0.5
HUGO	0.3613 ± 0.0041	0.3079 ± 0.0017	0.2418 ± 0.0021	0.1942 ± 0.0021	0.1822 ± 0.0024	0.1340 ± 0.0015
WOW	0.3548 ± 0.0030	0.2997 ± 0.0022	0.2326 ± 0.0017	0.1887 ± 0.0018	0.1637 ± 0.0015	0.1331 ± 0.0023
S-UNI	0.4150 ± 0.0022	0.3661 ± 0.0032	0.2930 ± 0.0037	0.2360 ± 0.0022	0.1936 ± 0.0023	0.1572 ± 0.0021
MiPOD	**0.4294 ± 0.0037**	**0.3747 ± 0.0014**	**0.3030 ± 0.0019**	0.2481 ± 0.0027	0.2038 ± 0.0039	0.1678 ± 0.0038
2DSSA-WMF	0.4084 ± 0.0029	0.3635 ± 0.0015	0.3006 ± 0.0033	**0.2521 ± 0.0034**	**0.2122 ± 0.0028**	**0.1779 ± 0.0029**

TABLE 6. Detectability of different steganographic methods under various payloads against the CNN model Xu-Net.

Steganography	0.1 bpp	0.4bpp
WOW	0.4078	0.1956
S-UNI	**0.4476**	0.2067
HILL	0.4380	0.2097
2DSSA-WMF	0.4446	**0.2410**

AMD CPU 4800H and 16GB of RAM on Windows 10, MATLAB version 2019b.

From Table 7, we can see that our method is the fastest among the model-based methods, which is twice as fast as MIPOD and MG, yet it has produced the best results in almost all experimental settings in defending the SRM and TLBP attacks. Note that model-based methods are not comparable to the convolution-based methods in terms of computation efficiency as they often require complex matrix analysis and take much more time than other approaches using convolutions.

TABLE 7. Computation time comparisons among different methods (seconds).

Steganography	Running time (s)
HUGO	14.67
MG	2.4
MIPOD	2.33
2DSSA-WMF	0.95
SUNI	0.41
HILL	0.33

V. CONCLUSION

In this paper, following the rules for ranking priority profile in [5], we proposed our 2DSSA-WMF method for image steganography.

The 2D-SSA method can effectively decompose an image by eigenvalues, which helps us to select the edges from the images automatically. We found 2D-SSA is particularly useful in clustering the embedding positions. We also used WMF in designing our cost function, which helps to smooth the reconstructed image produced by 2D-SSA. In this way, we prevent embedding positions from straying into the low-frequency area in the images. This two-step method achieves the best results on the well-known BOSSbase 1.01 dataset when compared with several state-of-the-art approaches against non-shared selection channel attack. In selection-channel aware scenarios, it also provides the best results when the payload is 0.3 bpp or larger. We also tested the detectability against the well-known CNN model, Xu-Net, and the results suggest our method does provide secure performance.

For future work, we will consider further image processing techniques in designing steganographic algorithms and adaptive feature selection as well as the most recent deep-learning-based approaches.

REFERENCES

[1] 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.

[2] 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.

[3] T. Pevnỳ, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. Int. Workshop Inf. Hiding*. Berlin, Germany: Springer, 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] 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.

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

[7] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP J. Inf. Secur.*, vol. 2014, no. 1, p. 1, 2014.

[8] J. Fridrich and J. Kodovskỳ, "Multivariate Gaussian model for designing additive distortion for steganography," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 2949–2953.

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

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

[11] 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.

[12] D. Hu, H. Xu, Z. Ma, S. Zheng, and B. Li, "A spatial image steganography method based on nonnegative matrix factorization," *IEEE Signal Process. Lett.*, vol. 25, no. 9, pp. 1364–1368, Jul. 2018.

[13] X. Qin, B. Li, and J. Huang, "A new spatial steganographic scheme by modeling image residuals with multivariate Gaussian model," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2617–2621.

[14] W. Su, J. Ni, X. Hu, and J. Fridrich, "Image steganography with symmetric embedding using Gaussian Markov random field model," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jun. 9, 2020, doi: 10.1109/TCSVT.2020.3001122.

[15] J. Zabalza, J. Ren, J. Zheng, J. Han, H. Zhao, and S. Li, "Novel two-dimensional singular spectrum analysis for effective feature extraction and data classification in hyperspectral imaging," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4418–4433, Aug. 2015.

[16] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter (WMF)," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2830–2837.
[17] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," *Proc. SPIE*, vol. 6505, Jan. 2007, Art. no. 650502.
[18] W. Zhang, X. Zhang, and S. Wang, "Near-optimal codes for information embedding in gray-scale signals," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1262–1270, Mar. 2010.
[19] J. Zabalza, C. Qing, P. Yuen, G. Sun, H. Zhao, and J. Ren, "Fast implementation of two-dimensional singular spectrum analysis for effective data classification in hyperspectral imaging," *J. Franklin Inst.*, vol. 355, no. 4, pp. 1733–1751, 2018.
[20] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: A tutorial," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 3, pp. 157–192, Mar. 1996.
[21] P. Bas, T. Filler, and T. Pevnỳ, "'Break our steganographic system': The ins and outs of organizing BOSS," in *Proc. Int. Workshop Inf. Hiding*. Berlin, Germany: Springer, 2011, pp. 59–70.
[22] 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.
[23] B. Li, Z. Li, S. Zhou, S. Tan, and X. Zhang, "New steganalytic features for spatial image steganography based on derivative filters and threshold LBP operator," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1242–1257, May 2018.
[24] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, "Selection-channel-aware rich model for steganalysis of digital images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2014, pp. 48–53.
[25] J. Kodovsky, 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.
[26] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1181–1193, May 2018.
[27] S. Wu, S.-H. Zhong, and Y. Liu, "A novel convolutional neural network for image steganalysis with shared normalization," *IEEE Trans. Multimedia*, vol. 22, no. 1, pp. 256–270, 2019.
[28] W. You, H. Zhang, and X. Zhao, "A Siamese CNN for image steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 291–306, Jul. 2020.
[29] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 708–712, May 2016.

GUOLIANG XIE received the M.Eng. degree in electronic and information engineering from Guangdong Polytechnic Normal University, Guangzhou, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, U.K. His research interests include image processing, image steganography, and machine learning.

JINCHANG REN (Senior Member, IEEE) received the B.E. degree in computer software, the M.Eng. degree in image processing, and the D.Eng. degree in computer vision from Northwestern Polytechnical University, Xi'an, China, and the Ph.D. degree in electronic imaging and media communication from the University of Bradford, U.K. He is currently a Professor with the National Subsea Centre, Robert Gordon University, Aberdeen, U.K. He has published over 300 peer-reviewed journal and conferences papers. His research interests include hyperspectral imaging, image processing, computer vision, big data analytics, and machine learning. He acts as an Associate Editor for several international journals, including IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING and *Journal of the Franklin Institute*.

STEPHEN MARSHALL (Senior Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the University of Nottingham, U.K., and the Ph.D. degree in image processing from the University of Strathclyde, Glasgow, U.K. He is currently a Professor with the Department of Electronic and Electrical Engineering, University of Strathclyde. With over 150 articles published, his research activities focus on nonlinear image processing and hyperspectral imaging. He is a Fellow of IET.

HUIMIN ZHAO was born in Shaanxi, China, in 1966. He received the B.Sc. and M.Sc. degrees in signal processing from Northwestern Polytechnical University (NWPU), in 1992 and 1997, respectively, and the Ph.D. degree in electrical engineering from Sun Yat-sen University, in 2001. He is currently a Professor and the Dean of the School of Computer Sciences, Guangdong Polytechnic Normal University. His research interests include image/video and information security technologies and applications.

HUIHUI LI received the Ph.D. degree in computer science and engineering from the South China University of Technology, Guangzhou, China. She is currently a Lecturer with the School of Computer Science, Guangdong Polytechnic Normal University. Her research interests include image processing, machine learning, and affective computing.