IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 14, 2019, accepted February 26, 2019, date of publication March 13, 2019, date of current version April 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2904174

Improved K-Pass Pixel Value Ordering Based Data Hiding

SHAOWEI WENG[1,2], (Member, IEEE), YI CHEN[1], BO OU[3], CHIN-CHEN CHANG[4], (Fellow, IEEE), AND CHUNYU ZHANG[5]

[1]School of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China
[2]Guangdong Key Laboratory of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China
[3]College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[4]Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan
[5]School of Information Engineering, Xizang Minzu University, Xianyang 712082, China

Corresponding author: Shaowei Weng (wswweiwei@126.com)

This work was supported in part by the National NSF of China under Grant 61872095, Grant 61872128, Grant 61571139, and Grant 61201393, in part by the New Star of Pearl River on Science and Technology of Guangzhou under Grant 2014J2200085, in part by the Open Project Program of Shenzhen Key Laboratory of Media Security under Grant ML-2018-03, in part by the Opening Project of GuangDong Province Key Laboratory of Information Security Technology under Grant 2017B030314131-15, and in part by the Natural Science Foundation of Xizang under Grant 2016ZR-MZ-01.

ABSTRACT K-pass pixel value ordering (PVO) is an effective reversible data hiding (RDH) technique. In k-pass PVO, the complexity measurement may lead to a weak estimation result because the unaltered pixels in a block are excluded to estimate block complexity. In addition, the prediction-error is computed without considering the location relationship of the second largest and largest pixels or the second smallest and smallest pixels. To this end, an improved RDH technique is proposed in this paper to enhance the embedding performance. The improvement mainly lies in the following two aspects. First, some pixels in a block, which are excluded from data hiding in some existing RDH methods, are exploited together with the neighborhood surrounding this block to increase the estimation accuracy of local complexity. Second, the remaining pixels in a block, i.e., three largest and three smallest pixels are involved in data embedding. Taking three largest pixels for example, when the difference between the largest and third largest pixels is relatively large (e.g., > 1), we improve k-pass PVO by considering the location relationship of the second largest and largest pixels. The advantage of doing this is that the difference valued 3 between the maximum and the second largest pixel which is shifted in k-pass PVO, is able to carry 1 bit data in our method. In other words, a larger amount of pixels are able to carry data bits in our scheme compared with k-pass PVO. Abundant experimental results reveal that the proposed method achieves preferable embedding performance compared with the previous work, especially when a larger payload is required.

INDEX TERMS K-pass PVO, reversible data hiding, relative location, complexity measurement.

I. INTRODUCTION

With the rapid development of the Internet and technology, behaviors on violation of ownership are becoming easier and more common. In order to prevent data from being tampered illegally, the protection of information is in urgent demand. To this end, information security, which includes information hiding, access control and cryptography, has become a popular research area undoubtedly. Typical applications in these disciplines are digital watermarking, firewall and digital signature, respectively. In particular, digital watermarking is a technique to hide secret bits into the cover images. It falls into two categories in the light of reversibility: irreversible data hiding and reversible data hiding(RDH). Compared to irreversible data hiding, the property of reversibility must be satisfied in RDH, which means the cover image is able to be restored without any distortion after the secret bits are exactly extracted from the marked image. RDH is essential for medical, military and legal images. On the one hand, these kind of images have to be embedded with data bits for copyright protection. On the other hand, the cover images are applied for diagnosis and exhibits so that any slight distortion

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Shorif Uddin.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

may bring improper medical treatment or unjust judgement. Therefore, it is necessary to recover the cover images from marked ones completely.

Over recent decades, a considerable number of researchers have been engaged in investigation on how to improve the embedding performance in RDH. Basically, there are two predominant contributions in RDH. The first breakthrough is difference expansion(DE) [1]. The basic idea of DE is to use two neighboring pixels as one embedding unit, and expand their difference to embed 1 bit. Unlike lossless compression [2], DE is capable of providing high embedding capacity so that it has made an unprecedented success in applications of RDH. Since DE was proposed, there have been substantial extensions of DE, such as integer transform. Stemming from DE, several RDH strategies named integer transform are developed in [3]–[6]. These methods exploit more than two pixels as an embedding unit. The advantage is that they bring higher payload compared with DE. Specifically, the maximal embedding rate approximates to 0.5, 0.75, 0.94 and 0.75 in [1], [3], [4], and [6] correspondingly. On the other hand, another contribution in RDH is histogram shifting(HS) proposed in [7]. In HS, the peak point of histogram is embedded with one secret bit while pixels on the right-hand side of the peak point are shifted by one unit. Although it offers smaller amount of embedded data bits, less distortion is guaranteed. In 2007, Thodi and RodrÍłguez [8] came up with a prediction error expansion(PEE) based technique, which is a combination of HS and DE. For one thing, prediction error instead of difference value is adopted for expansion to carry data bits, which contributes to more accurate prediction. And for another, histogram shifting(HS) is combined into PEE in order to tackle the problem existing in DE, i.e. low compression ratio of the location map. In other words, HS enables the location map to be effectively compressed. Later, extensive research has been concentrated on how to design predictors with higher accuracy [9]–[22].

Recently, a pixel value ordering(PVO) based scheme stands out from the PEE-HS-based strategies because it provides outstanding image quality when limited capacity is required [23]. In PVO, the second largest(smallest) pixel is utilized to predict the largest(smallest) pixel so as to obtain the prediction error. If the prediction error between the largest(smallest) and second largest(smallest) pixels is $1(-1)$, one data bit is embedded. Otherwise, the prediction error which is larger(smaller) than $1(-1)$ will be shifted by one unit to ensure reversibility. The superiority is that more accurate prediction is obtained with the adoption of the second largest pixel to predict the largest one. However, blocks whose prediction error is equal to 0 are abandoned in the embedding phase. Actually, these blocks are extremely smooth and suitable for data hiding. Therefore, IPVO is invented to compensate for the shortage in PVO by considering the relative location of the largest and second largest pixels [24]. Specifically, prediction error with value 1 or 0 is able to carry one bit data. In the same year, PVO-k is designed by taking k largest pixels as an embedding unit [25]. In this way, more smooth blocks can be exploited compared with PVO. The negative aspect in PVO-k is that if k largest pixels are shifted by 1 simultaneously, the distortion will be huge. Hence, He *et al.* put forward a scheme called k-pass PVO, where k largest pixels are respectively treated as independent data carriers [26]. As it turns out, less distortion is introduced compared with PVO-k based technique. In [23]–[25], only two data bits are embedded into a block at best. In order to achieve sufficient payload, small-sized blocks are selected at the expense of weaker accuracy in terms of prediction. In contrast, k-pass PVO enables four data bits to be embedded within a block. This property contributes to the improvement in embedding capacity so that larger-sized blocks can be adopted and higher accuracy is attained.

Motivated by k-pass PVO, a reversible data hiding (RDH) technique is proposed in this paper. For a block, its adjacent pixels are combined with the pixels excluded from embedding to measure the block complexity. Secondly, when the difference between the largest and third largest pixel values exceeds 1, the relative location of the largest and second largest pixel values is considered to calculate the prediction error between them. On this account, a larger amount of embeddable pixels are provided, which leads to higher payload so that larger-sized blocks are employed for accuracy upgrade. In this way, a sharper prediction error histogram is created. Based on the property that sharper distribution of histogram contributes to better embedding performance, our strategy offers better visual quality with high payload successfully, especially when larger payload is required.

The rest of this paper falls into three sections. Section 2 aims at delivering a brief introduction on He *et al.*'s k-pass PVO technique. Section 3 puts emphasis on detailed description of the proposed method. Section 4 gives the experimental results while concluding remarks are presented in Section 5.

II. RELATED WORK

In this section, we introduce a recently reported RDH work proposed by He *et al.* [26]. Ou *et al.* [25] extended PVO-k based scheme to k-pass PVO. When k is set 2, two-pass PVO is able to achieve the optimal embedding performance. Therefore, two-pass PVO, as the representive of k-pass PVO, is described in details.

In two-pass PVO, a cover image is divided into equal-sized non-overlapped blocks, and each block is composed of n pixels $\{x_1, x_2, \ldots, x_{n-1}, x_n\}$. These n pixels are sorted in ascending order to obtain $X = \{x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n-1)}, x_{\sigma(n)}\}$, where $\sigma(i)$, $1 \leq i \leq n$, denotes the original location of each pixel in $\{x_1, x_2, \ldots, x_{n-1}, x_n\}$. In the data embedding process, two largest and two smallest pixels within a block are modified for carrying at most four data bits. For three largest pixels, similar to PVO-k, the maximum $x_{\sigma(n)}$ is predicted by the third largest pixel $x_{\sigma(n-2)}$, and the prediction error p_n is equal to $x_{\sigma(n)} - x_{\sigma(n-2)}$. Different from PVO-k, the second largest pixel $x_{\sigma(n-1)}$ is predicted depending on the difference between $x_{\sigma(n)}$ and $x_{\sigma(n-2)}$. Specifically, the prediction error

p_{n-1} is generated via Eq. (1). From Eq. (1), it can be clearly seen that if the prediction error p_n between $x_{\sigma(n)}$ and $x_{\sigma(n-2)}$ is smaller than or equal to 1, the third largest pixel $x_{\sigma(n-2)}$ is utilized to predict the second largest one $x_{\sigma(n-1)}$. On the contrary, when p_n is larger than 1, $x_{\sigma(n-1)}$ is predicted by $x_{\sigma(n)} - 2$. Finally, the marked pixel $y_{\sigma(i)}$ is obtained depending on Eq. (2), where $i \in \{n-1, n\}$.

$$p_{n-1} = \begin{cases} x_{\sigma(n-1)} - x_{\sigma(n-2)}, & p_n \leq 1, \\ x_{\sigma(n-1)} - (x_{\sigma(n)} - 2), & p_n > 1. \end{cases} \tag{1}$$

$$y_{\sigma(i)} = \begin{cases} x_{\sigma(i)} + b, & p_i = 1, \\ x_{\sigma(i)} + 1, & p_i > 1. \end{cases} \tag{2}$$

For three smallest pixels, similar to PVO-k, the minimum $x_{\sigma(1)}$ is predicted by the third smallest pixel $x_{\sigma(3)}$, and the prediction error p_1 is equal to $x_{\sigma(1)} - x_{\sigma(3)}$. Different from PVO-k, the prediction error p_2 is calculated depending on the prediction error p_1 (refer to Eq. (3)). The marked pixel $y_{\sigma(i)}$ is then modified for probably carrying one bit according to Eq. (4), where $i \in \{1, 2\}$.

In k-pass PVO, except for the two largest and two smallest pixels in each block which may be modified for data embedding, the others remain unaltered. K-pass PVO only utilizes the neighborhood of each block to measure its local complexity, while ignores the unchanged pixels in each block, which play an important role in increasing measurement accuracy. Therefore, we conclude that there is still room for improvement in complexity measurement. On the other hand, after studying k-pass PVO deeply, we find that the payload in k-pass PVO can be further improved while introducing less distortion. In this paper, we name the pixels which can be embedded with data bits as "embeddable pixels". In contrast, the pixels, which have to be shifted during embedding process, are called as "shifted pixels". Here, we take a block $X = \{148, 149, 150, 151, 152, 155\}$ as an example. In k-pass PVO, four prediction errors are calculated as $p_6 = 4$, $p_5 = -1$, $p_1 = -2$, $p_2 = -1$ according to Eq. (1) and Eq. (3). Two largest and two smallest pixels are all shifted during data embedding process. However, when it is utilized for data embedding in the proposed method, two data bits can be embedded into this block. Compared with k-pass PVO, the payload in the proposed scheme is enhanced because more pixels, which are shifted in k-pass PVO, become embeddable in the proposed scheme. Besides, the increase in the number of embeddable pixels means that the proposed scheme is able to modify fewer blocks and reduce the amount of shifted pixels in order to reach the same capacity. In this way, the visual quality can be improved. Therefore, how to reduce the number of shifted pixels while increase the number of embeddable pixels is very important for the increase of capacity-distortion performance, and it is also the focus of our research.

$$p_2 = \begin{cases} x_{\sigma(2)} - x_{\sigma(3)}, & p_1 \geq -1, \\ x_{\sigma(2)} - (x_{\sigma(1)} + 2), & p_1 < -1. \end{cases} \tag{3}$$

$$y_{\sigma(i)} = \begin{cases} x_{\sigma(i)} - b, & p_i = -1, \\ x_{\sigma(i)} - 1, & p_i < -1. \end{cases} \tag{4}$$

III. PROPOSED SCHEME

In this paper, we propose a more accurate method to measure the local complexity of blocks by combining the unchanged pixels with the neighborhood. In addition, an efficient prediction mechanism is adopted considering the relative location of the largest and second largest pixel values (or the smallest and second smallest pixel values) in the process of calculating prediction errors. Depending on this prediction mechanism, the ratio between the embeddable and shifted pixels is increased compared with k-pass PVO, and consequently, our method achieves higher payload while induces less distortion.

A. PREDICTION MECHANISM

The key idea of the prediction mechanism is to increase the number of embeddable pixels while reduce the number of shifted pixels. In this subsection, we will give a detailed introduction of the prediction mechanism for three largest and three smallest pixels, respectively. After dividing the host image I into n-sized non-overlapped blocks, one block is scanned in raster scanning order to form a pixel list B, i.e., $B = \{x_1, x_2, \ldots, x_{n-1}, x_n\}$. Then, B is sorted in ascending order to obtain a sorted pixel list $X = \{x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n-1)}, x_{\sigma(n)}\}$, where $\sigma(i)$, $1 \leq i \leq n$ indicates the original location of each pixel in $\{x_1, x_2, \ldots, x_{n-1}, x_n\}$.

For three largest pixels, the largest pixel $x_{\sigma(n)}$ is predicted by the third largest pixel $x_{\sigma(n-2)}$, and then the prediction error $p_n = x_{\sigma(n)} - x_{\sigma(n-2)}$. Since $x_{\sigma(n)} \geq x_{\sigma(n-2)}$, $p_n \geq 0$. For the second largest pixel $x_{\sigma(n-1)}$, its prediction value is dependent on p_n. For ease of description, the prediction error p_{n-1} is formulated in Eq. (5).

$$p_{n-1} = \begin{cases} x_{\sigma(n-1)} - x_{\sigma(n-2)}, & p_n \leq 1, \\ x_{\sigma(n-1)} - (x_{\sigma(n)} - 2), & p_n > 1 \text{ and } \sigma(n-1) < \sigma(n), \\ (x_{\sigma(n)} - 2) - x_{\sigma(n-1)}, & p_n > 1 \text{ and } \sigma(n-1) > \sigma(n). \end{cases} \tag{5}$$

As described in Eq. (5), if p_n is smaller than or equal to 1, the second largest pixel $x_{\sigma(n-1)}$ is predicted by the third largest pixel $x_{\sigma(n-2)}$. $p_n > 1$ implies that the maximum $x_{\sigma(n)}$ is not strongly correlated to $x_{\sigma(n-2)}$, and therefore, $x_{\sigma(n-2)}$ is no longer suitable for predicting $x_{\sigma(n-1)}$. Under this consideration, $x_{\sigma(n)} - 2$, deemed to be more correlated with $x_{\sigma(n-1)}$ than $x_{\sigma(n-2)}$, is used to predict $x_{\sigma(n-1)}$. In the prediction process, we still need to take into account the relative location of $x_{\sigma(n-1)}$ and $x_{\sigma(n)}$. The advantage of doing this is to increase the number of embeddable pixels as many as possible while reduce the number of shifted pixels.

According to the aforementioned description, $p_n = x_{\sigma(n)} - x_{\sigma(n-2)}$. Since $x_{\sigma(n)} \geq x_{\sigma(n-2)}$, p_n is always greater than or equal to 0. According to Eq. (5), p_{n-1} is only determined by p_n when $p_n \in \{0, 1\}$, namely $p_{n-1} = x_{\sigma(n-1)} - x_{\sigma(n-2)}$.

Specifically, $p_n = 0$ implies $x_{\sigma(n-2)} = x_{\sigma(n-1)} = x_{\sigma(n)}$, which directly leads to $p_{n-1}=0$. Similarly, $p_n = 1$ means $x_{\sigma(n-1)} \leq x_{\sigma(n)}$ and $x_{\sigma(n)} - x_{\sigma(n-2)} = 1$, and thus $p_{n-1} \leq 1$. Correspondingly, Table 1 is generated to illustrate the embedding process of two largest pixels without consideration of the relationship between $\sigma(n-1)$ and $\sigma(n)$. When $p_n>1$, if $p_{n-1}=x_{\sigma(n-1)} - (x_{\sigma(n)} - 2) = 0$, it is unnecessary to consider the relationship between $\sigma(n-1)$ and $\sigma(n)$ because $x_{\sigma(n-1)}=x_{\sigma(n)} - 2$ means $x_{\sigma(n)} > x_{\sigma(n-1)}$, namely $\sigma(n-1)$ is always smaller than $\sigma(n)$. Therefore, we classify $p_n>1$ and $p_{n-1} = 0$ into Table 1.

TABLE 1. The detailed modification process of two largest pixels without consideration of the relationship between $\sigma(n-1)$ and $\sigma(n)$.

prediction error p_n	prediction error p_{n-1}	marked pixels
$p_n = 1$	$p_{n-1} = 1$	$y_{\sigma(n-1)} = x_{\sigma(n-1)} + w_1$ $y_{\sigma(n)} = x_{\sigma(n)} + w_2$
	$p_{n-1} = 0$	$y_{\sigma(n-1)} = x_{\sigma(n-1)}$ $y_{\sigma(n)} = x_{\sigma(n)} + w$
$p_n = 0$	$p_{n-1} = 0$	$y_{\sigma(n-1)} = x_{\sigma(n-1)}$ $y_{\sigma(n)} = x_{\sigma(n)}$
$p_n > 1$	$p_{n-1} = 0$	$y_{\sigma(n-1)} = x_{\sigma(n-1)}$ $y_{\sigma(n)} = x_{\sigma(n)} + 1$

When $p_n \leq 1$, a prediction error with value 1 is expanded to carry one data bit while a prediction error with value 0 keeps unaltered. Specifically, if $p_n=1$ and $p_{n-1}=1$, $y_{\sigma(n-1)}$ and $y_{\sigma(n)}$ are both expanded to embed two data bits w_1 and w_2, respectively.

In contrast, when $p_n > 1$, p_{n-1} is determined by two factors: p_n and the relationship between $\sigma(n-1)$ and $\sigma(n)$ according to Eq. (5). Since the location relationship between $x_{\sigma(n-1)}$ and $x_{\sigma(n)}$ is considered in the process of calculating p_{n-1}, p_{n-1} belongs to the range $(-\infty, \infty)$. To this end, Table 2 is created to contain all the cases of $p_n>1$ and $p_{n-1} \neq 0$. As illustrated in Table 2, if $p_{n-1}=1$ or $p_{n-1}= -1$, the prediction-error pair (p_{n-1}, p_n) is able to embed 1 bit data, and p_{n-1} remains unchanged if $p_{n-1} = 0$. To ensure reversibility, when p_{n-1} belongs to $(-\infty, -2] \cup [2, \infty)$, p_n is shifted by one or two units, while p_{n-1} is shifted by one unit or keeps unaltered.

From Table 2, it can be seen clearly that the largest pixel $x_{\sigma(n)}$ needs to be first embedded with 1 bit and then shifted outwards by 1 when $p_n > 1$, $p_{n-1} = 1$, and $\sigma(n-1) > \sigma(n)$. For better illustration, we take a pixel list $B = \{132, 138, 130, 143, 134, 140\}$ as an example. Three largest pixels $\{138, 140, 143\}$ are chosen from B, and their corresponding locations $\{\sigma(4), \sigma(5), \sigma(6)\}$ in B are $\{2, 6, 4\}$. The prediction error p_6 between the maximum and third largest pixel is computed as $p_6 = x_{\sigma(6)} - x_{\sigma(4)} = 143 - 138 = 5$. Because of $p_6 > 1$ and $\sigma(5) > \sigma(6)$, $p_5 = (x_{\sigma(6)} - 2) - x_{\sigma(5)} = (143 - 2) - 140 = 1$ in accordance with Eq. (5). $p_6 > 1$ implies that p_6 needs to be shifted by 1, while $p_5 = 1$ means that p_5 can be embedded with one bit. Guided by the idea above, if the maximum $x_{\sigma(6)}$ is shifted by 1 and the second largest pixel $x_{\sigma(5)}$ is embedded with 1 bit, then $y_{\sigma(6)} = x_{\sigma(6)} + 1 = 143 + 1 = 144$ and $y_{\sigma(5)} = x_{\sigma(5)} + b = 140 + b$. Let the to-be-embedded bit b be 1, then $y_{\sigma(5)} = 141$. Therefore, three marked largest pixels are $\{138, 141, 144\}$. On the decoding side, three largest pixels $\{138, 141, 144\}$ are selected from the marked pixel list. The marked prediction error p_6^w between the maximum and third largest pixel is calculated as $p_6^w = y_{\sigma(6)} - y_{\sigma(4)} = 144 - 138 = 6$. Since $p_6^w > 2$ and $\sigma(5) > \sigma(6)$, $p_5^w = (y_{\sigma(6)} - 3) - y_{\sigma(5)} = 0$ (referring to Eq. (8) for more details). Because p_5^w valued 0 is not processed during data embedding, no watermark is extracted from p_5^w valued 0. After data extraction, three largest pixels are recovered as $\{138, 141, 143\}$. Obviously, the recovered second largest pixel is not the same as that in data embedding. Therefore, to ensure reversibility, we need to modify the maximum $x_{\sigma(n)}$ as $x_{\sigma(n)} + b + 1$ while keep the second largest pixel unchanged in the proposed scheme.

On the other hand, for three smallest pixels, the prediction error p_1 is calculated as $x_{\sigma(1)} - x_{\sigma(3)}$. If $p_1 < -1$, we can conclude that the correlation between $x_{\sigma(3)}$ and $x_{\sigma(1)}$ is not high, so $(x_{\sigma(1)} + 2)$ is used to predict $x_{\sigma(2)}$. The detailed prediction process is given in Eq. (6). With the assist of two prediction errors p_1 and p_2, the marked pixel $y_{\sigma(i)}$ is then modified for probably embedding one data bit according to Tables 3 and 4, where $i \in \{1, 2\}$.

From Tables 3 and 4, it can be seen clearly that p_1 is smaller than or equal to 0 because of $p_1=x_{\sigma(1)} - x_{\sigma(3)}$ and $x_{\sigma(1)} \leq x_{\sigma(3)}$. When $p_1 \in \{-1, 0\}$, p_2 only depends on p_1 according to Eq. (6), namely $p_2=x_{\sigma(2)} - x_{\sigma(3)}$. Since $p_1 = 0$ indicates that $x_{\sigma(1)} = x_{\sigma(2)} = x_{\sigma(3)}$, $p_2=x_{\sigma(2)} - x_{\sigma(3)} = 0$. Similarly, $p_1 = -1$ denotes that $x_{\sigma(2)} \geq x_{\sigma(1)}$ and $x_{\sigma(1)} - x_{\sigma(3)} = -1$, and therefore $p_2 \geq -1$. Moreover, since $x_{\sigma(2)} \leq x_{\sigma(3)}$ and $p_2=x_{\sigma(2)} - x_{\sigma(3)}$, $p_2 \leq 0$. Hence, $p_2 \in \{-1, 0\}$ when $p_1 = -1$. Table 3 is generated to illustrate the embedding process of two smallest pixels without consideration of the relationship between $\sigma(1)$ and $\sigma(2)$. When $p_1 < -1$ and $p_2 = 0$, $\sigma(1)$ is always smaller than $\sigma(2)$. Therefore, we classify $p_1 < -1$ and $p_2 = 0$ into Table 3.

When $p_1 < -1$, p_2 depends on p_1 and the relationship between $\sigma(1)$ and $\sigma(2)$. Since the location relationship between $x_{\sigma(1)}$ and $x_{\sigma(2)}$ is considered in the process of calculating p_2, p_2 belongs to the range $(-\infty, \infty)$. To this end, Table 4 is created to contain all the cases of $p_1< -1$ and $p_2 \neq 0$.

$$p_2 = \begin{cases} x_{\sigma(2)} - x_{\sigma(3)}, & p_1 \geq -1, \\ x_{\sigma(2)} - (x_{\sigma(1)} + 2), & p_1 < -1 \textit{ and } \sigma(2) < \sigma(1), \\ (x_{\sigma(1)} + 2) - x_{\sigma(2)}, & p_1 < -1 \textit{ and } \sigma(2) > \sigma(1). \end{cases} \tag{6}$$

B. COMPLEXITY MEASUREMENT

It is well known that in order to guarantee high fidelity, the blocks located in smooth regions are preferentially selected for data embedding while the blocks in complex regions keep unchanged. Therefore, how to classify the smoothness of blocks accurately is very important for increas-

TABLE 2. The detailed modification process of two largest pixels considering the relationship between $\sigma(n-1)$ and $\sigma(n)$.

prediction error p_n	prediction error p_{n-1}	the relationship between $\sigma(n-1)$ and $\sigma(n)$	marked pixels
$p_n > 1$	$p_{n-1} = 1$	$\sigma(n-1) < \sigma(n)$	$y_{\sigma(n-1)} = x_{\sigma(n-1)} + w$ $y_{\sigma(n)} = x_{\sigma(n)} + 1$
		$\sigma(n-1) > \sigma(n)$	$y_{\sigma(n-1)} = x_{\sigma(n-1)}$ $y_{\sigma(n)} = x_{\sigma(n)} + w + 1$
	$p_{n-1} > 1$	$\sigma(n-1) < \sigma(n)$	$y_{\sigma(n-1)} = x_{\sigma(n-1)} + 1$ $y_{\sigma(n)} = x_{\sigma(n)} + 1$
		$\sigma(n-1) > \sigma(n)$	$y_{\sigma(n-1)} = x_{\sigma(n-1)}$ $y_{\sigma(n)} = x_{\sigma(n)} + 2$
	$p_{n-1} = -1$	$\sigma(n-1) < \sigma(n)$	$y_{\sigma(n-1)} = x_{\sigma(n-1)}$ $y_{\sigma(n)} = x_{\sigma(n)} + w + 1$
		$\sigma(n-1) > \sigma(n)$	$y_{\sigma(n-1)} = x_{\sigma(n-1)} + w$ $y_{\sigma(n)} = x_{\sigma(n)} + 1$
	$p_{n-1} < -1$	$\sigma(n-1) < \sigma(n)$	$y_{\sigma(n-1)} = x_{\sigma(n-1)}$ $y_{\sigma(n)} = x_{\sigma(n)} + 2$
		$\sigma(n-1) > \sigma(n)$	$y_{\sigma(n-1)} = x_{\sigma(n-1)} + 1$ $y_{\sigma(n)} = x_{\sigma(n)} + 1$

TABLE 3. The detailed modification process of two smallest pixels without consideration of the relationship between $\sigma(1)$ and $\sigma(2)$.

prediction error p_1	prediction error p_2	marked pixels
$p_1 = -1$	$p_2 = -1$	$y_{\sigma(1)} = x_{\sigma(1)} - w_1$ $y_{\sigma(2)} = x_{\sigma(2)} - w_2$
	$p_2 = 0$	$y_{\sigma(1)} = x_{\sigma(1)} - w$ $y_{\sigma(2)} = x_{\sigma(2)}$
$p_1 = 0$	$p_2 = 0$	$y_{\sigma(1)} = x_{\sigma(1)}$ $y_{\sigma(2)} = x_{\sigma(2)}$
$p_1 < -1$	$p_2 = 0$	$y_{\sigma(1)} = x_{\sigma(1)} - 1$ $y_{\sigma(2)} = x_{\sigma(2)}$

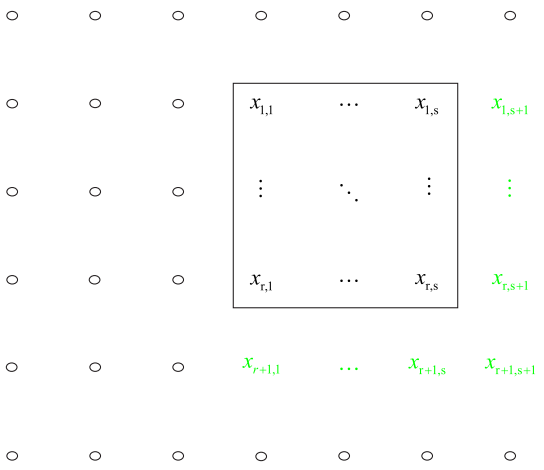

FIGURE 1. An $r \times s$-sized block along with its neighboring pixels, where pixels in black and green respectively represent pixels within a block and its neighboring pixels.

ing the embedding performance. The local complexity is an effective measurement for smoothness classification. The more precisely the local complexity is measured, the better embedding performance can be achieved. In this paper, two largest and two smallest pixels in a block are modified for data hiding, while the remaining pixels keep unaltered before and after data embedding. These unchanged pixels are very helpful for increasing the evaluation accuracy of local complexity, and thus, they are used along with the neighborhood of this block to measure the block complexity.

Specifically, we utilize an $r \times s$-sized block to illustrate the process of calculating the local complexity. Fig. 1 is given to show this $r \times s$-sized block along with its neighborhood. This block is sorted in an ascending order to generate an n-sized pixel list $X = \{x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n-1)}, x_{\sigma(n)}\}$. Except for two largest and two smallest pixels, the remaining $n-4$ pixels $\{x_{\sigma(3)}, x_{\sigma(4)}, \ldots, x_{\sigma(n-2)}\}$ are exploited for complexity measurement. As illustrated in Fig. 1, the pixels $x_{r+1,1}, \cdots, x_{r+1,s+1}, x_{1,s+1}, \cdots, x_{r,s+1}$ constitute the neighborhood. This neighborhood coupled with $n-4$ unmodified pixels are utilized to measure the local complexity of this block. The block complexity, denoted by Δ, is calculated referring to Eq. (7), as shown at the bottom of this page.

where u denotes the average value among $\{x_{\sigma(3)}, x_{\sigma(4)}, \ldots, x_{\sigma(n-2)}\}$ and $r+s+1$ neighboring pixels surrounding a block.

Based on the local variance Δ defined in Eq. (7), all the blocks are classified into three types: smooth ($\Delta < T_1$), normal ($T_1 \le \Delta < T_2$) and textured blocks $\Delta \ge T_2$ with the assist of two predefined thresholds T_1 and T_2. The detailed classification process is listed below, and three embedding modes are respectively performed.

Case 1. $\Delta < T_1$

The smaller the Δ, the smoother the blocks. When $\Delta < T_1$, the block is deemed as a smooth block and it is classified

$$\Delta = \sqrt{\frac{\sum_{k=1}^{s+1}(x_{r+1,k}-u)^2 + \sum_{k=1}^{r}(x_{k,s+1}-u)^2 + \sum_{k=3}^{n-2}(x_{\sigma(k)}-u)^2}{r+s+n-3}}, \tag{7}$$

TABLE 4. The detailed modification process of two smallest pixels considering the relationship between $\sigma(1)$ and $\sigma(2)$.

prediction error p_1	prediction error p_2	the relationship between $\sigma(1)$ and $\sigma(2)$	marked pixels
$p_1 < -1$	$p_2 = 1$	$\sigma(1) < \sigma(2)$	$y_{\sigma(1)} = x_{\sigma(1)} - 1$ $y_{\sigma(2)} = x_{\sigma(2)} - w$
		$\sigma(1) > \sigma(2)$	$y_{\sigma(1)} = x_{\sigma(1)} - 1 - w$ $y_{\sigma(2)} = x_{\sigma(2)}$
	$p_2 > 1$	$\sigma(1) < \sigma(2)$	$y_{\sigma(1)} = x_{\sigma(1)} - 1$ $y_{\sigma(2)} = x_{\sigma(2)} - 1$
		$\sigma(1) > \sigma(2)$	$y_{\sigma(1)} = x_{\sigma(1)} - 2$ $y_{\sigma(2)} = x_{\sigma(2)}$
	$p_2 = -1$	$\sigma(1) < \sigma(2)$	$y_{\sigma(1)} = x_{\sigma(1)} - 1 - w$ $y_{\sigma(2)} = x_{\sigma(2)}$
		$\sigma(1) > \sigma(2)$	$y_{\sigma(1)} = x_{\sigma(1)} - 1$ $y_{\sigma(2)} = x_{\sigma(2)} - w$
	$p_2 < -1$	$\sigma(1) < \sigma(2)$	$y_{\sigma(1)} = x_{\sigma(1)} - 2$ $y_{\sigma(2)} = x_{\sigma(2)}$
		$\sigma(1) > \sigma(2)$	$y_{\sigma(1)} = x_{\sigma(1)} - 1$ $y_{\sigma(2)} = x_{\sigma(2)} - 1$

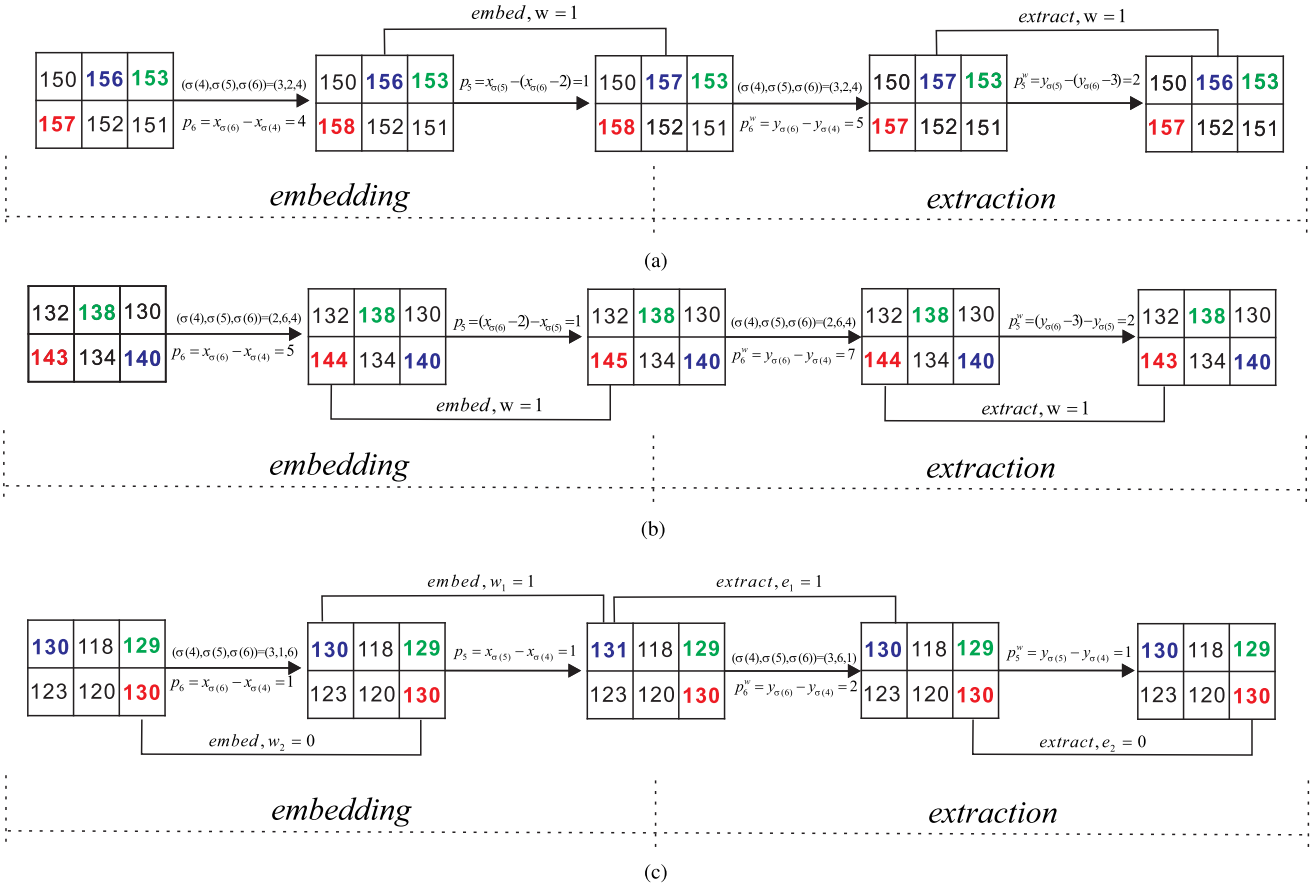

FIGURE 2. Examples of maximum-direction modification. (a) Example 1. (b) Example 2. (c) Example 3. (a) Example 1. (b) Example 2. (c) Example 3.

into Case 1. We perform the embedding method described in Tables 1, 2, 3, 4 so that each image block in Case 1 can carry four data bits at most.

Case 2. $T_1 \leq \Delta < T_2$

For a block with $T_1 \leq \Delta < T_2$, we classify it into a normal block. We take advantage of IPVO [24] proposed by Peng *et al.* as the embedding strategy. In IPVO, two bits at most can be embedded into the image block.

Case 3. $\Delta \geq T_2$

When $\Delta \geq T_2$, this image block belongs to Case 3 and it is located in the rough region. For a block in Case 3, it remains unchanged in the embedding phase.

C. SIMPLE EXAMPLES

In order to have a better understanding for the proposed scheme, three examples are given in Fig. 2 to illustrate our

strategy is feasible and effective. Here, we will give a brief explanation for each example shown in Fig. 2.

In Example 1, an original 2×3 block is scanned in raster scanning order to form a 6-sized pixel list $B = \{150, 156, 153, 157, 152, 151\}$. Three largest pixels are selected from the pixel list B, and the set of their corresponding locations $\{\sigma(4), \sigma(5), \sigma(6)\}$ in the pixel list B is $\{3, 2, 4\}$. As illustrated in Fig. 2(a), the maximum $x_{\sigma(6)}$ marked in red is predicted by the third largest pixel $x_{\sigma(4)}$ marked in green to generate the prediction error $p_6 = 157 - 153 = 4$. Since $p_6 > 1$ and $\sigma(5) < \sigma(6)$, the second largest pixel $x_{\sigma(5)}$ marked in purple is predicted by $x_{\sigma(6)} - 2$ according to Eq. (5), and the prediction error $p_5 = 156-(157-2) = 1$. When $p_5 = 1$, $p_6 > 1$ and $\sigma(5) < \sigma(6)$, referring to Table 2, the marked pixel $y_{\sigma(6)}$ is increased by 1, i.e., $y_{\sigma(6)} = x_{\sigma(6)} + 1$, and $y_{\sigma(5)}$ can be embedded with one bit. If the to-be-embedded bit w is 1, then $y_{\sigma(5)} = x_{\sigma(5)} + w = 157$. After two largest pixels have been modified, the marked list $\{150, 157, 153, 158, 152, 151\}$ is formed. At the extraction stage, the sorted marked list $\{150, 151, 152, 153, 157, 158\}$ is generated by sorting the marked list in ascending order. The maximum $y_{\sigma(6)}$ is predicted by the third largest pixel $y_{\sigma(4)}$ to yield the prediction error $p_6^w = 158 - 153 = 5$. Since $p_6^w > 2$, $x_{\sigma(6)}$ is retrieved by decreasing $y_{\sigma(6)}$ by 1, namely $x_{\sigma(6)} = y_{\sigma(6)} - 1$. Moreover, when $p_6^w > 2$ and $\sigma(5) < \sigma(6)$, the prediction error $p_5^w = 157 - (158 - 3) = 2$ via Eq. (8), as shown at the bottom of the next page. $p_5^w = 2$ indicates that the embedded data bit w is 1. Therefore, $x_{\sigma(5)} = y_{\sigma(5)} - w = 156$. After two largest pixels have been restored correctly, the original pixel list B is retrieved. Finally, the original block is obtained by scanning the pixel list B in raster order.

In Example 2 shown in Fig. 2(b), a 2×3-sized block is converted into $\{130, 132, 134, 138, 140, 143\}$ by sorting. The maximum $x_{\sigma(6)}$ is predicted by the third largest pixel $x_{\sigma(4)}$ to generate the prediction error $p_6 = 143 - 138 = 5$. Since $p_6 > 1$ and $\sigma(5) > \sigma(6)$, the second largest pixel $x_{\sigma(5)}$ is predicted by $x_{\sigma(6)} - 2$ according to Eq. (5), and the prediction error $p_5 = (143 - 2) - 140 = 1$. Referring to Table 2, when $p_6 > 1$, $p_5 = 1$ and $\sigma(5) > \sigma(6)$, $y_{\sigma(5)} = x_{\sigma(5)} = 140$ and $y_{\sigma(6)} = x_{\sigma(6)} + w + 1$. If $w = 1$, then $y_{\sigma(6)} = 145$. Finally, the marked pixel list is generated as $\{132, 138, 130, 145, 134, 140\}$. At the extraction stage, the maximum $y_{\sigma(6)}$ is predicted by the third largest pixel $y_{\sigma(4)}$ to generate the prediction error $p_6^w = 145 - 138 = 7$. Owing to $p_6^w > 2$ and $\sigma(5) > \sigma(6)$, the prediction error $p_5^w = (145 - 3) - 140 = 2$ according to Eq. (8). $p_5^w = 2$ indicates that the embedded data bit w is 1. Depending on $p_5^w = 2$, $p_6^w > 2$ and $\sigma(5) > \sigma(6)$, the original pixels are recovered based on Table 2 as follows: $x_{\sigma(6)} = y_{\sigma(6)} - w - 1 = 143$ and $x_{\sigma(5)} = y_{\sigma(5)} = 140$. Finally, the original pixel list B is recovered completely.

From Table 1, it can be known that when $p_n = p_{n-1} = 1$, $y_{\sigma(n-1)} = x_{\sigma(n-1)} + w_1$ and $y_{\sigma(n)} = x_{\sigma(n)} + w_2$, where $w_1 \in \{0, 1\}$ and $w_2 \in \{0, 1\}$. It should be mentioned that $p_n = p_{n-1} = 1$ holds when $x_{\sigma(n)} = x_{\sigma(n-1)}$ and $\sigma(n) > \sigma(n-1)$. However, when w_1 is set greater than w_2, $y_{\sigma(n-1)}$ exceeds $y_{\sigma(n)}$ so that the pixel value ordering (PVO) is changed. Example 3 is given to explain two embedded bits still can be correctly extracted even if the PVO is changed. From Fig. 2(c), we know $y_{\sigma(5)} = x_{\sigma(5)} + w_1 = 131$, and $y_{\sigma(6)} = x_{\sigma(6)} + w_2 = 130$ after data embedding. On the decoding side, the maximum $y_{\sigma(6)}$ is predicted by the third largest pixel $y_{\sigma(4)}$ to obtain the prediction error $p_6^w = 131 - 129 = 2$. Because of $p_6^w = 2$, $p_5^w = 130 - 129 = 1$ according to Eq. (8). $p_6^w = 2$ indicates that p_6^w has been embedded with one bit e_1, and $e_1 = 1$. Then, $x_{\sigma(6)}$ is recovered as $x_{\sigma(6)} = y_{\sigma(6)} - e_1 = 130$. Similarly, $p_5^w = 1$ implies that p_5^w has been embedded with one bit e_2, and $e_2 = 0$. Thus, $x_{\sigma(5)} = y_{\sigma(5)} - e_2 = 130$. According to the description above, e_1 is extracted from $y_{\sigma(6)}$, and therefore its corresponding position in B is $\sigma(6)$, while e_2 is extracted from $y_{\sigma(5)}$, and its corresponding position in B is $\sigma(5)$. Since $\sigma(6)$ is listed before $\sigma(5)$, e_1 should also be listed before e_2. Therefore, the extracted two bits are resorted as $1, 0$, which is the same as $(w_1, w_2) = 1, 0$ in embedding.

D. EMBEDDING PROCEDURE

To ensure reversibility, it is necessary to record the auxiliary information, which helps the receiver side for image recovery and data extraction. The auxiliary information is composed of the following parts:

- compressed location map $L(L_S$ bits): After the location map is generated, it is losslessly compressed by an arithmetic encoder.
- block size r(3 bits) and s(3 bits)
- thresholds T_1(8 bits) and T_2(8 bits)
- the location where the embedding process is terminated: 9 bits for row number and 9 bits for column number
- an end of symbol(EOS)(8 bits)

The detailed procedure is formulated as follows:

Step 1. Partition the cover image into blocks B_i of size $r \times s$, where $i \in \{1, \ldots, N\}$ and N denotes the last modified block.

Step 2. Create a location map(LM) to solve the potential overflow or underflow problem. For a smooth block, two largest and two smallest pixels are modified during data embedding. Specifically, the maximum $x_{\sigma(n)}$ or the minimum $x_{\sigma(1)}$ is modified by 2 at most and the second largest pixel $x_{\sigma(n-1)}$ or the second smallest pixel $x_{\sigma(2)}$ is modified by 1 at most. The overflow occurs when $x_{\sigma(n)}$ valued 255 or 254 may be modified to 257 or 256 or $x_{\sigma(n-1)}$ valued 255 may be modified to 256. Similarly, the underflow occurs when $x_{\sigma(1)}$ valued 1 or 0 may be modified to -1 or -2 or $x_{\sigma(2)}$ valued 0 may be modified to -1. For a normal block, the overflow or underflow occurs when $x_{\sigma(n)}$ valued 255 may be modified to 256 or $x_{\sigma(1)}$ valued 0 may be modified to -1. We call the pixels leading to overflow or underflow as overflow/underflow pixels. The location of this block in the location map is marked by 0 if a block contains overflow or underflow pixels and 1 otherwise.

Step 3. For each $r \times s$-sized block, when its corresponding location in LM is 1, perform different embedding modes described in section B:complexity measurement for data hiding.

Step 4. Extract the first $L_s + 48$ LSBs of the marked pixels and replace these vacant LSBs with auxiliary information. Append the extracted LSBs to the payload, and embed all of them into the cover image according to Step 3.

E. DATA EXTRACTION AND IMAGE RECOVERY PROCEDURE

At the extraction stage, auxiliary information plays a significant role in data extraction and image recovery. Therefore, extraction of auxiliary information should be initially performed on the receiver side. Once EOS is recognized, we collect the LSBs of the marked image before EOS and decompress them by an arithmetic decoder to recover the location map. The location map is then compressed by an arithmetic encoder to obtain L_s, and all the remaining components of auxiliary information can be identified. The detailed extraction procedure is listed as follows:

Step 1. Partition the marked image into blocks B_i^w of size $r \times s$, where $i \in \{1, 2, \ldots, N\}$ and N denotes the last modified block. We realize the data extraction in the light of the reversed order performed in data hiding. In other words, the last modified block is initially restored while the first modified block is eventually recovered at the extraction stage.

Step 2. For each $r \times s$-sized block $Y = \{y_{\sigma(1)}, y_{\sigma(2)}, \ldots, y_{\sigma(n-1)}, y_{\sigma(n)}\}$, when its corresponding location in LM is 1 and its block complexity satisfies $\Delta < T_1$, the prediction error p_i^w is obtained in Eq. (8), where $i \in \{1, 2, n-1, n\}$. The cover image can be perfectly restored based on the embedding method described in section A:prediction mechanism. On the other hand, if the complexity of a block is confined to $T_1 \leq \Delta < T_2$, then perform the extraction process according to IPVO.

IV. EXPERIMENTAL RESULTS

In this section, we compare the proposed method with other state-of-the-art works in terms of payload and the image quality. Peak signal-to-noise ratio (PSNR) is employed to measure the image quality. Eight 512×512-sized images including Baboon, Barbara, Boat, Lake, Elaine, Peppers, Lena and Airplane are utilized as test images.

A. COMPARISON WITH K-PASS PVO BASED TECHNIQUE

In the embedding phase, both our scheme and k-pass PVO enable an image block to carry at most 4 data bits. When the difference between the largest(smallest) and third largest(smallest) pixels is relatively large, we improve k-pass PVO by considering the relative location of the maximum(minimum) and second largest(smallest) pixel to calculate the prediction error between them. In our method, bins 1 and -1 are both employed for embedding while only bin 1 is utilized in k-pass PVO. In order to demonstrate the effectiveness of our method, Table 5 and Table 6 are employed in comparing k-pass PVO with the proposed method in terms of the PSNR and distortion.

TABLE 5. Comparison of distortion and PSNR with payload of 10,000 bits.

Image	distortion		PSNR	
	the proposed	k-pass PVO	the proposed	k-pass PVO
Barbara	14437	15862.5	60.7190	60.37
Baboon	58997	67450	54.6042	54
Boat	23494	26089	58.6103	58.28
Lake	17552	18705	59.8643	59.71
Lena	14582	15090	60.6807	60.64
Elaine	30383	29013	57.4922	57.67
Peppers	20768	22247.5	59.1435	59.29
Airplane	8020	7801.5	63.2788	63.45

TABLE 6. Comparison of distortion and PSNR with payload of 20,000 bits.

Image	distortion		PSNR	
	the proposed	k-pass PVO	the proposed	k-pass PVO
Barbara	36078	41837	56.7397	56.09
Baboon	-	-	-	-
Boat	64790	69030	54.2693	54.07
Lake	58205	60036.5	54.6629	54.53
Lena	36719	35569.5	56.6678	56.81
Elaine	82680	83786	53.1406	53.08
Peppers	54147	52491	54.9697	55.1
Airplane	20573	19226	59.1849	59.59

As shown in Table 5 and Table 6, the proposed scheme cannot achieve optimal performance for all types of images. For images with relatively higher texture like Baboon, Barbara, Boat and Lake, our method is able to provide more payload than k-pass PVO. Actually, the thresholds T_1 and T_2 are two payload-control parameters. When the high payload is required, two thresholds T_1 and T_2 need to be set larger values so that more blocks can be involved in data embedding.

$$p_i^w = \begin{cases} y_{\sigma(i)} - y_{\sigma(n-2)}, & i = n, \\ y_{\sigma(i)} - y_{\sigma(n-2)}, & i = n-1 \text{ and } p_n^w \leq 2, \\ y_{\sigma(i)} - (y_{\sigma(n)} - 3), & i = n-1 \text{ and } p_n^w > 2 \text{ and } \sigma(n-1) < \sigma(n), \\ (y_{\sigma(n)} - 3) - y_{\sigma(i)}, & i = n-1 \text{ and } p_n^w > 2 \text{ and } \sigma(n-1) > \sigma(n), \\ y_{\sigma(i)} - y_{\sigma(3)}, & i = 1, \\ y_{\sigma(i)} - y_{\sigma(3)}, & i = 2 \text{ and } p_1^w \geq -2, \\ y_{\sigma(i)} - (y_{\sigma(1)} + 3), & i = 2 \text{ and } p_1^w < -2 \text{ and } \sigma(2) < \sigma(1), \\ (y_{\sigma(1)} + 3) - y_{\sigma(i)}, & i = 2 \text{ and } p_1^w < -2 \text{ and } \sigma(2) > \sigma(1). \end{cases} \tag{8}$$

With two thresholds T_1 and T_2 increased, some blocks located in highly-textured regions are no longer excluded from data embedding, and can also help to increase the payload. Compared with k-pass PVO, our advantage lies in the fact that the difference valued 3 between the maximum and second largest pixel, is capable of embedding 1 data bit. With two thresholds increased, the number of differences valued 3 is largely increased. As a result, the proposed scheme can achieve remarkably superior embedding performance for large payload. Besides, for the cover images with high texture, a large amount of differences valued 3 can be used for data embedding, and therefore, our method can also achieve higher embedding performance than k-pass PVO.

B. COMPARISON OF PAYLOAD AND IMAGE QUALITY MEASURED BY PSNR

Fig. 3 shows the marked image quality of the eight test images measured by PSNR when k-pass PVO [26], PVO-k [25], IPVO [24], PVO [23], KPPE [27] and Sachnev *et al.*'s method [9] are applied.

As for Baboon, Barbara and Boat, superior performance can be achieved at all embedding rates in our method in contrast to k-pass PVO. Slight increase in PSNR is introduced in our algorithm at low embedding rates. On the other hand, when it comes to higher embedding rates, the proposed scheme significantly outperforms k-pass PVO. More specifically, the improvement in PSNR is up to 0.9 dB for Baboon and 2.05 dB for Barbara at maximum payload of 13,000 bits and 30,000 bits, respectively. Experimental results in other images illustrate that the distortion generated in our method is slightly higher than k-pass PVO at low embedding rates. In contrast, the performance is capable of exceeding that of k-pass PVO at higher payload. For example, when the payload is larger than 19,000 bits for Lake and 34,000 bits for Lena, higher PSNR is provided in the proposed scheme. Virtually, when the image is highly-textured or large payload is required, more pixels within rough regions should be classified into Case 1 and Case 2 in order that sufficient payload is achieved. For the difference with value 3 between the maximum and second largest pixel, it is shifted in k-pass PVO, while the prediction mechanism in our method enables it to be embeddable. Besides, the cover image with high texture or large payload provides a remarkably greater number of difference valued 3 between the maximum and second largest pixel. Therefore, in contrast to k-pass PVO, the proposed method increases the number of embeddable pixels so that the payload is enhanced dramatically, while introduces fewer shifted pixels correspondingly so that less distortion is generated. On the other hand, some pixels have to be shifted twice in our method to ensure reversibility. It seems that they may bring huge distortion. However, since these pixels account for a small proportion of all the modified pixels and the proposed scheme allows a significant increase in embedding capacity, the new algorithm can still be able to achieve more effective capacity-distortion trade-off, especially for the images with high texture or large payload.

Compared with IPVO and PVO-k, the proposed method always has preferable performance for almost all test images. In IPVO and PVO-k based schemes, a block can carry 2 data bits at best. In order to achieve ample payload, blocks with small size should be employed, which leads to weaker accuracy for prediction. On the other hand, 4 data bits are embedded within a block in our scheme. It contributes to larger payload so that our scheme can adopt larger-sized blocks. In this way, stronger prediction accuracy and a sharper prediction error histogram are provided. Stemming from the principle that the sharper the histogram distributes, the better the embedding performance is, better image quality with higher payload is offered compared to IPVO and PVO-k based schemes.

In Sachnev *et al.*'s method, rhombus prediction pattern and pixel selection are taken into consideration so that more accurate prediction is obtained. Specifically, the center pixel is predicted by its four neighboring pixels and pixels belonging to smooth regions are preferentially utilized for embedment. Basically, the method proposed by Sachnev *et al.* is dominant in the cover image with high texture or large embedding capacity. As shown in Fig. 3, when it comes to higher embedding rates, superior PSNR is achieved in Sachnev *et al.*'s method for Barbara, Baboon and Airplane compared with k-pass PVO, PVO-k and IPVO. On the contrary, our scheme is capable of achieving comparable or superior performance compared with Sachnev *et al.*'s method at all embedding rates. It is because the smoothness of a block is classified more precisely in the proposed scheme by employing more pixels for complexity measurement. Specifically, only four neighboring pixels are utilized to measure the complexity of the center pixel in Sachnev *et al.*'s method, while we adopt more than four pixels i.e. the neighborhood of a block and the unmodified pixels within this block for smoothness classification.

Recently, He et al. have proposed an RDH technique named KPPE [27], which combines k-pass PVO with pairwise PEE. Pairwise PEE helps KPPE to largely improve embedding performance by avoiding two adjacent embeddable prediction-errors being simultaneously increased or decreased by 1. As illustrated in Fig. 3, our proposed method can still achieve higher embedding performance than KPPE when the payload exceeds 40,000 bits for Airplane and 11,000 bits for Baboon. For Barbara, Boat and Lake, it is noticeable our proposed scheme can achieve comparable embedding performance for higher payloads when compared with KPPE. Therefore, we conclude our proposed method can still achieve comparable or better embedding performance than the papers [9], [23]–[27].

Our future work will focus on combining pairwise PEE with our embedding mechanism. We believe that if we introduce pairwise PEE into the proposed method, better visual quality with high payload can be realized compared with KPPE.

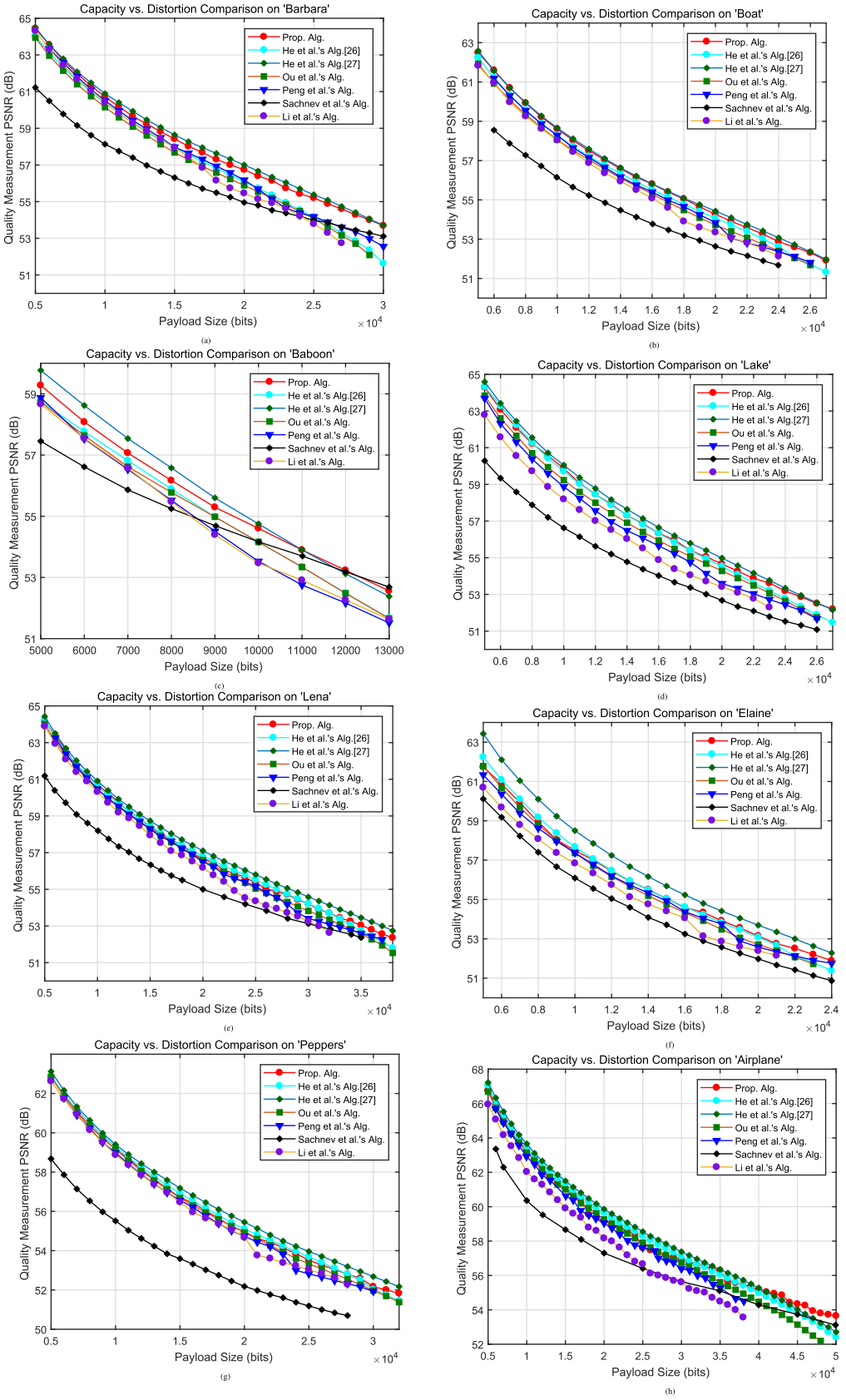

FIGURE 3. Comparison of capacity-distortion performance evaluated by PSNR among seven RDH schemes. (a) Peppers. (b) Airplane. (a) Barbara. (b) Boat. (c) Baboon. (d) Lake. (e) Lena. (f) Elaine. (g) Peppers. (h) Airplane.

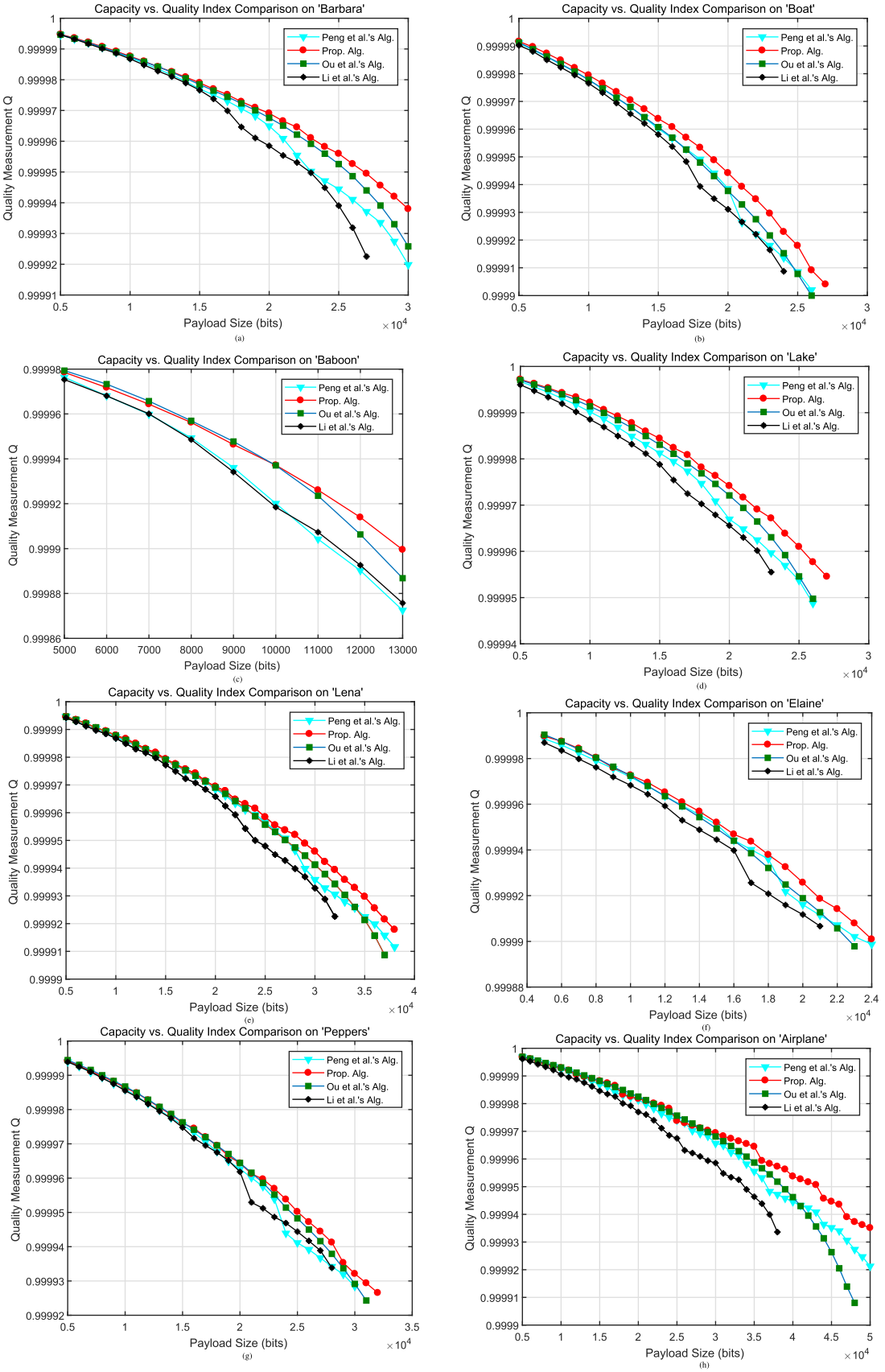

FIGURE 4. Comparison of capacity-distortion performance evaluated by quality index(Q) among four RDH schemes. (a) Barbara. (b) Boat. (c) Baboon. (d) Lake. (e) Lena. (f) Elaine. (g) Peppers. (h) Airplane.

C. COMPARISON OF PAYLOAD AND IMAGE QUALITY MEASURED BY QUALITY INDEX(Q)

Quality index(Q) is a quality measurement to measure the distortion between the host and marked images. The higher the quality index is, the lower distortion the embedding mechanism introduces. Suppose I and I' respectively represent an $A \times B$-sized host image and its corresponding marked image, and quality index(Q) is then calculated according to Eq. (9) utilized in [28]–[33].

$$Q = \frac{4 \times ave \times ave' \times \sigma_{12}}{(\sigma_1{}^2 + \sigma_2{}^2) \times (ave^2 + ave'^2)}, \tag{9}$$

where

$$ave = \frac{1}{A \times B} \sum_{i=1}^{A} \sum_{j=1}^{B} I(i,j),$$

$$ave' = \frac{1}{A \times B} \sum_{i=1}^{A} \sum_{j=1}^{B} I'(i,j),$$

$$\sigma_{12} = \frac{1}{A \times B - 1} \sum_{i=1}^{A} \sum_{j=1}^{B} (I(i,j) - ave)(I'(i,j) - ave'),$$

$$\sigma_1{}^2 = \frac{1}{A \times B - 1} \sum_{i=1}^{A} \sum_{j=1}^{B} [I(i,j) - ave]^2,$$

$$\sigma_2{}^2 = \frac{1}{A \times B - 1} \sum_{i=1}^{A} \sum_{j=1}^{B} [I'(i,j) - ave']^2.$$

We have simulated k-pass PVO and KPPE, but the resulting experimental data is worse than that in [26] and [27]. To this end, we had to contact with the author He, and He only provided the experimental data (containing PSNR values under given payloads) of these eight images to us. Since we did not obtain the source codes of k-pass PVO and KPPE, we can only give the performance comparisons evaluated by quality index(Q) among PVO-k [25], IPVO [24] and PVO [23] and our method.

Fig. 4 shows the marked image quality of the eight test images measured by quality index(Q) among PVO [23], IPVO [24] and PVO-k [25] and our proposed method. For Airplane and Baboon, the distortion introduced by the proposed method is slightly higher than Ou et al.'s method for low payloads. However, when it comes to higher payload, the proposed method predominantly outperforms Ou et al.'s, Peng et al.'s and Li et al.'s methods in terms of image visual quality measured by quality index(Q). As for other images, superior visual quality can be achieved at all embedding rates in our method compared with other methods. Specifically, with the increase of payload, the gap of quality index(Q) among our method and the other three methods is becoming more and more obvious and distinct. It is because when larger payload is required, the proposed method enables a larger amount of pixels, which are shifted in other PVO-based methods, to carry data bits while reduces the number of shifted pixels so that capacity-distortion performance is enhanced compared with other methods.

V. CONCLUSION

In this paper, we propose an improved k-pass PVO based scheme by exploring the location relationship of the largest and second largest pixels(or the smallest and second smallest pixels) to calculate the prediction errors. In this way, a larger amount of embeddable pixels are provided, while the number of shifted pixels is decreased dramatically. In addition, the proposed scheme combines the unchanged pixels within a block with its neighbors to enhance the prediction accuracy in terms of local complexity. The experimental results show that the proposed strategy provides better visual quality, especially when the cover image is highly-textured or requires large payload.

REFERENCES

[1] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[2] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 253–266, Feb. 2005.

[3] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1147–1156, Aug. 2004.

[4] X. Wang, X. Li, B. Yang, and Z. Guo, "Efficient generalized integer transform for reversible watermarking," *IEEE Signal Process. Lett.*, vol. 17, no. 6, pp. 567–570, Jun. 2010.

[5] F. Peng, X. Li, and B. Yang, "Adaptive reversible data hiding scheme based on integer transform," *Signal Process.*, vol. 92, no. 1, pp. 54–62, 2012.

[6] S. W. Weng and J.-S. Pan, "Integer transform based reversible watermarking incorporating block selection," *J. Vis. Commun. Image Represent.*, vol. 35, pp. 25–35, Feb. 2016.

[7] 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.

[8] 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.

[9] 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.

[10] Y. Hu, H.-K. Lee, and J. Li, "DE-based reversible data hiding with improved overflow location map," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 250–260, Feb. 2009.

[11] X. Gao, L. An, Y. Yuan, D. Tao, and X. Li, "Lossless data embedding using generalized statistical quantity histogram," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 8, pp. 1061–1070, Aug. 2011.

[12] 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.

[13] D. Coltuc, "Improved embedding for prediction-based reversible watermarking," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 873–882, Sep. 2011.

[14] 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.

[15] 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, 2009.

[16] W.-L. Tai, C.-M. Yeh, and C.-C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 6, pp. 906–910, Jun. 2009.

[17] 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, 2009.

[18] H.-T. Wu and J. Huang, "Reversible image watermarking on prediction errors by efficient histogram modification," *Signal Process.*, vol. 92, no. 12, pp. 3000–3009, Dec. 2012.
[19] G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, and C. Roux, "Reversible watermarking based on invariant image classification and dynamic histogram shifting," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 111–120, Jan. 2013.
[20] S.-W. Jung, L. T. Ha, and S.-J. Ko, "A new histogram modification based reversible data hiding algorithm considering the human visual system," *IEEE Signal Process. Lett.*, vol. 18, no. 2, pp. 95–98, Feb. 2011.
[21] W. Hong, T. Chen, and M. Wu, "An improved human visual system based reversible data hiding method using adaptive histogram modification," *Opt. Commun.*, vol. 291, pp. 87–97, Mar. 2013.
[22] S. Weng and J.-S. Pan, "Reversible watermarking based on multiple prediction modes and adaptive watermark embedding," *Multimedia Tools Appl.*, vol. 72, no. 3, pp. 3063–3083, 2014.
[23] X. Li, J. Li, B. Li, and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion," *Signal Process.*, vol. 93, no. 1, pp. 198–205, 2013.
[24] F. Peng, X. Li, and B. Yang, "Improved PVO-based reversible data hiding," *Digit. Signal Process.*, vol. 25, pp. 255–265, Feb. 2014.
[25] B. Ou, X. Li, Y. Zhao, and R. Ni, "Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion," *Signal Process., Image Commun.*, vol. 29, no. 7, pp. 760–772, 2014.
[26] W. He, K. Zhou, J. Cai, L. Wang, and G. Xiong, "Reversible data hiding using multi-pass pixel value ordering and prediction-error expansion," *J. Vis. Commun. Image Represent.*, vol. 49, pp. 351–360, Nov. 2017.
[27] W. He, G. Xiong, S. Weng, Z. Cai, and Y. Wang, "Reversible data hiding using multi-pass pixel-value-ordering and pairwise prediction-error expansion," *Inf. Sci.*, vol. 467, pp. 784–799, Oct. 2018.
[28] G. Swain, "Adaptive pixel value differencing steganography using both vertical and horizontal edges," *Multimedia Tools Appl.*, vol. 75, no. 21, pp. 13541–13556, 2016.
[29] A. K. Sahu and G. Swain, "An improved data hiding technique using bit differencing and LSB matching," *Internetworking Indonesia J.*, vol. 10, no. 1, pp. 17–21, 2018.
[30] A. K. Sahu and G. Swain, "Pixel overlapping image steganography using PVD and modulus function," *3D Res.*, vol. 9, no. 3, p. 40, 2018.
[31] A. K. Sahu and G. Swain, "A novel n-rightmost bit replacement image steganography technique," *3D Res.*, vol. 10, no. 1, p. 2, 2019.
[32] G. Swain, "Very high capacity image steganography technique using quotient value differencing and LSB substitution," in *Arabian Journal for Science and Engineering*. Berlin, Germany: Springer, 2018, pp. 1–10.
[33] A. K. Sahu and G. Swain, "Information hiding using group of bits substitution," *Int. J. Commun. Antenna Propag.*, vol. 7, no. 2, pp. 162–167, 2017.

Authors' photographs and biographies not available at the time of publication.