Hindawi
Security and Communication Networks
Volume 2022, Article ID 1400708, 17 pages
https://doi.org/10.1155/2022/1400708

Research Article

Toward Steganographic Payload Location via Neighboring Weight Algorithm

Tong Qiao, [1,2] Xiangyang Luo, [2] Binmin Pan, [1] Yuxing Chen, [1] and Xiaoshuai Wu [1]

[1] *School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China*
[2] *State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou Science and Technology Institute, Zhengzhou, Henan, China*

Correspondence should be addressed to Xiangyang Luo; xiangyangluo@126.com

Received 12 December 2021; Accepted 31 January 2022; Published 25 February 2022

Academic Editor: Beijing Chen

Modern steganalysis has been widely investigated, most of which mainly focus on dealing with the problem of detecting whether an inquiry image contains hidden information. However, few articles in the literature study the location of secret bits hidden by modern adaptive steganography. In this paper, we propose a novel algorithm for locating steganographic payload in the spatial domain. We first predict the steganographic scheme and its payload, which is used for generating a random bitstream. Then, the random bits are embedded in the stego image based on the cost matrix in the framework of Syndrome-Trellis Codes (STCs). Next, relying on the differences between two stego images, the extended modification map in couple with the neighboring weight algorithm can be acquired, leading to the location of the hidden bits. Compared with the prior art, the extensive experiments verify that our proposed locating algorithm performs better, in terms of locating accuracy and efficiency.

1. Introduction

Steganography is the science and art of covertly transmitting the secret message in a carrier, such as widely adopted multimedia content. In general, an empirical cover carries the secret message under the supervision of the warden while the recipient extracts it to accomplish covert communication. In the past two decades, image steganography has made great progress. To counter against steganography, the analysis technique of detecting a steganographic image, defined as steganalysis, has also been advanced.

To ensure the undetectability of image steganography, a practical and common manner is to change cover pixels slightly by ±1. In particular, in the early stage of the study in this field, LSBR (Least Significant Bit Replacement) is designed, which randomly spreads the modification changes to the whole cover image; LSBM (Least Significant Bit Matching) is proposed to avoid the asymmetry artifacts by randomly modifying LSBs. In the current image steganography, the study of image content-adaptive schemes is usually given the first priority. One of the most successful adaptive models rather treats the message embedding as a source coding problem with a fidelity constraint [1], instead of taking the cover source distribution into account. In this framework of minimizing the distortion caused by embedding, the establishment of the cost function becomes fundamentally important for the steganographer who prefers hiding information in the texture region of a cover image.

Many modern adaptive steganographic algorithms have been proposed, such as in spatial domain Highly Undetectable steGo (HUGO) [2], Wavelet Obtained Weights (WOW) [3], Spatial UNIversal WAvelet Relative Distortion (S-UNIWARD) [1], HIgh-pass, Low-pass, and Low-pass (HILL) [4], and in JPEG domain JPEG UIversal WAvelet Relative Distortion (J-UNIWARD) [1], Uniform Embedding Distortion (UED) [5], and Uniform Embedding Revisited Distortion (UERD) [6]. Moving the study from laboratory to real world, however, most of the current steganographic methods have poor performance of resisting JPEG compression or rescaling attack. Thus, some robust steganalysis detectors are recently proposed to address that challenge such as [7–10]. The steganographic algorithms always aim to hide the secret information in an imperceptible manner to ensure that the stego image visually and statistically behaves very similar to its counterpart cover source.

In the face of the challenge proposed by steganography, the task of steganalysis is to classify between the cover and stego source. Specifically, in the generalized framework of steganalysis, steganalysis aims to (1) detect the existence of hidden information (see [11–14]), namely, binary classification, (2) predict the size of the payload, also defined as quantitative steganalysis (see [15]), (3) locate the steganographic payload, and (4) extract the secret information (see [16–18]), also defined as forensic steganalysis. In the recent studies of steganalysis, most researchers focus on detecting if the secret information is hidden in an image. Relying on the rich models, together with the ensemble learning-based mechanism, the state of the arts (see [19–22]) perform very well in dealing with the problem of classifying between cover and stego images. Recently, in the framework of deep learning [23, 24], instead of hand-crafted feature extraction, the realization of end-to-end automatic image steganalysis gradually becomes widespread (see [25–33]). Furthermore, quantitative steganalysis algorithms have also been investigated (see [34]).

In this paper, we mainly study the algorithm of payload location, which currently receives less attention compared with both binary classification and quantitative steganalysis. By predicting the cover source (or specifically by calculating the differences between inquiry stego and predicted cover source), a series of steganalysis locators targeting LSBR or/and LSBM embedding steganography have been designed, such as [35–39]. Without loss of generality, the problem of locating hidden bits can be smoothly transferred as binary classification. Based on the prescribed threshold, each pixel is classified as an innocent or stego sample. Also, the following established algorithms obey the rule of binary classification by using hand-crated SPAM features [40] of [41] or deep-learning-based features [42]. Recently, [43, 44] propose locating hidden bits in the DCT domain, mainly targeting JSteg and F5 steganography, whose stego key can be recovered in [45].

In fact, most prior locating algorithms have two remarkable limitations. When dealing with modern adaptive steganography, it probably becomes invalid. Furthermore, most algorithms are designed for one targeted steganography, such as LSBR or LSBM, which cannot be used for universal location. To overcome the current limitations, let us establish a universal detector of locating the payload of adaptive steganography, only dependent on a single inquiry image. The core idea behind our proposed algorithm is that modern adaptive steganography is prone to embed secret bits into the texture region of an image. It should be noted that our proposed algorithm can only locate the flipped hidden bits (±1 happens) not including nonflipped bits. In fact, when the embedding procedure cannot modify the bits of a cover image, those nonflipped bits are hard to be located due to their unchanged property during embedding. Then, the main contributions are listed in the following:

(1) In virtue of the intrinsic property of adaptive steganography, we propose to design the steganalysis algorithm toward payload location relying on a single inquiry image

(2) Based on the proposed neighboring weight algorithm (NWA), we establish the extended modification map and its refined version for predicting the flipped-hidden-bit location, which further narrows down the prediction error and improves the location accuracy

(3) For practical use, we propose four cases of locating steganographic payload, referring to as KPKS, UPKS, KPUS, and UPUS (see details in Table 1)

(4) Numerical experiments empirically verify the effectiveness of the proposed location algorithm, which can deal with different modern adaptive steganographic algorithms such as WOW, S-UNIWARD, and HILL. Moreover, compared with the prior arts, our scheme performs its superiority

The rest of the paper is organized as follows. We first overview the state of the arts concerning the study of locating hidden bits. We present the core idea of designing a detector for payload location in Section 3. In Section 4, the detailed steps of locating hidden bits by adaptive steganography are extended. Furthermore, our proposed neighboring weight algorithm is specifically described. Next, the numerical experimental results are provided in Section 5, including the evaluation of our proposed algorithm as well as the comparison with the prior arts. Finally, we conclude this paper in Section 6.

2. State of the Arts

In this paper, we mainly focus on the study of locating payload. In general, the problem of locating hidden bits is always solved by biclassifying each pixel of the inquiry image. For clarity, let us define a cover image as a vector $\mathbf{c} = \{c_l\}$, $l \in \{1, \ldots, L\}$ and a corresponding stego image described as a vector $\mathbf{s} = \{s_l\}$, $l \in \{1, \ldots, L\}$. Then, the discrimination factor d_l, denoted as residual noise between stego and predicted cover source, is formulated as

$$
\begin{gathered}
d_l{=}_{\text{idc}} f l \in \{1, \ldots, L\} \\
\text{with } f_{\text{idc}}[a, b] = \begin{cases} 1, & \text{if } a \neq b, \\ 0, & \text{otherwise,} \end{cases}
\end{gathered}
\tag{1}
$$

where the indicator function $f_{\text{idc}}[\cdot]$ is used to label the predicted pixel with/without hidden bits, and the estimated cover pixel is denoted as $\widehat{c_l}$. On the assumption that all the hidden bits are embedded in the same position for a number of images, it holds true that the stego pixels can be successfully located by averaging the d_l. In this scenario, the expected value of the averaged differences is denoted as $E\overline{[d_l]}$. In detail, when the cover pixel is used for embedding, including the cases of flipping and nonflipping, $E\overline{[d_l]}$ equals 0.5; when the cover pixel is not selected for embedding, $E\overline{[d_l]}$ equals 0. Next, by calculating the average value of each pixel

Table 1: Abbreviation of location scenarios in the framework of our proposed algorithm.

	Known scheme	Unknown scheme
Known payload	KPKS	KPUS
Unknown payload	UPKS	UPUS

among a group of stego images, we can predict the position of hidden bits. Meanwhile, the optimal threshold arrives at 0.25 lying between two expected values. Immediately, let us extend the related works of the locating algorithms.

In [35], inspired by the Weighted Stego (WS) image steganalysis method, a steganalysis algorithm is designed for locating hidden bits embedded by LSBR. Specifically, the linear filter is used for predicting the average residual value of each pixel, in which a residual-based threshold is empirically prescribed for locating embedding positions. Next, to deal with the problem of LSBM steganography, [36] proposes adopting a Wavelet Absolute Moment (WAM) filer to extract the residual, which characterizes the distinguishable features between the cover and stego pixels. By estimating $\hat{c}_l$ or directly calculating the $\overline{d}_l$, two aforementioned methods have verified their effectiveness in locating hidden bits. Although the payload of the old steganography is successfully located, the limitations of the aforementioned methods are as followss: a large number of stego images have to share the same size; the secret bits are hidden in the same positions for each image; a locating algorithm is only applicable for one targeted steganographic algorithm, such as LSBR or LSBM.

In the following studies, high-order features represented by residuals are used to locate the steganographic payload embedded by LSBM (see [37]). To unify the location of hidden bits embedded by LSBR or LSBM, relying on the theory of Maximum A Posteriori (MAP) [38] designs a detector which remarkably improves the location performance. The accurately estimated cover source brings more discriminative residuals, which straightforwardly leads to improved location accuracy. Next, [39] designs an effective algorithm to extract the hidden bits independent of the embedding key. In fact, the algorithms mainly put the focus either on the estimation of cover source $\hat{c}_l$ [38, 39] or directly on calculating the weighted average of residual noise [37]. It makes sense that the fidelity of the cover source estimation directly impacts the accuracy of payload location (see [38] for details). Nevertheless, the methods still need large-scale stego images within embedding the same payload location.

Inspired by the work [40], [41] proposes dealing with the location problem by classifying each pixel into binary types: payload and nonpayload. Although the framework equation (1) is not used, the hidden bits can be successfully located by investigating the features of each pixel. The discriminative features (72-dimensional features for each pixel) characterized by neighboring pixel-value differences are used for training a Support Vector Machine (SVM) classifier, which serves for binary classification during the stage of locating hidden bits. Although the learning-based method improves the accuracy and efficiency of locating, the performance is degraded when the payload is increased. Recently, to solve the problem of inaccuracy location within the small payload, [42] proposes an efficient detector for locating hidden bits relying on the deep neural networks. However, still, it can only be applied to the stego image generated by old steganography, such as nonadaptive LSBM.

To our knowledge, few studies focus on locating adaptive steganographic payload. The article [46] opens a way to investigate the location of the steganographic payload embedded by modern adaptive algorithms. By reembedding randomly generated bits into the stego image, the hidden bits are generally located. However, when both the embedding scheme and the size of the payload are unknown, the accuracy of location cannot be guaranteed. Thus, in this paper, to further improve the location accuracy and reduce the prediction error, let us design an effective detector based on the proposed neighboring weight algorithm (NWA).

3. Statement of the Problem

In the community of data hiding, most literature studies focus on the establishment of locating algorithms for nonadaptive steganography while the challenging problem of payload location for adaptive steganography has not been widely investigated. For simplicity and clarity, it is proposed to illustrate the pipeline of our locating algorithm (see Figure 1). When an inquiry image is used for location, we first have to estimate its embedding payload and predict the embedding scheme. Because our proposed algorithm only works well in the scenario that the inquiry image has been confirmed as stego one with acquiring its steganographic scheme, subsequently, we can generate a random bitstream based on the payload. Then, let us reembed the random bits into the stego image based on the cost matrix in the framework of STCs. Next, relying on the differences between two stego images, the modification map can be obtained. By using the proposed neighboring weight algorithm, the modification map is further extended. Finally, our proposed algorithm is capable of locating flipped bits. For clarity, the main mathematical notations used in this paper are summarized in Table 2.

3.1. Establishment of Modification MAP.

By modifying pixels within texture regions, modern adaptive steganography performs very well and especially remains its high undetectability. That property inspires us to investigate if the modified pixels are selected again when reembedding happens, meaning that the locations of the steganographic payload are overlapped between a stego image and its reembedding version. That is because the cost matrix of the two images nearly remains unchanged.

First, let us embed a random bitstream into a grey-level cover image $\mathbf{C} = \left\{c_{i,j}\right\}, \quad i \in \{1, \ldots, I\}, j \in \{1, \ldots, J\}$, leading to the generation of a stego image $\mathbf{S}^{(1)} = \left\{s_{i,j}^{(1)}\right\}$. Next, by modifying the original stego image $\mathbf{S}^{(1)}$ acquired from $\mathbf{C}$, let us use the same bitstream to generate a new stego image $\mathbf{S}^{(2)} = \left\{s_{i,j}^{(2)}\right\}$. It should be noted that regardless of hidden bits (the same, flipped or random ones), we denote the first stego image generated from the cover $\mathbf{C}$ as $\mathbf{S}^{(1)}$ while the second stego image from $\mathbf{S}^{(1)}$ as $\mathbf{S}^{(2)}$. It is worth noting that

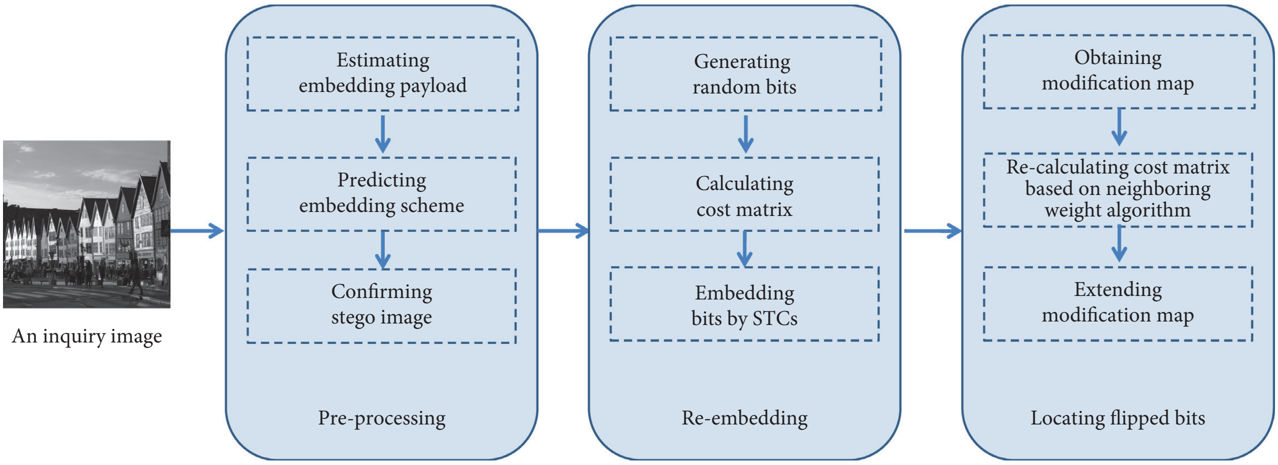

An inquiry image
Estimating embedding payload
Predicting embedding scheme
Confirming stego image
Pre-processing
Generating random bits
Calculating cost matrix
Embedding bits by STCs
Re-embedding
Obtaining modification map
Re-calculating cost matrix based on neighboring weight algorithm
Extending modification map
Locating flipped bits

Figure 1: Pipeline of our proposed method.

Table 2: Notations.

$\mathbf{C}$	Grey-level cover image
$\mathbf{S}^{(1)}$	Original stego image by the first embedding
$\mathbf{S}^{(2)}$	New stego image by reembedding
$\mathbf{M}^{(1)}$	Modification map
$\mathbf{M}^{(e)}$	Extended modification map
$\mathbf{M}^{(o)}$	Refined modification map
$\mathbf{m}$	Hidden bits
ρ	Cost matrix
ω	Weight factor
d	Euclidean distance

in the framework of STCs, the cost function (see [1] for instance) guides us to select the ready-to-embed pixels and to generate both $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$. Then, the modification map can be straightforwardly formulated as

$$\mathbf{M}^{(1)}(i,j)=\begin{cases}255, & \text{if the pixel } c_{i,j} \text{ is flipped,}\\ 0, & \text{otherwise,}\end{cases} \tag{2}$$

where both cover and stego images share the same size of the modification map $\mathbf{M}^{(1)}$. As Figure 2 illustrates, an 8-bit cover image with the size of 512 × 512 and the modification maps come from its corresponding stego images. $\mathbf{M}^{(1)}$ is acquired by making difference between $\mathbf{C}$ and $\mathbf{S}^{(1)}$, which visually labels the flipped pixels caused by embedding. Meanwhile, $\mathbf{M}^{(2)}$ can be obtained from both $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$. It should be noted that four different practical scenarios are considered in our proposed algorithms, referring to KPKS, UPKS, KPUS, and UPUS (see details in Table 1 of Section 3.2). Two embedding operations both prefer embedding the bits nearly at the same locations, referring to the texture region. Furthermore, few locations have the value 255 in the same position of two modification maps, meaning that few pixels at the positions experience twice modification. Thus, we need to extend the modification map $\mathbf{M}^{(2)}$ for digging out more hidden bits (see details in Section 4).

3.2. Practical Scenario of Locating Steganographic Payload. In this paper, to overall evaluate the effectiveness of the proposed locating algorithm, we intend to address that challenging problem in the four practical scenarios (see Table 1 for details). Before locating the flipped bits by adaptive steganography, it is proposed to emphasize a prerequisite that the inquiry image has been detected as stego one with secret information. Thus, we list two assumptions: *known payload* or *unknown payload*. When the payload is known, the steganalyst is capable of conducting the locating algorithm straightforward; when the payload is unknown, the prediction algorithm (or defined as quantitative steganalysis), such as [15], has to be conducted first. It is worth noticing that when the predicted payload α is larger than the given threshold τ, the inquiry image is detected as stego. Besides, in the procedure of locating flipped bits using the proposed algorithm, reembedding is obligatory. However, when a stego image is obtained, it hardly holds true that we can acquire the embedding scheme used for the stego image. Thus, another two assumptions should also be addressed, referring to as a *known scheme* or an *unknown scheme*. The specific experimental results are extended in Section 5.

In the framework of our proposed locating algorithm, the key point is how to confirm the adjacent regions for reembedding. If the large size of the adjacent region is selected, many incorrectly classified pixels will be included, leading to the decreased accuracy of the location. On the contrary, if the small size of the adjacent region is selected, possibly some missing-classified pixels that are actually flipped by adaptive steganography cannot be accurately located. To deal with that trade-off problem, we thus propose improving the performance of locating hidden bits based on the neighboring weight algorithm. In the following section, we first specifically describe our proposed locating algorithm. More importantly, the NWA is designed to further reduce location errors.

4. Proposed Work

In this section, we first introduce the general steps of locating a steganographic payload algorithm. Next, the establishment of the extended modification map is specifically presented.

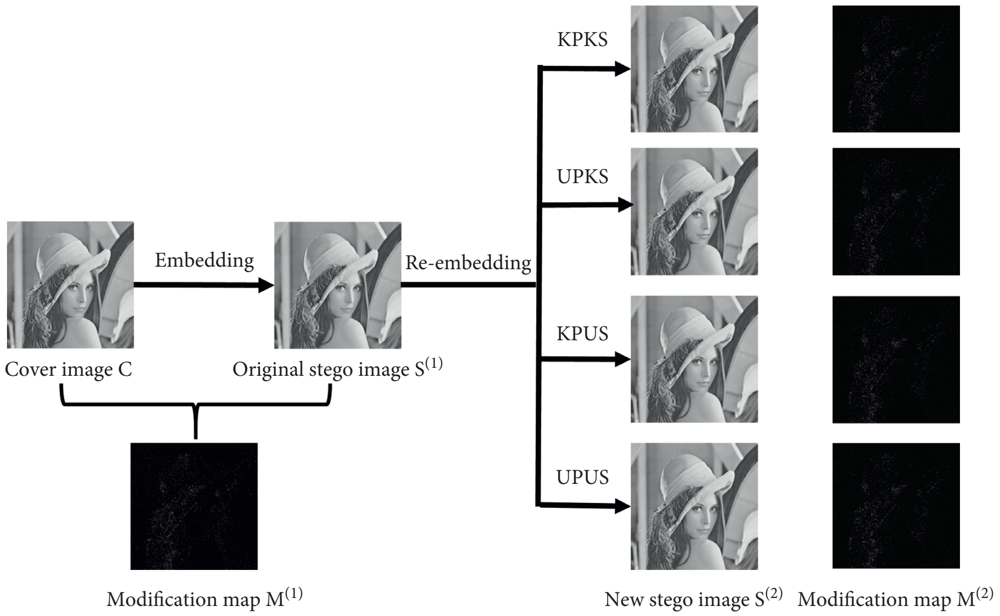

KPKS
UPKS
Embedding
Re-embedding
KPUS
UPUS
Cover image C
Original stego image $S^{(1)}$
Modification map $M^{(1)}$
New stego image $S^{(2)}$
Modification map $M^{(2)}$

FIGURE 2: Illustration of our proposed framework.

Then, we develop two novel schemes dealing with the problem of refining the extended modification map.

4.1. Description of Our Locating Algorithm. The description of our locating algorithm can be summarized in Algorithm 1.

Generating a random bit stream: a random bitstream $\mathbf{m}$ with the length L is generated. Note that $L = \text{Num} \times \alpha$ where Num denotes the total amount of pixels, and α is the relative payload. *Calculating the cost matrix*: relying on an adaptive steganographic algorithm, a bank of designed filters is then utilized to obtain the cost matrix of the stego image. For simplicity and clarity, let us denote the stego image as $\mathbf{S}$, and the cost matrix as ρ referring to [1]. *Embedding secret message using STCs*: without loss of generality, STCs are used to embed message $\mathbf{m}$ into the stego $\mathbf{S}^{(1)}$ on the principle of minimizing the distortion function based on the cost matrix ρ. In such a manner, the stego version of image $\mathbf{S}^{(1)}$ after modification is denoted as $\mathbf{S}^{(2)}$. *Obtaining the modification map*: based on the differences between the two stego images, the modification map $\mathbf{M}$ is obtained as described in Section 3.1. *Extending the modification map*: to locate hidden bits, we intend to extend the modification map $\mathbf{M}^{(o)}$ to $\mathbf{M}^{(e)}$ in a given margin value N. *Refining the extended modification map*: to locate the modified pixels as more as possible and reduce the number of incorrectly predicted bits, the refined extended modification map $\mathbf{M}^{(o)}$ is established based on the redesigned cost matrix ρ'.

Without loss of generality, the selection of N, denoted as margin value, is actually a trade-off problem in the design of the extended modification map. In detail, the value N would increase if we intend to locate the modified pixels as more as possible. While as the margin value N becomes larger, more and more innocent pixels (without being modified) would be also involved. In this context, the designed extended modification map should follow two requirements: (1) more hidden bits are contained in the map, denoting the location; (2) less innocent bits are excluded in the map.

In the following sections, let us specifically introduce the design of the extended modification map. More importantly, based on the proposed neighboring weight algorithm, the map is further refined for locating more hidden bits and abandoning more innocent bits.

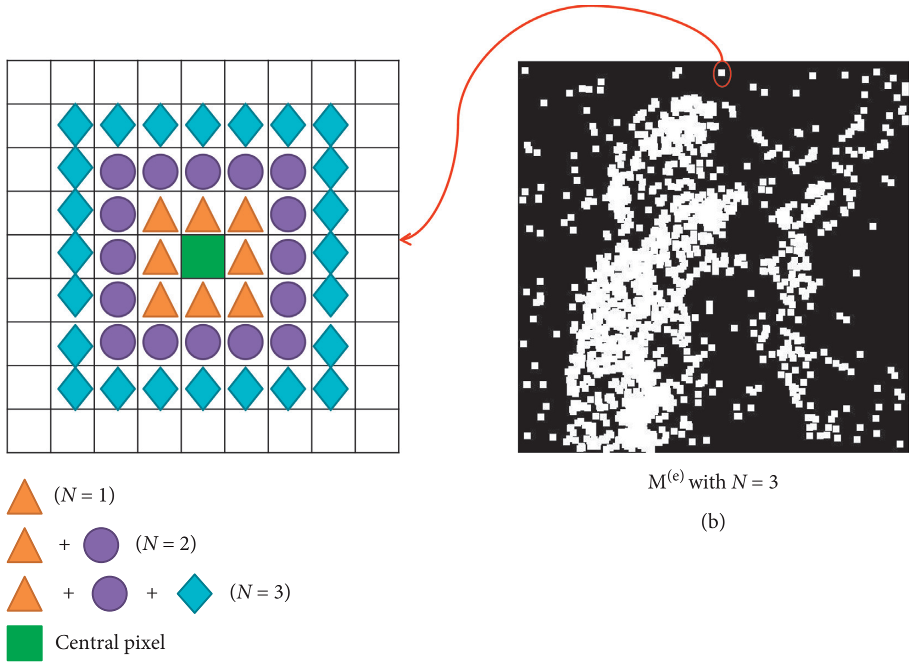
4.2. Design of Extended Modification MAP. In fact, based on the intrinsic property of the content-adaptive scheme, it hardly holds true that modern adaptive steganography modifies the pixel of the same location twice when embedding the same random bits. And meanwhile, the adjacent region of the pixel modified by the first embedding probably contains the modified pixels caused by the second embedding. Immediately, based on the results of Figure 2, it is proposed to extend the $\mathbf{M}^{(1)}$ by covering each pixel's neighbors, that is formulated as

$$\mathbf{M}^{(e)}(i+p, j+p) = \begin{cases} 255, & \text{if the pixel at } (i, j) \text{ is flipped}, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $p \in [-N, N]$ represents an integer controlled by the extension maximum, margin value N. Next, the adjacent regions of a pixel in variant margin value N are illustrated in Figure 3(a). Figure 3(b) illustrates $\mathbf{M}^{(e)}$ with the margin value $N = 3$, where the bright regions (the pixels in the regions equal to 255) definitely cover a large portion of pixels flipped by the first embedding. When the margin value equals N, the size of its adjacent region is calculated by the following function $(2N+1) \times (2N+1)$. Obviously, when $N = 0$, the modification map $\mathbf{M}^{(e)}$ is equivalent to $\mathbf{M}^{(2)}$. Furthermore, let us define the rate $r = m/n$, where n is the

```
Input: Stego image $\mathbf{S}^{(1)}$, steganographic payload $\alpha$
Output: Predicted locations of steganographic payload
(1) //Generating a random bit stream $\mathbf{m}\mathrm{Num} = f_{\mathrm{num}}[\mathbf{S}^{(1)}]$, function $f_{\mathrm{num}}[\cdot]$ for calculating the number of input data $L = \mathrm{Num} \times \alpha$,
    denoting the number of bits; $\mathbf{m} = G[L]$, function $G[\cdot]$ for generating random bits
(2) //Calculating the cost matrix $\rho$
(3) //Embedding secret message $\mathbf{mS}^{(2)} = f_{\mathrm{emb}}[\mathbf{S}^{(2)}, \mathbf{m}, \rho]$, function $f_{\mathrm{emb}}$ is used for embedding bits $\mathbf{m}$ into $\mathbf{S}^{(1)}$ in the framework of
    STCs
(4) //Obtaining the modification map $\mathbf{M}^{(1)} \leftarrow f_{\mathrm{diff}}[\mathbf{S}^{(1)}, \mathbf{S}^{(2)}]$, function $f_{\mathrm{diff}}[\cdot]$ calculates the differences between $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ to
    generate the modification map
(5) //Extending the modification map $\mathbf{M}^{(e)} \leftarrow \mathbf{M}^{(1)}$
(6) //Refining the extended modification map $\mathbf{M}^{(o)} = f[\mathbf{M}^{(e)}]$ by redesigning the cost matrix $\rho'$, function $f[\cdot]$ refines the original
    extended modification map $\mathbf{M}^{(e)}$. $\mathbf{M}^{(o)}$ labels all the predicted locations via our proposed algorithm
```

Algorithm 1:Procedure to Locate Steganographic Payload.



(a)

$\mathrm{M}^{(e)}$ with $N = 3$

(b)

Figure 3: Illustration of neighboring regions of a central pixel in different margin values, and the extended modification map $\mathbf{M}^{(e)}$ with margin value $N = 3$.

number of pixels flipped by the first embedding (those pixels equal to 255 in $\mathbf{M}^{(1)}$). Besides, m denotes the number of pixels that are flipped by the first and second embedding, in which the pixels equal 255 in both $\mathbf{M}^{(1)}$ and $\mathbf{M}^{(2)}$. r denotes the ratio of the number of pixels correctly predicted by the second embedding to the number of pixels modified by the first embedding.

To evaluate the feasibility of our proposed location algorithm, let us conduct the heuristic experiments over 10000 8-bit images from the BOSSbase ver.1.01 [47]. The experimental results in variant margin value are listed in Table 3. It should be noted that bpp denotes bits per pixel for abbreviation. One can observe that, at a fixed payload, the rate r can be increased as the margin value N becomes larger. That is because the larger adjacent regions can cover more pixels used for information hiding. Besides, in a fixed margin value N, the more the bits embedded into, the larger proportion the modified pixels can be located. In addition, as Table 3 reports, r nearly remains stable with the large N and payload. We assume that the cost value of the pixels modified by the first embedding is slightly changed (see [20] for details). That is because those pixels are merely modified by ±1. When we reembed the same bits into the stego image $\mathbf{S}$, some pixels carrying the payload might not be flipped again.

In fact, through investigating the possibility of locating a steganographic payload, we assume that the reembedding is conducted based on the known random bits used for the first embedding. However, it cannot hinder us from locating hidden bits, even the random bits are unknown or manually totally different from the first one. The results of those

Table 3: r statistics on p within the margin value N when reembedding same bits.

Payload α						N					
	0	1	2	3	4	5	6	7	8	9	10
0.05 bpp	0.064 5	0.344 1	0.558 7	0.688 4	0.770 1	0.823 9	0.860 9	0.887 2	0.906 3	0.920 5	0.931 0
0.10 bpp	0.086 3	0.434 0	0.665 5	0.790 6	0.862 1	0.905 0	0.931 8	0.948 8	0.959 7	0.966 9	0.971 7
0.20 bpp	0.117 0	0.544 7	0.779 9	0.887 5	0.939 1	0.964 5	0.977 1	0.983 6	0.987 0	0.989 0	0.990 2
0.30 bpp	0.142 7	0.624 3	0.849 8	0.936 4	0.970 1	0.983 5	0.989 0	0.991 6	0.992 9	0.993 6	0.994 1
0.40 bpp	0.165 9	0.687 7	0.897 2	0.963 4	0.984 1	0.991 0	0.993 6	0.994 8	0.995 4	0.995 8	0.996 0
0.50 bpp	0.189 0	0.743 1	0.931 4	0.979 1	0.991 2	0.994 7	0.996 1	0.996 7	0.997 0	0.997 2	0.997 3

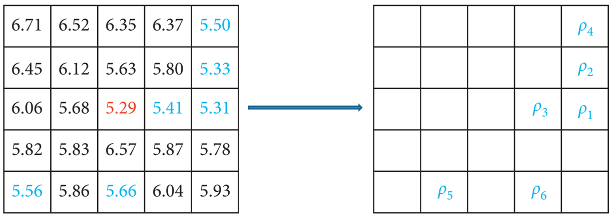
scenarios have been exemplified in our prior work (see [46] for details). Besides, the designed extended modification map $\mathbf{M}^{(e)}$ can to some degree predict the hidden bits but also incorrectly cover the innocent pixels. Therefore, in this paper, it is of great importance that we need to further refine the proposed extended modification map.

4.3. Refinement of Extended Modification MAP Based on Neighboring Weight Algorithm (NWA). As our aforementioned discussion, the selection of N is a trade-off problem. Although the increased N brings a high recall ratio, the number of incorrectly located bits is also raised. For clarity, it is worth noticing that in the design of $\mathbf{M}^{(e)}$, $(2N+1)^2$ pixels in the neighboring region (see the colored region in Figure 3 for instance) are contained, namely, labeled as the predicted hidden bits. To refine the extended modification map, we need to choose the bits, which are probably used for the first embedding. Then, let us formulate the refined modification map as

$$\mathbf{M}^{(\mathrm{o})} = f\left[\mathbf{M}^{(\mathrm{e})}\right], \tag{4}$$

where function $f[\cdot]$ refines the original extended modification map. To this end, the problem of predicting steganographic payload transfers to designing the manner of refining the extended modification map.

For simplicity, inspired by the calculation of the cost matrix ρ of stego image $\mathbf{S}^{(1)}$, we intuitively sort the cost value ρ of each pixel in ascending order. In the stage of generating a stego image $\mathbf{S}^{(1)}$, based on the calculated cost value of $\mathbf{C}$, the modern adaptive algorithm tries its best to embed the hidden bits into the locations carrying low-cost values, with modification as less as possible. Thus, it makes sense that we assume the cost value of each pixel from $\mathbf{S}^{(1)}$ is similar to that of $\mathbf{C}$. Then, the cost value ρ guides us to complete the design of the refinement function $f[\cdot]$. Specifically, K-minimum ρ is selected, corresponding to the predicted location in the extended modification map $\mathbf{M}^{(e)}$. In other words, the locations of the set $\{\rho_1, \ldots, \rho_K\}$ are selected as the optimal position. As Figure 4 illustrates, a portion of the cost matrix of a natural grey-level image is extracted for a clear demonstration. In particular, when the margin value N is set as 2 in the map $\mathbf{M}^{(e)}$, only six ρ $(K=6)$ are selected for refining the extended modification map. It should be noted that if the value K equals $(2N+1)^2 - 1$, the refined map $\mathbf{M}^{(\mathrm{o})}$ degenerates back to the original extended map $\mathbf{M}^{(\mathrm{e})}$. The effectiveness of our proposed refinement scheme will be verified in the extensive experiments.



6.71	6.52	6.35	6.37	5.50
6.45	6.12	5.63	5.80	5.33
6.06	5.68	5.29	5.41	5.31
5.82	5.83	6.57	5.87	5.78
5.56	5.86	5.66	6.04	5.93

				ρ_4
				ρ_2
			ρ_3	ρ_1
	ρ_5		ρ_6	

Figure 4: Illustration of the selection for K-minimum ρ, where $K=6$, and the margin value $N=2$ (at the center of value 5.29). The portion of a cost matrix (left) corresponds to the selected location in the extended modification map (right) in ascending order.

In fact, with increasing N, a labeled region where the value equals 255 (see (3)) contains more and more bits nearly irrelevant to the central pixel (labeled as 255 in the modification map $\mathbf{M}^{(\mathrm{o})}$), possibly leading to incorrectly predicting the hidden bits when still using the aforementioned refinement function $f[\cdot]$. For instance, the pixels located far from the central pixel carrying a low-cost value are probably selected for refinement while they have a low possibility for the first embedding. In our assumption, the hidden bits usually are embedded in the neighboring region around the central pixel. In this context, we need to consider the neighboring weight to redesign the refinement function $f[\cdot]$. Immediately, let us recalculate the cost value of each pixel by

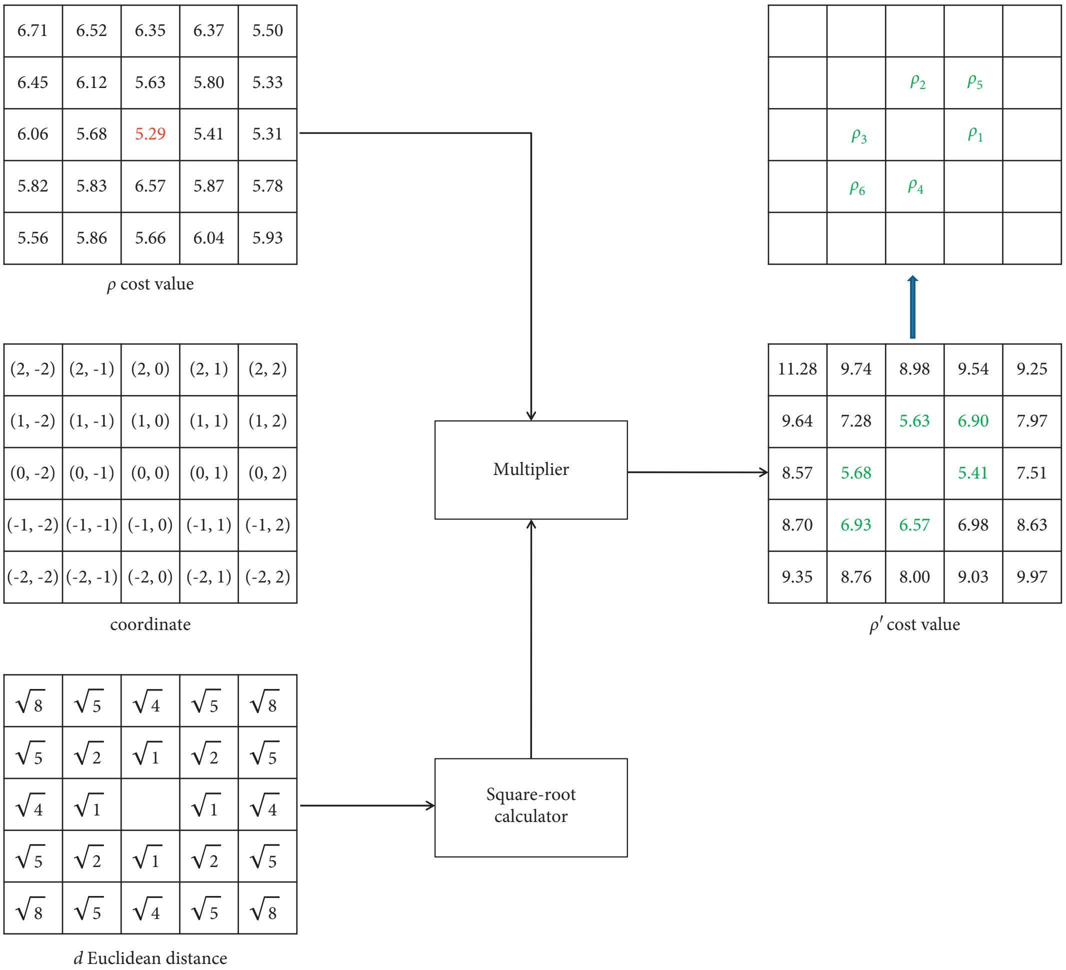
$$\rho' = \omega \cdot \rho, \tag{5}$$

where ρ denotes the original cost value while ρ' denotes a weighted cost value, and ω denoting neighboring weight factor that is formulated by

$$\omega = \sqrt{d}, \tag{6}$$

where $d = \sqrt{(x_p - x_0)^2 + (y_p - y_0)^2}$ represents the Euclidean distance between the central pixel (x_0, y_0) and any extended pixel (x_p, y_p), $p \in \{1, \ldots, P\}$ in the extended modification map $\mathbf{M}^{(e)}$. Obviously, P is the number of pixels carrying the recalculated cost value ρ' which equals $(2N+1)^2 - 1$. Still, among all ρ' in each map, we select the K-minimum ρ'.

For clarity, let us give an exemplary flowchart (see Figure 5) to illustrate the procedure of calculating ρ' in each extended modification map. As Figure 5 reports, although the original cost value ρ of Figure 5 is the same as that of Figure 4, the refined modification map using our proposed



6.71	6.52	6.35	6.37	5.50
6.45	6.12	5.63	5.80	5.33
6.06	5.68	5.29	5.41	5.31
5.82	5.83	6.57	5.87	5.78
5.56	5.86	5.66	6.04	5.93

ρ cost value

(2, -2)	(2, -1)	(2, 0)	(2, 1)	(2, 2)
(1, -2)	(1, -1)	(1, 0)	(1, 1)	(1, 2)
(0, -2)	(0, -1)	(0, 0)	(0, 1)	(0, 2)
(-1, -2)	(-1, -1)	(-1, 0)	(-1, 1)	(-1, 2)
(-2, -2)	(-2, -1)	(-2, 0)	(-2, 1)	(-2, 2)

coordinate

$\sqrt{8}$	$\sqrt{5}$	$\sqrt{4}$	$\sqrt{5}$	$\sqrt{8}$
$\sqrt{5}$	$\sqrt{2}$	$\sqrt{1}$	$\sqrt{2}$	$\sqrt{5}$
$\sqrt{4}$	$\sqrt{1}$		$\sqrt{1}$	$\sqrt{4}$
$\sqrt{5}$	$\sqrt{2}$	$\sqrt{1}$	$\sqrt{2}$	$\sqrt{5}$
$\sqrt{8}$	$\sqrt{5}$	$\sqrt{4}$	$\sqrt{5}$	$\sqrt{8}$

d Euclidean distance

11.28	9.74	8.98	9.54	9.25
9.64	7.28	5.63	6.90	7.97
8.57	5.68		5.41	7.51
8.70	6.93	6.57	6.98	8.63
9.35	8.76	8.00	9.03	9.97

ρ' cost value

		ρ_2	ρ_5	
	ρ_3		ρ_1	
	ρ_6	ρ_4		

Figure 5: Illustration of the selection for K-minimum ρ', where $K = 6$, and the margin value $N = 2$ (at the center of value 5.29) as Figure 4. The square-root calculator is designed based on equation (6); the multiplier is designed based on equation (5).

neighboring weight algorithm (see (6)) is different from the strategy by directly sorting the cost value ρ in ascending order (see Figure 4).

Furthermore, it is proposed to establish a more general weight factor by reformulating (6) as

$$\omega = d^n, \tag{7}$$

where n denotes the exponent of d, which is represented by

$$\begin{cases} n = 0, & \text{if } \rho \text{ is directly sorted in ascending order;} \\ n \neq 0, & \text{else.} \end{cases} \tag{8}$$

Obviously, when $n \neq 0$ holds, $n = 1/2$ represents a typical case of our redefined neighboring weight algorithm. Similarly, when $n = 0$ holds, the weight factor ω acts as a constant identity, leading to the fact that the recalculated cost value $\rho\prime$ equals its original version ρ. Therefore, we propose studying the neighboring weight algorithm in the general unified framework. Then, the "square-root calculator" is replaced by a "n-root calculator" by unifying all possible cases in our proposed framework. In the following section, we first discuss the selection of parameters n (see Section 5.2) based on the empirical experiments. Next, it is proposed to verify the effectiveness of the proposed steganographic payload location algorithm. Finally, we compare our location algorithm with some prior arts to further validate the superiority of our algorithm.

5. Experimental Results

5.1. Experiment Setups. It is proposed to conduct numerical experiments on the baseline BOSSbase ver.1.01 [47], where all 10000 8-bit grey-level images are acquired from eight different digital still cameras in the size of 512 × 512. The experimental settings are illustrated in Table 4. Besides, to comprehensively evaluate the performance of the steganographic payload location algorithm, we propose using the following metrics:

(i) Precision $\mathscr{V}_{\mathscr{P}}$ is defined as the percentage of correctly located samples among the total number of samples (all predicted pixels containing positive and negative samples). It is formulated by

$$\mathscr{V}_{\mathscr{P}} = \frac{D_{tp}}{D_{tp} + D_{fp}}, \tag{9}$$

Table 4: Experimental settings.

Image source	BOSSbase ver.1.01
Image color	Grey-level
Image size	512 × 512
Image format	Uncompressed
Number of original images	10 000
Payload	0.05 ~ 0.5 bpp
Steganographic schemes	WOW, S-UNIWARD, HILL, LSBR, LSBM
Locating method	[35, 36, 38, 39, 46], ours
CPUs	4 × intel xeon E7-4820 2.0 GHz CPUs
RAM	16G

where the number of true positive samples is denoted as D_{tp}, and the number of false-positive samples (the incorrectly located pixels without flipping when embedding) is denoted as D_{fp}.

(ii) Recall $\mathcal{V}_{\mathcal{R}}$ is the ratio of the number of samples D_{tp} to D_{tp} plus D_{fn}; it is given by

$$\mathcal{V}_{\mathcal{R}} = \frac{D_{tp}}{D_{tp} + D_{fn}}, \tag{10}$$

where D_{fn} denotes the number of false-negative samples (the flipped pixels without being correctly located).

(iii) F1-score $\mathcal{V}_{\mathcal{F}}$ considers both precision and recall, and it is calculated by

$$\mathcal{V}_{\mathcal{F}} = 2 \times \frac{\mathcal{V}_{\mathcal{P}} \times \mathcal{V}_{\mathcal{R}}}{\mathcal{V}_{\mathcal{P}} + \mathcal{V}_{\mathcal{R}}}. \tag{11}$$

It is worth noticing that the averaged value of each metric for all inquiry images is used to evaluate the performance of the proposed locating algorithm.

5.2. Parameter Selection of Neighboring Weight Algorithm. In this section, we empirically verify the selection of the neighboring weight parameter for optimal location. First, it is proposed to randomly choose 1000 grey-level images from the benchmark dataset BOSSbase. Next, by adopting S-UNIWARD steganography, we embed secret bits into the cover source with a 0.3 payload. In virtue of our proposed algorithm, the weight factor ω mainly controls the cost value ρ (see (5) and (7)) for each ready-to-located pixel. Moreover, the dimension of the ready-to-located region containing both flipped and nonflipped pixels is directly decided by the parameter K. Therefore, let us empirically select the optimal parameters for the proposed locating algorithm. To comprehensively evaluate the performance of the proposed neighboring weight algorithm, we report the location results using three metrics, referring to as precision, recall, and F1-score (see Figure 6).

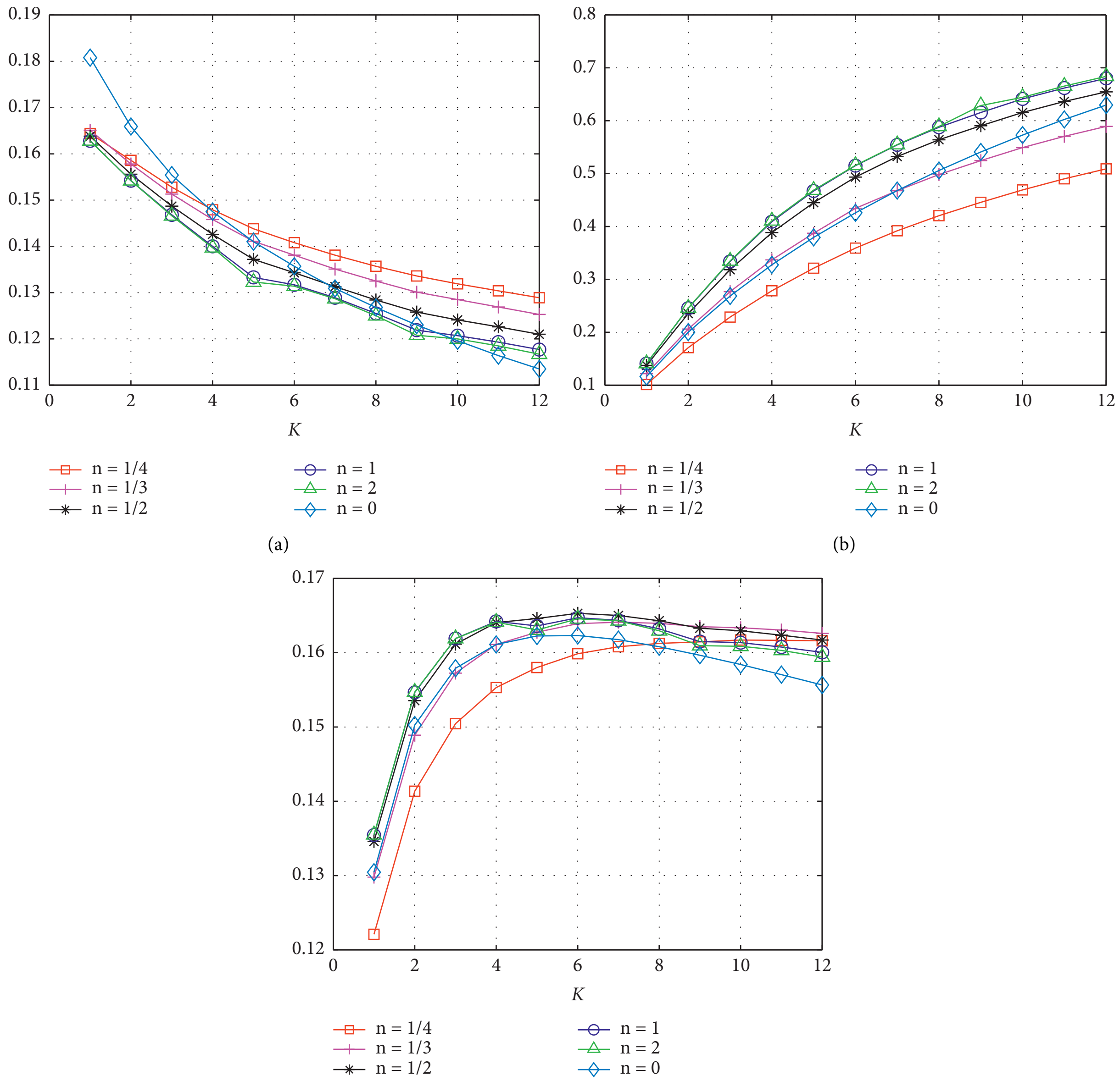
As Figure 6(a) illustrates, with increasing the K value, the $\mathcal{V}_{\mathcal{P}}$ is gradually falling down, meaning that more and more nonflipped (or innocent) pixels are incorrectly located. Since the large K probably generates the high-dimensional region including innocent pixels, the locating precision is deceased when the increased number of correctly located pixels cannot match the increased number of incorrectly located pixels. Additionally, at the small K (not larger than 6), the differences of the $\mathcal{V}_{\mathcal{P}}$ using various n behave very similarly. When K equals 1, $\mathcal{V}_{\mathcal{P}}$ with $n = 0$ is remarkably better than the others. That is because the proposed algorithm carefully selects one minimum cost ρ' without considering the neighboring weight factor ω. In this scenario, it cannot hold true that the information of distance impacts the precision of payload location. On the contrary, when the K is enlarged, the performance of the locating algorithm is obviously declined. It is worth noting that the slope of $\mathcal{V}_{\mathcal{P}}$ with $n = 0$ is steeper than the others. That is because when more payloads need to be located, we intend to centralize them around the central pixel while not locating pixels with low cost possibly in the far edge (the case of $n = 0$), which are impossibly used for embedding in our assumption. Accordingly, only relying on the empirical analysis of precision $\mathcal{V}_{\mathcal{P}}$, the selection of $n = 1/4$ is capable of bringing us the optimal locating result.

In Figure 6(b), we also investigate the performance of the proposed locating algorithm by comparing the K ranging from 1 to 12 and n lying between 0 and 2. With increasing the K value, the recall $\mathcal{V}_{\mathcal{R}}$ parameterized with different weight factors is improved. In fact, when calculating the value of recall rate, the D_{fp} is not counted. In this scenario, as the dimension of locating region is enlarged, more D_{tp} is counted while ignoring the number of pixels incorrectly located. Obviously, based on the investigation of locating performance relying on the recall $\mathcal{V}_{\mathcal{R}}$, the K equal to 12, together with $n = 2$, is our optimal choice, which is totally different from the result of parameter selection based on $\mathcal{V}_{\mathcal{P}}$ (see Figure 6(a)).

Without loss of generality, the precision $\mathcal{V}_{\mathcal{P}}$ denotes the rate of locating accuracy, and the recall $\mathcal{V}_{\mathcal{R}}$ represents if all the flipped pixels are comprehensively located. To strike the balance of two metrics for ideal selection, let us demonstrate the results of the F1-score $\mathcal{V}_{\mathcal{F}}$ in Figure 6(c). As we expected, when the K approaches 4, the F1-score value basically remains stable while achieving the maximum value at K equal to 6. Meanwhile, the n equal to 1/2 is the optimal choice for our proposed locating algorithm, which will be applied in the following experiments.

5.3. Case Studies for Locating Hidden Bits. Let us first evaluate the performance of our proposed locating algorithm in four cases, which has been specifically described in Section 3.2.

5.3.1. KPKS (Known Payload and Known Scheme). 10000 cover images from BOSSbase ver.1.01 are used for generating stego images, among which we adopt well-performed S-UNIWARD and HILL steganography, respectively. To overall verify the effectiveness of the locating algorithm, it is proposed to use various payloads ranging from 0.1 to 0.5 at step 0.1. Besides, as the prior work [46], it is compared with our proposed algorithm. For simplicity and clarity, let us name the algorithm [46] as LAS (see the description in the



(a)

(b)

(c)

Figure 6: Illustration of location performance using different n, which mainly controls the general weight factor $\omega = d^n$, where d denotes the Euclidean distance; three metrics, namely, precision, recall, and F1-score, are used for evaluation: (a) precision, (b) recall, and (c) F1-score.

following section) without NWA while our proposed scheme LAS with NWA. That is because the main differences between them are whether the locating scheme is designed based on the neighboring weight algorithm (NWA).

As Figure 7 illustrates, our proposed LAS with NWA slightly performs better than that of LAS without NWA at all the given payloads. Meanwhile, with increasing the payload, the performances of both algorithms are gradually improved. That is because the large payload brings more hidden bits embedded in the region, where the extended modification map can cover. Moreover, by assigning the weight, LAS with NWA further improves the performance of locating algorithms targeting modern adaptive steganography. It should be noted that the performance gap of two compared locating algorithms is gradually narrowed down as payload increases. That is because more hidden bits (payload 0.5 for instance) embedded into the carrier source nearly cover both texture and nontexture region, leading to the fact that the effectiveness of the selection of pixels with minimizing embedding distortion is not as remarkable as that of the small payload (a 0.1 payload for instance). In fact, when designing adaptive steganographic schemes, a similar case also happens.

Besides, by comparing S-UNIWARD with HILL, obviously, the hidden bits from stego image adopted by HILL are easier to be located. To our knowledge, the detection error of steganalysis (only targeting the problem of binary classification between the cover and stego source), referring to as

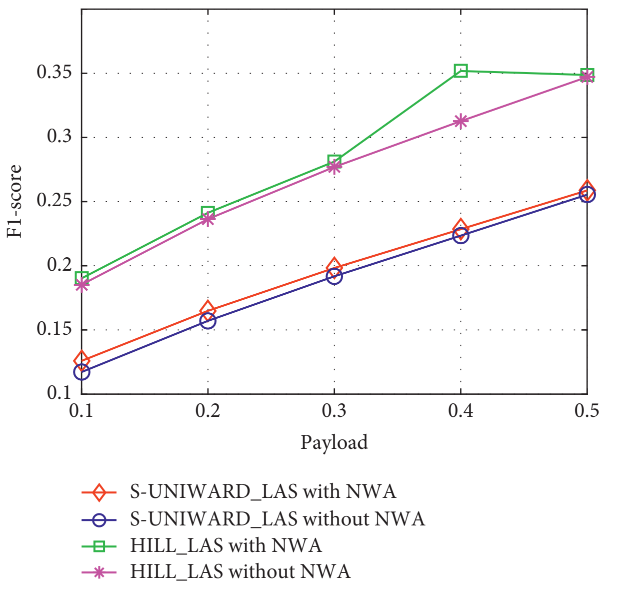


Figure 7: Averaged F1-score comparison between LAS without NWA [46] and our proposed LAS with NWA, in the case of KPKS.

E_{oob} or P_E, is usually adopted to evaluate the undetectability of steganography. In such a manner, HILL steganography is always regarded as the better choice than its opponent S-UNIWARD [4] while it hardly holds true that S-UNIWARD performs worse than HILL as the localization resistance of steganography is considered at the same time, which is empirically verified via our proposed locating algorithm.

5.3.2. KPUS (Known Payload and Unknown Scheme). Practically, a steganalyzer possibly has no idea of the specific algorithm used for data hiding. In such a case, it is proposed to evaluate the performance of the proposed LAS with NWA using the cost function from different adaptive steganography methods, also compared to LAS without NWA. 10000 cover images are used for generating stego images with a 0.3 payload, among which WOW, S-UNIWARD, and HILL are adopted.

As Table 5 reports, we list 9 pairs of comparison data, where the former data corresponds to F1-score from LAS with NWA, and the latter underlined data are obtained from LAS without NWA. It can be obviously observed that our proposed LAS with NWA performs better than LAS without NWA. Basically, when predicting the embedding scheme correctly, we can acquire the larger F1-score, meaning the better location result. Even if the steganographic method is predicted incorrectly, the performance is not decreased sharply, implying that the cost function from various adaptive steganographic methods cannot serve as a decisive factor for locating hidden bitts. In fact, whatever adaptive scheme is adopted, it always searches for the texture region in the image for minimizing embedding cost.

In addition, when adopting the cost function from WOW steganography, LAS with NWA performs best in not only the correctly predicted label but also the mismatched label, S-UNIWARD for instance. That is because WOW is prone to modify pixels centralized in the regions that are difficult to model while S-UNIWARD to some extent disperses its modification for security. Nevertheless, when not knowing the adaptive scheme, it can be predicted as WOW steganography.

5.3.3. UPKS (Unknown Payload and Known Scheme). Before locating the hidden bits, it is required to know the specific amount of payload of an inquiry image. However, in the more practical case that the payload is unknown, we have to predict it prior to locating hidden bits. Then, effective quantitative steganalysis [15] is adopted for accurately predicting the payload. To verify the effectiveness of the prediction algorithm, 4000 stego images are experimentally tested by, respectively, using S-UNIWARD and HILL steganography, in which half of them is with the payload 0.3 and half of them with the payload 0.5. Thus, the number of each type of stego images is 1000. Then, the histograms of prediction error are illustrated in Figures 8(a) and 9(a). It can be observed that the prediction error is relevantly small, where most of the data are concentrated around zero (perfectly correct prediction). Thus, the quantitative steganalysis is reliable enough, which can serve our proposed locating algorithm. Besides, with increasing payload, the overall error is narrowed down, meaning that the more payload is given, the more accurate prediction we can obtain.

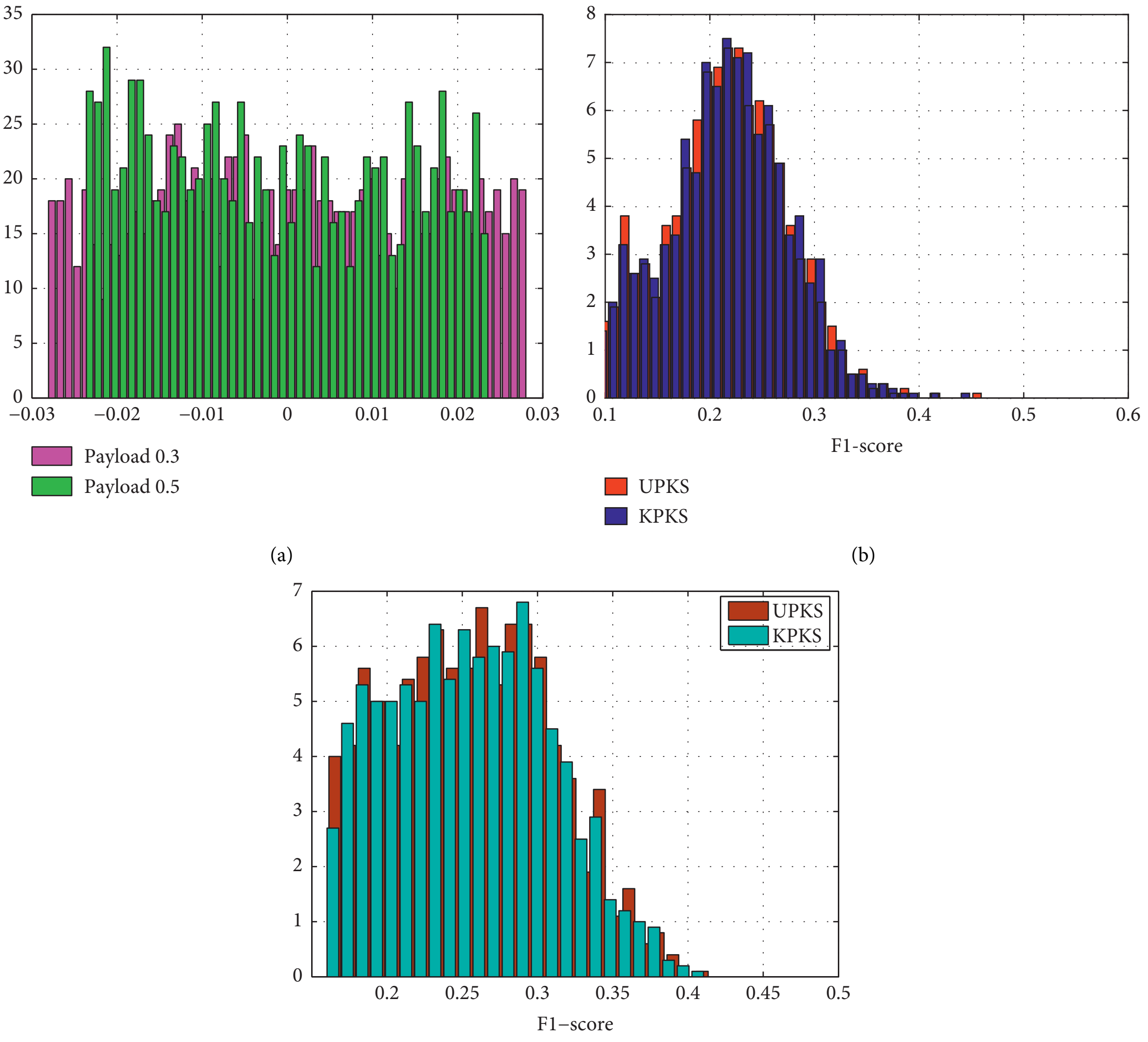
Next, let us further investigate whether the hidden bits can be successfully located relying on the predicted payload. In such a case, it is proposed to compare the F1-score result of UPKS with that of KPKS serving as the baseline ground truth. When the payload is 0.3, two modern steganographic schemes are adopted. Two histograms in each figure are nearly overlapped, meaning that the F1-score of UPKS basically matches that of KPKS (see Figures 8(b) and 9(b) for details). Besides, at the payload 0.5, the comparison results are illustrated in Figures 8(c) and 9(c), respectively, which also verify the effectiveness of our proposed locating algorithm. Moreover, we calculate the statistical parameters of the histogram, referring to mean and variance values of both compared histograms. In Figure 8(c), for instance, both mean and variance values of KPKS and UPKS are equal to 0.2556 and 0.0028. Therefore, the experimental results empirically verify that thanks to the accurate prediction of payload, our proposed LAS with NWA can still work very well for locating hidden bits in the case of UPKS.

5.3.4. UPUS (Unknown Payload and Unknown Scheme). Finally, let us evaluate the effectiveness of the proposed locating algorithm in the most difficult scenario, referring to as neither knowing payload nor specific adaptive embedding scheme. In this case, relying on the empirical analysis from KPUS and UPKS, it is proposed to first predict the payload and then locate hidden bits by using our proposed LAS with NWA based on the cost function of WOW steganography.

Also, the results of KPKS serve as the baseline for comparison. As Figure 10 illustrates, by comparing the F1-score between KPKS and UPUS, the overall result of KPKS is

Table 5: Averaged F1-score comparison between LAS with NWA and LAS without NWA [46] at the payload 0.3, in the case of KPUS.

True scheme	Predicted scheme		
	WOW	S-UNIWARD	HILL
WOW	**0.318 5**, 0.307 6	0.231 1, 0.253 2	0.263 6, 0.272 8
S-UNIWARD	**0.255 5**, 0.219 2	0.198 3, 0.191 7	0.220 9, 0.200 2
HILL	0.268 7, 0.265 1	0.196 6, 0.224 6	**0.281 2**, 0.277 0



(a)

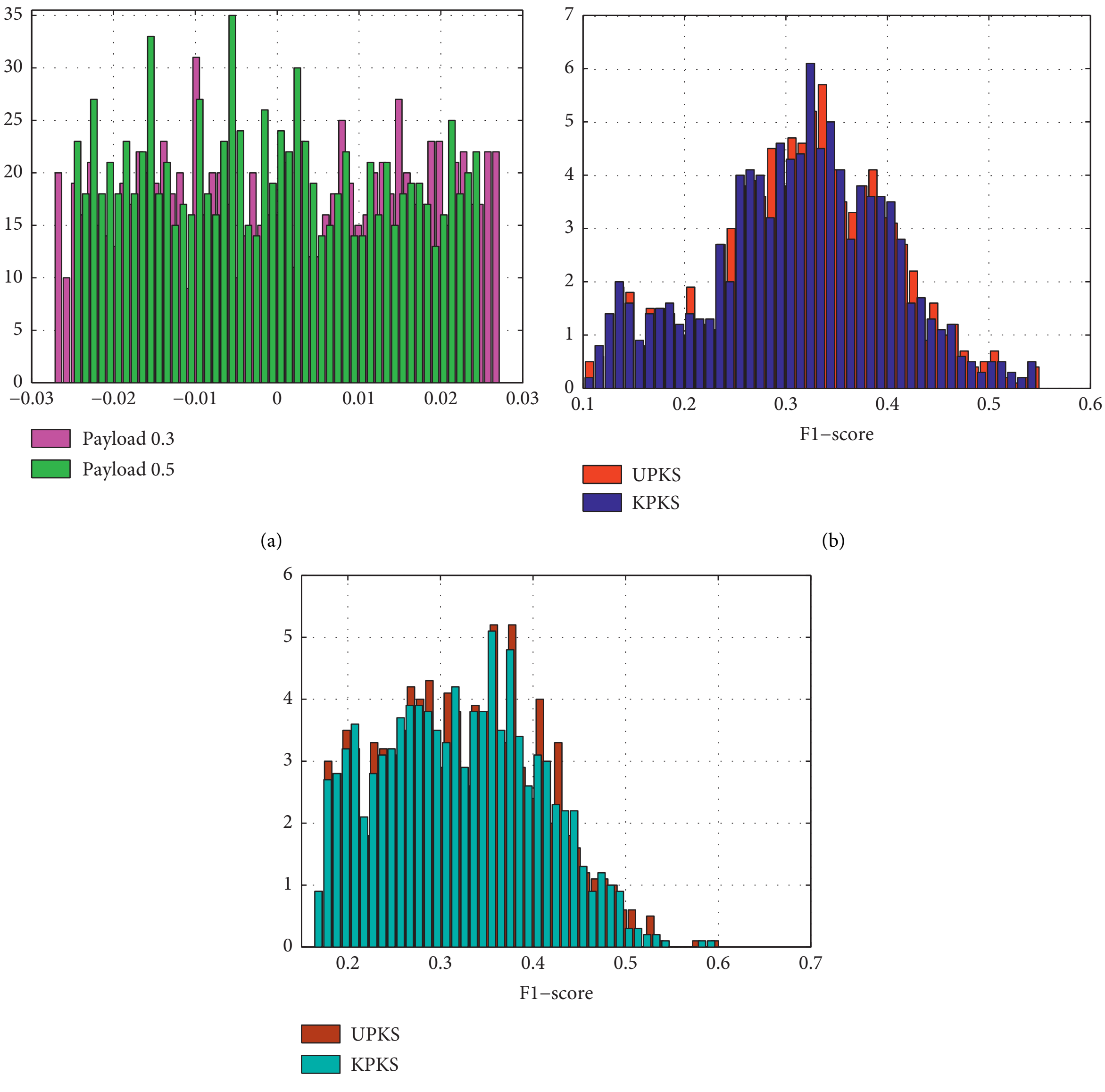
(b)

(c)

Figure 8: : Performance of our proposed locating algorithm targeting S-UNIWARD. (a) Error histogram between the predicted payload and its ground truth. (b) Histogram of F1-score in the case UPKS and KPKS at the payload 0.3. (c) Histogram of F1-score in the case UPKS and KPKS at the payload 0.5.

obviously superior to that of UPUS, especially at the large-value bin of histogram, meaning that the performance of locating is slightly degraded in the case UPUS. Moreover, the error histogram is also illustrated by making differences between F1-scores of two cases (the results of KPKS minus that of UPUS). As Figure 11 reports, most of the data larger than zero directly validates the better performance of the locating algorithm in the case KPKS. Lack of enough information about a specific amount of payload and embedding scheme unavoidably leads to the fact that the extended modification map is hardly constructed, which more or less impacts the accuracy of locating hidden bits.

5.4. Comparison with Prior Arts. Compared with the baseline prior arts, the superiority of our proposed locating algorithm is experimentally verified. For simplicity and clarity, let us describe the prior arts, referring to as 5 algorithms



(a)

(b)

(c)

Figure 9: Performance of our proposed locating algorithm targeting HILL. (a) Error histogram between the predicted payload and its ground truth. (b) Histogram of F1-score in the case UPKS and KPKS at the payload 0.3. (c) Histogram of F1-score in the case UPKS and KPKS at the payload 0.5.

[35, 36, 38, 39, 42] toward nonadaptive steganography, a novel algorithm [46] toward adaptive steganography, and adaptive steganalysis (not originally designed for locating hidden bits) [48, 49]. Each algorithm is elaborated as follows.

WSR [35]: by using the linear filter, each cover pixel can be approximately estimated. Then, each residual noise is calculated by making the differences between the stego pixel and its estimated cover one. The stego pixel carrying hidden bit is located by comparing the averaged residual noise with the preset threshold, such as 0.25 for instance. Furthermore, by assigning weight to residual noise, the performance of the Weighted Stego Residual (WSR) algorithm is improved. The limitation of it is that the secret key for each image should remain unchanged; it is designed only for LSBR. WAM [36]: relying on an 8-tap Daubechies kernel, pixels in the spatial domain are converted to coefficients in the wavelet domain. After removing low-frequency coefficients (corresponding to subband **LL**), the remaining residual coefficients in subbands **LH**, **HL**, and **HH** are required by adopting Wavelet Absolute Moment (WAM) filter. Similar to WAR, the inversely converted residual noise in the spatial domain is used for location by comparing its magnitude with the preset threshold. The limitation of WAM is that all possible stego images share the same secret key; it is designed only for LSBM. MAP [38]: dependent on the theory Maximum A Posteriori, together with the Viterbi algorithm, the estimation of cover pixels is optimized. Similar to WSR, the residual noise is calculated between stego and cover source. Like WAM and WAS, the limitation of it is that all hidden bits are embedded in the same position for all stego images.

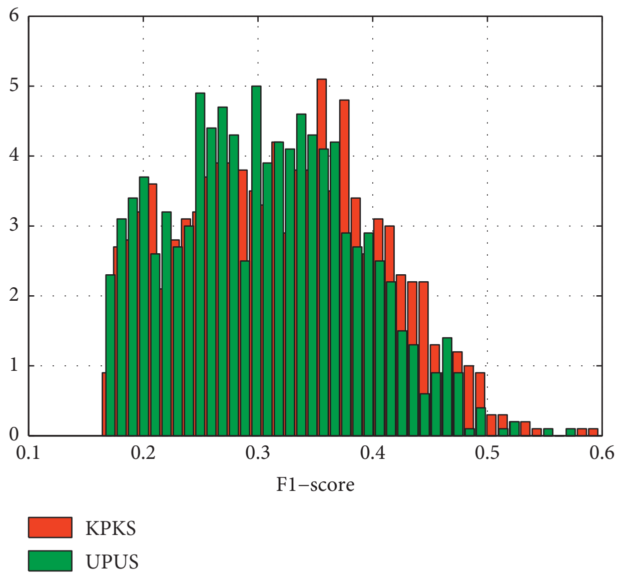


Figure 10: Histogram of F1-score comparison between LAS with NWA in the case UPUS and the baseline KPKS.

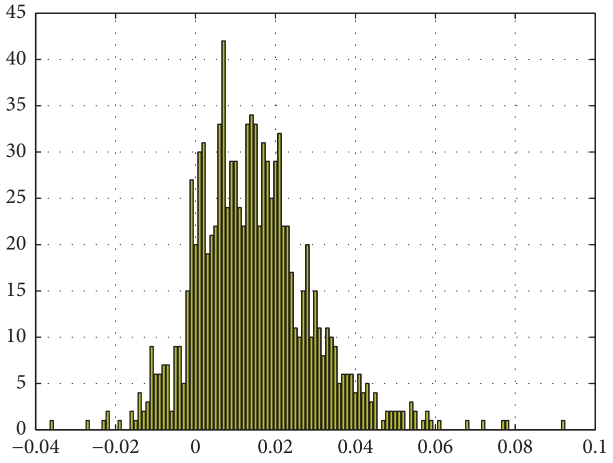


Figure 11: Error histogram of F1-score between LAS with NWA in the case of UPUS and the baseline KPKS.

It is worth noticing that MAP is available for both LSBR and LSBM. MRF [39]: in virtue of the Markov Random Field (MRF), a cover image is predicted by using a given stego one. In particular, dependent on pairwise constraints, the statistical features of a cover image are captured. Then, the designed locator is well performed targeting both LSBR and LSBM. DNN [42]: by taking the problem of payload location as binary classification, relying on the trained model, each pixel is treated as the predicted sample (carrying hidden bit or not), where the mean square of neighboring pixel differences serves as the key element for feature extraction. Moreover, the hand-crafted features are fed into the well-designed DNN for training an efficient model, which is available for both LSBR and LSBM. LAS [46]: by reembedding the bits into the stego image, the modification map is obtained. With the help of embedding cost, the extended version of the modification map guides us to locate the flipped bits in the stego image. The strength of this locating algorithm is to directly target adaptive steganography (LAS), which is totally different from prior arts such as WSR, WAM, WAP, or MRF. Hu's method [48] and Tang's method [49]: these two methods were originally designed for image steganalysis, where the regions [48] or bits [49] with high embedding probability are preferably selected for training an efficient classifier. Thus, we insist on conducting comparison experiments with them.

Also, 10000 grey-level images from BOSSbase ver.1.01 [47] are used for comparing the performance of different locating algorithms. Here, two payloads 0.3 and 0.5 are used to generate the stego images. To enrich the experimental data, it is proposed to adopt both modern adaptive steganography and old nonadaptive steganography. It is worth noting that the number of pixels with hidden bits is fixed when LSBR or LSBM is adopted. For instance, 78643 locations need to be predicted in a 512×512 stego image with a 0.3 payload. It should be noted that MAP [38] and MRF [39] are supervised algorithms, which need to construct the trained model prior to locating hidden bits. Thus, in that case, at the given payload, half the number of images is used for training; another half is used for testing while the remaining algorithms are training-free. For a fair comparison, we should ensure that all the same 5000 images with the same payload are used for locating. Still, the F1-score serves as the comparison metric for evaluating the performance of the locating algorithm.

As Table 6 illustrates, in the case of payload 0.3, WSR is good at locating hidden bits embedded by LSBR while not LSBM. On the contrary, WAM performs very well when LSBM is used for embedding. Those results perfectly match those of [35, 36]. For supervised locating algorithms, MAP cannot only locate secret bits hidden by LSBR but also by LSBM. In addition, DNN performs very close to MAP. However, MRF cannot perform very well. The default setting of the MRF model parameter ω_1 equals 0.9986, which is acquired from the database BOSSbase ver.0.92 of [39]. That probably leads to unsatisfying results. When the payload is increased, 0.5 for instance, the performance of MRF can be further improved, which nearly matches the results of [39]. All the hidden bits embedded by LSBR or LSBM nearly can both be located (see Table 7 for details). That empirically verifies that with increasing the payload, the impact of the inaccurate MRF model parameter is able to be mitigated.

Moreover, it is noticeable that the aforementioned locating algorithms [35, 36, 38, 39, 42] only work when the stego images with hidden bits are embedded in the same position in the spatial domain. The strong assumption largely limits the extension to hidden bits location of adaptive steganography. Since modern adaptive steganography prefers to embed bits relying on the content of the cover image, it hardly holds true that the pixels in the same positions are used for embedding toward different cover images. Thus, the performance of locating algorithms [35, 36, 38, 39, 42] targeting modern steganography is not illustrated. For clarity, we utilize the notation "/" in Tables 6 and 7 denoting the invalid results. However, when targeting old steganography, our proposed LAS algorithms and two

Table 6: F1-score comparison of locating performance from different steganalysis methods, using both modern adaptive steganography (WOW, S-UNIWARD, and HILL) and old nonadaptive steganography (LSBR and LSBM) at the given payload 0.3.

Steganalysis locating method, steganography	WOW	S-UNIWARD	HILL	LSBR	LSBM
WSR [35]	/	/	/	1.000 0	0.280 5
WAM [36]	/	/	/	0.304 2	1.000 0
MAP [38]	/	/	/	1.000 0	1.000 0
MRF [39]	/	/	/	0.625 2	0.803 1
DNN [42]	/	/	/	0.946 4	0.941 2
Hu's method [48]	0.124 9	0.105 9	0.126 5	/	/
Tang's method [49]	0.221 4	0.172 2	0.208 5	/	/
LAS without NWA [46]	0.307 6	0.191 8	0.277 0	/	/
LAS with NWA (ours)	**0.318 4**	**0.198 3**	**0.281 2**	/	/

Table 7: F1-score comparison of locating performance from different steganalysis methods, using both modern adaptive steganography (WOW, S-UNIWARD, and HILL) and old nonadaptive steganography (LSBR and LSBM) at the given payload 0.5.

Steganalysis locating method, steganography	WOW	S-UNIWARD	HILL	LSBR	LSBM
WSR [35]	/	/	/	1.000 0	0.808 1
WAM [36]	/	/	/	0.498 4	1.000 0
MAP [38]	/	/	/	1.000 0	1.000 0
MRF [39]	/	/	/	0.988 9	0.963 6
DNN [42]	/	/	/	0.941 8	0.936 8
Hu's method [48]	0.193 7	0.167 2	0.195 3	/	/
Tang's method [49]	0.356 4	**0.276 7**	0.338 0	/	/
LAS without NWA [46]	0.372 5	0.255 4	0.347 1	/	/
LAS with NWA (ours)	**0.375 3**	0.258 7	**0.348 7**	/	/

adaptive steganalysis types [48, 49] fail. Due to the fact that the hidden bits are embedded randomly, the methods designed by the characteristic of modern steganography become invalid when dealing with LSBR or LSBM. Nevertheless, as Table 6 reports, when locating secret bits hidden by modern steganography, our proposed LAS with NWA outperforms the prior arts.

Additionally, we illustrate the F1-score performance of locating algorithms at the given payload 0.5 in Table 7. As we expected, whatever modern or old steganography is adopted, the performance of locating algorithms is improved compared to the results in Table 6. In fact, when more secret bits are embedded into the cover image, more location hints caused by bit modification can be provided, which definitely results in better detection. It is worth noting that our proposed LAS with NWA performs better than the others when dealing with WOW and HILL and slightly worse than Tang's method [49] when dealing with S-UNIWARD.

6. Conclusion and Limitation

In this paper, we address the problem of locating the hidden bits embedded by modern adaptive steganography. In virtue of the intrinsic property of adaptive steganography, through reembedding secret bits into stego images, we acquire the modification map. Next, based on the extended modification map, together with the neighboring weight algorithm (NWA), the location of hidden bits is further refined, leading to better performance. More importantly, for practical use, we verify the effectiveness of locating hidden bits in the four possible cases. Prior to our study, most literature focused on locating hidden bits embedded by old steganography while ignoring the research of modern adaptive steganography. Meanwhile, a strong assumption should be given, referring to as secret bits embedded in the same position in the spatial domain for many stego images while, in our locating algorithm, only one single stego image is enough to be used for locating hidden bits.

The main limitation of the proposed algorithm is that the predicted flipped bits should be embedded by modern adaptive steganography. In other words, it fails when the old steganographic algorithm is adopted. When comparing the F1-score, we have to admit that the location accuracy of our proposed algorithm is not as good as that of the algorithms specialized in targeting old steganography (see Tables 6 and 7). In further study, we need to further improve the location accuracy targeting adaptive steganography.

Additionally, in the more generalized framework of steganalysis, on the one hand, the steganalyzer usually passively completes the task of binary classification (cover versus stego source), the amount of payload prediction (quantitative steganalysis), payload location, and hidden bits extraction (forensic steganalysis); on the other hand, he/she can also adopt the strategy of actively attacking towards steganography [50], such as interruption of covert communication or disturbing the stego carrier. However, the active disturbance is possibly nontargeted, leading to the fact that if the disturbance is too strong, referring to as randomly adding noise to overwrite the hidden bits in the stego image, for instance, the distortion of stego carrier is not acceptable;

on the contrary, too weak disturbance can hardly achieve the task of active attack.

Nevertheless, in this context, our proposed algorithm raises the promising study of payload location targeting modern adaptive steganography. Although the locations of hidden bits are not very accurately predicted, the modification region caused by embedding can be accurately located, which can indeed further help the steganalyzer actively and purposely disturb the stego image over the targeted region carrying hidden bits while mitigating the distortion caused by additional noise. Thus, it is of great importance that further steps are taken to achieve the goal of active steganalysis. Besides, we can also extend the proposed locating method to the adaptive steganalysis instead of overall feature extraction, such as [48, 49], whose effectiveness has been verified in two references, namely, channel-aware or channel-selection steganalysis for more accurate detection.

Data Availability

Data are available on request to the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Provincial Universities of Zhejiang under grant no. GK219909299001-007, the National Natural Science Foundation of China (Grant nos. U1804263 and 62172435), and the Zhongyuan Science and Technology Innovation Leading Talent Project of China (Grant no. 214200510019).

References

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

[2] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 705–720, 2010.

[3] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proceedings of the 2012 IEEE International workshop on information forensics and security (WIFS)*, pp. 234–239, IEEE, Tenerife, Spain, December 2012.

[4] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proceedings of the Image Processing (ICIP), 2014 IEEE International Conference on, IEEE*, pp. 4206–4210, Paris, France, October 2014.

[5] L. Linjie Guo, J. Jiangqun Ni, and Y. Q. Yun Qing Shi, "Uniform embedding for efficient jpeg steganography," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 814–825, 2014.

[6] L. Guo, J. Ni, W. Su, C. Tang, and Y.-Q. Shi, "Using statistical image model for jpeg steganography: uniform embedding revisited," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2669–2680, 2015.

[7] Z. Zhao, Q. Guan, H. Zhang, and X. Zhao, "Improving the robustness of adaptive steganographic algorithms based on transport channel matching," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1843–1856, 2018.

[8] J. Tao, S. Li, X. Zhang, and Z. Wang, "Towards robust image steganography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 594–600, 2019.

[9] Y. Zhang, X. Luo, Y. Guo, C. Qin, and F. Liu, "Multiple robustness enhancements for image adaptive steganography in lossy channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2750–2764, 2019.

[10] T. Qiao, S. Wang, X. Luo, and Z. Zhu, "Robust steganography resisting jpeg compression by improving selection of cover element," *Signal Processing*, vol. 183, Article ID 108048, 2021.

[11] T. Qiao, C. Zitzmann, F. Retraint, and R. Cogranne, "Statistical detection of jsteg steganography using hypothesis testing theory," in *Proceedings of the Image Processing (ICIP), 2014 IEEE International Conference on,*, pp. 5517–5521, IEEE, Paris, France, October 2014.

[12] T. Qiao, C. Ziitmann, R. Cogranne, and F. Retraint, "Detection of jsteg algorithm using hypothesis testing theory and a statistical model with nuisance parameters," in *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*, pp. 3–13, ACM, Salzburg, Austria, June 2014.

[13] T. Qiao, F. Retraint, R. Cogranne, and C. Zitzmann, "Steganalysis of jsteg algorithm using hypothesis testing theory," *EURASIP Journal on Information Security*, vol. 2015, no. 1, p. 2, 2015.

[14] T. Qiao, X. Luo, T. Wu, M. Xu, and Z. Qian, "Adaptive steganalysis based on statistical model of quantized dct coefficients for jpeg images," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 6, pp. 2736–2751, 2019.

[15] T. Pevny, J. Fridrich, and A. D. Ker, "From blind to quantitative steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 445–454, 2012.

[16] J. Fridrich, M. Goljan, and D. Soukal, "Searching for the stego-key," *Security, Steganography, and Watermarking of Multimedia Contents VI*, vol. 5306, pp. 70–83, 2004.

[17] J. Liu, Y. Tian, T. Han, J. Wang, and X. Luo, "Stego key searching for lsb steganography on jpeg decompressed image," *Science China Information Sciences*, vol. 59, no. 3, Article ID 32105, 2016.

[18] C. Xu, J. Liu, J. Gan, and X. Luo, "Stego key recovery based on the optimal hypothesis test," *Multimedia Tools and Applications*, vol. 77, no. 14, pp. 17973–17992, 2018.

[19] J. Kodovsky̌, J. J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.

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

[21] Y. Ma, X. Luo, X. Li, Z. Bao, and Y. Zhang, "Selection of rich model steganalysis features based on decision rough set α -positive region reduction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 336–350, 2019.

[22] Z. Wang, Z. Qian, X. Zhang, and S. Li, "An improved steganalysis method using feature combinations," in *Proceedings of the International Conference on Artificial Intelligence and*

Security, pp. 115–127, Springer, New York,NY, USA, July 2019.

[23] B. Chen, W. Tan, G. Coatrieux, Y. Zheng, and Y. Q. Shi, "A serial image copy-move forgery localization scheme with source/target distinguishment," *IEEE Transactions on Multimedia*.

[24] B. Chen, X. Liu, Y. Zheng, G. Zhao, and Y.-Q. Shi, *A Robust gan-generated Face Detection Method Based on Dual-Color Spaces and an Improved Xception*, IEEE Transactions on Circuits and Systems for Video Technology, New York, NY, USA.

[25] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2545–2557, 2017.

[26] W. Tang, B. Li, M. Barni, J. Li, and J. Huang, "An automatic cost learning framework for image steganography using deep reinforcement learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 952–967, 2020.

[27] W. Tang, B. Li, M. Barni, J. Li, and J. Huang, "Improving cost learning for jpeg steganography by exploiting jpeg domain knowledge," https://arxiv.org/abs/2105.03867.

[28] J. Butora, Y. Yousfi, and J. Fridrich, "How to pretrain for steganalysis," in *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security*, pp. 143–148, Brussels, Belgium, July 2021.

[29] Y. Yousfi, J. Butora, J. Fridrich, and C. Fuji Tsang, "Improving efficientnet for jpeg steganalysis," in *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security*, pp. 149–157, Brussels, Belgium, July 2021.

[30] W. You, H. Zhang, and X. Zhao, "A siamese cnn for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 291–306, 2020.

[31] W. Ren, L. Zhai, J. Jia, L. Wang, and L. Zhang, "Learning selection channels for image steganalysis in spatial domain," *Neurocomputing*, vol. 401, pp. 78–90, 2020.

[32] J. Zhang, K. Chen, C. Qin, W. Zhang, and N.-H. Yu, "Distribution-preserving-based automatic data augmentation for deep image steganalysis," *IEEE Transactions on Multimedia*, pp. 1–13, 2021.

[33] H. Yang, H. He, W. Zhang, and X. Cao, "Fedsteg: a federated transfer learning framework for secure image steganalysis," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1084–1094, 2020.

[34] M. Chen, M. Boroumand, and J. Fridrich, "Deep learning regressors for quantitative steganalysis," *Electronic Imaging*, vol. 7, pp. 1–7, 2018.

[35] A. D. Ker, "Locating steganographic payload via ws residuals," in *Proceedings of the 10th ACM workshop on Multimedia and security*, pp. 27–32, ACM, Oxford, UK, August 2008.

[36] A. D. Ker and I. Lubenko, "Feature reduction and payload location with wam steganalysis," *Media forensics and security*, vol. 7254, Article ID 72540A, 2009.

[37] Y. Luo, X. Li, and B. Yang, "Locating steganographic payload for lsb matching embedding," in *Proceedings of the 2011 IEEE International Conference on Multimedia and Expo*, pp. 1–6, IEEE, Barcelona, Spain, July 2011.

[38] T.-T. Quach, "Optimal cover estimation methods and steganographic payload location," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, pp. 1214–1222, 2011.

[39] T.-T. Quach, "Cover estimation and payload location using Markov random fields," *Media Watermarking, Security, and Forensics 2014*, vol. 9028, Article ID 90280H, 2014.

[40] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

[41] X. Yan, T. Zhang, L. Xi, and X. Ping, "New method for paylaod location aimed at lsb matching," *Journal of Data Acquisition & Processing*, vol. 31, no. 1, pp. 145–151, 2016.

[42] Y. Sun, H. Zhang, T. Zhang, and R. Wang, "Deep neural networks for efficient steganographic payload location," *Journal of Real-Time Image Processing*, vol. 16, no. 3, pp. 635–647, 2019.

[43] J. Wang, C. Yang, P. Wang, X. Song, and J. Lu, "Payload location for jpeg image steganography based on co-frequency sub-image filtering," *International Journal of Distributed Sensor Networks*, vol. 16, no. 1, Article ID 1550147719899569, 2020.

[44] J. Wang, C. Yang, M. Zhu, X. Song, Y. Liu, and Y. Lian, "Jpeg image steganography payload location based on optimal estimation of cover co-frequency sub-image," *EURASIP Journal on Image and Video Processing*, vol. 2021, no. 1, pp. 1–14, 2021.

[45] J. Liu, C. Yang, J. Wang, and Y. Shi, "Stego key recovery method for f5 steganography with matrix encoding," *EURASIP Journal on Image and Video Processing*, vol. 2020, no. 1, pp. 1–17, 2020.

[46] Q. Liu, T. Qiao, M. Xu, and N. Zheng, "Fuzzy localization of steganographic flipped bits via modification map," *IEEE Access*, vol. 7, pp. 74157–74167, 2019.

[47] P. Bas, T. Filler, and T. Pevný, ""Break our steganographic system": the ins and outs of organizing BOSS," in *Information Hiding*Springer, New York, NY, USA, 2011.

[48] D. Hu, Q. Shen, S. Zhou, X. Liu, Y. Fan, and L. Wang, "Adaptive steganalysis based on selection region and combined convolutional neural networks," *Security and Communication Networks*, vol. 20179 pages, 2017.

[49] W. Tang, H. Li, W. Luo, and J. Huang, "Adaptive steganalysis against wow embedding algorithm," in *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security, ACM*, pp. 91–96, 2014.

[50] J. M. Ettinger, "Steganalysis and game equilibria," in *International Workshop on Information Hiding*Springer, New York, NY, USA, 1998.