

# Reversible Data Hiding in Color Image with Grayscale Invariance

Dongdong Hou, Weiming Zhang, Kejiang Chen, Sian-Jheng Lin and Nenghai Yu

**Abstract**—Different from all the previous reversible data hiding schemes, a completely novel one for the color image is proposed, which reversibly embeds messages into the color host image without modifying its corresponding gray version. The property of grayscale invariance is valuable, because many applications and image processing algorithms for color images are based on the corresponding gray versions, such as black and white printing, producing reading materials for color blind people, single-channel image processing, and so on. Thus, in terms of these applications and image processing algorithms, the presented scheme will make the generated color marked image be free for its further uses. In this paper, the unchanged gray version is utilized efficiently in both the embedding processes and the extracting processes. Messages are embedded into the red and blue channels of color image, and then the green channel is adjusted adaptively to remove the offsets from the gray version caused by modifying its red and blue channels. To return the adjusted green channel, error correcting bits guaranteeing the reversibility are regarded as one part of payloads to be recursively embedded. Therefore, the reversibility and the property of grayscale invariance are both achieved.

**Index Terms**—color image, grayscale invariance, free for image processing, free for applications, reversible data hiding.

## I. INTRODUCTION

Data hiding embeds messages into digital multimedia such as image, audio, video etc through an imperceptible way. Such technology plays the important role in protecting the privacy, which can be roundly classified as robust watermarking [1] having the ability to resist many attacks, fragile watermarking [2], [3] being fragile to any modifications, and steganography [4] with the strong undetectability. From the application side, robust watermarking, steganography and fragile watermarking are mainly used for copyright protection, covert communication and integrity authentication respectively. A review of data hiding techniques and a few emerging innovative solutions using data hiding are well introduced in [5].

However, robust watermarking, steganography and many fragile watermarking algorithms like [2] will destroy the host signal irreversibly. Some special signals are so precious and cannot be damaged such as medical imagery, military imagery and law forensics. Therefore, recovering the host signal from the marked signal completely is rather important.

This work was supported in part by the Natural Science Foundation of China under Grant 61572452, U1636201.

D. Hou, W. Zhang, K. Chen, S. Lin and N. Yu are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, 230026, China. (Email: houdd@mail.ustc.edu.cn, zhangwm@ustc.edu.cn, chenkj@mail.ustc.edu.cn, sjlin@ustc.edu.cn, ynh@ustc.edu.cn.)

Corresponding author: Weiming Zhang

Reversible data hiding (RDH) [3] is one special type of fragile watermarking, and usually is used for integrity authentication, by which the host signal as well as the embedded data can be both restored from the marked signal without any loss. Many RDH algorithms have been proposed in the past decade, which can be roughly classified into three fundamental strategies: lossless compression appending scheme [6], difference expansion [7] and histogram shift [8]. The state of the art of RDH techniques combine these strategies to the residuals of images such as prediction errors (PEs) [9]–[15] to achieve a better performance.

All the RDH algorithms mentioned above are presented for gray images. Indeed, color images are much more popular than the gray ones in practice, because color is a powerful visual descriptor which simplifies the identification of an object from a scene. There are many watermarking algorithms in the color space, such as luminance-channel-based algorithms [16], [17], chrominance-channel-based algorithms [18], [19], three-color-channel-based algorithms [20], [21] and blue-channel-based algorithms [22]. However, these watermarking algorithms will destroy color images irreversibly. To embed watermarking into color images and also restore images without any loss, some researchers begin to explore the characteristics of color images and design special RDH algorithms for color images. The existing RDH algorithms for color images [23]–[26] mainly focus on generating the more shaper prediction error (PE) histogram and minimizing the total distortion by exploring the correlations within each color channel and among three color channels (R, G, B).

RDH will cause distortion for the host image, and the generated marked image will be deemed as a noisy image compared to the original host image. Therefore, after RDH on the image, its further image processing will be interfered. For example, in the system of object recognition, a marked image will affect the feature extraction and thus result in the decrease of recognition accuracy. For some important applications such as police system and military system, slight decrease of processing accuracy may involve fateful consequences. Of course, as for RDH, the original copy can be restored from the marked image before the further processing. However, besides the computing cost of recovery algorithm, in many cases it is just rather difficult for the processors to recover each original image due to that the embedding method and secret key are mastered by image owners who perhaps do not want to share this information, and the processors even do not know whether the input image is a marked copy or not among the huge number of digital images on the Internet. Therefore, a novel RDH scheme for images not interfering with the further image

processing is desired.

There are various of processing methods for color images, among which some methods only operate on luminance channel (green version) and entirely ignore chromatic information. For example, there are various of feature descriptors based on gray images such as the well known Haar-like descriptors [27], HOG (Histogram of Oriented Gradients) descriptors [28], SIFT (scale invariant feature transform) descriptors [29], and so on. Taking SIFT descriptors as example, which can be applied to image compression [30]–[32], pattern recognition [33], [34], image stitching [35], [36], and so on. As far as we know, almost all of SIFT or SIFT-extended descriptors like SURF (Speeded Up Robust Features) [37] and their applications only operate on the luminance channel. That is to say as for many processing algorithms before processing color images we usually convert them into gray versions. The reasons are clearly, on the one hand, comparing to color image processing, dealing with gray image reduces the computational costs greatly. On the other hand, gray version can well represent the structures and the contents of one image, thus for some processing methods operating on gray version is enough. Therefore, as for these color image processing methods, to avoid interfering with the further processing of color image we only need to hold its corresponding gray version.

Besides not interfering with the further image processing, gray versions of color images are widely used in many other applications, including black and white printing (e-ink-based book readers), producing reading materials for color blind people, nonphotorealistic rendering with black-and-white media, and so on. For some devices even only gray images are allowed, such as electron microscope, which measures the strength of an electron beam at various space points. Indeed, due to the importance of gray images, how to better convert color images to gray ones becomes a hot topic in academic circles [38]–[40]. All in all, RDH in color image with grayscale invariance to avoid interfering with the further image processing and applications is a very meaningful scheme.

In this paper, a novel RDH scheme in color image is proposed. Different from all the previous color image RDH algorithms [23]–[26], the corresponding gray version is kept unchanged by the presented scheme. Based on the unchanged gray version, we present a novel causal predictor and pixel selection strategy to generate PE sequence with a shape histogram for embedding. Messages are recursively embedded into red and blue channels, and then we adjust green channel to eliminate the offsets on grayscales caused by embedding messages, thus grayscale invariance is achieved. To return the adjusted green channel, additional error correcting bits are necessary and regarded as one part of payloads to be recursively embedded, then the reversibility is also guaranteed. The experimental results further show that, the proposed method as a novel RDH scheme for the color image not only holds the gray version, but also well keeps the quality of color marked image. Taking SIFT feature extraction as example, we demonstrate that the further processing is not interfered by the presented method.

The rest of this paper is organized as follows. We briefly introduce the relative works in Section II. Section III elaborates the proposed embedding and extracting algorithms. The experimental results are given in Section IV to show the feasibility of the proposed method, and finally this paper is concluded with a discussion in Section V.

## II. RELATIVE WORKS

Throughout this paper, we denote pure message as  $m^p$ , error correcting message as  $m^e$ , and mathematical function as  $f$  with a subscript. Matrices and vectors are denoted by boldface fonts, and a color pixel  $\{r, g, b\}$  is denoted as a triplet  $\mathbf{T} = \{r, g, b\}$ . The scales at the position  $\{i, j\}$  from channels R, G, B and gray version are denoted as  $r_{i,j}$ ,  $g_{i,j}$ ,  $b_{i,j}$  and  $v_{i,j}$  respectively, and the corresponding PEs from channels R and B are denoted as  $e_{i,j}^R$  and  $e_{i,j}^B$ .

Nearly all of the RDH algorithms consist of two steps. The first step is to generate a host sequence with the small entropy, i.e., a host sequence has a sharp histogram. Then in the second step, the users reversibly embed messages into the host sequence by modifying its histogram with methods like difference expansion and histogram shift.

PEs of image pixels are the most commonly adopted host sequence because their entropy is small. Therefore, many scholars [9]–[15] devote to generating host PEs with sharp histogram by exploring the correlations within neighboring pixels. Because one color image consists of three color channels and the correlations among three color channels are also high, researchers [24]–[26] begin to explore such correlations to generate a much sharper PE histogram and correspondingly reduce the total embedding distortion.

After generating host PEs, we will reversibly modify PEs to accommodate messages. The commonly used embedding methods are difference expansion [7] and histogram shift [8]. Taking difference expansion as example, we embed a binary bit denoted as  $m \in \{0, 1\}$  into the PE  $e$  by expanding it to  $e'$  as

$$e' = 2e + m. \quad (1)$$

Obviously, we can restore  $e$  and  $m$  from  $e'$  by

$$\begin{cases} m = e' \bmod 2 \\ e = \lfloor e'/2 \rfloor \end{cases}, \quad (2)$$

where  $\lfloor x \rfloor$  returns to the nearest integer less than or equal to  $x$ .

Although the existing color image RDH algorithms [23]–[26] well reduce the embedding distortion by exploring correlations within color image, the gray version of host color image is still polluted. As mentioned in the Introduction, the property of grayscale invariance will introduce many conveniences in the further color image applications. Next, we will elaborate how to design color image RDH with grayscale invariance.

## III. PROPOSED SCHEME

### A. Algorithm overview

The fundamental difference between color image and gray image is that, a vector consisting of three components is assigned to a pixel for color image, while a scalar grayscale is assigned to a pixel for gray image. As mentioned above,

there are various of algorithms [38]–[41] for converting color images to gray versions, among which the most widely used one is the following famous formula from [41]:

$$v = \lfloor 0.299r + 0.587g + 0.114b \rfloor, \quad (3)$$

where  $\lfloor x \rfloor$  rounds the element  $x$  to its nearest integer,  $r$ ,  $g$ ,  $b$  are scales from channels R, G, B respectively, and  $v$  is the generated grayscale. For convenience, we define

$$f_v(r, g, b) = \lfloor 0.299r + 0.587g + 0.114b \rfloor. \quad (4)$$

Through  $v = f_v(r, g, b)$  we convert a color pixel  $\{r, g, b\}$  to the gray one  $v$ . Based on the generated gray version, a shaper host histogram can be generated from each color channel by a pixel selection and prediction strategy, and then we modify host histogram to marked histogram to embed messages. In both the embedding processes and the extracting processes the gray version is kept unchanged, thus marked histogram can be regenerated, from which we restore embedded messages and host histogram.

The color pixel  $\{r, g, b\}$  is modified to  $\{r', g', b'\}$  after embedding messages, to achieve the property of grayscale invariance Eq. (5) needs to be held.

$$f_v(r, g, b) = f_v(r', g', b') \quad (5)$$

The main idea to keep Eq. (5) is embedding messages into  $r$  and  $b$ , and then adjusting  $g$ . In detail, after  $\{r, b\}$  is modified to  $\{r', b'\}$ , we generate  $g'$  by solving the linear equation Eq. (3) with  $r'$ ,  $g'$  and  $v$  as the inputs. That is

$$g' = \lfloor (v - 0.299r' - 0.114b') / 0.587 \rfloor. \quad (6)$$

For convenience, we define

$$f_g(r', b', v) = \lfloor (v - 0.299r' - 0.114b') / 0.587 \rfloor. \quad (7)$$

At the receiver's side, we firstly regenerate the gray version from color marked image, based on which marked histogram is regenerated. Then we recover  $\{r, b\}$  from  $\{r', b'\}$  after extracting the embedded messages, with the help of  $r$ ,  $b$ ,  $v$  and one bit error correcting message denoted as  $m^c$ ,  $g$  is restored. Error correcting messages are essential for recovery and need to be recursively embedded, which will be elaborated later. The overview of the proposed algorithm is shown as Fig. 1.

### B. Polynomial predictor based on gray version

As done in most of RDH algorithms, we should firstly generate a sequence having a sharp histogram as the host sequence of RDH. Eq. (3) implies that the gray version has a great correlation with each color channel. By utilizing such correlations, a shaper host histogram can be generated.

In this subsection, based on the unchanged gray version, we propose a novel adaptive causal predictor which can provide satisfactory results for predicting red scales and blue scales. Taking the prediction of red scale  $r_{i,j}$  as example, we predict  $r_{i,j}$  based on the input grayscale  $v_{i,j}$  with a 3th degree polynomial formulated as

$$f_p(v_{i,j}) = a + bv_{i,j} + cv_{i,j}^2. \quad (8)$$

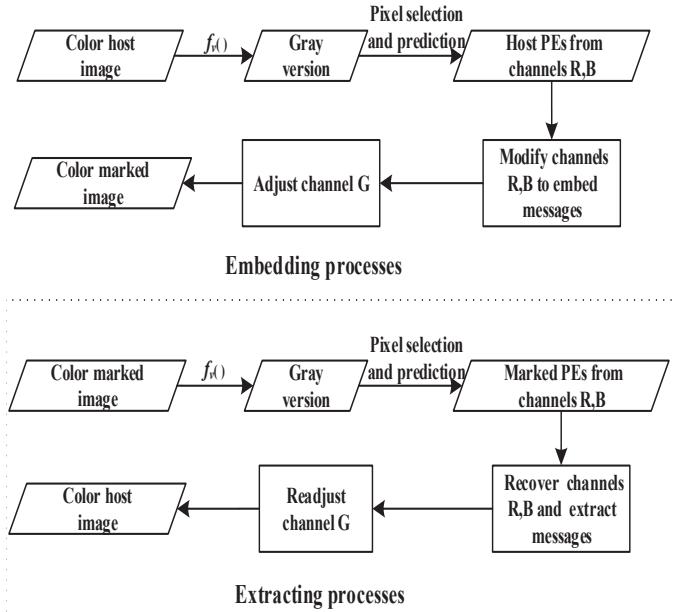


Fig. 1: Overview of the proposed algorithm.

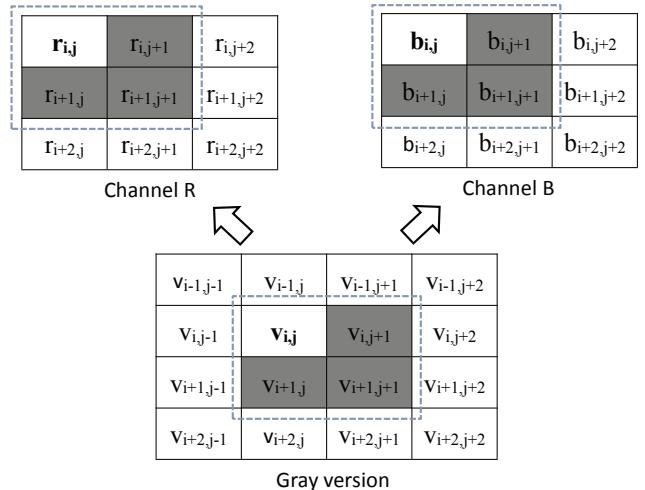


Fig. 2: Predicting  $r_{i,j}$  or  $b_{i,j}$  based on  $v_{i,j}$  by exploring correlations within  $\{v_{i+1,j}, v_{i,j+1}, v_{i+1,j+1}\}$  and  $\{r_{i+1,j}, r_{i,j+1}, r_{i+1,j+1}\}$  or  $\{b_{i+1,j}, b_{i,j+1}, b_{i+1,j+1}\}$ .

To provide a good prediction for  $r_{i,j}$  based on  $v_{i,j}$ , the correlations within their nearby scales are explored. We search the optimal polynomial coefficients which minimize the fitting errors with  $\{v_{i+1,j}, v_{i,j+1}, v_{i+1,j+1}\}$  as inputs and  $\{r_{i+1,j}, r_{i,j+1}, r_{i+1,j+1}\}$  as outputs. That is

$$\beta^* = \arg \min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2, \quad (9)$$

where  $\mathbf{y} = [r_{i+1,j} \ r_{i,j+1} \ r_{i+1,j+1}]^T$ ,  $\mathbf{X} = \begin{bmatrix} 1 & v_{i+1,j} & v_{i+1,j}^2 \\ 1 & v_{i,j+1} & v_{i,j+1}^2 \\ 1 & v_{i+1,j+1} & v_{i+1,j+1}^2 \end{bmatrix}$

and  $\beta = [a \ b \ c]^T$ .

By taking the partial derivatives of Eq. (9) with respect to the polynomial coefficients  $\beta$  and setting them equal to zero we get

$$\beta^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (10)$$

We firstly initialize the prediction of  $r_{i,j}$  as  $f_p(v_{i,j})$  after finding the optimal polynomial coefficients  $\beta^*$ . However, there may exist some cases that  $r_{i,j}$  cannot be well predicted by the polynomial prediction because of over fitting or under fitting. To solve this problem, we also utilize the correlations among red scales nearing  $r_{i,j}$  to restrict such prediction. Inspired by median-edge detector (MED) [42], the ultimate prediction of  $r_{i,j}$  denoted by  $\hat{r}_{i,j}$  is limited in the range of  $r_{i+1,j}$  and  $r_{i,j+1}$ . That is

$$\hat{r}_{i,j} = \begin{cases} \min(r_{i+1,j}, r_{i,j+1}), & \text{if } f_p(v_{i,j}) \leq \min(r_{i+1,j}, r_{i,j+1}) \\ \max(r_{i+1,j}, r_{i,j+1}), & \text{if } f_p(v_{i,j}) \geq \max(r_{i+1,j}, r_{i,j+1}) \\ f_p(v_{i,j}), & \text{otherwise} \end{cases}. \quad (11)$$

By subtracting  $\hat{r}_{i,j}$  from the original scale  $r_{i,j}$ , we obtain its PE  $e_{i,j}^R$  as

$$e_{i,j}^R = r_{i,j} - \hat{r}_{i,j}. \quad (12)$$



Fig. 3: Tested color images of size 510×510. (a) Airplane. (b) Barbara. (c) Lena. (d) Baboon.

Empirically, we adopt 3th polynomial predictor and select the nearest 3 pairs of scales labelled in dark in Fig. 2 for predicting  $r_{i,j}$ . Obviously, the adopted 3th polynomial predictor can be easily extended to the higher order. Similar with MED, the presented predictor moves prediction window in the raster order, i.e., from left to right and from top to bottom, which is also the half-enclosing causal one. The receiver will inversely scan and process scales to get the same prediction values in the same way, so there is no need to record these optimal polynomial coefficients. By the same prediction method, the PEs of blue scales can be generated.

To show that the presented polynomial predictor based on the unchanged gray version can generate shaper host histograms, we compare it with the classic casual MED predictor [42] operating on single channel. The average PEs from airplane, barbara, lena and baboon are shown in Fig. 4, from which we see that the presented predictor utilizing the gray version outperforms MED a lot. The reason is that the unchanged gray version consists of three color channels and there exists high correlations among them.

### C. Select host triplets for modification based on gray version

As for each triplet  $\mathbf{T}_{i,j} = \{r_{i,j}, g_{i,j}, b_{i,j}\}$ , we get the PEs  $\{e_{i,j}^R, e_{i,j}^B\}$  and expand them to  $\{e'_{i,j}^R, e'_{i,j}^B\}$  by Eq. (1) with carrying 2 bits. Accordingly,  $\{r_{i,j}, b_{i,j}\}$  are modified to  $\{r'_{i,j}, b'_{i,j}\}$  as

$$\begin{cases} r'_{i,j} = \hat{r}_{i,j} + e'_{i,j}^R \\ b'_{i,j} = \hat{b}_{i,j} + e'_{i,j}^B \end{cases}. \quad (13)$$

Finally, we adjust  $g_{i,j}$  to  $g'_{i,j}$  with  $g'_{i,j} = f_g(r'_{i,j}, b'_{i,j}, v_{i,j})$  to remove the offset from  $v_{i,j}$ .

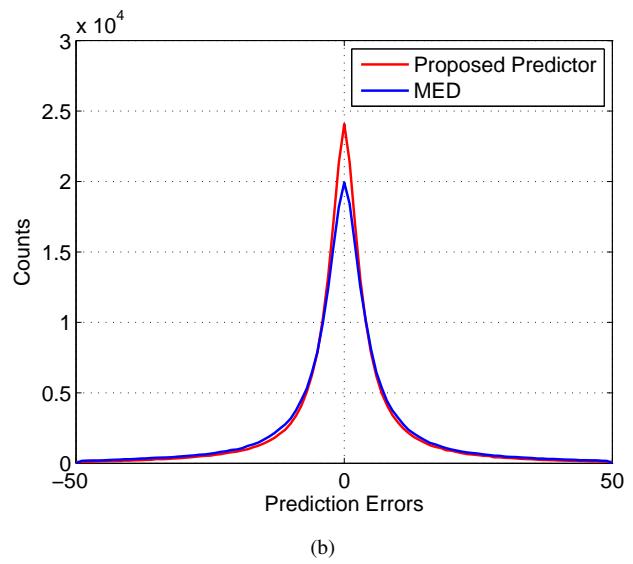
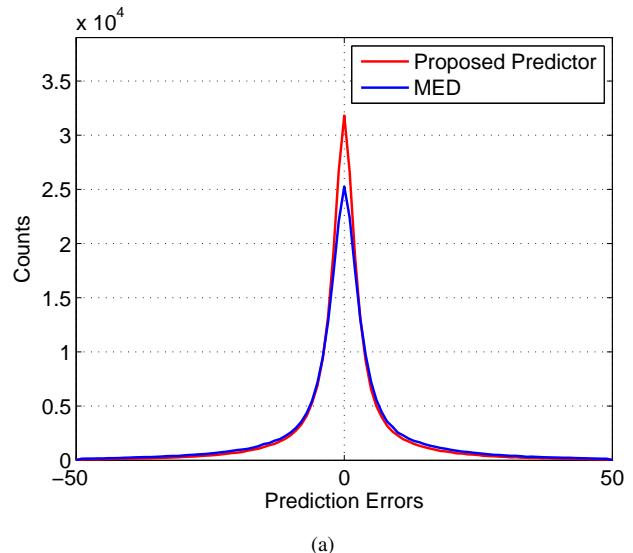


Fig. 4: (a) Average PEs from channel R of Fig. 3. (b) Average PEs from channel B of Fig. 3.

The recipient will recalculate  $\{\hat{r}_{i,j}, \hat{b}_{i,j}\}$  by the same polynomial predictor, and thus regain  $\{e'_{i,j}^R, e'_{i,j}^B\}$  by subtracting the predicted scales  $\{\hat{r}_{i,j}, \hat{b}_{i,j}\}$  from the marked scales  $\{r'_{i,j}, b'_{i,j}\}$ . We will extract the embedded two bits and restore  $\{e_{i,j}^R, e_{i,j}^B\}$  from  $\{e'_{i,j}^R, e'_{i,j}^B\}$  with Eq. (2), and then restore  $\{r_{i,j}, b_{i,j}\}$  as

$$\begin{cases} r_{i,j} = \hat{r}_{i,j} + e_{i,j}^R \\ b_{i,j} = \hat{b}_{i,j} + e_{i,j}^B \end{cases}. \quad (14)$$

After restoring  $\{r_{i,j}, b_{i,j}\}$ ,  $g_{i,j}$  can be roughly calculated with

$$g_{i,j}^c = f_g(r_{i,j}, b_{i,j}, v_{i,j}). \quad (15)$$

Note that the calculated result denoted as  $g_{i,j}^c$  may not equal to  $g_{i,j}$  due to the round function, but the magnitude of their offset is 1 or 0, because the weight of color channel G is more than 0.5. That is also the reason why we adjust channel G to hold the gray version. We need an error correcting bit  $m_{i,j}^c$  to

record such offset, where

$$m_{i,j}^c = |g_{i,j} - f_g(r_{i,j}, b_{i,j}, v_{i,j})|, \quad (16)$$

which needs to be recursively embedded into host triples.

To reduce the distortion, we usually select the PEs with the smaller magnitudes for modification firstly. As we all know, the magnitude of each PE is proportional to its local variance, and a small variance usually indicates a small magnitude of PE. In order to decrease the image distortion, we prefer to embed messages into triplets with the relatively smaller local variances. To better represent the local variance of  $\mathbf{T}_{i,j}$  denoted by  $\rho_{i,j}$ , we calculate  $\rho_{i,j}$  based on the gray version (see Fig. 2) as

$$\rho_{i,j} = \frac{\Delta v_{i-1,j}^2 + \Delta v_{i,j-1}^2 + \Delta v_{i,j}^2 + \Delta v_{i+1,j}^2 + \Delta v_{i,j+1}^2}{4}, \quad (17)$$

where  $\Delta v_{i-1,j} = v_{i-1,j} - v_{i,j}$ ,  $\Delta v_{i,j-1} = v_{i,j-1} - v_{i,j}$ ,  $\Delta v_{i,j} = v_{i,j} - v_{i,j}$ ,  $\Delta v_{i+1,j} = v_{i+1,j} - v_{i,j}$ ,  $\Delta v_{i,j+1} = v_{i,j+1} - v_{i,j}$ , and  $v_{i,j} = (v_{i-1,j} + v_{i,j-1} + v_{i,j} + v_{i+1,j} + v_{i,j+1})/5$ . Of course, at the receiver's side we can recalculate these local variances based on the unchanged gray version.

Some selected triplets may yield serious distortion by the embedding method Eq. (1) because of the large magnitudes of PEs. Traditional RDH algorithms usually group PEs into the inner region and the outer region by a threshold, and the PEs in outer region are shifted without carrying messages to avoid causing large distortion. Different from the traditional RDH algorithms, it will be rather wasteful to shift the PEs in outer region for the presented novel scheme. Because as long as one triplet is modified, not only distortion is caused, but also it will yield one bit error correcting message for recovery, even such triplet does not accommodate any pure messages. Therefore, we need to skip these triplets yielding large distortion.

The distortion caused by modifying  $\mathbf{T}_{i,j} = \{r_{i,j}, g_{i,j}, b_{i,j}\}$  to  $\mathbf{T}'_{i,j} = \{r'_{i,j}, g'_{i,j}, b'_{i,j}\}$  is

$$D_{i,j} = \sqrt{(r_{i,j} - r'_{i,j})^2 + (g_{i,j} - g'_{i,j})^2 + (b_{i,j} - b'_{i,j})^2}. \quad (18)$$

We give a threshold denoted by  $D_T$  to control the distortion. In detail, we modify  $\mathbf{T}_{i,j}$  to  $\mathbf{T}'_{i,j}$  by embedding one bit pure message and one bit error correcting message. If the distortion  $D_{i,j}$  between  $\mathbf{T}_{i,j}$  and  $\mathbf{T}'_{i,j}$  is larger than  $D_T$ , we just skip this triplet for modification. Clearly that, the number of skipped triplets will decrease with the increase of  $D_T$ . Note that, usually  $D_T$  is initialized as 20 empirically. As for some images with rich textures, we can enlarge  $D_T$  to expand the number of legal triplets. After embedding messages, some triplets perhaps will yield overflow/underflow scales which are less than 0 or more than 255, and these triplets are also skipped for modification. That is, as long as current triplet meets one of the following two illegal conditions, we label such illegal triplet with “tag = 0”; otherwise, current triplet is legal and labelled with “tag = 1”. These tags are recorded as auxiliary parameters, which are easy to be compressed due to that the number of skipped illegal triplets is usually fewer.

### Two illegal conditions:

- $D_{i,j} > D_T$ ;
- Overflow/underflow scale exists.

### D. Recursive embedding algorithm

We reserve some triplets for recording auxiliary parameters whose content will be elaborated at the end of this subsection. These parameters are embedded into the LSBs (least significant bits) of blue scales of reserved triplets, and these substituted LSBs will be regarded as a part of pure payloads to be embedded. Note that, the reserved triplets cannot be selected arbitrarily, because LSB substitutions of blue scales perhaps will also modify the corresponding grayscales. Our objective is holding the gray version completely, thus we must reserve the triplets which can hold the corresponding grayscales after LSB substitutions of their blue scales. The triplet with ability resisting disturbance is called invariant triplet, and how to select these invariant triplets is the first step we need to do.

Indeed, the modified magnitude by LSB substitution is 1 or 0. As long as Eq. (19) is held by one triplet, after LSB substitution of its blue scale Eq. (19) still holds, and then such triplet is invariant. As for an invariant triplet, it is still invariant at the receiver's side. Therefore, if we reserve some invariant triplets in a fixed region such as the corner or the edges of image, the receiver will relocate these reserved invariant triplets to extract auxiliary parameters. The reason why we select blue scales for LSB substitution is that the weight of blue channel is the least and so the ratio of invariant triplet is the maximum.

$$f_v(r, g, 2\lfloor \frac{b}{2} \rfloor) = f_v(r, g, 2\lfloor \frac{b}{2} \rfloor + 1) \quad (19)$$

Assume the messages to be embedded are  $\{m_1, m_2, \dots, m_L\}$ , and the number of reserved invariant triplets in a fixed region is  $L_r$ . The local variance  $\rho$  of each triplet apart from these border triplets and reserved invariant triplets is calculated with Eq. (17) firstly, and then we select a threshold denoted as  $\rho_T$  to determine the positions of selected host triplets.  $\rho_T$  is initialized as  $\rho_0$ , where  $\rho_0$  is the minimum integer value with  $L + L_r$  triplets' local variances less than it. If the triplets with local variances less than  $\rho_T$  cannot provide enough capacity, then we increase  $\rho_T$  by 1. And so on, until selecting  $N$  host triplets, which can include  $L + L_r$  legal triplets and some skipped illegal triplets.

After  $\rho_T$  is determined, we scan host image in the raster order starting from the upper left corner, and collect  $N$  triplets  $\{\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_N\}$  with variances less than  $\rho_T$ . The prediction window with  $2 \times 2$  size is also moved in the raster order from the upper left corner, and PEs of red scales and blue scales of collected triplets are calculated with the proposed causal polynomial predictor.

As shown in Fig. 5, for each host triplet  $\mathbf{T}_i = \{r_i, g_i, b_i\}$ , with the method of Eq. (1) we embed one bit pure message denoted as  $m_i^p$  into  $r_i$  by expanding its PE  $e_i^R$  to  $e'_i^R$ . At the same time, one error correcting bit denoted as  $m_{i-1}^c$  is embedded into  $b_i$  by expanding  $e_i^B$  to  $e'_i^B$ , where  $m_{i-1}^c$  is utilized for recovering  $g_{i-1}$ . After embedding  $m_i^p$  and  $m_{i-1}^c$  we modify  $\{r_i, b_i\}$  to  $\{r'_i, b'_i\}$ . To hold the corresponding grayscale  $v_i$ , we generate  $g'_i$  with  $g'_i = f_g(r'_i, b'_i, v_i)$ . Then  $g_i$  is modified to  $g'_i$ , and  $\mathbf{T}_i = \{r_i, g_i, b_i\}$  becomes  $\mathbf{T}'_i = \{r'_i, g'_i, b'_i\}$ .

Next, we will check the validity of  $\mathbf{T}'_i$ . If  $\mathbf{T}'_i$  is illegal, then we label it with  $tag_i = 0$  and skip it for modification. Correspondingly,  $m_i^c = m_{i-1}^c$  and  $\mathbf{T}'_i = \mathbf{T}_i$ ; If  $\mathbf{T}_i$  is legal, one error correcting bit  $m_i^c = |g_i - f_g(r_i, b_i, v_i)|$  together with  $m_{i+1}^p$  are embedded into  $\mathbf{T}_{i+1}$ . That is we embed  $m_{i-1}^c$  and  $m_i^p$  into  $\mathbf{T}_i$ , and generate  $\mathbf{T}'_i$ ,  $m_i^c$  and  $tag_i$ , which is formulated as

$$(\mathbf{T}'_i, m_i^c, tag_i) = f_{emb}(\mathbf{T}_i, m_i^p, m_{i-1}^c), i = 1, 2, \dots, N. \quad (20)$$

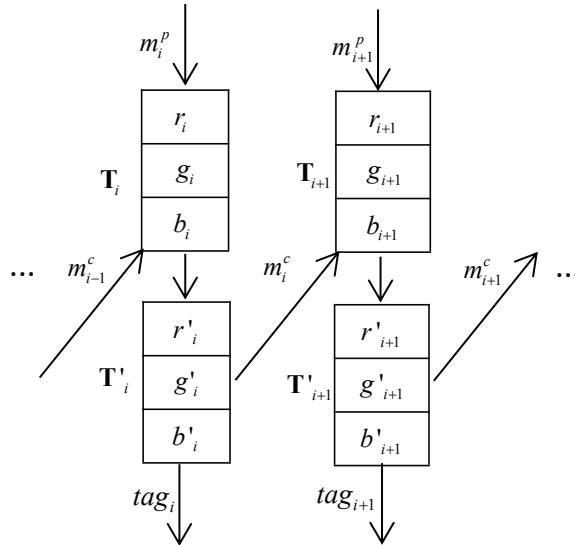
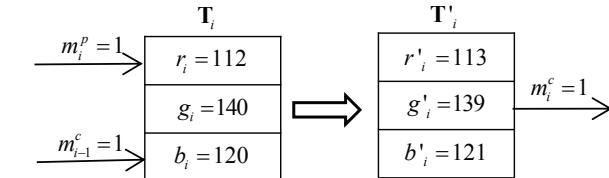


Fig. 5: Recursive embedding processes.

$$v_i = 129 = f_v(112, 140, 120) = f_v(113, 139, 121)$$



$$\begin{cases} e_i^R = 1 \xrightarrow{m_i^p=0} e'^R = 2 \\ e_i^B = 0 \xrightarrow{m_{i-1}^c=1} e'^B = 1 \end{cases} \Rightarrow \begin{cases} r_i = 112 \xrightarrow{m_i^p=0} r'_i = 113 \\ b_i = 120 \xrightarrow{m_{i-1}^c=1} b'_i = 121 \end{cases}$$

$$\begin{cases} g'_i = 139 = f_g(113, 121, 129) \\ m_i^c = 1 = |140 - f_g(112, 120, 129)| \end{cases}$$

Fig. 6: Example of the proposed embedding algorithm.

We give a simple example depicted in Fig. 6 to show the embedding processes of one legal triplet. We expand  $e_i^R = 1$  to  $e'^R = 2$  by embedding  $m_i^p = 0$ , and  $e_i^B = 0$  to  $e'^B = 1$  by embedding  $m_{i-1}^c = 1$ , then  $r'_i = 113$ ,  $b'_i = 121$ . To keep  $v_i = 129$  unchanged,  $g_i$  is adjusted to  $g'_i = 139$ , thus  $\mathbf{T}'_i = \{113, 139, 121\}$ . The error correcting bit  $m_i^c = 1$  for restoring  $g_i$  is calculated with  $m_i^c = |140 - f_g(112, 120, 129)|$ .

The receiver needs some auxiliary parameters to inversely decode marked triplets to restore host triplets and embedded messages. The auxiliary parameters include the number of

collected triplets  $N$ , the threshold of local variance  $\rho_T$ , triplets  $tags$  and the error correcting bit of the last triplet  $m_N^c$ , which are compressed and embedded into the LSBs of blue scales of reserved invariant triplets.

### E. Recursive extracting algorithm

At the decoder's side, we firstly relocate invariant triplets with Eq. (19) in a fixed region. Then the parameters, including  $N$ ,  $\rho_T$ ,  $tags$  and  $m_N^c$  are extracted, from the blue scales of reserved invariant triplets. The local variance  $\rho$  of each triplet apart from these border triplets and reserved invariant triplets is calculated with Eq. (17), and then according to  $N$  and  $\rho_T$ , we scan the image in the raster order to collect  $N$  marked triplets  $\{\mathbf{T}'_1, \mathbf{T}'_2, \dots, \mathbf{T}'_N\}$ .

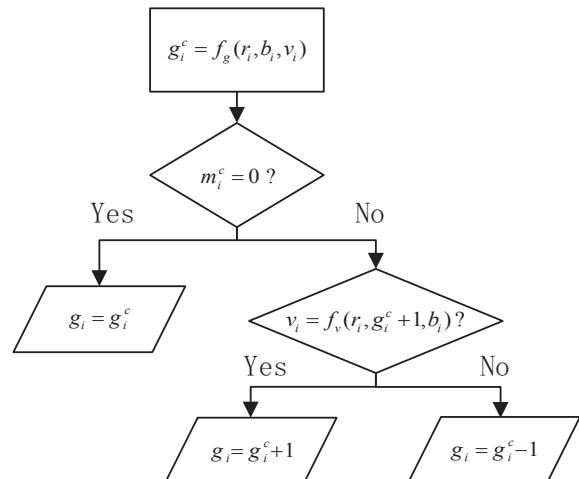
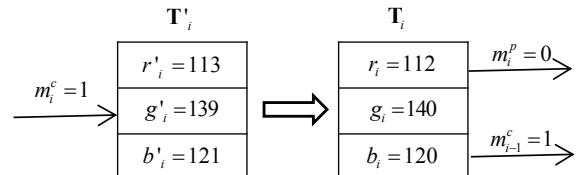


Fig. 7: The processes of recovering green scale.

$$v_i = 129 = f_v(112, 140, 120) = f_v(113, 139, 121)$$



$$\begin{cases} e_i^R = 2 \xrightarrow{(e_i^R=1, m_i^p=0)} e'^R = 1 \\ e_i^B = 0 \xrightarrow{(e_i^B=0, m_{i-1}^c=1)} e'^B = 0 \end{cases} \Rightarrow \begin{cases} r'_i = 113 \xrightarrow{} r_i = 112 \\ b'_i = 121 \xrightarrow{} b_i = 120 \end{cases}$$

$$g_i^c = 139 = f_g(112, 120, 129)$$

$$\left( m_i^c = 1, \begin{cases} f_v(112, 140, 120) = 129 \\ f_v(112, 138, 120) = 128 \end{cases} \right) \Rightarrow g_i = 140$$

Fig. 8: Example of the proposed extracting algorithm.

Beginning with  $\mathbf{T}'_N$  we decode  $\{\mathbf{T}'_1, \mathbf{T}'_2, \dots, \mathbf{T}'_N\}$  in the inverse order. The prediction window with  $2 \times 2$  size is also moved in the inverse raster order, and we get marked PEs of red scale

and blue scale of  $\mathbf{T}'_N$  with the proposed causal polynomial predictor firstly. After restoring  $\mathbf{T}_N$  we further calculate marked PEs of red scale and blue scale of  $\mathbf{T}'_{N-1}$ , and so on.

According to  $tag_i$  we get whether  $\mathbf{T}'_i$  is legal or not. If  $\mathbf{T}'_i$  is illegal, then  $m_{i-1}^c = m_i^c$ ,  $\mathbf{T}_i = \mathbf{T}'_i$  and  $m_i^p$  is empty. For legal  $\mathbf{T}'_i$ ,  $\{e_i^R, e_i^B\}$  are calculated from  $\{r'_i, b'_i\}$  by the proposed polynomial predictor, from which  $m_i^p$ ,  $m_{i-1}^c$  and  $\{e_i^R, e_i^B\}$  are restored with Eq. (2). Further,  $r_i$  and  $b_i$  can be restored. Since  $v_i = f_v(r'_i, g'_i, b'_i)$ , we will restore  $g_i$  from the calculated  $g_i^c = f_g(r_i, b_i, v_i)$  (Eq. 6) according to  $m_i^c$ . To be detailed, if  $m_i^c = 0$ , then  $g_i = g_i^c = f_g(r_i, b_i, v_i)$ ; Otherwise, we get that the magnitude of the offset between  $g_i$  and  $g_i^c$  is 1, that is  $g_i = g_i^c + 1$  or  $g_i = g_i^c - 1$ . Since the weight of  $g_i$  to generate  $v_i$  is 0.587, which is larger than 0.5, the wrong one must will modify  $v_i$  through  $f_v(\cdot)$  while the right one will not, i.e.,  $f_v(r_i, g_i^c + 1, b_i) \neq v_i$  or  $f_v(r_i, g_i^c - 1, b_i) \neq v_i$ . Therefore, it is easy to eliminate the wrong one and get the right one. The processes to get  $g_i$  according to  $m_i^c$  are described in Fig 7. From each marked triplet  $\mathbf{T}'_i$ , with the help of  $m_i^c$  and  $tag_i$  we restore  $m_{i-1}^c$ ,  $m_i^p$  and  $\mathbf{T}_i$ , that is

$$(\mathbf{T}_i, m_i^p, m_{i-1}^c) = f_{ext}(\mathbf{T}'_i, m_i^c, tag_i), i = N, N-1, \dots, 1. \quad (21)$$

As depicted in Fig. 8, we continue the above example to show how  $\mathbf{T}'_i = \{113, 139, 121\}$  is decoded with the help of the error correcting bit  $m_i^c = 1$ . We restore  $\{e_i^R = 1, e_i^B = 0\}$  from  $\{e_i^R = 2, e_i^B = 1\}$  after extracting  $m_i^p = 0$  and  $m_{i-1}^c = 1$ , and then further restore  $\{r_i = 112, b_i = 120\}$ .  $v_i = 129$  can be got with  $v_i = f_v(113, 139, 121)$ , thus we get  $g_i^c = 139 = f_g(112, 120, 129)$ . Since  $m_i^c = 1$ , and  $f_v(112, 139 + 1, 120) = 129$  while  $f_v(112, 139 - 1, 120) = 128 \neq 129$ , we get  $g_i = 140$ . Therefore,  $\mathbf{T}_i = \{112, 140, 120\}$ ,  $m_i^p = 0$  and  $m_{i-1}^c = 1$  are all restored.

After extracting the embedded messages and restoring the host triplets, we further reconstruct the blue scales of reserved invariant triplets with the extracted LSBs.

#### F. Set parameters

There are some auxiliary messages for restoring host image and embedded messages from marked image. Invariant triplets are located and reserved to record auxiliary messages by LSB substitution, and the substituted LSBs will be embedded as one part of pure payloads. The number of reserved invariant triplets  $L_r$  needs to be larger than the length of auxiliary messages denoted by  $L_a$ . Indeed, triplet tags account for the vast majority of auxiliary messages. Apart from triplet tags, 50 bits messages are used to record the rest of auxiliary parameters, including the number of collected triplets  $N$ , the threshold of local variance  $\rho_T$  and the error correcting bit of the last triplet  $m_N^c$ . In this subsection, we discuss how to properly select the parameters  $\rho_T$ ,  $L_r$ , and the threshold of embedding distortion (see Eq. (18))  $D_T$ .

Empirically,  $D_T$  is initialized as 20, and  $L_r$  is initialized as  $0.02L$ , where  $L$  is the length of given embedded messages. For some cases,  $0.02L$  bits may not accommodate the auxiliary messages. If that happens we firstly enlarge  $D_T$  to reduce the number of illegal triplets, thus making triplet tags to be compressed more efficiently and reducing the length of

auxiliary messages. We cannot enlarge  $D_T$  without limits, once  $D_T \geq 50$ , we keep  $D_T$  unchanged, and enlarge the number of reserved triplets  $L_r$ . Note that, there is no need to record  $D_T$ , because the receiver will relocate legal triplets according to triplet tags, the number of collected triplets  $N$  and the threshold of local variance  $\rho_T$ . We initialize the threshold of local variance  $\rho_T$  as  $\rho_0$ , which is the minimum integer value with  $L + L_r$  triplets' local variances less than it. If the triplets with local variances less than  $\rho_T$  cannot provide  $L + L_r$  legal triplets, we increase  $\rho_T$  by 1. And so on, until selecting  $N$  host triplets, which include  $L + L_r$  legal triplets and  $N - L - L_r$  skipped illegal triplets. The pseudo codes of selecting  $L_r$ ,  $\rho_T$  and  $D_T$  are shown as Algorithm 1.

---

#### Algorithm 1 Pseudo code of selecting $L_r$ , $\rho_T$ and $D_T$

---

```

1:  $L_r = 0.02L$ ,  $D_T = 20$ ,  $flag_1 = true$  and  $flag_2 = true$ ;
2: Initialize  $\rho_T = \rho_0$ ;
3: while  $flag_1$  do
4:   while  $flag_2$  do
5:     Locate the triplets whose local variances are less
       than  $\rho_T$ ;
6:     Embed  $L + L_r$  bits messages into the located triplets
       in the raster order;
7:     if Legal triplets are enough then
8:        $flag_2 = false$ ;
9:     else
10:       $\rho_T = \rho_T + 1$ ;
11:    end if
12:  end while
13:  Compress auxiliary parameters and generate  $L_a$  bits
       auxiliary messages;
14:  if  $L_a \leq L_r$  then
15:     $flag_1 = false$ ;
16:  else
17:    if  $D_T < 50$  then
18:       $D_T = D_T + 5$ ;
19:    else
20:       $L_r = L_r + 0.01L$ ;
21:    end if
22:  end if
23: end while
24: Record the number of scanned triplets  $N$ , including legal
       triplets and illegal triplets;
25: return  $L_r$ ,  $\rho_T$  and  $N$ ;

```

---

## IV. EXPERIMENTAL RESULTS

### A. Payload-distortion performance

Firstly, we need to determine the threshold of embedding distortion (see Eq. (18))  $D_T$ , the threshold of local variance  $\rho_T$  and the number of reserved invariant triplets  $L_r$  with Algorithm 1. After  $D_T$ ,  $\rho_T$  and  $L_r$  are determined, the length of auxiliary messages  $L_a$  and the number of scanned triplets  $N$  will be also determined. With typical images (Fig. 3) as test images, the selected  $D_T$ ,  $\rho_T$ ,  $L_r$  and the corresponding generated  $L_a$ ,  $N$  under different payloads are shown in TABLE I, from which we see that for general images like airplane and barbara,

setting  $L_r = 0.02L$  and  $D_T = 20$  can usually accommodate auxiliary messages. For lena, we can reduce the length of auxiliary messages through enlarging  $D_T$ . But for some images like baboon, due to its rich textures, enlarging  $D_T$  cannot reduce the length of auxiliary messages efficiently. In such case, the step of enlarging the number of reserved invariant triplets is executed.

The presented RDH algorithm is the first one for color image with grayscale invariance. A good algorithm should not only hold the gray version, but also keep the quality of color marked image as far as possible. Therefore, one key assessment indicator to evaluate such novel RDH scheme is the visual quality of color marked image, and peak signal to noise ratio (PSNR) is the generally adopted assessment indicator. PSNR is defined as

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right), \quad (22)$$

and MSE is computed by

$$MSE = \frac{1}{M \times N \times C} \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^N (s(i, j, k) - s'(i, j, k))^2, \quad (23)$$

where  $M$  and  $N$  are the width and the height of host image,  $C$  means the number of color channel which will be 3 for color image and 1 for gray image,  $s(i, j, k)$  represents the scale value at the position  $(i, j)$  in the  $k$ th color channel of host image and  $s'(i, j, k)$  is the scale value after modification.

The payload-distortion performances of the proposed method with Fig. 3 as test images are shown in Fig. 9. Of course, all the generated color marked images will have the same gray versions as original color host images. Fig. 9 shows that by the proposed method not only the gray versions are unchanged, but also the quality of color marked images are well kept. From Fig. 9 we also see that baboon's performance is relatively poorer than the others'. The reasons are mainly about two aspects: one is that the PE histograms generated from baboon are not sharp enough due to its rich texture, and the other is that the number of illegal triplets are rather large, which causes much more auxiliary messages for embedding (see TABLE I).

By the proposed method, one triplet accommodates one bit pure message but also will yield one bit error correcting message. Obviously, the large number of error correcting bits limit the embedding capacity. However, in this paper the error correcting messages are difficult to be compressed. For example, the ratios of "0" in the binary error correcting messages for Airplane, Barbara, Lena and Baboon are all about 0.587. That is because the error correcting messages imply whether green scales are completely restored or not and the weight of green channel is 0.587. Of course, compressing the error correcting messages is a potential solution to enlarge the embedding capacity in the future.

The design of invariant triplets is important for grayscale invariance. We embed auxiliary parameters into blue scales of invariant triplets by LSB substitution. Since the weight of blue channel is 0.114, the ratio of invariant triplets is about 0.886. Therefore, we think the number of invariant triplets is enough to accommodate these auxiliary parameters.

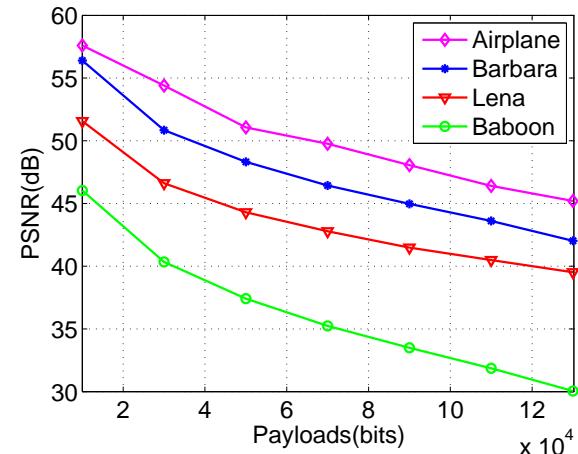


Fig. 9: Payload-distortion performances on color image of the proposed algorithm.

### B. Application example: SIFT feature extraction

As mentioned in Introduction, there are various of applications for the gray versions of color images. In this subsection, we just take SIFT feature extraction as example to show that how previous RDH schemes in color image [23]–[26] interfere with the feature extraction while the presented method does not. SIFT descriptors are widely applied to image compression [30]–[32], pattern recognition [33], [34], image stitching [35], [36], and so on. To the best of our knowledge, almost all of SIFT or SIFT-extended descriptors and their applications for color images only operate on the corresponding gray versions.

The previous RDH schemes in color image [23]–[26] mainly focus on minimizing the total distortion for the yielded color marked image, among which Ou *et al.*'s method [26] is the state-of-the-art one. By Ou *et al.*'s method the total payload is adaptively allocated to three color channels according to their PEs so that well keeping image quality. However, as shown in Fig. 10, by Ou *et al.*'s method the gray versions of color marked images will be polluted. Naturally, after modifications the earlier RDH schemes [23]–[25] will also change the gray versions greatly. Once the corresponding gray versions of color images are polluted, there is no doubt that the further applications based on gray versions will be disturbed.

To extract SIFT features from color image, we usually convert the color image to the gray version, and then extract features from the converted gray one. The first step for SIFT feature extraction is keypoint location. Response is an indicator of how good a point is, and a bigger response means a stronger SIFT point. For lena image, we locate the top eight points with the strongest responses as signed in Fig. 11. Since the presented method will hold the corresponding gray version, SIFT features from the generated marked image will be the same as those from the host image. However, by Ou *et al.*'s method [26], SIFT features from the generated marked image will be modified due to the modification on the corresponding grayscales. For example, after embedding 140000 bits messages, 200000 bits messages and 260000 bits messages into lena by Ou *et al.*'s method respectively,

TABLE I  
PARAMETERS UNDER DIFFERENT PAYLOADS.

Payloads (bits)	10000	30000	50000	70000	90000	110000	130000
airplane	$D_T =$ $\rho_T =$ $N =$ $L_r =$ $L_a =$	20 1 10199 200 51	20 1 30599 600 51	20 2 50999 1000 51	20 2 71399 1400 51	20 3 91800 1800 73	20 5 112201 2200 89
							7 7 132608 2600 190
barbara	$D_T =$ $\rho_T =$ $N =$ $L_r =$ $L_a =$	20 1 10203 200 101	20 3 30616 600 258	20 4 51033 1000 457	20 6 71446 1400 614	20 9 91886 1800 1049	25 15 112348 2200 1688
							30 30 132775 2600 1985
lena	$D_T =$ $\rho_T =$ $N =$ $L_r =$ $L_a =$	40 2 10212 200 193	25 3 30630 600 403	25 4 51039 1000 520	25 5 71451 1400 661	25 7 91866 1800 844	25 9 112302 2200 1236
							13 13 132754 2600 1783
baboon	$D_T =$ $\rho_T =$ $N =$ $L_r =$ $L_a =$	50 9 12043 1800 1770	50 21 34402 3900 3836	50 36 56199 5500 5482	50 59 78602 7700 7163	50 94 101123 9900 9598	50 160 126164 14300 14058
							313 313 159783 26000 25877

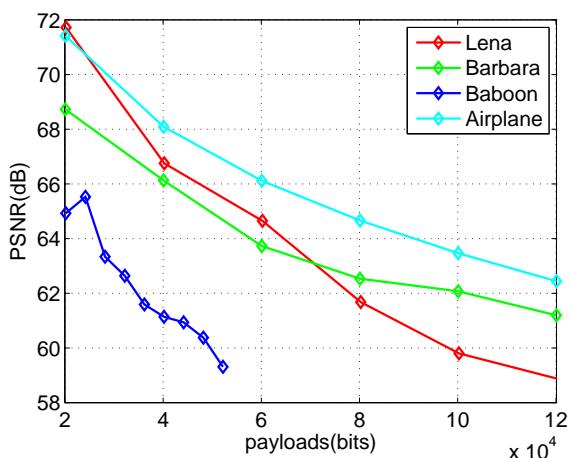


Fig. 10: Payload-distortion performances on gray versions of color marked images generated from Ou *et al.*'s method.

the Euclidean distances between SIFT features from marked image and host image at the same signed positions of Fig. 11 are shown in TABLE II.

Once the extracted SIFT features are modified, the further SIFT applications will be interfered. Continuing with the example of lena image, SIFT points with responses larger than 0.5 in both host image and yielded marked image are located, and these located SIFT points in host image and marked image are matched. The results of SIFT matching for Ou *et al.*'s method [26] and the proposed method are shown in Fig. 12. Note that, the horizontal line between two SIFT points means such two SIFT points are matched correctly, while the oblique line means a false match between two SIFT points. From Fig. 12(a)-(c) we can see that the number of false matches will increase with the increase of embedded messages. However, as long as the given payload is allowed no matter how many



Fig. 11: The top eight SIFT points with the strongest responses.

messages are embedded into the color host image by the presented method, the matching results will be the same as Fig. 12(d), which carrying 260000 bits messages. That is because Ou *et al.*'s method will destroy both color host images and corresponding gray versions more seriously with the increase of embedded messages, and thus correspondingly interfere with SIFT feature extraction more seriously, but the proposed method will keep the gray versions unchanged throughout. Therefore, a novel RDH scheme in color image not interfering with its further image processing is achieved by the presented method.

TABLE II  
EUCLIDEAN DISTANCES OF SIFT FEATURES FROM HOST IMAGE AND MARKED IMAGE.

keypoints (in scan order)	1	2	3	4	5	6	7	8
After embedding 140000 bits	4.795832	2.449490	2.236068	3.162278	7.745967	1.732051	3.162278	5.916080
After embedding 200000 bits	5.916080	3.000000	2.000000	3.316625	4.690416	1.414214	4.123106	3.741657
After embedding 260000 bits	18.681541	3.741657	2.236068	7.483315	7.000000	2.000000	5.916080	4.472136



Fig. 12: (a) SIFT matching after embedding 140000 bits messages by Ou *et al.*'s method. (b) SIFT matching after embedding 200000 bits messages by Ou *et al.*'s method. (c) SIFT matching after embedding 260000 bits messages by Ou *et al.*'s method. (d) SIFT matching after embedding 260000 bits messages by the presented method.

## V. CONCLUSION

Different from all the previous RDH algorithms, we propose a completely novel one for color image, which can hold the corresponding gray version of color host image. Grayscale invariance for RDH in color image is valuable in practice, because it does not affect the further applications and image processing of the yielded marked image in terms of many cases.

In the proposed method, the large number of error correcting bits limit the embedding capacity, and thus compressing the error correcting messages is a potential solution to improve the proposed scheme in the future. On the other hand, beside the famous formula Eq. (3), there are still many other methods [38]–[40] for converting color images to the gray ones. When adopting these advanced converting algorithms, how to hold corresponding gray version for the color image after RDH is a more challenging but interesting problem.

## VI. ACKNOWLEDGEMENTS

The authors thank the anonymous reviewers for their valuable comments and thank Ou *et al.* for offering the source

code in the paper [26]. To help readers to use the proposed method, we will post the Matlab implementation of this paper on our website: <http://home.ustc.edu.cn/%7Ehoudd>.

## REFERENCES

- [1] X. Zhu, J. Ding, H. Dong, K. Hu and X. Zhang, "Normalized Correlation-Based Quantization Modulation for Robust Watermarking," *IEEE Trans. Multimedia*, vol. 16, no. 7, pp. 1888-1904, Nov. 2014.
- [2] A. Ahmad, M. Pant, and C. Ahn, "SVD based fragile watermarking scheme for tamper localization and self-recovery," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 6, pp. 1225-1239, 2016.
- [3] A. Khan, A. Siddiqi, S. Munib, and S. A. Malik, "A recent survey of reversible watermarking techniques," *Information sciences*, vol. 279, pp. 251–272, 2014.
- [4] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in 2014 IEEE International Conference on Image Processing (ICIP), pp. 4206-4210, 2014.
- [5] M. Asikuzzaman and M. R. Pickering, "An Overview of Digital Video Watermarking," *IEEE Trans. Circuits System and Video Technology*, vol., no. 99, pp. 1-1. doi: 10.1109/TCSVT.2017.2712162.
- [6] J. Fridrich and M. Goljan, "Lossless Data Embedding for All Image Formats," in SPIE Proceedings of Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents, vol. 4675, pp. 572-583, San Jose, Jan. 2002.
- [7] J. Tian, "Reversible Data Embedding Using a Difference Expansion," *IEEE Trans. Circuits System and Video Technology*, vol. 13, no. 8, pp. 890-896, Aug. 2003.

- [8] Z. Ni, Y. Shi, N. Ansari, and S. Wei, "Reversible Data Hiding," *IEEE Trans. Circuits System and Video Technology*, vol. 16, no. 3, pp. 354-362, 2006.
- [9] P. Tsai, Y.C. Hu, and H.L. Yeh, "Reversible Image Hiding Scheme Using Predictive Coding and Histogram Shifting," *Signal Processing*, vol. 89, pp. 1129-1143, 2009.
- [10] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Shi, "Reversible Watermarking Algorithm Using Sorting and Prediction," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989-999, July 2009.
- [11] X. Li, B. Yang, and T. Zeng, "Efficient Reversible Watermarking Based on Adaptive Prediction-Error Expansion and Pixel Selection," *IEEE Trans. Image Processing*, vol. 20, no. 12, pp. 3524-3533, Dec. 2011.
- [12] B. Ou, X. Li, Y. Zhao, R. Ni and Y. Q. Shi, "Pairwise Prediction-Error Expansion for Efficient Reversible Data Hiding," *IEEE Trans. Image Processing*, vol. 22, no. 12, pp. 5010-5021, Dec. 2013.
- [13] X. Hu, W. Zhang, X. Li, and N. Yu, "Minimum rate prediction and optimized histograms modification for reversible data hiding," *IEEE Trans. Information Forensics and Security*, vol. 10, no. 3, pp. 653-664, 2015.
- [14] B. Ou, X. Li, J. Wang and F. Peng, "High-fidelity reversible data hiding based on geodesic path and pairwise prediction-error expansion," *Neurocomputing*, vol. 226, pp. 23-34, Feb. 2017.
- [15] J. Wang, J. Ni, X. Zhang and Y. Shi, "Rate and distortion optimization for reversible data hiding using multiple histogram shifting," *IEEE Trans. cybernetics*, vol. 47, no. 2, pp. 315-326, Feb. 2017.
- [16] M. Kutter and S. Winkler, "A vision-based masking model for spread-spectrum image watermarking," *IEEE Trans. Image Processing*, vol. 11, no. 1, pp. 16-25, Jan. 2002.
- [17] P. Bao and X. Ma, "Image adaptive watermarking using wavelet domain singular value decomposition," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 96-102, Jan. 2005.
- [18] M. Asikuzzaman, M. J. Alam, A. J. Lambert, and M. R. Pickering, "Imperceptible and robust blind video watermarking using chrominance embedding: A set of approaches in the DT CWT domain," *IEEE Trans. Information Forensics and Security*, vol. 9, no. 9, pp. 1502-1517, Sep. 2014.
- [19] M. Asikuzzaman, M. J. Alam, A. J. Lambert, and M. R. Pickering, "A blind and robust video watermarking scheme using chrominance embedding," *International Conference on Digital Image Computing: Techniques and Applications*, pp. 1-6, Nov. 2014.
- [20] C. H. Chou and K. C. Liu, "A perceptually tuned watermarking scheme for color images," *IEEE Trans. Image Processing*, vol. 19, no. 11, pp. 2966-2982, Nov. 2010.
- [21] Y. He, W. Liang, J. Liang, and M. Pei, "Tensor decompositionbased color image watermarking," *Proceedings of SPIE*, vol. 9069, pp. 90690U-90690U-6, Jan. 2014.
- [22] M. R. A. Lari, S. Ghofrani, and D. McLernon, "Using curvelet transform for watermarking based on amplitude modulation," *Signal, Image and Video Processing*, vol. 8, no. 4, pp. 687-697, May 2014.
- [23] C. Chang, C. Lin, Y. Fan, "Lossless data hiding for color images based on block truncation coding," *Pattern Recognition*, vol. 41, no. 7, pp. 2347-2357, 2008.
- [24] W. Yang, K. Chung, and H. Liao, "Efficient reversible data hiding for color filter array images," *Inf. Sci.*, vol. 190, pp. 208-226, 2012.
- [25] J. Li, X. Li, and B. Yang, "Reversible data hiding scheme for color image based on prediction-error expansion and cross-channel correlation," *Signal Processing*, vol. 93, no. 9, pp. 2748-2758, 2013.
- [26] B. Ou, X. Li, Y. Zhao, and R. Ni, "Efficient color image reversible data hiding based on channel-dependent payload partition and adaptive embedding," *Signal Processing*, vol. 108, pp. 642-657, 2015.
- [27] Viola, Paul, and Michael J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137-154, 2004.
- [28] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, vol. 1, pp. 886-893, 2005.
- [29] D. G. Lowe, "Distinctive image feautres from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91-110, 2004.
- [30] Chandrasekhar V. Makar M, Takacs G, et al., "Survey of SIFT compression schemes," In Proc. Int. Workshop Mobile Multimedia Processing, pp. 35-40, 2010.
- [31] H. Yue, X. Sun, J. Yang and F. Wu, "Cloud-Based Image Coding for Mobile Devices!Toward Thousands to One Compression," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 845-857, June 2013.
- [32] Z. Shi, X. Sun and F. Wu, "Photo Album Compression for Cloud Storage Using Local Features," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, no. 1, pp. 17-28, March 2014.
- [33] A. P. Psyllos, C. N. E. Anagnostopoulos and E. Kayafas, "Vehicle Logo Recognition Using a SIFT-Based Enhanced Matching Scheme," *IEEE Trans. Intelligent Transportation Systems*, vol. 11, no. 2, pp. 322-328, June 2010.
- [34] J. Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade and B. L. Lu, "Person-Specific SIFT Features for Face Recognition," *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, Honolulu, HI, pp. II-593-II-596, 2007.
- [35] Brown, M., Lowe, D. G., "Automatic panoramic image stitching using invariant features," in *International journal of computer vision*, vol. 74, no. 1, pp. 59-73, 2007.
- [36] Z. Hua, Y. Li and J. Li, "Image stitch algorithm based on SIFT and MVSC," *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, Yantai, Shandong, pp. 2628-2632, 2010.
- [37] H. Bay, T. Tuytelaars, and L. J. V. Gool, SURF: Speeded up robust features, in *Proc. Eur. Conf. Comput. Vis.*, pp. 404-417, 2008.
- [38] Amy A. Gooch, Sven C. Olsen, Jack Tumblin, and Bruce Gooch, "Color2Gray: salience-preserving color removal," *ACM Siggraph 2005 Papers*, Markus Gross (Ed.), ACM, New York, NY, USA, 634-639, 2005.
- [39] Mark Grundland, Neil A. Dodgson, "Decolorize: Fast, contrast enhancing, color to grayscale conversion," *Pattern Recognition*, vol. 40, no. 11, pp. 2891-2896, 2007.
- [40] Sowmya V, Govind D, Soman K P, "Significance of incorporating chrominance information for effective color-to-grayscale image conversion," *Signal, Image and Video Processing*, vol. 11, pp. 129-136, 2017.
- [41] Rec. 601: [https://en.wikipedia.org/wiki/Rec.\\_601](https://en.wikipedia.org/wiki/Rec._601).
- [42] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The loco-i lossless image compression algorithm: principles and standardization into jpegls," *IEEE Trans. Image Processing*, vol. 9, no. 8, pp. 1309-1324, 2000.

**Dongdong Hou** received his B.S. degree in 2014 from Hefei University of Technology, Hefei, China. He is now pursuing the Ph.D. degree in University of Science and Technology of China. His research interests include multimedia security, image processing, and deep learning.



**Weiming Zhang** received his M.S. degree and PH.D. degree in 2002 and 2005 respectively from the Zhengzhou Information Science and Technology Institute, Zhengzhou, China. Currently, he is a professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include multimedia security, information hiding, and privacy protection.



**Kejiang Chen** received his B.S. degree in 2015 from Shanghai University, Shanghai, China. He is now pursuing the Ph.D. degree in University of Science and Technology of China. His research interests include data hiding, and deep learning.





**Sian-Jheng Lin** received the B.Sc., M.Sc., and Ph.D. degrees in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2004, 2006, and 2010, respectively. From 2010 to 2014, he was a postdoc with the Research Center for Information Technology Innovation, Academia Sinica. From 2014 to 2016, he was a postdoc with the Electrical Engineering Department at King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. He was a part-time lecturer at Yuanpei University from 2007 to 2008, and at Hsuan Chuang University From 2008 to 2010. He is currently a project researcher with the School of Information Science and Technology at University of Science and Technology of China (USTC), Hefei, China. In recent years, his research focuses on the algorithms for MDS codes and its applications to storage systems.



**Nenghai Yu** received his B.S. degree in 1987 from Nanjing University of Posts and Telecommunications, M.E. degree in 1992 from Tsinghua University and Ph.D. degree in 2004 from University of Science and Technology of China, where he is currently a professor. His research interests include multimedia security, multimedia information retrieval, video processing, information hiding and security, privacy and reliability in cloud computing.