Efficient Reversible Data Hiding Simultaneously Exploiting Adjacent Pixels

HANZHOU WU, (Member, IEEE)
School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China
Shanghai Institute for Advanced Communication and Data Science, Shanghai 200444, China
e-mail: h.wu.phd@ieee.org

This work was supported in part by the National Natural Science Foundation of China under Grant 61902235, and in part by the Shanghai Municipal Education Commission and Shanghai Education Development Foundation through the Chen Guang Project under Grant 19CG46.

ABSTRACT As an effective means to content authentication and privacy protection, reversible data hiding (RDH) permits us to hide a payload such as authentication data in a media file. The resulting marked content will not introduce noticeable artifacts. In order to achieve superior payload-distortion performance, the conventional RDH algorithms often exploit the smooth content for data embedding. Since RDH allows both the embedded payload and the raw content to be perfectly reconstructed, it is required that, the altered smooth regions within the cover should be identified without error by a data receiver. Therefore, a core work in RDH is to design an efficient content-aware algorithm that can enable a data hider to take advantages of the smooth cover elements as much as possible while the detection procedure for the marked elements should be invertible to a data receiver. This has motivated the authors to present a novel patch-level selection and breadth-first prediction strategy for efficient RDH in this paper. However, different from many conventional RDH works, the proposed approach allows a data hider to preferentially and simultaneously use adjacent smooth elements as many as possible, which can benefit data embedding procedure a lot. Experiments show that our work significantly outperforms a part of advanced RDH algorithms in terms of the payload-distortion performance, which has demonstrated the superiority and applicability.

INDEX TERMS Reversible data hiding, watermarking, fragile, patch, breadth-first search.

I. INTRODUCTION

Reversible data hiding (RDH) [1]–[3] allows us to modify an image to carry a secret payload without introducing artifacts. Different from steganography [4]–[7], RDH ensures that both the hidden data and the original media content can be perfectly reconstructed. RDH is a fragile technique, meaning that, when a media file containing extra data was manipulated, one will find it is not authentic and the original media content may be not fully reconstructed. RDH has been applied to sensitive applications requiring no degradation of the cover such as medical imagery and military imagery.

Many RDH algorithms are designed to images. A straightforward idea [3], [8] is to losslessly compress pixel values to reserve space for data embedding. To avoid obvious artifacts, the compression often executes on the LSBs of pixels, where the compression ratio is relatively low, which may not provide a sufficient payload. More efficient techniques such as histogram shifting (HS) [1] and difference expansion (DE) [2] have been proposed to achieve a high payload or a low distortion. Most existing RDH systems [9]–[14] use HS or its variants, e.g., prediction error (PE) [9] and prediction error expansion (PEE) [13], [14]. The information-theoretic analysis of RDH [15]–[18] has also been deeply studied.

RDH into encrypted images [19]–[22] has also been studied. The motivation is, a content owner may share images only with an authorized receiver. Thus, before uploading an image to the cloud, the owner may encrypt the original image. After receiving an encrypted image, the manager appends extra data to the encrypted image. RDH in encrypted images is an intuitive and effective way to deal with it. In addition, RDH has also been combined in multimedia signal processing (MSP) applications such as image contrast enhancement [23]. Comparing with conventional MSP methods, a significant advantage for RDH-based MSP methods is that it can protect the original content from permanently distorted.
A RDH system can be evaluated by the payload-distortion performance [24], [25]. Namely, for a payload, it is required to keep distortion as low as possible. In other words, for a fixed distortion level, it is desirable to embed as many message bits as possible. Strong spatial correlations exist between neighboring pixels in an image. It inspires researchers to design prediction based RDH algorithms, which use the prediction-errors (PEs) to carry a payload. A common operation [11], [24] used in RDH is therefore to generate a prediction error histogram (PEH) for data embedding. On the one hand, the PEs are noise-like components of the cover, meaning that, the slight modification to PEs would not result in obvious artifacts. On the other hand, as a pooled vector, the PEH allows us to embed secret data in an effective way. It indicates that, to provide superior performance, the design of a system may focus on PEH generation and its modification. Lots of works [26]–[32] are developed along this line.

In this paper, we present a new pixel selection and prediction strategy to PEH generation for efficient RDH. Different from many previous works, our method allows a hider to preferentially and simultaneously use adjacent smooth elements as many as possible, which can benefit data embedding procedure a lot. Our experiments show that the proposed work ensures reversibility and significantly outperforms related works in terms of payload-distortion performance, which shows superiority.

This rest are organized as follows. We revisit prior arts and clarify motivation in Section II. We show the framework of proposed work in Section III. The details of proposed work are given in Section IV, followed by experiments and analysis in Section V. Finally, we conclude this paper in Section VI. This work is an extension version of the paper [33].

II. PRIOR ARTS REVISITED AND MOTIVATION

Many RDH works generate a Laplacian-like PEH [34], [35], which benefits the payload-distortion performance a lot. This requires a well-designed predictor. Regardless of designing a predictor, it is desirable to choose smooth content for RDH since the prediction accuracy would be very high, namely, many small PEs can be collected. Accordingly, a necessary work for advanced RDH systems is to identify smooth content. However, to reconstruct the embedded payload and the original content, a receiver should be able to identify the smooth content. Thus, a RDH system should guarantee that, detecting marked locations is reconstructable.

There are two classical methods to deal with the aforementioned problem. One [14] is to divide the cover pixels into two sets, denoted by $S_1$ and $S_2$ where $|S_1| \approx |S_2|$. We take Fig. 1 (a) for example. The pixels in $S_1$ are used for predicting the pixels in $S_2$. The PEs of pixels in $S_2$ are determined to constitute a PEH used for data embedding. Thereafter, one may use the marked $S_2$ to predict $S_1$ and generate another PEH used for carrying the rest payload. It is seen that, the prediction accuracy may decline when to use the marked $S_2$ to predict $S_1$ since the pixels in $S_2$ have been altered, which may not benefit the payload-distortion performance.

Another method [26] is to set $|S_1| \ll |S_2|$, for which the pixels in $S_1$ are usually unchanged (in some cases, one may use some pixels in $S_1$ to store system parameters, e.g., the secret key). We take Fig. 1 (b) for example. One may use the pixels in the first row and first column to constitute $S_1$ and the rest belong to $S_2$. Thus, the pixels in $S_2$ can be orderly predicted and embedded, e.g., from left to right and from top to bottom. However, in some cases, it may not benefit the performance since the smoothness of pixels may not match the embedding order, e.g., assuming that “Region I” in Fig. 1 (b) is relatively more complex than “Region II”, if we orderly process the cover pixels from left to right and from top to bottom, we may alter “Region I” before “Region II”, which is not desirable from performance optimization view.

One can use a local-complexity function [14] to evaluate the smoothness of a pixel such that smooth pixels can be embedded first. However, in many existing works, the prediction context for a pixel heavily relies on its neighbors. It indicates that, though a smooth pixel may be embedded, its neighbors which may be also smooth pixels should be unchanged during embedding, e.g., in Fig. 1 (a), \{u, v, w, x\} are unchanged before predicting A. Namely, many adjacent smooth pixels may not be simultaneously used for data embedding, e.g., \{u, v, A\} in Fig. 1 (a) may not be embedded at the same time, which provides the room for improvement.

It motivates the authors in this paper to introduce a novel patch-level selection and breadth-first prediction strategy to take advantages of smooth pixels as much as possible. We expect to preferentially and simultaneously use many adjacent smooth pixels for data embedding. And, a data receiver can identify the marked smooth pixels without error, ensuring that, both the secret message and the original image content can be perfectly recovered. In the following section, we will introduce the proposed method in detail.

III. GENERAL FRAMEWORK

With an image $\mathbf{x} = \{x_{i,j}\}_{i,j=0}^{w} \in \mathcal{I} = \{0, 1, \ldots, 2^d - 1\}$, $d > 0$, e.g., $d = 8$ for 8-bit grayscale image, the proposed
work embeds a payload \( m = \{ m_i \}_{i=1}^c \subset \{ 0, 1 \}^c \), into \( x \), resulting in a marked image \( y = \{ y_{i,j} \}_{i=1,j=1}^{m} \), \( y_{i,j} \in \mathbb{T} \). For compactness, we sometimes consider \( x \) as a set including all pixels and \( x_{i,j} \) as the pixel at \((i,j)\) whose value is \( x_{i,j} \).

A patch is defined as a small matrix sized \( r \times c \), where \( 1 \leq r \leq h, 1 \leq c \leq w \). The proposed work first collects a number of disjoint patches from \( x \), each of which is then classified as “a smooth patch” or “a complex patch”. Only smooth patches are used for pixel prediction and data embedding. The complex patches are unchanged. Then, by using proposed breadth-first prediction procedure, the pixels to be potentially embedded in smooth patches are predicted, which allows a data hider to collect all required PEs. By computing the local-complexity of each pixel to be potentially embedded, an ordered PE sequence can be generated. Thereafter, with optimized HS parameters, \( m \) can be successfully embedded into the corresponding PEH, finally resulting in a marked image \( y \). Notice that, to avoid the underflow and overflow problem, a losslessly compressed location map has also been self-embedded into the PEH together with \( m \).

At the data receiver side, according to the necessary parameters extracted from \( y \), one can successfully identify the marked pixels and determine the marked PE sequence. Thus, with the HS parameters extracted from \( y \), \( m \) can be perfectly reconstructed. Meanwhile, the original image \( x \) can be also recovered according to the extracted location map and other necessary information. Fig. 2 shows the sketch for the proposed work. In Section IV, we detail each important part.

**IV. DETAILS OF PROPOSED METHOD**

**A. PATCH-LEVEL SELECTION**

The proposed patch-level selection procedure divides \( x \) into two disjoint subsets, denoted by \( S_A \) and \( S_B \). Both \( S_A \) and \( S_B \) are further divided into two disjoint subsets, i.e., \( S_A^{(0)} \), \( S_A^{(1)} \), \( S_B^{(0)} \), and \( S_B^{(1)} \). We have \( x = S_A \cup S_B \), \( S_A = S_A^{(0)} \cup S_A^{(1)} \), and \( S_B = S_B^{(0)} \cup S_B^{(1)} \). \( S_A^{(0)} \) will be unchanged and used for predicting the pixels in \( S_A^{(1)} \). The pixels in \( S_B^{(1)} \) will carry \( m \). \( S_B^{(0)} \) will be unchanged, and \( S_B^{(1)} \) will be used for storing the secret key and data embedding parameters. For the sake of simplicity, in default, we set \( b^{(1)} = \{ 0, 1 \}^r \times c \). Notice that, one may redefine \( S_B^{(1)} \).

With the patch size, by processing \( x \) from left to right and from top to bottom, we can collect \( N_p = \lfloor \frac{h-1}{r} \rfloor \times \lfloor \frac{w-1}{c} \rfloor \) patches. Let \( p_k = \{ p_{i,j}^{(k)} \}_{r \times c} \) be the \( k \)-th patch indexed by a row-by-row manner. For each \( p_k \), we randomly generate a binary matrix \( b_k = \{ b_{i,j}^{(k)} \}_{r \times c} = \{ 0, 1 \}^r \times c \) by a key. And,

\[
\sum_{i=1}^{r} \sum_{j=1}^{c} b_{i,j}^{(k)} = a^{(k)},
\]

where \( a^{(k)} \geq 1 \) is a predetermined integer parameter.

For each \( p_k \), we determine a set \( s_k \) including such elements that they are corresponding to \( 1 \) in \( b_k \), namely,

\[
s_k = \{ p_{i,j}^{(k)} | 1 \leq i \leq r, 1 \leq j \leq c, b_{i,j}^{(k)} = 1 \}.
\]

For \( s_k \), the difference between the maximum element and the minimum element can be therefore computed as:

\[
d^{(k)} = \max \{ s | s \in s_k \} - \min \{ s | s \in s_k \}.
\]

We classify \( p_k \) as a smooth patch if \( d^{(k)} \leq \beta^{(k)} \). Otherwise, \( p_k \) is considered as a complex patch. Here, \( \beta^{(k)} \geq 0 \) is also a predetermined integer parameter. A smooth patch implies that, it would be suitable for data embedding. A complex patch will not be embedded. Accordingly, we determine:

\[
S_A = \bigcup_{d^{(k)} \leq \beta^{(k)}} p_k.
\]
Then, $S^{(0)}_A$ and $S^{(1)}_A$ are determined as:

$$
S^{(0)}_A = \bigcup_{\alpha(\beta(x)) = 0} \beta(x),
S^{(1)}_A = \bigcup_{\alpha(\beta(x)) = 1} \beta(x).
$$

Furthermore, $S_B$ and $S^{(0)}_B$ are determined as:

$$
S_B = x \setminus S_A,
S^{(0)}_B = S_B \setminus S^{(1)}_B.
$$

The patch-level selection procedure finally generates four disjoint pixel-sets depending on a sequence of parameters $\{\alpha^{(1)}, \ldots, \alpha^{(N_p)}\}$ and $\{\beta^{(1)}, \ldots, \beta^{(N_p)}\}$. For simplicity, in default, we will use $\alpha^{(1)} = \ldots = \alpha^{(N_p)} = \alpha$ and $\beta^{(1)} = \ldots = \beta^{(N_p)} = \beta$. Thus, we only set two parameters.

### B. BREADTH-FIRST PREDICTION

Once we collect four pixel-sets, we predict the pixels in $S^{(1)}_A$ and collect PEs for data embedding. Since we expect to simultaneously use adjacent smooth pixels for data embedding, we cannot directly use the conventional predictors. We propose a breadth-first prediction algorithm to determine the prediction values of pixels in $S^{(1)}_A$ by only using $S^{(0)}_A$.

The proposed algorithm corresponds to an iterative process. During the prediction process, we will select a specific pixel from $S_A$. And, we will predict all its adjacent non-predicted pixels belonging to $S^{(1)}_A$ simultaneously. The “breadth” prediction process will be terminated when all required pixels are predicted. This “breadth-first” perspective requires a low computational cost, i.e., $O(4|S_A|)$, since each pixel will be selected and processed at most once.

Let $\hat{x}_{i,j}$ denote the prediction value of $x_{i,j}$. We first append all pixels in $S^{(0)}_A$ to an empty queue $Q$, and mark pixels in $S^{(0)}_A$ as processed. Then, iteratively, we select a pixel $x_{i,j}$ from $Q$ and perform the following steps until $Q$ is empty.

**Step 1** Collect all the adjacent pixels of $x_{i,j}$. The collected pixels should be in $S^{(1)}_A$ and have not been marked as processed. A pixel $x_{u,v}$ is adjacent to $x_{i,j}$ if $|u - i| + |v - j| = 1$.

**Step 2** For each collected pixel $x_{u,v}$ (if any), if $x_{u,v} \in S^{(0)}_A$, we predict $x_{u,v}$ by $\hat{x}_{u,v} = x_{i,j}$, otherwise we must have $x_{u,v} \in S^{(1)}_A$ and predict $x_{u,v}$ by $\hat{x}_{u,v} = \hat{x}_{i,j}$.

**Step 3** Append all collected pixels (if any) to $Q$, and remove $x_{i,j}$ from $Q$. Mark the collected pixels as processed.

Algorithm 1 shows the pseudocode. We take Fig. 3 for explanation. Fig. 3 (a) shows the original pixel values. The patch size is $3 \times 3$, and $\alpha = 2$. Moreover, $S^{(0)}_A = \{x_{1,3}, x_{2,6}, x_{3,3}, x_{3,4}\}$. Initially, we have $Q = \{x_{1,3}, x_{2,6}, x_{3,3}, x_{3,4}\}$. Then, we select $x_{1,3}$ out from $Q$, and compute $\hat{x}_{1,2} = \hat{x}_{1,4} = \hat{x}_{2,3} = x_{1,3} = 162$. We further update $Q$ as shown in Fig. 3 (b). We continue to select $x_{2,6}$ out from $Q$, and compute $\hat{x}_{1,6} = \hat{x}_{2,5} = x_{3,6} = x_{2,6} = 157$, which is described in Fig. 3 (c). Fig. 3 (c) also shows the updated $Q$ after processing. Fig. 3 (d, e, f, g) show the subsequent prediction operations, e.g., in Fig. 3 (f), $x_{1,1}$ is predicted from $x_{1,2}$ by $\hat{x}_{1,1} = \hat{x}_{1,2}$.

### Algorithm 1 Pseudocode for Breadth-First Prediction

**Input:** cover image $x$ and four sets $S^{(0)}_A$, $S^{(1)}_A$, $S^{(0)}_B$, $S^{(1)}_B$.

**Output:** prediction values of pixels in $S^{(1)}_A$.

1. Append all pixels in $S^{(0)}_A$ to an empty $Q$.
2. Mark all pixels in $S^{(0)}_A$ as processed.
3. While $Q$ is not empty do
4. Select a pixel $x_{u,v}$ from $Q$, collect all its adjacent pixels (if any) in $S^{(0)}_A$ that have not been processed.
5. For each collected pixel $x_{u,v}$ (if any) do
6. If $x_{u,v}$ is in $S^{(0)}_A$ then
7. $\hat{x}_{u,v} = x_{i,j}$
8. Else
9. $\hat{x}_{u,v} = \hat{x}_{i,j}$
10. End if
11. End for
12. Append all collected pixels (if any) to $Q$, and remove $x_{i,j}$ from $Q$.
13. Mark all collected pixels (if any) as processed.
14. End while
15. Return prediction values of pixels in $S^{(1)}_A$.

As shown in Fig. 3 (g), we have $Q = \{x_{2,3}, x_{1,6}, x_{2,5}, x_{3,6}, x_{3,2}, x_{2,4}, x_{3,5}, x_{1,1}, x_{2,2}, x_{1,5}\}$. Thereafter, we will select $x_{2,3}$ out from $Q$. However, we will not use $x_{2,3}$ to predict its neighbors since all its neighbors have been processed. Moreover, no pixel will be appended to $Q$, and $x_{2,3}$ should be further removed from $Q$. The same operation will be orderly applied to $x_{1,6}, x_{2,5}$ and $x_{3,6}$. In this way, we finally have $Q = \{x_{3,2}, x_{2,4}, x_{3,5}, x_{1,1}, x_{2,2}, x_{1,5}\}$.

We continue to select $x_{3,2}$ out from $Q$, and compute $\hat{x}_{3,1} = \hat{x}_{3,2} = 162$. Fig. 3 (h) shows the corresponding result. Then, $x_{3,2}$ and $x_{3,5}$ will not be used to predict their neighbors. Thus, $Q$ is updated as $\{x_{1,1}, x_{2,2}, x_{1,5}, x_{3,1}\}$. The subsequent operations are straightforward. Fig. 3 (f) shows all the required prediction values, which are close to the raw values.

The prediction algorithm ensures that, the prediction value of a pixel in $S^{(1)}_A$ is predicted from its nearest pixel that belongs to $S^{(0)}_A$. Obviously, the Manhattan distance is at least 1. And, its upper bound is $r + c$. When we keep $r + c$ low, the prediction accuracy will be satisfactory. It relies on the assumption that strong spatial correlations exist between smooth pixels with a small distance.

### C. DATA EMBEDDING

After predicting all the pixels in $S^{(1)}_A$, we can determine the PEs of the pixels. For each $x_{i,j} \in S^{(1)}_A$, the PE is defined as:

$$
e_{i,j} = x_{i,j} - \hat{x}_{i,j}.
$$

We need to generate an ordered PE sequence for embedding $m$. It is quite desirable to further sort the PEs according to their local-complexities such that small PEs can be used first. This perspective has been utilized in many existing works [11], [14]. We use the prediction values of pixels in
The local-complexity for each $x_{i,j} \in S_A^{(1)}$, we determine

$$N(x_{i,j}) = \{x_{i',j'} \in S_A | i - i' + |j - j'| \leq 1\}. \quad (8)$$

The local-complexity for $x_{i,j}$ is then computed as:

$$\rho_{i,j} = \max\{x_{u,v}^o \in N(x_{i,j})\} - \min\{x_{u,v}^o \in N(x_{i,j})\}, \quad (9)$$

where

$$x_{u,v}^o = \begin{cases} x_{u,v}, & \text{if } x_{u,v} \notin S_A^{(0)}, \\ \hat{x}_{u,v}, & \text{if } x_{u,v} \in S_A^{(1)}. \end{cases} \quad (10)$$

We take $x_{3,6}$ in Fig. 3 (i) for explanation. $\rho_{3,6}$ is computed as $\rho_{3,6} = \max\{\hat{x}_{3,6}, x_{2,6}, \hat{x}_{3,5}\} - \min\{\hat{x}_{3,6}, x_{2,6}, \hat{x}_{3,5}\} = 4$.

A smaller local-complexity implies better prediction accuracy. Accordingly, we can sort all PEs in an increasing order of the local-complexity so that small PEs can be embedded first. We use $e = \{e_i\}_{i=1}^{n_e}$ to represent the sorted PE sequence. $m$ will be embedded into $e$ by using two pairs of peak-zero points. Let $(l_p, l_z)$ and $(r_p, r_z)$ denote the two pairs of peak-zero points. It meets that $l_z < l_p < r_p < r_z$. For a

$$e_i + b, \quad \text{if } e_i = r_p, \quad \hat{e}_i = \begin{cases} e_i + 1, & \text{if } r_p < e_i < r_z, \\ e_i - 1, & \text{if } l_z < e_i < l_p, \\ e_i, & \text{otherwise}. \end{cases} \quad (11)$$

$\hat{e}_i$ will be added to the prediction value of the corresponding pixel so as to generate the marked pixel. It is mentioned that, we will terminate the data embedding procedure when $m$ is fully embedded. It means that, there is a PE position $t \leq n_e$ that all PEs $\{e_i | i > t\}$ are unchanged. It is necessary to optimize $(l_p, l_z)$ and $(r_p, r_z)$ such that the distortion by embedding $m$ into $\{e_i\}_{i=1}^{n_e}$ can be kept low. For a specific $t$, we can easily determine the corresponding PEH, i.e.,

$$h(k) = \frac{1}{|\{e_i = k | 1 \leq i \leq t\}|}. \quad (12)$$

It is inferred that, to carry $m$, we have $h(l_p) + h(r_p) = l$, where $l$ is the length of $m$. Moreover, $e_i \in \{l_p, r_p\}$. It is always assumed that $h(l_z) + h(r_z) = 0$ for natural images. Since $m$ looks like a random bit-stream, we can estimate the data.
embedding distortion, denoted by
\[ D_l(l_p, l_z; r_p, r_z) = \frac{l}{2} + \sum_{h(l_p, r_p, l_z, r_z)} h(k). \] (13)

We call \( t \) usable if there exist \((l_p, l_z)\) and \((r_p, r_z)\) meeting that \( h(l_p) + h(r_p) = l \) and \( e_i \in \{ l_p, r_p \} \). Accordingly, we can collect all usable \( t \), which can be done during the process of orderly collecting PEs [11]. For each usable \( t \), we can further determine the suboptimal \((l_p, l_z)\) and \((r_p, r_z)\) by minimizing Eq. (13) with a complexity \( O(4^2l) \). Then, we can get global-optimal \((l_p, l_z)\) and \((r_p, r_z)\) from different usable \( t \) with a complexity \( O(n_r) \). In this way, \( m \) can be embedded. It is mentioned that, when the payload size is large, one may use multiple PEH bins to carry the secret payload. Moreover, the used multiple PEH bins can be optimized as well.

D. PARAMETER EMBEDDING AND PARAMETER EXTRACTION

For reversibility, the system parameters including the secret key, \( r, c, \alpha, \beta, (l_p, l_z) \) and \((r_p, r_z)\) will be embedded into the LSBS of some pixels belonging to \( S^{A(1)} \) (by LSB replacement), the original LSBS of which are considered as a part of \( m \) that is embedded into \( S^{A(1)} \). We use 16 bits to store the secret key. And, \( r \) (4 bits), \( c \) (4 bits), \( \alpha \) (6 bits) and \( \beta \) (4 bits) are stored with 18 bits. \( 4d + 4 \) bits are used to store \( l_p, l_z, r_p \) and \( r_z \), each of which requires \( d + 1 \) bits. In addition, before data embedding, a part of boundary pixels with a value of 0 or \( 2^d - 1 \) in \( S^{A(1)} \) are adjusted into the range \([1, 2^d - 2]\) and recorded to constitute a location map, which will be losslessly compressed by arithmetic coding. The compressed location map is also considered as a part of \( m \). Thus, the size of pure payload is \( L = l - 38 - 4d - C_{LM} \), where \( C_{LM} \) is the size of losslessly compressed location map. In most cases, \( C_{LM} \) for a natural image could be quite small, e.g., the standard Lena image has no boundary pixels in our experiments, meaning that, its impact on \( L \) can be roughly ignored. In our experiments, we will use \( l \) to denote the payload measurement. For a receiver, after obtaining the marked image \( y \), he should extract the embedded parameters at first for subsequent operations. The parameter extraction operation is straightforward.

E. DATA EXTRACTION AND IMAGE RECOVERY

By extracting \((r, c)\) from LSBS of some pixels, a receiver can collect all \( N_p \) patches. The \( N_p \) binary matrices are computed by the secret key and \( \alpha \). With \( \beta \) and the binary matrices, the receiver can identify smooth patches. Accordingly, \( S^{A(0)}_A, S^{A(1)}_A, S^{A(0)}_B \) and \( S^{A(1)}_B \) can be perfectly reconstructed.

Since \( S^{A(1)}_A \) is unchanged, the receiver can reconstruct the prediction values of pixels in \( S^{A(1)}_A \). And, the corresponding marked PEs can be all collected. As the local-complexity of a pixel only relies on the prediction values of pixels in \( S^{A(1)}_A \) and the original values of pixels in \( S^{A(0)}_A \), the receiver can generate a marked PE sequence identical to the sender side. Thus, the receiver can extract \( m \) without error. For a marked PE \( \tilde{e}_i, \) “0” is collected if \( \tilde{e}_i \in \{ l_p, r_p \} \); otherwise, “1” is extracted if \( \tilde{e}_i \in \{ l_p - 1, r_p + 1 \} \). The original PE \( e_i \) is reconstructed by:

\[
e_i = \begin{cases} 
\hat{e}_i - 1, & \text{if } r_p < e_i \leq r_z, \\
\hat{e}_i + 1, & \text{if } l_z \leq e_i < l_p, \\
\hat{e}_i, & \text{otherwise}.
\end{cases}
\] (14)

\( e_i \) will be added to the prediction value of the corresponding pixel to reconstruct the preprocessed pixel value at the sender side. With \( m \), the receiver can further obtain the original LSBS of specific pixels in \( S_B^{A(1)} \) as well as the losslessly compressed location map, which allows \( x \) to be finally recovered without error. Obviously, the secret message can be parsed as well.

V. EXPERIMENTS AND ANALYSIS

In this section, we conduct experiments for evaluation. We first take standard images1 Airplane, Lena, Tiffany, Peppers, Baboon, Boat shown in Fig. 4 varying from smooth to complex for experiments. We apply proposed patch-level selection procedure to the images. Fig. 5 shows the resulting distribution maps. It is seen that, the proposed work selects smooth content out from an image. The selected pixels to be potentially embedded could be adjacent to each other. It implies that, many adjacent smooth pixels could be simultaneously embedded, which has the potential to provide superior performance.

FIGURE 4. Six standard images sized 512 x 512 x 8: (a) Airplane, (b) Lena, (c) Tiffany, (d) Peppers, (e) Baboon, and (f) Boat.

FIGURE 5. The resulting distribution maps by using patch-level selection procedure: (a) Airplane, (b) Lena, (c) Tiffany, (d) Peppers, (e) Baboon, and (f) Boat. Here, \( r = c = \alpha = \beta = 4 \). The white pixels belong to \( S_B^{A(1)} \) and black for \( S_B^{A(0)} \).

It is also observed that, due to the diversity, different images have a different number of smooth pixels, e.g., Baboon has a less number of smooth pixels comparing with other images. It indicates that, generally, different images will result in different payload-distortion performance. From the performance optimization view, to provide satisfactory performance, it is necessary to choose suitable parameters so that the payload can be fully carried and extracted without error. Meanwhile, the distortion should be as low as possible.

1 Available at http://sipi.usc.edu/database/database.php?volume=misc
The overall smoothness for different \((\alpha, \beta)\): (a) Airplane, \(r = c = 4\), (b) Airplane, \(r = c = 8\), (c) Baboon, \(r = c = 4\), (d) Baboon, \(r = c = 8\).

**A. PARAMETER SELECTION**

We should guarantee that, the preselected parameters, i.e., \((r, c, \alpha, \beta)\), enable a hider to generate a sharp PEH that can fully carry the payload. This can be done by tuning \((r, c, \alpha, \beta)\) within a small range. On the other hand, the aforementioned HS optimization algorithm can be adopted to keep distortion low. Actually, the setting of \((r, c, \alpha, \beta)\) does not only affect the size of embeddable payload, but also affect the performance since different \((r, c, \alpha, \beta)\) often result in different \(S_A\) having different statistical characteristics. Accordingly, it is desirable to analyze the statistical characteristics of \(S_A\) due to different \((r, c, \alpha, \beta)\), which could guide us to select suitable parameters.

For an image, we collect four disjoint pixel-sets. Since only \(S_A^{(1)}\) carries \(m\), we therefore define the overall smoothness as:

\[
\rho = \left( \sum_{x_{i,j} \in S_A^{(1)}} \frac{|x_{i,j} - f(x_{i,j})|^2}{|S_A^{(1)}|} \right)^{1/2},
\]

where

\[
f(x_{i,j}) = \left( x_{i-1,j} + x_{i+1,j} + x_{i,j+1} + x_{i,j-1} \right)/4.
\]

Eq. (16) shows the average value of neighbors. It is easy to slightly modify Eq. (16) when a pixel position is out of the image. We apply the pixel selection algorithm to Airplane and Baboon, and compute the corresponding overall smoothness. Fig. 6 shows the results due to different \((\alpha, \beta)\) in cases \(r = c = 4\) and \(r = c = 8\). For the smooth image Airplane, for a fixed \(\alpha\), when \(\beta\) increases, \(\rho\) converges to a certain level. However, for the complex image Baboon, when \(\beta\) increases, \(\rho\) becomes relatively larger, especially for \(r = c = 8\). A small \(\rho\) can be achieved by using small \(\alpha\) and \(\beta\). A small \(\rho\) also means that the selected pixels are more smooth. Therefore, for RDH, we could use small \(\alpha\) and \(\beta\). Since different images may use different \(\alpha\) and \(\beta\), from performance optimization view, it is desirable to optimize \(\alpha\) and \(\beta\) within a small range. Based on this perspective, in our experiments, we will vary \(\alpha\) from 1 to \(\lfloor \frac{r \times c}{2} \rfloor\), and \(\beta\) from 0 to 8 for all images. The \((\alpha, \beta)\) resulting in the best performance will be chosen for RDH. Notice that, the parameter optimization is only available to a data hider. For a receiver, he only extracts the parameters.

It is also seen that, the smoothness in cases \(r = c = 4\) and \(r = c = 8\) are near to each other (though intuitively Baboon has relatively large fluctuation of \(\rho\) compared to Airplane). It indicates that small \(r\) and \(c\) correspond to similar
FIGURE 10. Performance comparison between proposed work and the methods introduced by Dragoi et al. [27], Hsu et al. [35], Luo et al. [29], Sachnev et al. [14], Tsai et al. [9] and Hong et al. [26].

Smoothness. Moreover, as mentioned in Section IV-B, small $r$ and $c$ ensure that, a pixel can be predicted from a pixel (in $S_A^{(0)}$) that has a small Manhattan distance to the present pixel, which can benefit data embedding operation. Based on this perspective, in our experiments, for optimization, we will vary both $r$ and $c$ from 1 to 8. It is also required that $r \times c \geq 2$. And, in our experiments, we use $2\alpha \leq r \times c$, which ensures that $|S_A^{(0)}| \leq |S_A^{(1)}|$, so as to provide a sufficient payload.
After applying the patch-level selection procedure, we predict the pixels in $S_A^{(1)}$. Since the pixels in $S_A^{(1)}$ are relatively smooth, the prediction accuracy will be high. We take Airplane for example. Fig. 7 shows the PEHs for Airplane due to different $\beta$ in case $r = c = \alpha = 4$. It is seen that, the resulting PEHs are quite sharp. We further identify such pixel positions in $S_A^{(1)}$ having a PE value of “0”. Fig. 8 shows the distribution map with $r = c = \alpha = \beta = 4$. The white pixels have a PE value of “0”. It is observed that, many adjacent pixels have a PE of “0”, implying that, the proposed work has good ability to simultaneously use adjacent smooth pixels.

In addition, we use a secret key to randomly generate binary matrices. Different binary matrices correspond to different divisions of the cover. We take Airplane and Baboon for statistical analysis. For specific $(\alpha, \beta, r, c)$, we randomly generate a key and produce the binary matrices for 100 times. In each time, we further determine the overall smoothness. Fig. 9 shows the statistical results. It is seen that, for fixed $(\alpha, \beta, r, c)$, the overall smoothness due to different divisions are relatively stable, implying that, the impact of randomly generating binary matrices can be roughly ignored. It is also observed that, Baboon has relatively larger fluctuation of the overall smoothness comparing with Airplane, which is due to the complex content of Baboon and working mechanism of proposed work. It indicates that, due to the between-difference of images, we need to optimize parameters $(\alpha, \beta, r, c)$, which has been analyzed previously.

B. PERFORMANCE EVALUATION

The above analysis shows that our work has the potential to well predict smooth pixels and exploit smooth pixels for RDH. To evaluate the payload-distortion performance, we compare the proposed work with a part of recently reported HS based RDH works introduced by Dragoi et al. [27], Hsu et al. [35], Luo et al. [29], Sachnev et al. [14], Tsai et al. [9] and Hong et al. [26]. In our experiments, we use a random bit-string to represent the secret message. Fig. 10 shows the comparison results. It is observed that, the PSNR gain is significant at relatively low embedding rates for test images, which indicates that the proposed work significantly outperforms the related works in terms of payload-distortion performance. When the size of payload increases, the performance will decline, resulting in a low PSNR gain.

The performance for Baboon declines significantly, resulting in relatively worse performance at relatively high embedding rates. It is due to the working mechanism of proposed work. In detail, the proposed work aims to use smooth content for data embedding. However, due to the between-image difference, different images actually have different smoothness, meaning that, different images result in different performance (always true for content-dependent works). Moreover, the proposed pixel prediction algorithm is effective for smooth images since in smooth images, the values of adjacent pixels (or pixels with a small distance to each other) are quite close to each other. However, for a complex image, e.g., Baboon, the difference between two adjacent pixels could be very high, meaning that, the number of smooth pixels is limited, which cannot benefit the proposed prediction procedure. From the viewpoint of real-world, more natural images are likely to have relatively sufficient smooth pixels, indicating that, the proposed work has the potential to provide superior performance.

C. SYSTEM SECURITY

Many conventional RDH methods use prespecified content-aware procedure, which, however, according to Kerckhoff’s principle, may allow an unauthorized receiver to identify the marked pixels and further reconstruct the embedded message. E.g., the division patterns shown in Fig. 1 may be not secure, especially when the content-dependent operations are prefixed. For the proposed work, a key is used to generate a random binary matrix for each patch. It ensures that, one cannot fully distinguish between marked pixels and non-marked pixels without required parameters. It means that, it is almost impossible for an unauthorized receiver to reconstruct the embedded message within a short time, which has shown the security of proposed work. It can be said that, a goal of generating random binary matrices is to secure the work.

VI. CONCLUSION AND FUTURE WORK

Improving the utilization of smooth content has been a core topic in RDH. It motivates the authors in this paper to introduce a novel patch-level selection and breadth-first prediction strategy to further take advantages of smooth pixels as much as possible. The proposed system divides a cover image into a number of disjoint patches, which are further classified as smooth patches and complex patches. The pixels in smooth patches have strong spatial correlations and therefore are used for data embedding since the PEs could be kept low. The prediction value of each cover pixel is determined from a pixel in a specific pixel-set that has the minimum Manhattan distance, which relies on the reasonable assumption that the values of adjacent smooth pixels are rather close to each other. Unlike many traditional works, the proposed work enables a data hider to preferentially and simultaneously use many adjacent smooth pixels to carry a payload, which can benefit payload-distortion performance a lot. Experiments have shown that, the proposed work provides superior performance at low embedding rates. A future work is to improve the performance on complex images at high embedding rates.

REFERENCES

[1] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, “Reversible data hiding,” IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar. 2006.
[2] J. Tian, “Reversible data embedding using a difference expansion,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890–896, Aug. 2003.
[3] M. U. Celik, G. Sharma, and A. M. Tekalp, “Lossless watermarking for image authentication: A new framework and an implementation,” IEEE Trans. Image Process., vol. 15, no. 4, pp. 1042–1049, Apr. 2006.
H.-Z. Wu, Y.-Q. Shi, H.-X. Wang, and L.-N. Zhou, “Separable reversible steganography based on DCT and LDA topic classification,” IEEE Trans. Multimedia, vol. 6, no. 3, pp. 501–509, Jun. 2004.

G.-S. Lin, Y.-T. Chang, and W.-N. Lie, “A framework of enhancing image steganography with picture quality optimization and anti-steganalysis based on simulated annealing algorithm,” IEEE Trans. Multimedia, vol. 12, no. 5, pp. 345–357, Aug. 2010.

X. Zhang, F. Feng, and M. Long, “Robust coverless image steganography based on DCT and LDA topic classification,” IEEE Trans. Multimedia, vol. 20, no. 12, pp. 3223–3238, Dec. 2018.

I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, Digital Watermarking and Steganography. San Mateo, CA, USA: Morgan Kaufmann, 2008.

J. Fridrich, M. Goljan, and R. Du, “Invertible authentication,” Proc. SPIE, vol. 3971, pp. 197–208, Aug. 2001.

P. Tsai, Y.-C. Hu, and H.-L. Yeh, “Reversible image hiding scheme using predictive coding and histogram shifting,” Signal Process., vol. 89, no. 6, pp. 1129–1143, Jun. 2009.

X. Li, B. Li, B. Yang, and T. Zeng, “General framework to histogram-shifting-based reversible data hiding,” IEEE Trans. Image Process., vol. 22, no. 6, pp. 2181–2191, Jun. 2013.

H. Wu, H. Wang, and Y. Shi, “PPE-based reversible data hiding,” in Proc. ACM Workshop Inf. Hiding Secur., Jun. 2016, pp. 187–188.

X. Li, W. Zhang, X. Gui, and B. Yang, “Efficient reversible data hiding based on multiple histograms modification,” IEEE Trans. Inf. Forensics Security, vol. 10, no. 9, pp. 2016–2027, Sep. 2015.

X. Li, B. Yang, and T. Zeng, “Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection,” IEEE Trans. Image Process., vol. 20, no. 12, pp. 3524–3533, Dec. 2011.

V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, “Reversible watermarking algorithm using sorting and prediction,” IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 7, pp. 989–999, Jul. 2009.

T. Kalker and F. M. J. Willems, “Capacity bounds and constructions for reversible data-hiding,” Proc. SPIE, vol. 5020, pp. 604–611, Jun. 2003.

F. Willems and T. Kalker, “Coding theorems for reversible embedding,” DIMACS Ser. Discrete Math. Theor. Comput. Sci., vol. 66, pp. 61–78, Apr. 2004.

W. Zhang, B. Chen, and N. Yu, “Improving various reversible data hiding schemes via optimal codes for binary covers,” IEEE Trans. Image Process., vol. 21, no. 6, pp. 2991–3003, Jun. 2012.

S.-J. Lin and W.-H. Chung, “The scalar scheme for reversible information-embedding in gray-scale signals: Capacity evaluation and code constructions,” IEEE Trans. Inf. Forensics Security, vol. 7, no. 4, pp. 1155–1167, Aug. 2012.

X. Zhang, “Reversible data hiding in encrypted image,” IEEE Signal Process. Lett., vol. 18, no. 4, pp. 255–258, Apr. 2011.

H.-Z. Wu, Y.-Q. Shi, H.-X. Wang, and L.-N. Zhou, “Separable reversible data hiding for encrypted palette images with color partitioning and flipping verification,” IEEE Trans. Circuits Syst. Video Technol., vol. 27, no. 8, pp. 1620–1631, Aug. 2017.

Z. Qian, X. Zhang, and S. Wang, “Reversible data hiding in encrypted JPEG bitstream,” IEEE Trans. Multimedia, vol. 16, no. 5, pp. 1486–1491, Aug. 2014.

W. Zhang, H. Wang, D. Hou, and N. Yu, “Reversible data hiding in encrypted images by reversible image transformation,” IEEE Trans. Multimedia, vol. 18, no. 8, pp. 1469–1479, Aug. 2016.

H.-T. Wu, J.-L. Dugelay, and Y.-Q. Shi, “Reversible image data hiding with contrast enhancement,” IEEE Signal Process. Lett., vol. 22, no. 1, pp. 81–85, Jan. 2015.

H.-Z. Wu, H.-X. Wang, and Y.-Q. Shi, “Dynamic content selection-and-prediction framework applied to reversible data hiding,” in Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS), Dec. 2016, pp. 1–6.

B. Ma and Y. Q. Shi, “A reversible data hiding scheme based on code division multiplexing,” IEEE Trans. Inf. Forensics Security, vol. 11, no. 9, pp. 1914–1927, Sep. 2016.

W. Hong, T.-S. Chen, and C.-W. Shiu, “Reversible data hiding for high quality images using modification of prediction errors,” J. Syst. Softw., vol. 82, no. 11, pp. 1833–1842, Nov. 2009.

I.-C. Dragoi, D. Coltuc, and I. Caciula, “Gradient based prediction for reversible watermarking by difference expansion,” in Proc. 2nd ACM Workshop Inf. Hiding Multimedia Secur. (IH&MMSec), Jun. 2014, pp. 35–40.

C. Yang and M. Tsai, “Improving histogram-based reversible data hiding by interleaving prediction,” IET Image Process., vol. 4, no. 4, pp. 223–234, Aug. 2010.

L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, “Reversible image watermarking using interpolation technique,” IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187–193, Mar. 2010.

D. M. Thodi and J. J. Rodriguez, “Expansion embedding techniques for reversible watermarking,” IEEE Trans. Image Process., vol. 16, no. 3, pp. 721–730, Mar. 2007.

I.-C. Dragoi and D. Coltuc, “Local-prediction-based difference expansion reversible watermarking,” IEEE Trans. Image Process., vol. 23, no. 4, pp. 1779–1790, Apr. 2014.

B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, “Pairwise prediction-error expansion for efficient reversible data hiding,” IEEE Trans. Image Process., vol. 22, no. 12, pp. 5010–5021, Dec. 2013.

H. Wu, “Patch-level selection and breadth-first prediction strategy for reversible data hiding,” in Proc. ICASSP-IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 2837–2841.

C. Wang, X. Li, and B. Yang, “Efficient reversible image watermarking by using dynamical prediction-error expansion,” in IEEE Trans. Inf. Forensics Image Process., Sep. 2010, pp. 3673–3676.

F.-H. Hsu, M.-H. Wu, and S.-J. Wang, “Reversible data hiding using side-match predictions on steganographic images,” Multimedia Tools Appl., vol. 67, no. 3, pp. 571–591, Dec. 2013.

X. Zhang, “Reversible data hiding with optimal value transfer,” IEEE Trans. Multimedia, vol. 15, no. 2, pp. 316–325, Feb. 2013.

HANZHOU WU (Member, IEEE) received the B.S. and Ph.D. degrees from Southwest Jiaotong University, Chengdu, China. He was a Visiting Scholar with the New Jersey Institute of Technology, USA. He was a Research Staff with the Institute of Automation, Chinese Academy of Sciences. He is currently an Assistant Professor with Shanghai University. He has published around 20 papers in peer journals and conferences, such as the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE International Workshop on Information Forensics and Security, ACM IH&MMSec, and IS&T Electronic Imaging, Media Watermarking, Security and Forensics. His research interest includes information hiding.

* * *