Dynamic Content Selection-and-Prediction Framework Applied to Reversible Data Hiding

Han-Zhou Wu
School of Inf. Science & Technology
Southwest Jiaotong University
Chengdu 611756, P. R. China
Email: h.wu.phd@ieee.org

Hong-Xia Wang
School of Inf. Science & Technology
Southwest Jiaotong University
Chengdu 611756, P. R. China
Email: hxwang@home.swjtu.edu.cn

Yun-Qing Shi
Dept. of Electrical & Computer Engineering
New Jersey Institute of Technology
Newark, NJ 07102, United States
Email: shi@njit.edu

***Abstract*—The existing reversible data hiding (RDH) methods often use a fixed pixel preselection pattern and predictor to generate prediction errors that are then utilized for embedding secret data. According to Kerckhoffs's principle, this deterministic operation may allow an illegal decoder to successfully reconstruct the marked prediction-error histogram from a marked image, which is not desirable in application scenarios. This has prompted us to propose a dynamic content selection-and-prediction framework for the RDH in this paper. The proposed framework aims to auto-preselect the complex pixels out from a given image to predict the rest pixels that are thereafter exploited to carry the secret data. Comparing with some state-of-the-art algorithms, the proposed technique guarantees that, the illegal decoder will hardly locate the whole marked pixels and determine the marked prediction errors, which can ensure the security level. In our designed framework, after pixel selection and prediction, there exists a lot of freedom to design the data embedding procedure, meaning that, the proposed framework can be applied to the design of an RDH scheme. In the experiments, we simply employ an optimized histogram shifting operation for data embedding after applying the proposed framework. Our experimental results have shown that, the data embedding process can benefit from the proposed pixel selection and prediction procedure with relatively low embedding rates, and therefore significantly outperform some related works in terms of the payload-distortion performance, especially for images with more smooth regions.**

***Index Terms*—Reversible data hiding (RDH), watermarking, histogram shifting, dynamic, multi-layer selection, prediction.**

I. Introduction

Reversible data hiding (RDH) [1] aims to embed extra data such as source information into a host signal (e.g., image), by slightly altering the insignificant components of the host signal, and; the hidden data as well as the original host signal can be fully recovered on the receiver side. Since the RDH allows the original content to be perfectly recovered, as a means of information hiding, it is quite desirable in some sensitive scenarios such as military and remote sensing.

Difference expansion (DE) [1] and histogram shifting (HS) [2] are two common techniques for the RDH. Since better rate-distortion behavior can be achieved by applying DE and/or HS, various RDH algorithms have been developed along this line, e.g., prediction error (PE) [3], [4], integer wavelet transform (IWT) [5], and prediction error expansion (PEE) [6]–[8]. For these algorithms, in most cases, the data hiding procedure can be roughly generalized as two basic steps:

- *Pixel Prediction*: A part of the image pixels are predicted to generate a prediction error histogram (PEH) which follows a Laplacian-like distribution [8] centered at zero.
- *Data Embedding*: The PEH is then exploited to carry the secret data. E.g., a PEH can be divided into expanding regions and shifting regions [7], where the bins in expanding regions are expanded to carry the secret bits, and that in shifting regions are shifted to ensure reversibility.

The pixel prediction process could be separated into two stages. First, a part of the pixels are selected out according to a predetermined rule. Then, the rest pixels are orderly predicted to generate the PEH with a well-designed predictor. In the first stage, the preselected pixels are usually unchanged throughout the pixel prediction, which ensures that the encoder and decoder can compute out the identical prediction. And, the existing methods often use a fixed preselection pattern such as first-row-first-column [4], parity-column [9], chessboard [10], and block [11], e.g., Fig. 1 (c) represents the chessboard pattern, in which pixels in the black region are kept unchanged to predict pixels located at the white region. In the second stage, a suitable predictor is required to obtain accurate estimation of the pixels to be embedded. A common use is the median edge detector (MED) [12], in which the prediction context is composed of the left, top, and top-left pixel of a pixel. This predictor tends to select the vertical/horizontal neighbor when a vertical/horizontal edge is detected, or a linear combination of neighbors if no edge is detected. Another predictor, gradient-adjusted predictor (GAP) [13], works on a context of seven pixels and produces an output by taking account into the existence of the horizontal/vertical edge and its strength. Since the GAP has a more complex prediction mode, it outperforms the MED, which has prompted us to use the GAP or improved versions [8], [14]–[16] for predicting pixels. Instead of a well-designed predictor, the arithmetic average of neighboring pixels is also a good predictor because of its simplicity and efficiency.

The above-mentioned algorithms have moved the RDH field ahead rapidly. Through they work well, there is still room for improvement. For them, the pixel prediction corresponds to such a deterministic procedure that, the prediction of a pixel is a *static function* of the neighboring pixels, meaning that, the

978-1-5090-1138-4/16/$31.00 ©2016 IEEE

predictor always depends on a fixed number of the neighbors. For example, Sachnev *et al*. [3] adopt the chessboard pattern to predict a pixel by using the four neighboring pixels. Only the four neighboring pixels of a pixel are used for prediction and should be kept unchanged through the prediction process at a time. In fact, the prediction can be processed as a *dynamic function* of the neighboring pixels. It means that a pixel will be predicted from an indefinite number of its neighbors, e.g., a pixel may be predicted from its four neighbors, while another one may be predicted with eight neighbors. This will facilitate the prediction accuracy since more context are provided. For example, as shown in Fig.1 (c), if A has been predicted from $\{u, v, w, x\}$, instead of $\{w, x, y, z\}$, B could be predicted from $\{A, w, x, y, z\}$. This perspective has been utilized in our proposed framework. On the other hand, as the conventional methods use fixed preselection patterns (see Fig.1), it allows an unauthorized receiver to determine out the prediction of the marked pixels according to Kerckhoffs's principle, which may lead an attacker to recover the directly embedded information. To address this issue, we propose to utilize a *random* preselection pattern in this paper, meaning that, the used pattern here is always changing due to a key, which will significantly enhance the security level. In addition, as a more sharply distributed PEH often provides a better payload-distortion performance, the proposed framework further attempts to select the complex pixels out (according to the preselection pattern) to predict the smooth pixels to be embedded. The reason is that, the pixels at smooth regions are more likely to be predicted with a smaller prediction error than those at complex regions, instead of embedding bits into pixels at complex regions, the pixels at complex regions are more desirable for prediction (as long as the smooth pixels can carry a required payload), which is helpful to generate a sharply distributed PEH.

The remainder of this paper are organized as follows. In Section II, we give a detailed introduction about the proposed dynamic content selection-and-prediction framework (DCSPF) that can be applied to reversible data hiding. In Section III, we will present a detailed RDH design by utilizing the DCSPF, followed by some experimental results and analysis in Section IV to show the DCSPF performance. Finally, we conclude this paper in Section V.

II. Proposed Dynamic Content Selection-and-Prediction Framework

The proposed DCSPF aims to select a part of the image pixels out from the given image to predict the rest (smooth) pixels, which will be utilized for data hiding. Let $\mathcal{X}$ represent a grayscale image with $n = h \times w$ pixels; for compactness, we will sometimes consider $\mathcal{X}$ as the set containing all pixels in an image and say "pixel $x_{i,j}$" meaning a pixel located at position (i, j) whose grayscale value is $x_{i,j}$. The DCSPF procedure corresponds to a *multi-layer* process. Let m and $\mathcal{S}_k$ $(0 \leq k < m)$ denote the number of layers and the computed pixel-set in the k-th layer. During the k-th $(k > 1)$ layer, the DCSPF utilizes all $\mathcal{S}_t$ $(0 \leq t < k)$ to generate $\mathcal{S}_k$. In this way, all the pixels in $\cup_{t=0}^{m-1}\mathcal{S}_t$ are finally selected out, and the pixels in $\mathcal{X} \setminus \cup_{t=0}^{m-1}\mathcal{S}_t$ will be used for data hiding. In the following, we will give a detailed introduction.

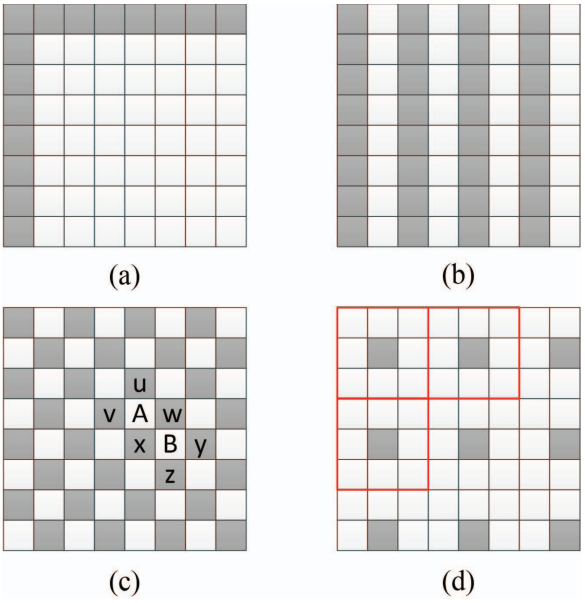

Fig. 1. Example for different pixel preselection patterns: (a) first-row-first-column (b) parity-column (c) chessboard and (d) block.

First, we use a secret key to generate the preselection pattern, where the selected pixels are pseudo-randomly distributed through the given image. Compared to the previous works, it guarantees that, an unauthorized receiver will never produce the correct preselection pattern without the secret key. Noting that, the way to produce the preselection pattern in the existing methods can be considered as a special case of our method.

Let $\mathcal{S}_0$ be the set containing all pixels in the preselection pattern. For the DCSPF, in the k-th $(k \geq 1)$ layer, we first use the pixels in $\cup_{t=0}^{k-1}\mathcal{S}_t$ to predict the rest pixels in $\mathcal{X} \setminus \cup_{t=0}^{k-1}\mathcal{S}_t$. A total of $|\mathcal{S}_k|$ in $\mathcal{X} \setminus \cup_{t=0}^{k-1}\mathcal{S}_t$ are then chosen to constitute $\mathcal{S}_k$ according to their local complexities. Here, $|\mathcal{S}_k|$ denotes the size of $\mathcal{S}_k$. We will always use $|*|$ to denote the size of a set. It is noted that, both m and $|\mathcal{S}_t|$ $(0 \leq t < m)$ are a part of the DCSPF parameters. In the following, we introduce the detailed pixel prediction and selection for the k-th layer.

Pixel Prediction. In the k-th layer, we propose a method called *degree-first prediction* (DFP) to predict the pixels. We collect all the pixels of $\cup_{t=0}^{k-1}\mathcal{S}_t$, and we are to only exploit these pixels to predict the pixels in $\mathcal{X} \setminus \cup_{t=0}^{k-1}\mathcal{S}_t$. The pixels are orderly predicted according to the associated degrees. The *degree* of a pixel is defined as the size of its *degree-set*, which is a subset of its *neighbor-set*. The neighbor-set of a pixel $x_{i,j}$ is defined as:

$$\mathcal{N}_{i,j} = \left\{x_{i+u,j+v} | 1 \leq u^2 + v^2 \leq r^2; u, v \in \mathbf{Z}\right\}. \quad (1)$$

In default, we will use $r = \sqrt{2}$. Thus, except for boundary positions, the neighbor-set of a pixel consists of eight pixels. The degree-set of $x_{i,j}$ is then determined by:

$$\mathcal{D}_{i,j} = \mathcal{N}_{i,j} \bigcap \left(\cup_{t=0}^{k-1}\mathcal{S}_t \bigcup \mathcal{A}_{i,j}\right). \quad (2)$$

where $\mathcal{A}_{i,j}$ represents a pixel-set consisting of the pixels that have been predicted prior to $x_{i,j}$, which will be shown in the following DFP procedure.

A pixel with a larger degree will be predicted prior to that with a smaller one. The reason is, only pixels in the degree-set are utilized to predict a pixel. Thus, a pixel with a larger degree can be predicted from more pixels, meaning that, the pixel can be well predicted as more context are provided. That is why we consider the prediction as "degree-first prediction". In this paper, the prediction of $x_{i,j}$ is defined as:

$$\hat{x}_{i,j} = \frac{\sum\limits_{x_{u,v} \in \mathcal{N}_{i,j} \bigcap \mathcal{A}_{i,j}} \hat{x}_{u,v}}{|\mathcal{D}_{i,j}|} + \frac{\sum\limits_{x_{u,v} \in \mathcal{D}_{i,j} \setminus (\mathcal{N}_{i,j} \bigcap \mathcal{A}_{i,j})} x_{u,v}}{|\mathcal{D}_{i,j}|}. \quad (3)$$

Based on the above description, we describe the proposed DFP procedure for the k-th ($k \geq 1$) layer as follows.

Step 1) For all $x_{i,j} \in \mathcal{X} \setminus \cup_{t=0}^{k-1} \mathcal{S}_t$, set $\mathcal{A}_{i,j} = \emptyset$, and obtain $\mathcal{D}_{i,j}$ with Eq. (2). Mark all the $x_{i,j}$ as unprocessed.

Step 2) Select such a *unprocessed* pixel $x_{i,j} \in \mathcal{X} \setminus \cup_{t=0}^{k-1} \mathcal{S}_t$ that has the largest degree in $\mathcal{X} \setminus \left(\cup_{t=0}^{k-1} \mathcal{S}_t \bigcup \mathcal{A}_{i,j} \right)$. If there are multiple pixels that have the largest degree, choose one according to a key or specified rule. Find $\hat{x}_{i,j}$ with Eq. (3).

Step 3) Mark $x_{i,j}$ as processed. For each *unprocessed* pixel $x_{u,v} \in \mathcal{X} \setminus \cup_{t=0}^{k-1} \mathcal{S}_t$, update $\mathcal{A}_{u,v}$ as $\mathcal{A}_{u,v} \cup \{x_{i,j}\}$ and further update $\mathcal{D}_{i,j}$ with Eq. (2).

Step 4) Terminate the procedure if all pixels in $\mathcal{X} \setminus \cup_{t=0}^{k-1} \mathcal{S}_t$ are processed; otherwise, go to Step 2).

Pixel Selection. After all required pixels are predicted, we are to select a part of the predicted pixels out to constitute $\mathcal{S}_k$. It relies on the local complexities of the pixels. Here, we define the local complexity of a predicted pixel $x_{i,j}$ as:

$$\rho_{i,j} = \frac{\sum\limits_{x_{u,v} \in \mathcal{P}_{i,j}} (x_{u,v} - \hat{x}_{i,j})^2}{|\mathcal{N}_{i,j}|} + \frac{\sum\limits_{x_{u,v} \in \mathcal{Q}_{i,j}} (\hat{x}_{u,v} - \hat{x}_{i,j})^2}{|\mathcal{N}_{i,j}|}. \quad (4)$$

where $\mathcal{P}_{i,j} = \cup_{t=0}^{k-1} \mathcal{S}_t \bigcap \mathcal{N}_{i,j}$, $\mathcal{Q}_{i,j} = \mathcal{N}_{i,j} \setminus \cup_{t=0}^{k-1} \mathcal{S}_t$.

A larger local complexity indicates that, the pixel is likely to be located at a more complex region. Every predicted pixel is associated with its local complexity. We sort the pixels by their local complexities in an increasing order. In this way, the top $|\mathcal{S}_k|$ pixels are chosen to constitute $\mathcal{S}_k$, and the selected pixels are likely located at relatively complex regions. Now, we are to describe the DCSPF procedure as follows.

Step 1) Choose the DCSPF parameters, namely, m, $|\mathcal{S}_t|$ for $0 \leq t < m$, and the secret key. Set $k = 1$, initially.

Step 2) Determine the preselection pattern with a secret key, only shared between the sender and receiver.

Step 3) Apply the above DFP and pixel selection procedure to construct $\mathcal{S}_k$. Set $k = k + 1$; repeat Step 3) until $k \geq m$.

Step 4) Collect all the pixels in $\cup_{t=0}^{m-1} \mathcal{S}_t$ and terminate the DCSPF procedure.

Therefore, a total of $|\cup_{t=0}^{m-1} \mathcal{S}_t|$ *relatively complex* pixels are selected out from the given image. When to apply DCSPF for the RDH, the selected pixels will be unchanged and the *relatively smooth* pixels in $\mathcal{X} \setminus \cup_{t=0}^{m-1} \mathcal{S}_t$ are utilized for data hiding. Since for the k-th layer, we only use the original values of pixels obtained from previous layers, both the data hider and data receiver should be able to construct the same $\cup_{t=0}^{m-1} \mathcal{S}_t$ according to the shared parameters.

III. DCSPF Application in Reversible Data Hiding

In this section, we present a practical RDH scheme based on the DCSPF. Note that, it is open to apply the DCSPF for RDH. One may design other RDH schemes by using DCSPF. The RDH scheme involves three aspects: data embedding, data extraction and image recovery. For data embedding, at the very beginning, we should adopt the DCSPF procedure to determine $\cup_{t=0}^{m-1} \mathcal{S}_t$ out from $\mathcal{X}$. Since only the pixels in $\mathcal{X} \setminus \cup_{t=0}^{m-1} \mathcal{S}_t$ are modified to carry extra data, to avoid the underflow/overflow problem, boundary pixels in $\mathcal{X} \setminus \cup_{t=0}^{m-1} \mathcal{S}_t$ are adjusted into the reliable range and recorded to constitute a location map that will be embedded into the image together with the secret data.

Then, with the DFP procedure, the prediction of pixels in $\mathcal{X} \setminus \cup_{t=0}^{m-1} \mathcal{S}_t$ can be computed by only utilizing the pixels in $\cup_{t=0}^{m-1} \mathcal{S}_t$. Note that, when utilizing Eq. (3), the prediction value should be rounded to a nearest integer. We thereafter sort the pixels according to their local complexities, i.e., Eq. (4). In this way, we can generate a pixel sequence in an increasing order of the local complexities. Also, we can obtain the corresponding prediction error sequence (PES). Let $\mathbf{e} = (e_1, e_2, ..., e_{n_l})$ be the resultant PES, where e_i $(1 \leq i \leq n_l)$ is the difference between the original value and prediction value, i.e., $p_i - \hat{p}_i$. For a payload, we modify $\mathbf{e}$ by reversibly shifting the PEH.

Specifically, by using two bin-pairs (l_z, l_p) and (r_p, r_z), we orderly scan and process the prediction errors until the required payload is completely embedded. Note that, we have $l_z < l_p < r_p < r_z$. The operation to a prediction error $e_i = p_i - \hat{p}_i$ is:

$$e_i{}' = \begin{cases} e_i + \text{sgn}(e_i - \frac{l_p + r_p}{2}) \cdot b, & \text{if } e_i = l_p \text{ or } r_p; \\ e_i + \text{sgn}(e_i - \frac{l_p + r_p}{2}), & \text{if } e_i \in [l_z, l_p) \cup (r_p, r_z]; \\ e_i, & \text{otherwise.} \end{cases} \quad (5)$$

where $b \in \{0, 1\}$ is the secret bit, and sgn(t) = $t/|t|$. In this way, the marked version of the corresponding pixel p_i is:

$$p_i{}' = \hat{p}_i + e_i{}'. \quad (6)$$

By replacing the original cover pixels with the corresponding marked version, a marked image can be finally generated. In application scenarios, it is required to find the optimal values for (l_z, l_p) and (r_p, r_z). We here employ the optimization algorithm introduced in [10] to determine (l_z, l_p) and (r_p, r_z). Also, one may choose (l_z, l_p) and (r_p, r_z) with other efficient methods or manually tuned.

On the receiver side, it is straightforward to fully retrieve the secret data and recover the original image content. First, with the DCSPF parameters, the decoder can perfectly reconstruct $\cup_{t=0}^{m-1} \mathcal{S}_t$. The prediction of the rest pixels are then computed. The corresponding marked PES can be reconstructed as well. Thereafter, as the data hiding operation ensures reversibility, both the hidden bits and cover pixels can be recovered without

Fig. 2. The selected pixels due to the DCSPF procedure for the *Airplane*, *Lena*, *Baboon* and *Boat* images: $m = 6$ and $r = \frac{|\cup_{t=0}^{m-1} \mathcal{S}_t|}{h \cdot w} \times 100\%$.

error. For example, to compute the original pixel value p_i, the *non-marked* version of ${e_i}'$ is computed as:

$$e_i = \begin{cases} {e_i}' - \text{sgn}({e_i}' - \frac{l_p + r_p}{2}), & \text{if } {e_i}' \in [l_z, l_p) \cup (r_z, r_p]; \\ {e_i}', & \text{otherwise.} \end{cases} \tag{7}$$

Then, the cover pixel value can be reconstructed as:

$$p_i = \hat{p}_i + e_i. \tag{8}$$

According to the extracted location map, the original image content can be completely recovered. Therefore, the proposed DCSPF has been successfully applied to the design of an RDH scheme. It is noted that, all the data embedding parameters such as m, $|\mathcal{S}_t|$ $(0 \le t < m)$, (l_z, l_p), (r_p, r_z), and the secret key, should be shared between the data hider and data receiver. Since they require extra storage, the data hider can embed the parameters into some LSBs of specified pixels in advance. The original LSBs should be embedded into the image together with the secret data. Meantime, these pixels should be unchanged in the subsequent procedure so that, a receiver can fully retrieve the parameters. Furthermore, it is possible to iteratively use the presented RDH scheme, i.e., a given image could be used to data hiding for several times.

IV. Performance Evaluation and Analysis

In this section, we present some experiments and analysis to evaluate the performance of the DCSPF applied to the RDH.

A. Security and DCSPF Parameters Selection

According to Kerckhoff's principle, for conventional RDH methods, an unauthorized receiver has the preselection pattern and even the potential to reconstruct the prediction errors and PEH, which is not desirable in applications. In our method, with the DCSPF, one cannot fully distinguish the *marked* and *non-marked* regions if he has not the parameters, meaning that, it is almost impossible for an unauthorized receiver to compute the prediction errors and PEH. Therefore, our method is indeed more secure than the conventional methods.

The DCSPF parameters mainly includes m, $|\mathcal{S}_t|$ $(0 \le t < m)$, and the secret key to generate $\mathcal{S}_0$. For a required payload, there exists a lot of freedom to choose m and $|\mathcal{S}_t|$ $(0 \le t < m)$. For convenience, we consider $|\mathcal{S}_0| = |\mathcal{S}_1| = ... = |\mathcal{S}_{m-1}|$ in default, which may be not optimal. In the DCSPF, a total of $|\cup_{t=0}^{m-1} \mathcal{S}_t|$ pixels are selected from a given image, and used for predicting the rest pixels in the data hiding process. Fig.2

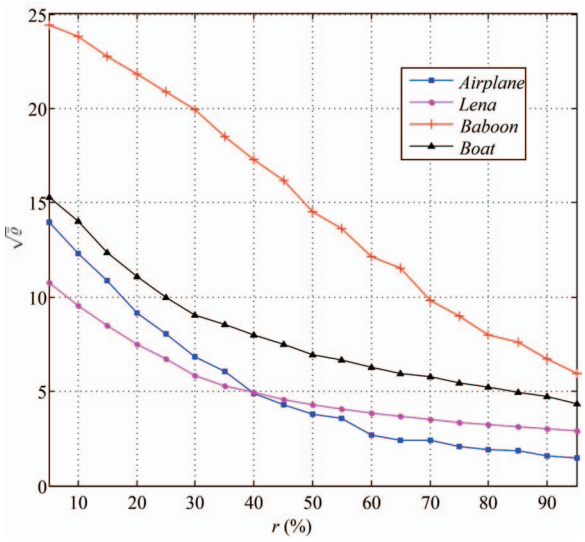

Fig. 3. Relationship between r and $\sqrt{\bar{\varrho}}$ for different images with $m = 6$.

shows an example to choose a certain number of pixels from the *Airplane*, *Lena*, *Baboon* and *Boat* images (all grayscale with a size of $512 \times 512 \times 8$) with the DCSPF procedure. In Fig.2, the selected pixels are located at the *black* regions. It can be seen that, the DCSPF procedure can *auto-capture* the relatively smooth pixels (please refer to the *white* regions in Fig.2) for data hiding, which is very desirable for an RDH system since the smooth pixels can be well predicted and therefore provide a good data embedding performance. To evaluate the overall smoothness for pixels in $\mathcal{X} \setminus \cup_{t=0}^{m-1} \mathcal{S}_t$, we define the smoothness of $x_{i,j}$ as:

$$\varrho_{i,j} = \frac{1}{|\mathcal{N}_{i,j}|} \cdot \sum_{x_{u,v} \in \mathcal{N}_{i,j}} (x_{u,v} - x_{i,j})^2. \tag{9}$$

The overall smoothness is then denoted by:

$$\bar{\varrho} = \frac{1}{|\mathcal{X} \setminus \cup_{t=0}^{m-1} \mathcal{S}_t|} \cdot \sum_{x_{i,j} \in \mathcal{X} \setminus \cup_{t=0}^{m-1} \mathcal{S}_t} \varrho_{i,j}. \tag{10}$$

Fig.3 shows the percentage of selected pixels r (refer to its definition in Fig.2) versus $\sqrt{\bar{\varrho}}$. It can be seen that, when more pixels are chosen, the rest pixels are likely located at smoother regions. Therefore, for a required payload, we usually hope to choose r as large as possible for data hiding, as long as

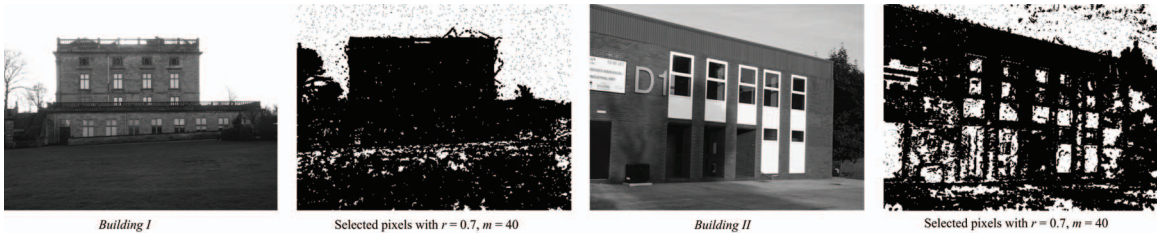

Fig. 4. Example to show the selected pixels (at black regions) due to the DCSPF procedure for two images containing *near-white* background.

the payload can be completely carried. For example, if we want to embed 5000 message bits into the *Lena*, we can set $r = 90\%$. For a required payload and fixed r, it is necessary to select a suitable m for data hiding. According to our empirical experiments, a larger m usually provides a relatively better data-embedding performance. However, a larger m will require a higher computational complexity. In applications, we recommend the data hider to use $m \leq 40$. For the preselection pattern, a larger $|\mathcal{S}_0|$ usually gives a better subsequent pixel-selection and pixel-prediction performance since more original context are provided. However, on the other hand, a larger $|\mathcal{S}_0|$ will block more smooth pixels (from the viewpoint of probabilistic analysis). Therefore, it is necessary to keep $|\mathcal{S}_0|$ within a reasonable level. As we have $|\mathcal{S}_0| = h \cdot w \cdot \frac{r}{m}$ here, we recommend that, when r tends to be larger, m should be larger as well, and vice versa.

B. Pixel-Selection and Payload-Distortion Evaluation

We first want to show that the DCSPF has the good property to *well-capture* the smooth regions out for data hiding. Fig.4 shows an example to process two images that have the *near-white* background (corresponding to smooth regions) with the DCSPF procedure. It can be seen that, the near-white regions (i.e., the smooth regions) in both images are well-captured. Since the pixels in smooth regions can be well-predicted, it indicates that the subsequent data hiding procedure will significantly benefit from the proposed DCSPF procedure.

We have implemented the RDH scheme [1] in Section III and applied it to the standard testing images shown in Fig.2. Though we have proposed the detailed RDH scheme, it is still open to design more efficient DCSPF-based RDH algorithms. In the experiments, we use Peak Signal-to-Noise Ratio (PSNR, dB) as the measurement of introduced distortion. The preselection pattern was always generated according to a random seed in each time. For a payload, we performed the RDH operation by randomly choosing $r \in [5\%, 95\%]$ and $m \in [2, 40]$ for several times. Thereafter, the highest PSNR value was used as the distortion measurement for the payload, since in applications, the data hider always has the freedom to generate a marked image with a better quality.

It should be noted that, since we did not take all possible combinations between r and m for experiments, the payload-distortion behavior in our experiments may be not optimal. However, we can still draw out some valuable conclusions. Fig.5 shows the payload-distortion performance comparison between some of the state-of-the-art methods and the RDH scheme presented in Section III. It can be observed from Fig.5 that, the used data hiding operation can benefit from the DCSPF procedure, and therefore provide a relatively good data-embedding performance. In Fig.5, with relatively low embedding rates, the proposed scheme significantly outperforms the related works. This indicates that, the proposed DCSPF procedure indeed has the ability to well-capture the smooth pixels out for data hiding. It can be also seen that, when the embedding payload increases, all the PSNRs are likely to be relatively lower than some of the related works. For example, the proposed RDH scheme has a weaker performance for the *Baboon* image after embedded with more than 1.1×10^4 bits. The reason is that, the proposed DCSPF procedure aims to select the complex pixels for prediction and smooth pixels for data hiding, while the amount of smooth pixels within an image is actually limited due to the image content. For example, the *Baboon* image is full of complex content so that, many complex pixels are finally selected out for data hiding. However, the complex pixels are likely to be predicted with a larger prediction error, which therefore cannot keep a good payload-distortion performance. It also implies that, the proposed pixel predictor and/or data hiding mechanism should be further improved when a larger payload is required and/or the number of smooth pixels is limited, which is the next work.

[1] we will release our simulation code at https://hzwu.github.io/

V. Conclusion and Discussion

Compared with the related works, the DCSPF guarantees that, an illegal decoder will hardly access the whole marked pixels and determine the marked prediction errors. Meantime, the DCSPF has the property to well-capture the smooth regions out for data hiding, which can collect many pixels that may be the neighbor of each other (see Fig.4). Experimental results have shown that, the used RDH operation will benefit from the DCSPF, and provide a good payload-distortion performance as well as a high security level. It is true that, the DCSPF can be applied to the design of an RDH system, which will provide an efficient way to enhance the security level and improve the payload-distortion performance. And, there is still room for improvement such as by designing a better pixel predictor and more accurate evaluation function for local complexity.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (NSFC) under the grant No. U1536110.

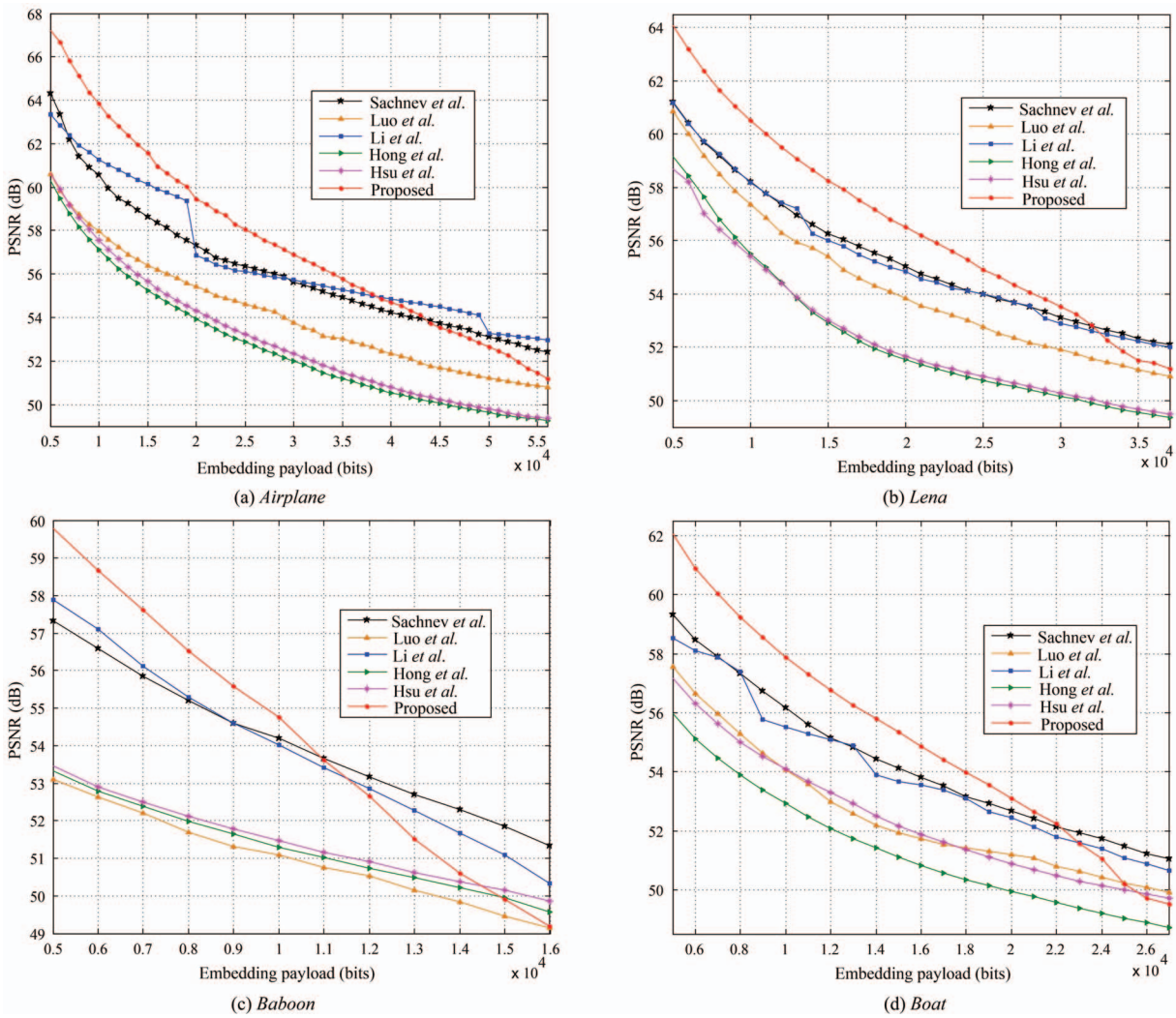

Fig. 5. The payload-distortion performance comparison between the state-of-the-art methods of Sachnev *et al.* [3], Hong *et al.* [4], Luo *et al.* [6], Li *et al.* [7], Hsu *et al.* [17] and the RDH scheme presented in Section III.

REFERENCES

[1] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, 13(8): 890-896, Aug. 2003.

[2] Z. Ni, Y. Shi, N. Ansari and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, 16(3): 354-362, Mar. 2006.

[3] V. Sachnev, H. Kim, J. Nam, S. Suresh and Y. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Trans. Circuits Syst. Video Technol.*, 19(7): 989-999, Jul. 2009.

[4] W. Hong, T. Chen and C. Shiu, "Reversible data hiding for high quality images using modification of prediction errors," *J. Syst. Softw.*, 82(11): 1833-1842, Nov. 2009.

[5] S. Lee, C. Yoo and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Trans. Inf. Forensics Security*, 2(3): 321-330, Sept. 2007.

[6] L. Luo, Z. Chen, M. Chen, X. Zeng and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Trans. Inf. Forensics Security*, 5(1): 187-193, Mar. 2010.

[7] X. Li, B. Yang and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Trans. Image Process.*, 20(12): 3524-3533, Dec. 2011.

[8] X. Li, W. Zhang, X. Gui and B. Yang, "A novel reversible data hiding scheme based on two-dimensional difference-histogram modification," *IEEE Trans. Inf. Forensics Security*, 8(7): 1091-1100, Jul. 2013.

[9] C. Yang and M. Tsai, "Improving histogram-based reversible data hiding by interleaving prediction," *IET Image Process.*, 4(4): 223-234, Aug. 2010.

[10] H. Wu, H. Wang and Y. Shi, "PPE-based reversible data hiding," In: *ACM Workshop Inf. Hiding Multimed. Security*, pp. 187-188, Jun. 2016.

[11] P. Tsai, Y. Hu and H. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Process.*, 89(6): 1129-1143, Jun. 2009.

[12] D. Thodi and J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, 16(3): 721-730, Mar. 2007.

[13] M. Fallahpour, "Reversible image data hiding based on gradient adjusted prediction," *IEICE Electr. Express*, 5(20): 870-876, Aug. 2008.

[14] I. Dragoi and D. Coltuc, "Local-prediction-based difference expansion reversible watermarking," *IEEE Trans. Image Process.*, 23(4): 1779-1790, Apr. 2014.

[15] D. Coltuc, "Improved embedding for prediction-based reversible watermarking," *IEEE Trans. Inf. Forensics Security*, 6(3): 873-882, Sept. 2011.

[16] M. Chen, Z. Chen, X. Zeng and Z. Xiong, "Reversible data hiding using additive prediction-error expansion," In: *ACM Workshop Multimed. Security*, pp. 19-24, Sept. 2009.

[17] F. Hsu, M. Wu, and S. Wang, "Reversible data hiding using side-match predictions on steganographic images," *Multimed. Tools Appl.*, 67(3): 571-591, Dec. 2013.