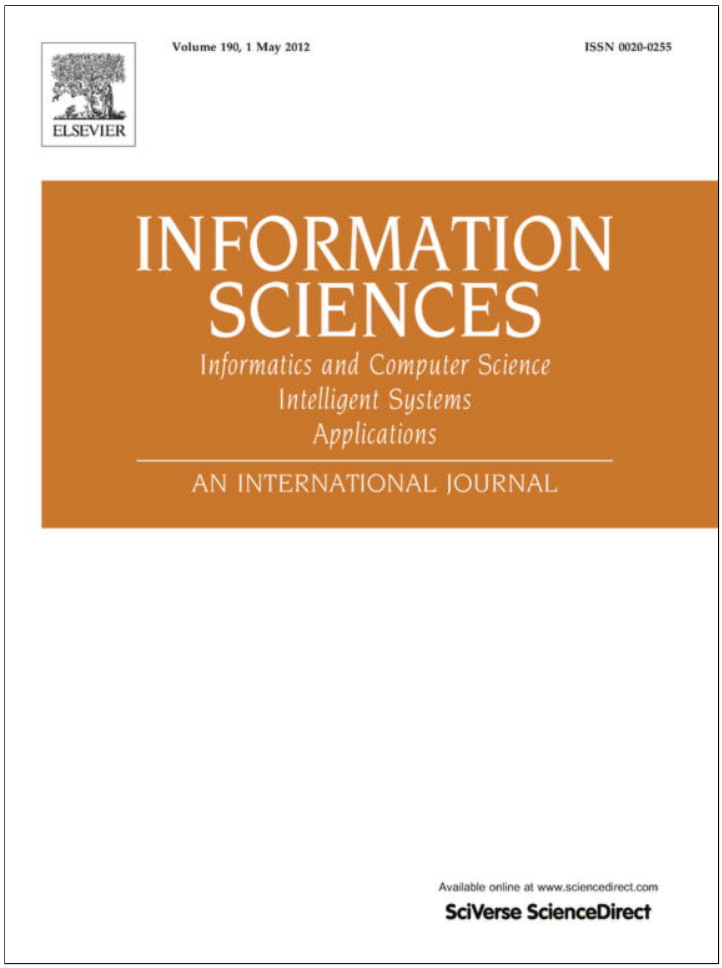

Information Sciences 190 (2012) 208–226

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Efficient reversible data hiding for color filter array images ☆

Wei-Jen Yang [a], Kuo-Liang Chung [a,*], Hong-Yuan Mark Liao [b]

[a] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 10672, Taiwan, ROC
[b] Institute of Information Science, Academia Sinica, No. 128, Section 2, Academia Road, Taipei 11529, Taiwan, ROC

ARTICLE INFO

Article history:
Received 12 April 2011
Received in revised form 29 October 2011
Accepted 26 November 2011
Available online 13 December 2011

Keywords:
Color difference
Color filter array
Difference expansion
Digital time delay and integration
Prediction errors
Reversible data hiding

ABSTRACT

A reversible data hiding algorithm which uses prediction errors in the color difference domain for mosaic images with the Bayer color filter array (CFA) is proposed. Furthermore, the proposed algorithm can be extended to deal with the digital time delay and integration (DTDI) mosaic images and Lukac and Plataniotis (LP) mosaic images. Experimental results on CFA, DTDI, and LP mosaic images demonstrate that the proposed algorithm can achieve high embedding capacity while maintaining good image quality.

1. Introduction

Data hiding is an important technique for embedding secret data in a host image [12]. In the last decade, a large number of data hiding algorithms have been proposed for black/white images and color images, e.g. [9,10,12,16,20]. A common drawback of data hiding algorithms is that the original image cannot be recovered completely. The problem becomes more serious when dealing with certain sensitive images, such as military, medical, and artwork images. To completely recover the original images, the reversible data hiding technique was first developed by Barton [1]. Researchers on reversible data hiding algorithms emphasized how to increase the embedding capacity and enhance the quality of the marked images [4,6–8, 13–15,17–19,25,27,29–34].

Tian [32] proposed a reversible data hiding algorithm based on an integer version of the Haar wavelet transform [26]. Kamstra and Heijmans [17] introduced an algorithm, which used the low-pass subband to determine the location for embedding, to resolve the image distortion problem in Tian's algorithm. Kim et al. [18] developed a difference expansion transform to improve the performance of Kamstra and Heijmans' algorithm. Ni et al. [27] and Chang et al. [4] developed reversible data hiding algorithms based on the use of peak-valley pairs in an image histogram and the outcome of side match vector quantization, respectively. Chang et al. [5] introduced a reversible data hiding algorithm, which modified the coefficients of the medium-frequency components to embed the hidden data, for DCT-based compressed images. Lin et al. [19] proposed a multilevel reversible data hiding algorithm by modifying the difference image histogram. Fallahpour [13] presented a reversible data hiding algorithm based on gradient adjusted prediction. Based on joint neighboring coding scheme, two reversible data hiding

☆ Supported by the National Science Council of the ROC under contract NSC99-2221-E-011-078-MY3. K-L Chung conducted his research while visiting the Institute of the Information Science, Academia Sinica in 2010.

* Corresponding author.
E-mail address: k.l.chung@mail.ntust.edu.tw (K.-L. Chung).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.11.046

algorithms were, respectively, developed by Chang et al. [6] and Wang and Lu [34]. Tseng and Hsieh [33] proposed a prediction-based reversible data hiding algorithm. Chang and Kieu [7] proposed a reversible data hiding algorithm based on the complementary embedding strategy; while Chang et al. [8] introduced a frequency-based reversible image hiding algorithm. Luo et al. [25] proposed a reversible data hiding algorithm exploiting a multi-level histogram shifting mechanism. Gao et al. [14] developed a reversible data hiding algorithm by using the generalized statistical quantity histogram. In [29–31], three reversible data hiding algorithms based on the concepts of prediction error expansion and histogram modification were proposed. The three algorithms utilized prediction schemes to predict the pixel values and embedded hidden data in the pixels by modifying the corresponding prediction errors. Thodi and Rodriguez [31] used the LOCO-I predictor [35], which utilized the information of the neighboring pixels, to predict the value of the pixel. Then, a histogram shifting scheme based on the prediction error expansion was used to embed the hidden data. Furthermore, a two-pass testing method with a flag bit stream was applied to resolve the overflow and underflow problems. Tai et al. [30] scanned an image by an inverse *s*-order and predicted each pixel using the scanned value of the previous pixel. Then, the histogram of the prediction errors was used to embed the hidden data. Without using the two-pass testing method, the pixel histogram was contracted from both sides to ensure that the embedding would not cause the overflow and underflow problems. The algorithm could also solve the problem of communicating multiple peak points to recipients, which was the major drawback in Lin et al.'s algorithm [19]. Using different prediction scheme, Sachnev et al. [29] predicted the pixel value by averaging the gray values of the four neighboring pixels in a rhombus shape. The hidden data were embedded, using the difference expansion, into the pixels with prediction errors falling into the range of the two specified thresholds T_n and T_p. Furthermore, in order to enhance the quality of the marked image, the shifting scheme in [31] is used and the embedding order is determined by a sorting strategy.

Most reversible data hiding algorithms are often applied on the black/white and color images but seldom on the mosaic images, which are captured by the digital cameras. Since digital cameras are becoming increasingly popular in the consumer electronics market, equipping them with a secret data embedding capability is an important issue. To reduce costs, most digital cameras capture color information by using a single charge-coupled device/complementary metal–oxide–semiconductor (CCD/CMOS) sensor with the Bayer color filter array (CFA) structure [2,22–24]. As shown in Fig. 1, each pixel in the Bayer CFA structure only has one measured color. This kind of image is called a *mosaic image* [11,36]. Because the green (G) channel is the most important factor to determine the luminance of a color image, half of the pixels in the Bayer CFA structure are assigned to the G channel whereas the red (R) and blue (B) channels, which share the other half of the Bayer CFA structure, contribute to the chrominance information. Since the structure of a mosaic image is quite different from that of gray and color images, existing reversible data hiding algorithms do not work well on the mosaic images. Although we can partition a mosaic image into four color planes, and then apply existing reversible data hiding algorithms to each of the rearranged color planes, the most important property of spectral–spatial correlation [28,21,37] in a CFA mosaic image cannot be exploited. A reversible data hiding algorithm exploiting the spectral–spatial correlation can efficiently reduce the prediction errors, resulting in large embedding capacity and good quality of marked images. In addition to CFA mosaic images, there exist other types of mosaic images. A digital time delay and integration (DTDI) mosaic image [3], which has two color channels in each pixel, can be decomposed into two CFA mosaic images. A Lukac and Plataniotis (LP) mosaic image [22], which is another type of the mosaic arrangement exploited in the single sensor digital camera, can be converted into a CFA mosaic image. Therefore, we can apply the proposed algorithm to the DTDI and LP mosaic images. The main motivation of this work is twofold. In the first place, develop a reversible data hiding algorithm, which exploits spectral–spatial correlation, for CFA mosaic images. Second, apply the proposed algorithm to DTDI mosaic images, widely used in industrial printer applications, and LP mosaic images.

In this paper, we propose an efficient reversible data hiding algorithm for CFA mosaic images. The proposed algorithm utilizes the spectral–spatial correlation to achieve small prediction errors in the color difference domain for embedding hidden data. Since such prediction errors tend to follow a Laplacian distribution with relatively small variance, the proposed algorithm achieves high embedding capacity and good quality of marked images. The proposed algorithm can also be applied to DTDI and LP mosaic images. Experimental results on typical CFA, DTDI, and LP mosaic images demonstrate that with the

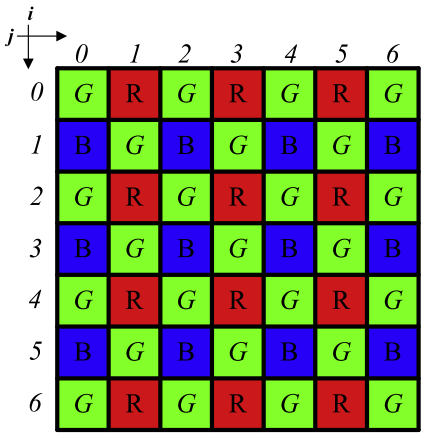

Fig. 1. The Bayer CFA structure [2].

same embedding capacity, the proposed data hiding algorithm yields better quality of marked images than the algorithms presented in [29–31]. We compare the proposed algorithm with the above three reversible algorithms for two reasons. First, like the proposed algorithm, the three compared algorithms are based on the concepts of prediction error expansion and histogram modification. Second, they are regarded as state-of-the-art reversible data hiding algorithms.

The major novel contributions of this work can be stated as follows. First, we propose a novel reversible algorithm, which explicitly exploits spectral–spatial correlation, for the CFA mosaic image. Second, to further enhance the quality of the marked image, a variance-sorting strategy the color difference domain is used to determine the embedding order and a smoother color difference domain is selected to embed the hidden data. Third, since the DTDI and LP images can be decomposed or converted to CFA mosaic images, the proposed algorithm can also process DTDI and LP mosaic images.

The remainder of this paper is organized as follows. In Section 2, a brief introduction to spectral–spatial correlation in the color difference domain is given. In Section 3, we discuss the prediction errors in the color difference domain and the associated Laplacian distribution. In Section 4, we present the proposed reversible data hiding algorithm for CFA mosaic images. Section 5 explains how to apply the proposed algorithm to DTDI and LP mosaic images. In Section 6, we report the experimental results on the embedding capacity and the quality of marked images. Section 7 gives the concluding remarks.

2. Spectral–spatial correlation in the color difference domain

In this section, we introduce spectral–spatial correlation in the color difference domain. Fig. 2(a)–(c) illustrate, respectively, the *G* color channel, the *G-R* color difference plane, and the *G-B* color difference plane of the Carving image. From the three figures, it is obvious that the contrasts of the *G-R* and *G-B* color difference planes are much flatter than that of the *G* channel, indicating that the color difference plane is more suitable for predicting the pixel values. Since spectral–spatial correlation in the *G-R* color difference domain is the same as that in the *G-B* color difference domain, we only discuss how to predict the pixel values in the *G-R* color difference domain.

As shown in Fig. 1, denote by $I^K_{mo}(i,j)$ the color value in channel $K(K \in \{R,G,B\})$ of the pixel located at position (i,j) in a mosaic image of size $M \times N$ with $i \in \{0,1,\ldots,M-1\}$ and $j \in \{0,1,\ldots,N-1\}$. For simplicity, consider how to predict the *G-R* color difference value of the pixel at position (3,3). Denote by $\widetilde{I}^R_{mo}(i,j)$ the predicted color value of channel *R*. Due to the flat property of a color difference domain, the assumption of spectral–spatial correlation suggests

$$I^G_{mo}(3,3) - \widetilde{I}^R_{mo}(3,3) = I^G_{mo}(2,2) - \widetilde{I}^R_{mo}(2,2), \tag{1}$$

where $\widetilde{I}^R_{mo}(2,2)$ is predicted by $\frac{I^R_{mo}(2,1)+I^R_{mo}(2,3)}{2}$. Then, the predicted *G-R* color difference value of the pixel at position (3,3) along upper-left direction can be approximated by

$$\mathcal{G}^{ul}_R(3,3) = I^G_{mo}(2,2) - \widetilde{I}^R_{mo}(2,2) = I^G_{mo}(2,2) - \frac{I^R_{mo}(2,1) + I^R_{mo}(2,3)}{2}. \tag{2}$$

Similarly, the predicted *G-R* color difference value of the pixel at position (3,3) along upper-right, lower-left, and lower-right directions can be approximated by

$$\begin{aligned}
\mathcal{G}^{ur}_R(3,3) &= I^G_{mo}(2,4) - \frac{I^R_{mo}(2,3) + I^R_{mo}(2,5)}{2},\\
\mathcal{G}^{ll}_R(3,3) &= I^G_{mo}(4,2) - \frac{I^R_{mo}(4,1) + I^R_{mo}(4,3)}{2},\\
\mathcal{G}^{lr}_R(3,3) &= I^G_{mo}(4,4) - \frac{I^R_{mo}(4,3) + I^R_{mo}(4,5)}{2}.
\end{aligned} \tag{3}$$

Thus, the predicted *G-R* color difference value for the *G* pixel at position (3,3) can be expressed as

$$\overline{\mathcal{G}}_R(3,3) = \frac{1}{4} \sum_{d \in \{ul,ur,ll,lr\}} \mathcal{G}^d_R(3,3). \tag{4}$$

In addition to predicting the color difference values for *G* pixels, we can also apply the concept of spectral–spatial correlation to predict the color difference values for both *R* and *B* pixels.

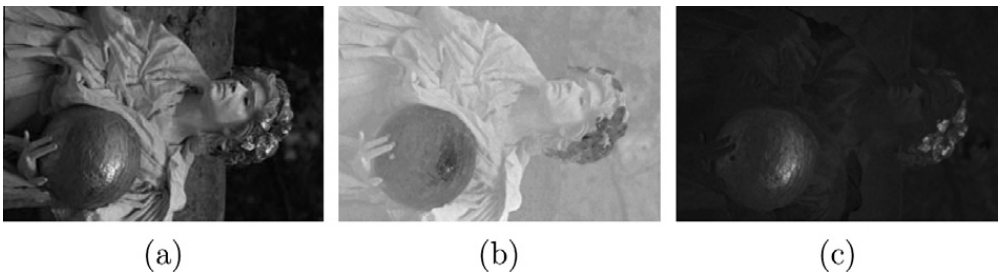

Fig. 2. (a) *G* color channel, (b) *G-R* color difference plane, and (c) *G-B* color difference plane of the Carving image.

3. Prediction errors in the color difference domain and the associated Laplacian distribution

According to the concept of spectral–spatial correlation, in small regions, the contrast of an image's color difference plane is much flatter than that of the spacial channel, indicating that the color difference domain is more appropriate for error prediction. Thus, performing prediction in the color difference domain instead of the spacial domain can enhance the embedding capacity and yield better quality of the marked images. In the following sub-sections, we explain how to obtain the prediction errors and derive the associated Laplacian distribution.

3.1. Prediction errors in G pixels

Based on the spectral–spatial correlation concept, we can predict the color difference values for the G pixels and obtain the associated prediction errors. For embedding hidden data in each color channel, two independent pixel sets in each channel will be constructed. For simplicity, we use Fig. 3 to show how to obtain the prediction errors of the G pixels in a mosaic image of size $M \times N$. In Fig. 3, for a generic G pixel at (i,j), divide the G pixels into two sets, denoted by $\Omega_{G1} = \{(i \pm 2m, j \pm 2n) | \forall m,n \in \mathbb{N}, 0 \leqslant i \pm 2m \leqslant M-1, 0 \leqslant i \pm 2n \leqslant N-1\}$ and $\Omega_{G2} = \{(i \pm (2m+1), j \pm (2n+1)) | \forall m,n \in \mathbb{N}, 0 \leqslant i \pm (2m+1) \leqslant M-1, 0 \leqslant i \pm (2n+1) \leqslant N-1\}$, and mark, respectively, the set elements by the symbols "▲" and "○." Since Ω_{G1} and Ω_{G2} are disjoint, the color difference values of the pixels in Ω_{G1} can be predicted by using the color difference values of the pixels in Ω_{G2} and vice versa. Two reasons exist for dividing the G pixels into two independent sets. The first reason is that since embedding hidden data in the pixels in one set would not affect the pixels in the other set, the precise predicted color difference value of each pixel can be obtained to extract the hidden data and then recover the original pixel values in the proposed extraction strategy. The second reason is that the embedding order for one set, determined by the other set, can be maintained for recovering. Since the color difference prediction process is the same for both sets, we only discuss the prediction process for Ω_{G1}.

In the sub-image shown in Fig. 4, the color difference prediction error corresponding to the central pixel located at position (i,j) is the difference between the predicted color difference value, which is the average color difference value of four neighboring corner pixel with positions $(i \pm 1, j \pm 1)$, and its own color difference. For each corner pixel at position (x,y), denoting respectively by $\mathcal{G}_R(x,y) = I^G_{mo}(x,y) - \frac{1}{2}\sum_{k\in\{\pm1\}} I^R_{mo}(x,y+k)$ and $\mathcal{G}_B(x,y) = I^G_{mo}(x,y) - \frac{1}{2}\sum_{k\in\{\pm1\}} I^B_{mo}(x+k,y)$ the G-R and G-B color difference values, the predicted G-R color difference value $\overline{\mathcal{G}}_R(i,j)$ and predicted G-B color difference value $\overline{\mathcal{G}}_B(i,j)$ for the pixel at position (i,j) can be expressed as

$$\begin{aligned} \overline{\mathcal{G}}_R(i,j) &= \frac{1}{4}\sum_{(x,y)\in\Psi_G} \mathcal{G}_R(x,y), \\ \overline{\mathcal{G}}_B(i,j) &= \frac{1}{4}\sum_{(x,y)\in\Psi_G} \mathcal{G}_B(x,y), \end{aligned} \tag{5}$$

where $\Psi_G = \{(i \pm 1, j \pm 1)\}$. Since the predictor with less variation yields better prediction, only one of $\overline{\mathcal{G}}_R(i,j)$ and $\overline{\mathcal{G}}_B(i,j)$ that has lower variance is used as the predicted color difference value for the central pixel at position (i,j). Thus, the predicted color difference value $\overline{\mathcal{G}}(i,j)$ for the central pixel is determined by

$$\overline{\mathcal{G}}(i,j) = \begin{cases} \overline{\mathcal{G}}_R(i,j), & \text{if } \sigma^2_{\mathcal{G}_R}(i,j) \leqslant \sigma^2_{\mathcal{G}_B}(i,j), \\ \overline{\mathcal{G}}_B(i,j), & \text{otherwise}, \end{cases} \tag{6}$$

where $\sigma^2_{\mathcal{G}_R}(i,j) = \frac{1}{4}\sum_{(x,y)\in\Psi_G}[\mathcal{G}_R(x,y) - \overline{\mathcal{G}}_R(i,j)]^2$ and $\sigma^2_{\mathcal{G}_B}(i,j) = \frac{1}{4}\sum_{(x,y)\in\Psi_G}[\mathcal{G}_B(x,y) - \overline{\mathcal{G}}_B(i,j)]^2$. The G-R and G-B color difference values of the central pixel at position (i,j) can be expressed as

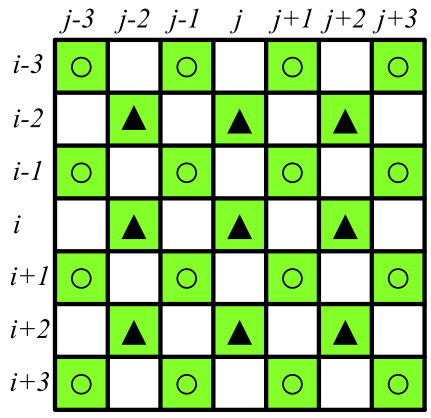

Fig. 3. Pixels in Ω_{G1} and Ω_{G2} with ▲ denoting pixels in Ω_{G1} and ○ in Ω_{G2}.

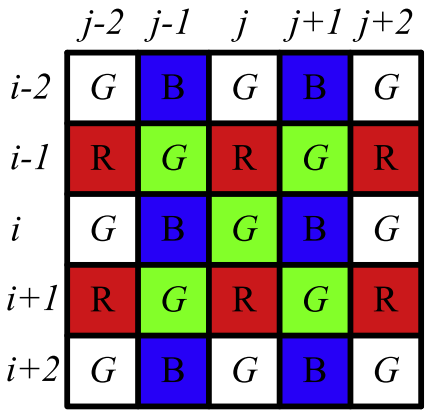

Fig. 4. Typical colors in Ω_{G1}.

$$\begin{aligned}\widehat{\mathcal{G}}_R(i,j) &= I^G_{mo}(i,j) - \frac{1}{2}\sum_{k=\pm1} I^R_{mo}(i+k,j),\\ \widehat{\mathcal{G}}_B(i,j) &= I^G_{mo}(i,j) - \frac{1}{2}\sum_{k=\pm1} I^B_{mo}(i,j+k).\end{aligned} \tag{7}$$

Then, the prediction error for the central pixel at position (i,j) is determined as

$$\Delta\mathcal{G}(i,j) = \begin{cases} \widehat{\mathcal{G}}_R(i,j) - \overline{\mathcal{G}}_R(i,j), & \text{if } \sigma^2_{\mathcal{G}_R}(i,j) \leqslant \sigma^2_{\mathcal{G}_B}(i,j),\\ \widehat{\mathcal{G}}_B(i,j) - \overline{\mathcal{G}}_B(i,j), & \text{otherwise.}\end{cases} \tag{8}$$

Next, we consider the prediction errors in R and B pixels.

3.2. Prediction errors in R and B pixels

Since the process for obtaining the prediction errors in R and B pixels is the same, we only discuss the prediction errors in R pixels. Similar to the prediction process for G pixels, as shown in Fig. 5, for a generic R pixel at (i,j), divide R pixels into two sets, denoted by $\Omega_{R1} = \{(i \pm 4m, j \pm 4n) \cup (i \pm (4m+2), j \pm (4n+2)) | \forall m,n \in \mathbb{N}, 0 \leqslant i \pm 4m, i \pm (4m+2) \leqslant M-1, 0 \leqslant j \pm 4n, j \pm (4n+2) \leqslant N-1\}$ and $\Omega_{R2} = \{(i \pm 4m, j \pm (4n+2)) \cup (i \pm (4m+2), j \pm 4n) | \forall m,n \in \mathbb{N}, 0 \leqslant i \pm 4m, i \pm (4m+2) \leqslant M-1, 0 \leqslant j \pm 4n, j \pm (4n+2) \leqslant N-1\}$, and mark the set elements by the symbols "▲" and "○", respectively. Again, since the color difference prediction process for the pixels in Ω_{R1} and Ω_{R2} is the same, we only discuss the prediction process for Ω_{R1}.

In the sub-image shown in Fig. 6, the prediction error of the central pixel at position (i,j) is the difference between the predicted color difference value, which is the average color difference value of four neighboring pixel with positions $\{(i \pm 2, j), (i, j \pm 2)\}$, and its own color difference. For each neighboring (around (i,j)) pixel at position (x,y), denoting by $\mathcal{R}_G(x,y) = I^R_{mo}(x,y) - \frac{1}{4}\sum_{(x,y)\in\xi_1} I^G_{mo}(x,y)$ with $\xi_1 = \{(x \pm 1, y), (x, y \pm 1)\}$ the R-G difference values, the predicted R-G color difference value $\overline{\mathcal{R}}_G(i,j)$ for the pixel at position (i,j) can be expressed as

$$\overline{\mathcal{R}}(i,j) = \frac{1}{4}\sum_{(x,y)\in\Psi_R} \mathcal{R}_G(x,y), \tag{9}$$

where $\Psi_R = \{(i \pm 2, j), (i, j \pm 2)\}$. The color difference value, $\widehat{\mathcal{R}}(i,j)$, of the central pixel at (i,j) can be expressed as

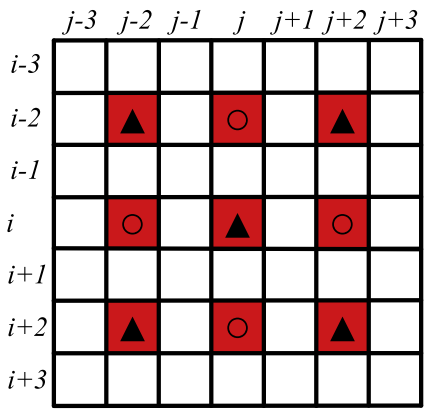

Fig. 5. Pixels in Ω_{R1} and Ω_{R2} with ▲ denoting pixels in Ω_{R1} and ○ in Ω_{R2}.

	j-3	*j-2*	*j-1*	*j*	*j+1*	*j+2*	*j+3*
i-3	B	*G*	B	*G*	B	*G*	B
i-2	*G*	R	*G*	R	*G*	R	*G*
i-1	B	*G*	B	*G*	B	*G*	B
i	*G*	R	*G*	R	*G*	R	*G*
i+1	B	*G*	B	*G*	B	*G*	B
i+2	*G*	R	*G*	R	*G*	R	*G*
i+3	B	*G*	B	*G*	B	*G*	B

Fig. 6. Typical colors in Ω_{R1}.

$$\widehat{\mathcal{R}}(i,j) = I^R_{mo}(i,j) - \frac{1}{4}\sum_{(x,y)\in\xi_2} I^G_{mo}(x,y), \tag{10}$$

with $\xi_2 = \{(i \pm 1, j), (i, j \pm 1)\}$, which is the same as $\mathcal{R}(i,j)$. Thus, the prediction error $\Delta\mathcal{R}(i,j)$ can be calculated by

$$\Delta\mathcal{R}(i,j) = \widehat{\mathcal{R}}(i,j) - \overline{\mathcal{R}}(i,j). \tag{11}$$

Since the color difference domain has smaller measuring scale due to less variation, the distribution of the prediction errors in the color difference domain tends to have smaller variance than that in the spatial domain, implying that higher embedding capacity and better quality of marked images can be achieved.

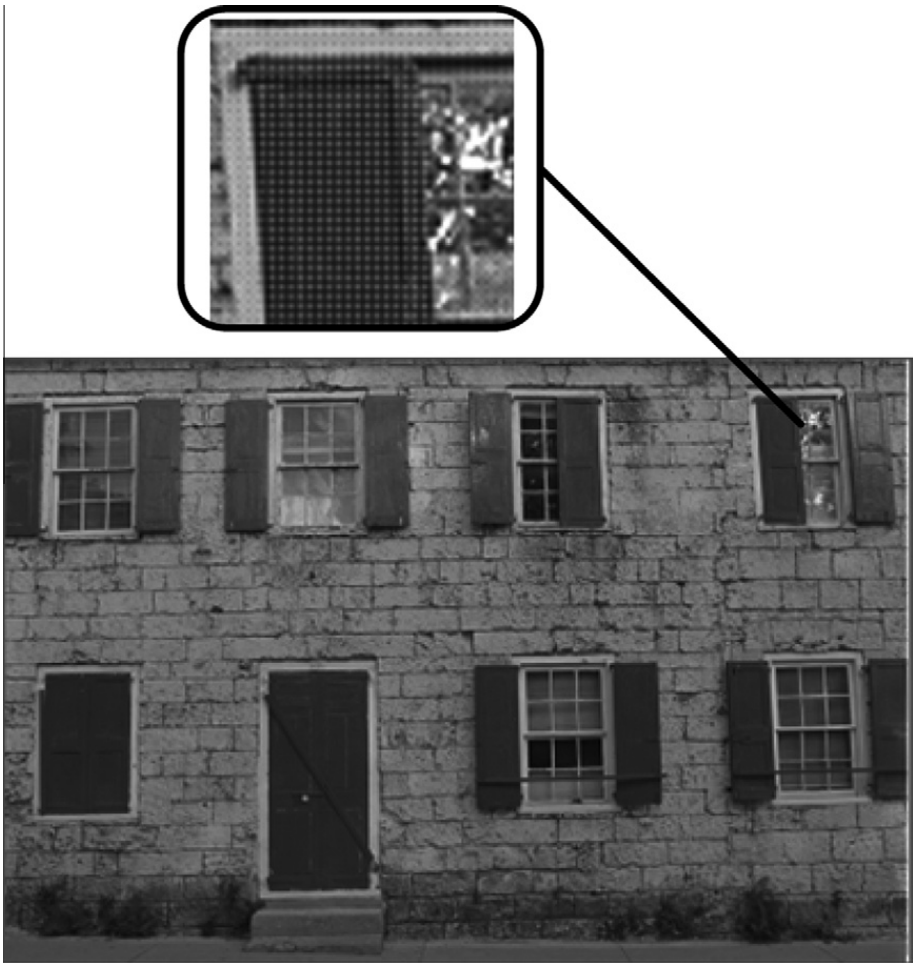

Fig. 7. The mosaic wall image.

3.3. Laplacian distribution of prediction errors and the potential embedding benefits

The prediction errors generally follow a Laplacian distribution on which the performance of prediction-error-based reversible data hiding algorithms heavily depend. It is known that less variation of the Laplacian distribution implies better embedding performance [29]. In this sub-section, we compare specifically the variance of the prediction error distribution associated with the proposed scheme with those corresponding to the previously proposed schemes by Tai et al. [30], Thodi and Rodriguez [31], and Sachnev et al. [29].

Since the above three previous schemes were designed to operate in the spatial domain, for the purpose of comparison, we partition a CFA mosaic image into four color planes, $\mathbf{G}_0$, $\mathbf{G}_1$, $\mathbf{R}_0$ and $\mathbf{B}_0$, according to the following rules [24]:

$$\begin{aligned}
\mathbf{G}_0 &= \left\{ I^G_{mo}(i,j) | i \text{ and } j \text{ are even} \right\}, \\
\mathbf{G}_1 &= \left\{ I^G_{mo}(i,j) | i \text{ and } j \text{ are odd} \right\}, \\
\mathbf{R}_0 &= \left\{ I^R_{mo}(i,j) | i \text{ is even and } j \text{ is odd} \right\}, \\
\mathbf{B}_0 &= \left\{ I^B_{mo}(i,j) | i \text{ is odd and } j \text{ is even} \right\}.
\end{aligned} \tag{12}$$

The mosaic Wall image shown in Fig. 7 is used as an example and the associated four color planes are shown in Fig. 8. The histograms of the prediction errors associated with the three compared schemes and the proposed scheme are shown, respectively, in Fig. 9(a)–(d). Clearly, the histogram associated with the proposed scheme has smaller variance, implying that the proposed algorithm can achieve better embedding performance.

4. The proposed reversible data hiding algorithm

In this section, we describe the proposed reversible data hiding algorithm for CFA mosaic images. Since the application of the algorithm is the same for the *G*, *R*, and *B* color pixels, only *G* color pixels are considered. In Section 4.1, we present the embedding and extraction strategies. Section 4.2 explains how to determine an appropriate threshold to guide the data hiding process. In Section 4.3, we describe how the embedding order determined by a sorting strategy is used to reduce the distortion in the marked images. The detailed procedure of the proposed algorithm is given in Section 4.4.

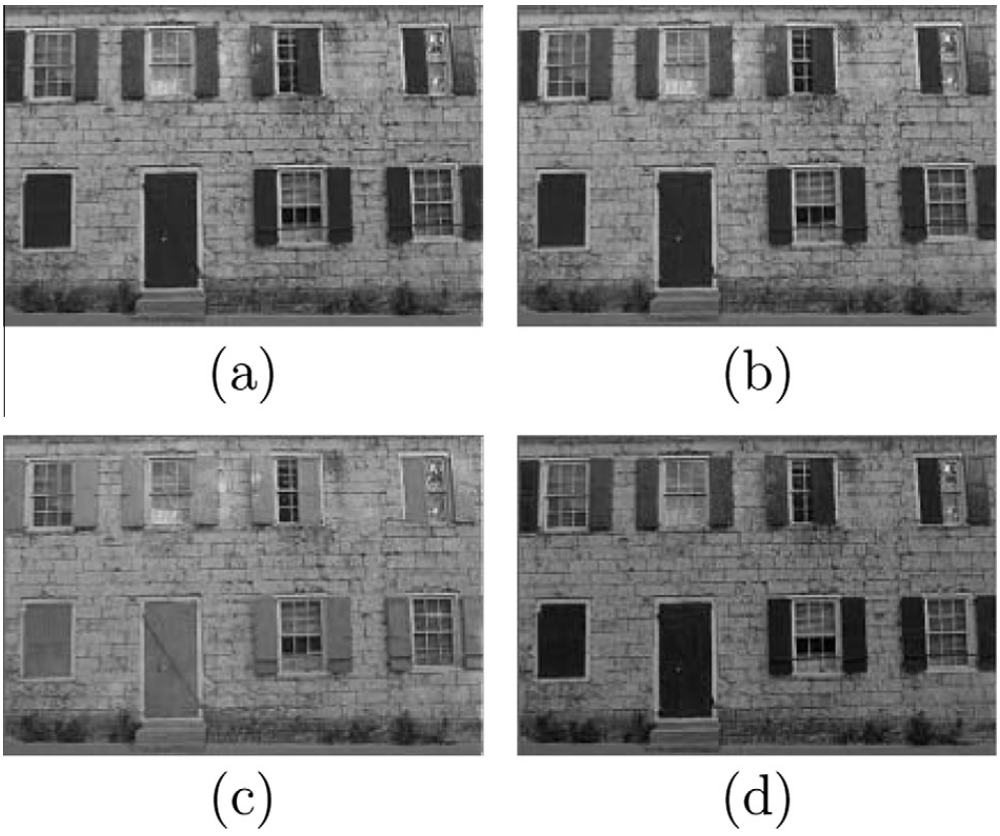

Fig. 8. Four partitioned color planes (a) $\mathbf{G}_0$, (b) $\mathbf{G}_1$, (c) $\mathbf{R}_0$, and (d) $\mathbf{B}_0$ of the mosaic Wall image.

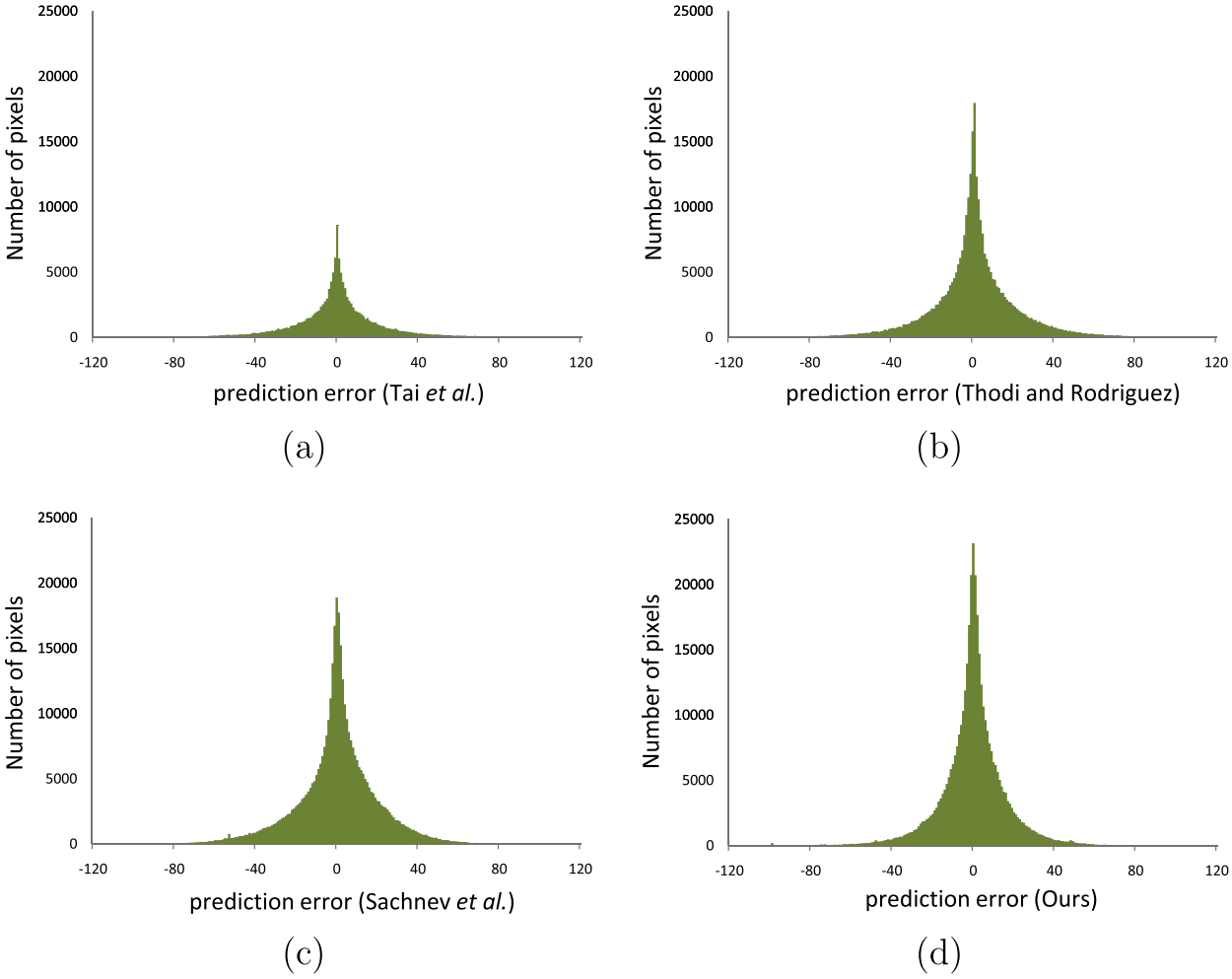

Fig. 9. Prediction error histograms of (a) Tai et al.'s scheme. (b) Thodi and Rodriguez's scheme. (c) Sachnev et al.'s scheme, and (d) the proposed scheme for the mosaic wall image.

4.1. The embedding and extraction strategies

For the proposed algorithm, the G pixels in a CFA mosaic image I^G_{mo} are first divided into two sets Ω_{G1} and Ω_{G2}, as shown in Fig. 3. The embedding strategy states that we first embed, as much as possible, the hidden data in the pixels in Ω_{G1}, and then embed the remaining hidden data in the pixels in Ω_{G2}. Given an input pixel $I^G_{mo}(i,j)$, as shown in Fig. 4, the embedding strategy involves three steps:

Step 1: Calculate, by Eq. (8), the prediction error $\Delta\mathcal{G}(i,j)$ which can be re-expressed as

$$\Delta\mathcal{G}(i,j) = I^G_{mo}(i,j) - \widehat{C}(i,j) - \overline{\mathcal{G}}(i,j), \tag{13}$$

where

$$\widehat{C}(i,j) = \begin{cases} \frac{1}{2}\sum\limits_{k\in\{\pm1\}} I^R_{mo}(i+k,j), & \text{if } \sigma^2_{\mathcal{G}_R}(i,j) \leqslant \sigma^2_{\mathcal{G}_B}(i,j), \\ \frac{1}{2}\sum\limits_{k\in\{\pm1\}} I^B_{mo}(i,j+k), & \text{otherwise}, \end{cases} \tag{14}$$

implying that $I^G_{mo}(i,j) = \Delta\mathcal{G}(i,j) + \widehat{C}(i,j) + \overline{\mathcal{G}}(i,j)$.

Step 2: Embed, by difference expansion [32], the hidden data in the pixels with small magnitude of prediction errors and calculate the modified prediction error as follows:

$$\Delta\mathcal{G}'(i,j) = \begin{cases} \Delta\mathcal{G}(i,j) + T + 1, & \text{if } \Delta\mathcal{G}(i,j) > T, \\ \Delta\mathcal{G}(i,j) - T - 1, & \text{if } \Delta\mathcal{G}(i,j) < -T - 1, \\ 2\Delta\mathcal{G}(i,j) + h, & \text{otherwise}, \end{cases} \tag{15}$$

where $h \in \{0,1\}$ represents the hidden bit; and T ($\geqslant 0$) denotes a predefined threshold, which will be discussed in Section 4.2.

Step 3: Generate the marked mosaic G pixel $I'^{G}_{mo}(i,j)$ by

$$I'^{G}_{mo}(i,j) = \Delta\mathcal{G}'(i,j) + \widehat{C}(i,j) + \overline{\mathcal{G}}(i,j) = I^{G}_{mo}(i,j) + \mathcal{G}'(i,j) - \mathcal{G}(i,j), \tag{16}$$

where $\mathcal{G}'(i,j) - \mathcal{G}(i,j)$ denotes the distortion of the marked image.

For illustration, consider the case of $I^{G}_{mo}(i,j) = 50, \widehat{C}(i,j) = 47, \overline{\mathcal{G}}(i,j) = 2, h = 1$, and $T = 1$ which implies $\Delta\mathcal{G}(i,j) = 50 - 47 - 2 = 1$. For $(-T-1) \leqslant \Delta\mathcal{G}(i,j) \leqslant T$, we embed the hidden bit $h = 1$ in $\Delta\mathcal{G}(i,j)$ and obtain $\Delta\mathcal{G}'(i,j) = 2\Delta\mathcal{G}(i,j) + h = 3$. Thus, we have the marked mosaic G pixel $I'^{G}_{mo}(i,j) = \Delta\mathcal{G}'(i,j) + \widehat{C}(i,j) + \overline{\mathcal{G}}(i,j) = 52$.

After modifying the input image by embedding the hidden data in the pixels in Ω_{G1}, embed similarly the remaining hidden data in the pixels in Ω_{G2} based on the modified pixels in Ω_{G1} to yield the marked image. Furthermore, embedding the hidden data based on the prediction errors may cause the problems of overflow and underflow. The problems may be resolved using the two-pass method proposed in [31,29] where an extra flag bit stream is stored for use in the extraction process. Empirically, the size of the flag bit stream is often small and the deterioration on the embedding capacity can be negligible.

For extracting the hidden data, first extract from Ω_{G2} and then Ω_{G1}. Since the same strategy for extracting the hidden data from the pixels in Ω_{G1} and Ω_{G2}, for illustration, we only consider the case of extracting the hidden data from the pixels in Ω_{G1}. Consider extracting hidden data from the central pixel at position (i,j) in Fig. 4. Given marked mosaic G pixel $I'^{G}_{mo}(i,j)$, the data extraction process is executed in four steps:

Step 1: Calculate, according to Eq. (16), the modified prediction error

$$\Delta\mathcal{G}'(i,j) = I'^{G}_{mo}(i,j) - \widehat{C}(i,j) - \overline{\mathcal{G}}(i,j). \tag{17}$$

Step 2: If $\Delta\mathcal{G}'(i,j) \in [-2T-2, 2T+1]$, extract the hidden bit h in $I'^{G}_{mo}(i,j)$ by

$$h = \Delta\mathcal{G}'(i,j) \bmod 2; \tag{18}$$

otherwise, go to Step 3.

Step 3: Obtain the original prediction error by

$$\Delta\mathcal{G}(i,j) = \begin{cases} \Delta\mathcal{G}'(i,j) - T - 1, & \text{if } \Delta\mathcal{G}'(i,j) > 2T+1, \\ \Delta\mathcal{G}'(i,j) + T + 1, & \text{if } \Delta\mathcal{G}'(i,j) < -2T-2, \\ \lfloor \Delta\mathcal{G}'(i,j)/2 \rfloor, & \text{otherwise.} \end{cases} \tag{19}$$

Step 4: Recover the original mosaic G pixel by

$$I^{G}_{mo}(i,j) = \Delta\mathcal{G}(i,j) + \widehat{C}(i,j) + \overline{\mathcal{G}}(i,j). \tag{20}$$

4.2. Threshold determination

The embedding capacity and the quality of a marked image are affected by the threshold T in Eq. (15). Larger T provides higher embedding capacity but results in more degradation in the marked images. Based on the mosaic Wall image and the mosaic Girl image, Fig. 10(a) and (b) plot, respectively, the quality of the marked image against the embedding capacity for different values of thresholds.

To embed hidden data, a header for recording the size of the hidden data, the size of flag bit stream, and the value of threshold is required. Given the hidden data of size $|\mathbb{P}|$, the accompanying flag bit stream of size $|\mathbb{F}|$ which depends on the threshold used, and the header stream of size $|\mathbb{H}|$, find the required capacity for accommodating these information of size $|\mathbb{T}| = |\mathbb{H}| + |\mathbb{F}| + |\mathbb{P}|$ and then determine the smallest threshold that yields the best quality of the marked image. Since a header of size $|\mathbb{H}|$ is embedded, using the least significant bit (LSB) replacement method [32], in the first $|\mathbb{H}|$ pixels of a mosaic image, an extra correction bit stream of size $|\mathbb{H}|$ is required for recording the $|\mathbb{H}|$ LSB values that are replaced by the header, implying that the embedding phase starts from the $(|\mathbb{H}|+1)$th pixel.

The determination of the threshold can be summarized as follows.

Step 1: Set $T = 0$.

Step 2: Based on the current threshold T, calculate the number $|\mathbb{C}|$, usually referred to the usable capacity, of pixels whose prediction errors are within the range of $[-T-1, T]$ and in which embedding will not cause the problems of overflow and underflow, and $|\mathbb{T}| = |\mathbb{H}| + |\mathbb{F}| + |\mathbb{P}|$.

Step 3: If $|\mathbb{C}| \geqslant |\mathbb{T}|$, output T as the appropriate threshold and stop; otherwise, set $T = T + 1$ and go to Step 2.

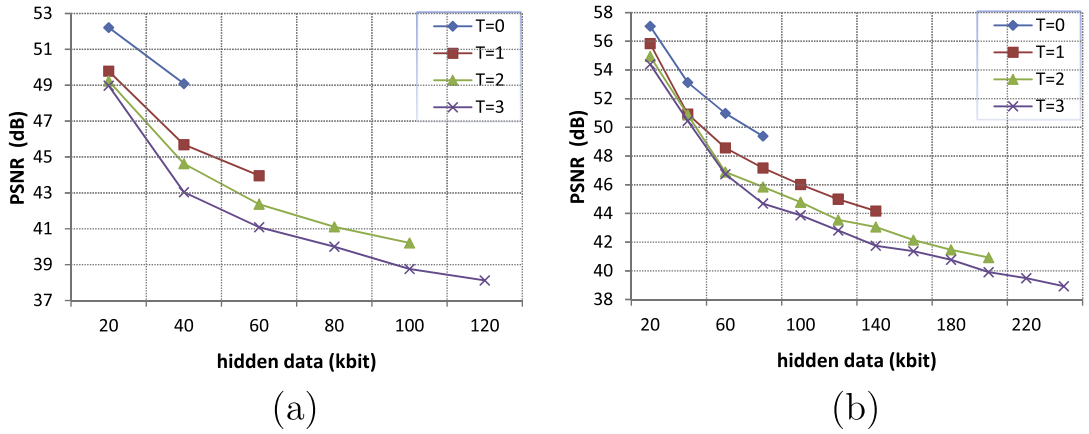

Fig. 10. Quality of the marked image (PSNR) against embedding capacity for different values of thresholds for CFA mosaic images: (a) wall and (b) girl.

4.3. Determination of embedding order to reduce distortion in marked images

The quality of the marked images depends on the embedding order. Small variance of color difference usually suggests small prediction errors, implying that we should first search for the pixels with small local variance to embed hidden data. Therefore, an embedding order, based on the local variances, is determined to reduce the distortion of the marked images.

For a pixel $I^G_{mo}(i,j)$, the corresponding local variance $\sigma^2_{\mathcal{G}}(i,j)$ is determined by

$$\sigma^2_{\mathcal{G}}(i,j) = \min\left(\sigma^2_{\mathcal{G}_R}(i,j), \sigma^2_{\mathcal{G}_B}(i,j)\right), \tag{21}$$

where $\sigma^2_{\mathcal{G}_R}(i,j)$ and $\sigma^2_{\mathcal{G}_B}(i,j)$ are defined in Eq. (6). When more than one pixel has the same local variance, the embedding order is determined according to their positions in the lexicographic order. Since Ω_{G1} and Ω_{G2} are disjoint, there is no need to record the embedding orders for extraction. The embedding order of the pixels in Ω_{G1} is determined based on the local variance calculated from the pixels in Ω_{G2}. After embedding the hidden data in the pixels of Ω_{G1}, the embedding order of the pixels in Ω_{G2} is determined based on the local variance calculated from the modified pixels in Ω_{G1}. For extraction, first extract the hidden data in Ω_{G2} where embedding order can be retrieved from the modified Ω_{G1}; then extract the hidden data in Ω_{G1} with embedding order retrieved from the recovered Ω_{G2}.

In addition, the embedding order for the R pixels is determined similarly, with local variance of each R pixel $I^R_{mo}(i,j)$ calculated by $\sigma^2_{\mathcal{R}}(i,j) = \frac{1}{4}\sum_{(x,y)\in\Psi_R}[\mathcal{R}_G(x,y) - \overline{\mathcal{R}}(i,j)]^2$, where $\mathcal{R}_G(x,y)$, $\overline{\mathcal{R}}(i,j)$, and Ψ_R are defined in Eq. (9). The embedding order for the B pixels is similar to that for the R pixels. Consequently, a pixel with smaller local variance has higher priority for embedding the hidden data, and the embedding order is based on the sorted local variances.

4.4. The steps of the proposed reversible data hiding algorithm

We now describe the steps of the proposed reversible data hiding algorithm. Since the operations applied to the G, R, and B color pixels are the same, we only describe the operations on G pixels. Similarly, since the embedding and extraction processes for the pixels in Ω_{G1} and Ω_{G2} are the same, we only consider the processes for pixels in Ω_{G1}. Note that the size of the header stream $|\mathbb{H}|$ is known in advance by both the embedding and extraction processes.

The embedding process for all the pixels in Ω_{G1} composes of five steps:

Step 1: Preserve the first $|\mathbb{H}|$ pixels in Ω_{G1}.
Step 2: From the $(|\mathbb{H}|+1)$th pixel to the last pixel in Ω_{G1}, determine the embedding order of the pixels by the sorting strategy described in Section 4.3.
Step 3: Determine an appropriate threshold T by the threshold determination process described in Section 4.2, and then obtain the flag bit stream.
Step 4: Embed the header stream, which records the threshold value T and the sizes of hidden data and flag bit stream, in the first pixels by the LSB replacement method, and obtain the correction bit stream.
Step 5: According to the embedding order of the pixels, embed the correction bit stream, the flag bit stream, and the hidden data into the pixels in Ω_{G1} using the embedding strategy described in Section 4.1.

The extraction process for the pixels in Ω_{G1} consists of four steps:

Step 1: Find all the pixels in Ω_{G1}. Extract the header stream from the first $|\mathbb{H}|$ pixels to obtain the threshold T and the sizes of hidden data and flag bit stream.

Step 2: From the $(|\mathbb{H}|+1)$th pixel to the last pixel in Ω_{G1}, determine the extraction order of the pixels by the method described in Section 4.3.

Step 3: According to the extraction order of the pixels, extract the correction bit stream, the flag bit stream, and the hidden data from the pixels in Ω_{G1} using the strategy described in Section 4.1.

Step 4: Based on the correction bit stream, recover the LSB values of the first $|\mathbb{H}|$ pixels.

Because the human visual system is more sensitive to the *G* channel than the other channels, the number of *G* pixels in a CFA mosaic image is designed to be twice the number of *R* pixels and *B* pixels. Thus, the accuracy of the prediction process for *G* pixels is higher than that for *R* and *B* pixels, implying that embedding a hidden bit in a *G* pixel often results in less distortion than in an *R* pixel or a *B* pixel. For reducing the distortion in a marked image, we first embed the hidden data in the *G* pixels, and then the *R* and *B* pixels. The extraction process is the inverse of the embedding process; that is, first extract the hidden data from *R* and *B* pixels, and then from *G* pixels. The flowchart of the proposed data hiding algorithm for CFA mosaic images is shown in Fig. 11. Empirically, the proposed embedding algorithm yields better quality of the marked images.

5. Application of the proposed algorithm to DTDI and LP mosaic images

We now extend the proposed algorithm to tackle the reversible data hiding in DTDI mosaic images [3] and LP mosaic images [22]. In contrast to mosaic images captured by the Bayer CFA structure, two colors (*G* and *R* or *G* and *B* colors) are

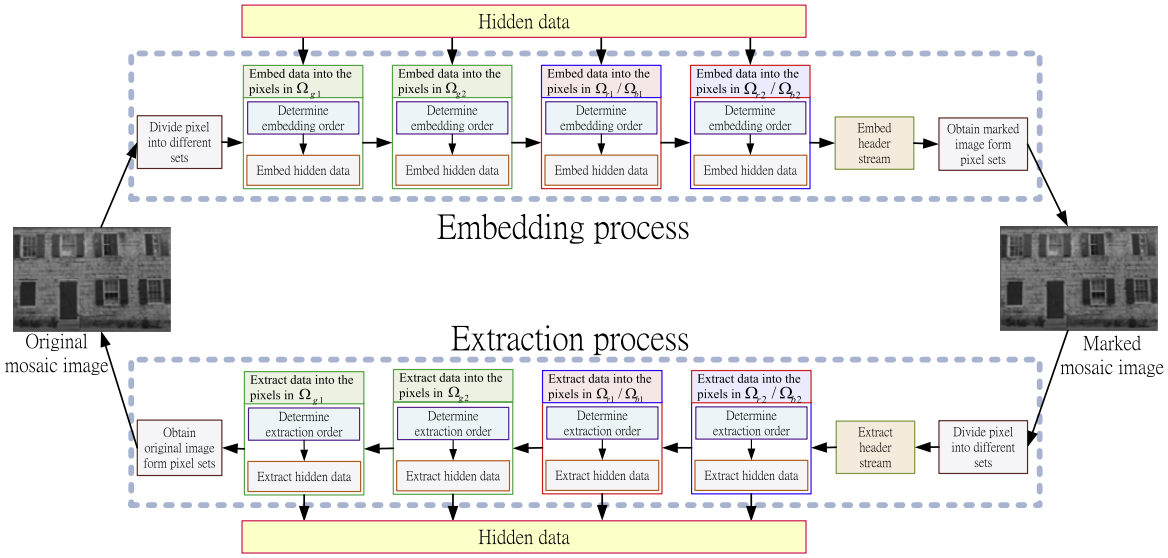

Fig. 11. The flowchart of the proposed algorithm.

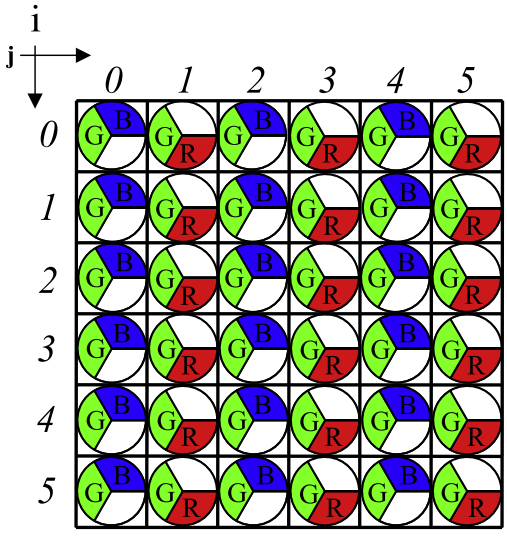

Fig. 12. A DTDI mosaic structure.

measured in each pixel of a DTDI mosaic image. As shown in Fig. 12, denote by $I_D^K(i,j)$ the color value in channel $K(K \in \{R,G,B\})$ of the pixel located at position (i,j) in a DTDI mosaic image of size $M \times N$ with $i \in \{0,1,\ldots,M-1\}$ and $j \in \{0,1,\ldots,N-1\}$. Since a DTDI image can be regarded as the composition of two CFA mosaic images, the input DTDI mosaic image is first decomposed into two CFA mosaic images, denoted by I_{mo_1} and I_{mo_2}, as shown in Fig. 13. Then, apply individually the proposed data hiding algorithm to each decomposed CFA mosaic image. Finally, merge two marked CFA mosaic images to generate the marked DTDI mosaic image. The DTDI decomposition method can be formulated as follows:

$$\begin{cases} I_{mo_1}^G(i,j) = I_D^G(i,j), & \text{if } (i+j) \text{ is even} \\ I_{mo_1}^R(i,j) = I_D^R(i,j), & \text{if } i \text{ is even and } j \text{ is odd} \\ I_{mo_1}^B(i,j) = I_D^B(i,j), & \text{if } i \text{ is odd and } j \text{ is even} \end{cases} \tag{22}$$

$$\begin{cases} I_{mo_2}^G(i,j) = I_D^G(i,j), & \text{if } (i+j) \text{ is odd} \\ I_{mo_2}^R(i,j) = I_D^R(i,j), & \text{if } i \text{ is odd and } j \text{ is odd} \\ I_{mo_2}^B(i,j) = I_D^B(i,j), & \text{if } i \text{ is even and } j \text{ is even.} \end{cases} \tag{23}$$

For applying the proposed data hiding algorithm to an LP mosaic image, first convert an LP mosaic image to a CFA mosaic image by pixel shifting. The LP mosaic structure can be illustrated in Fig. 14. Denote by $I_{lp}^K(i,j)$ the color value in channel $K(K \in \{R,G,B\})$ with $i \in \{0,1,\ldots,M-1\}$ and $j \in \{0,1,\ldots,N-1\}$ of the pixel located at position (i,j) in an LP mosaic image of size $M \times N$. The conversion by pixel shifting can be briefly described as follows.

$$\begin{cases} I_{mo}^G(i,j) = I_{lp}^G(i,j-1), & \text{if } i \bmod 4 = 1 \text{ and } j \text{ is odd} \\ I_{mo}^G(i,j) = I_{lp}^G(i,j+1), & \text{if } i \bmod 4 = 2 \text{ and } j \text{ is even} \\ I_{mo}^B(i,j) = I_{lp}^B(i,j+1), & \text{if } i \bmod 4 = 1 \text{ and } j \text{ is even} \\ I_{mo}^R(i,j) = I_{lp}^R(i,j-1), & \text{if } i \bmod 4 = 2 \text{ and } j \text{ is odd} \\ I_{mo}^C(i,j) = I_{lp}^C(i,j), & \text{otherwise,} \end{cases} \tag{24}$$

where $C \in \{R,G,B\}$. After an LP mosaic image is converted to a CFA mosaic image, apply the proposed data hiding algorithm to the CFA mosaic image. Then, the marked CFA mosaic image is converted back to the marked LP image by the inverse pixel shifting.

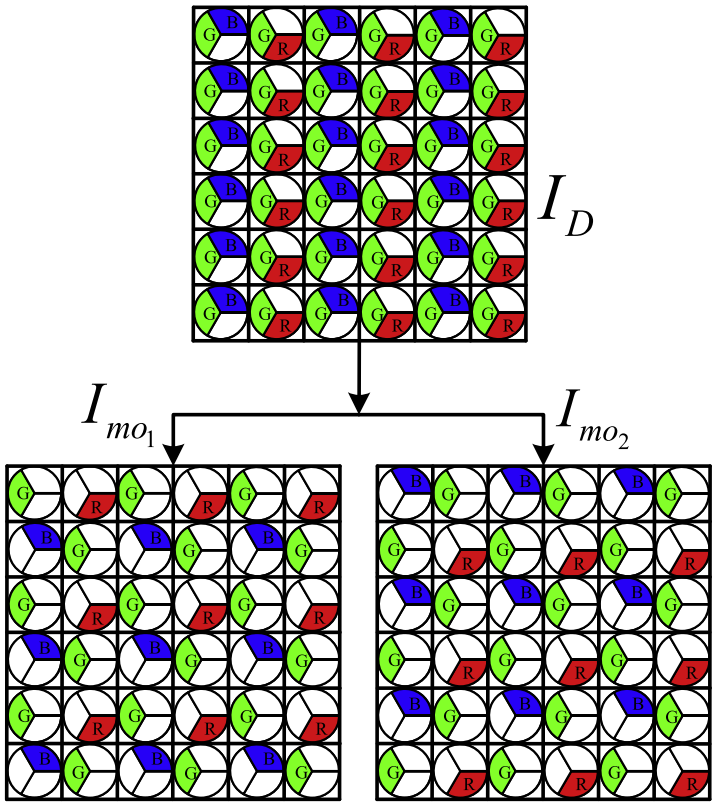

Fig. 13. The DTDI decomposition procedure.

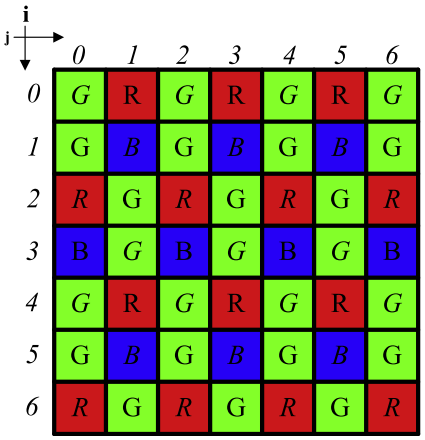

Fig. 14. Lukac and Plataniotis' mosaic structure [22].

6. Experimental results

For comparison, the proposed algorithm was compared with three state-of-the-art reversible data hiding algorithms [29–31] which are based on the prediction error expansion and histogram modification. Since the three compared algorithms were designed to operate in the spatial domain, we partitioned each test mosaic image into four color planes by Eq. (12). Then, three compared algorithms were first applied to each individual color plane and marked color planes were combined to generate the marked mosaic image. The proposed algorithm was directly applied to the input mosaic images to generate the marked mosaic images. The experiments were conducted on the 10 test CFA mosaic images shown in Fig. 15 which were obtained by downsampling the original full color images in the Kodak photo CD [38] and the website [39]. The original sizes of Fig. 15(a)–(h), Fig. 15(i), and Fig. 15(j) were 512×768,1024×1024, and 3456×2304, respectively. All the algorithms were implemented on an IBM compatible computer with an Intel Core 2 Duo CPU 1.83 GHz and a 2 GB RAM. The operating system was MS-Windows XP; the program development environment was Borland C++ Builder 6.0; and the hidden data were generated by the function *rand*() in C++ language.

The comparisons were based on two performance measures, the peak signal-to-noise ratio (PSNR) and the Capacity. The PSNR measures the quality of the marked image whereas the Capacity, measured in bits per pixel (bpp), represents the amount of hidden data. The PSNR of a marked CFA mosaic image can be expressed as

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[I_{mo}(i,j)-I'_{mo}(i,j)]^2}, \tag{25}$$

where $I_{mo}(i,j)$ and $I'_{mo}(i,j)$ denote, respectively, the color values of the pixels at position (i,j) in an input and marked CFA mosaic images of size $M \times N$. Higher values of the PSNR indicate better quality of the marked images. The Capacity of a marked CFA mosaic image of size $M \times N$ is defined as

$$\mathrm{Capacity} = \frac{\#\{\text{hidden bits}\}}{MN}, \tag{26}$$

where #{hidden bits} denotes the number bits with hidden data. Large values of the Capacity indicate larger embedding capacity. For the convenience of comparison, the PSNR is measured for different values of the Capacity for each test image, as shown in Fig. 16. It is clear that the proposed algorithm delivered the marked image with the best quality in terms of the PSNR.

For applying the proposed algorithm to the DTDI mosaic images, downsample the 10 test color images as DTDI mosaic images, shown in Fig. 17. The comparisons were based on two performance measures, PSNR in the DTDI mosaic images, denoted by DPSNR, and the Capacity in the DTDI mosaic images, denoted by DCapacity. The DPSNR of a marked DTDI mosaic image of size $M \times N$ is defined as

$$\mathrm{DPSNR} = 10\log_{10}\frac{255^2}{\frac{1}{2MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\sum_{C\in\Phi}\delta_C(i,j)\left[I_D^C(i,j)-I_D'^C(i,j)\right]^2}, \tag{27}$$

where $\Phi = \{R,G,B\}$, $I_D^K(i,j)$, and $I_D'^K(i,j)$ denote, respectively, the $K(K \in \Phi)$ color values of the pixel at location (i,j) in an input and marked DTDI mosaic images, and

$$\begin{cases} \delta_G(i,j) = 1, \\ \delta_B(i,j) = (j+1) \bmod 2, \\ \delta_R(i,j) = j \bmod 2. \end{cases} \tag{28}$$

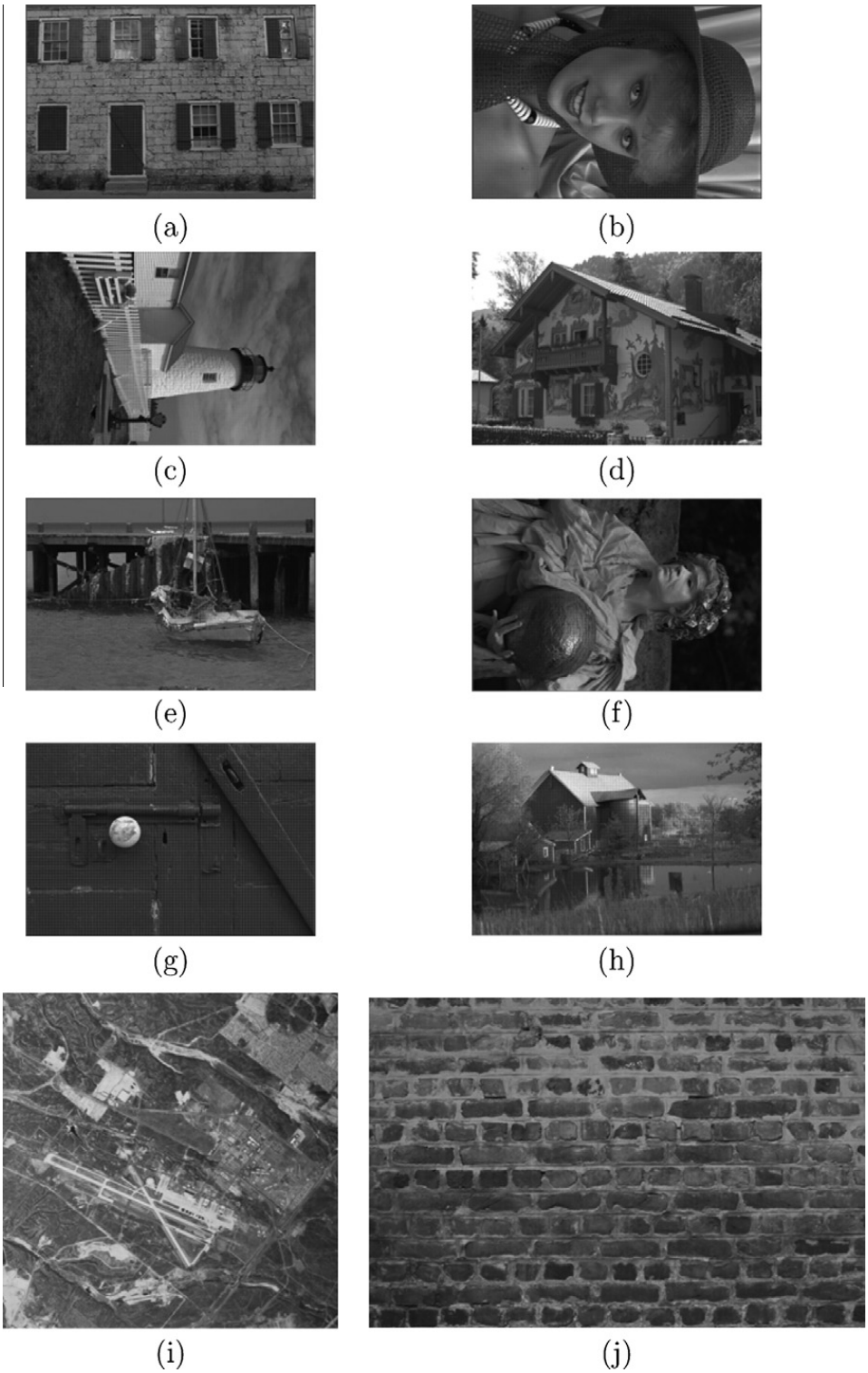

Fig. 15. Ten CFA mosaic images: (a) wall, (b) girl, (c) lighthouse (d) house, (e) boat, (f) carving, (g) door, (h) village, (i) satellite-map, and (j) brick-texture.

Higher values of the DPSNR indicate better quality of the marked images. The DCapacity of a marked DTDI mosaic image of size $M \times N$ can be expressed as Capacity/2 since there exist two channels, for embedding data, in the pixels of a DTDI image. In Fig. 18, plot DPSNR against different values of the DCapacity for each test image. It is clear that the proposed algorithm delivered the marked DTDI mosaic images with the best quality in terms of the DPSNR.

To apply the proposed reversible data hiding algorithm to the LP mosaic images, first convert, by pixel shifting, LP mosaic images to CFA mosaic images. Based on the same 10 test images, Table 1 gives the average, over 10 marked images, PSNR for different values of Capacity. It is clear that the proposed algorithm produced better marked image quality when compared with the other algorithms.

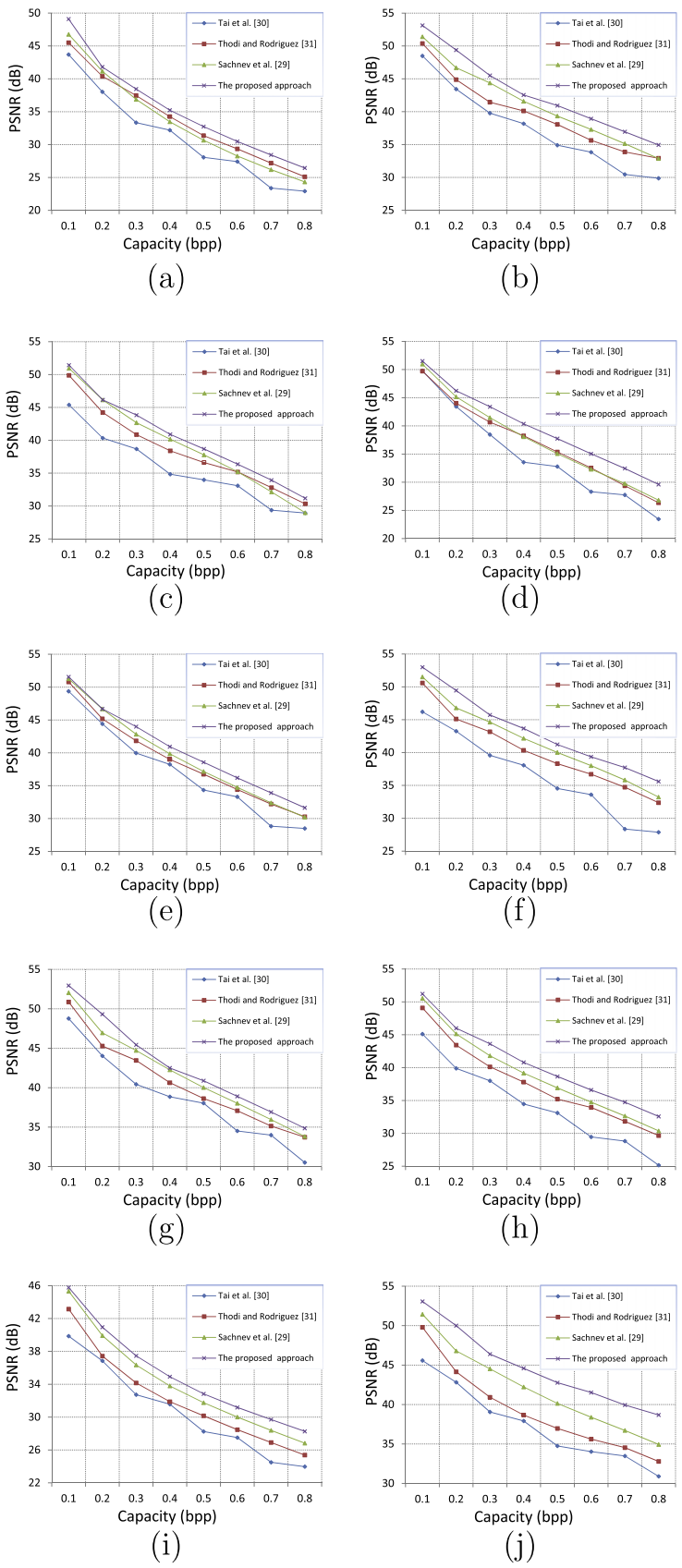

Fig. 16. The PSNR for different values of Capacity for mosaic images (a) wall, (b) girl, (c) lighthouse, (d) house, (e) boat, (f) carving, (g) door, (h) village, (i) satellite-map, and (j) brick-texture.

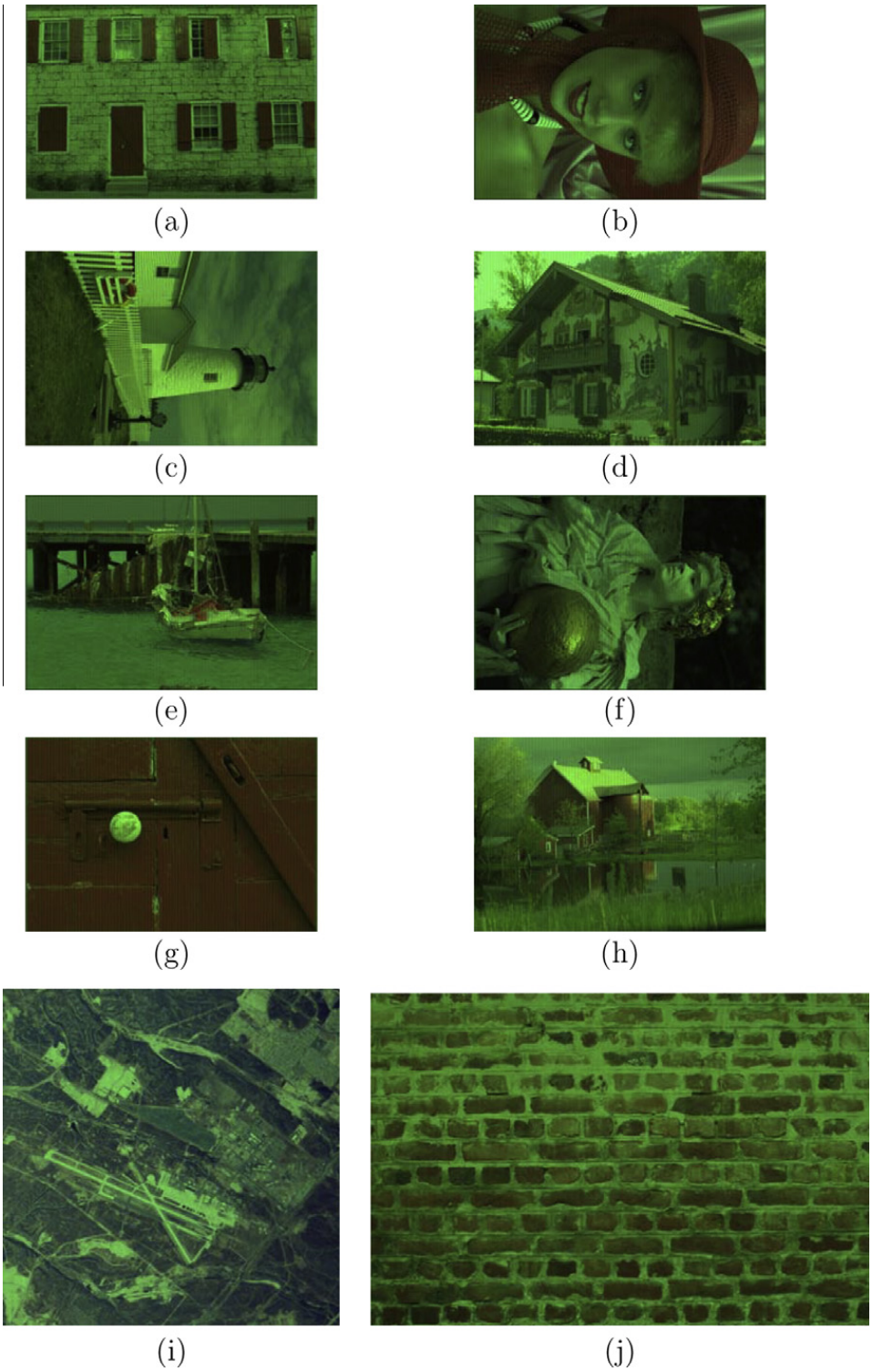

Fig. 17. Ten test DTDI mosaic images: (a) wall, (b) girl, (c) lighthouse, (d) house, (e) boat, (f) carving, (g) door, (h) village, (i) satellite-map and (j) brick-texture.

The threshold used in the proposed algorithm is sequentially increased from zero. To embed more amount of hidden data, more iterations may be required to find the appropriate threshold. Table 2 gives the number of iterations to determine the threshold T for different values of the embedding capacity. It is clear that more iterations are required to determine the threshold when embedding more amount of hidden data. The execution time for each iteration mostly depends on the size of the image. For an image of size 512×768, the average execution time of each iteration is only around 0.16 s, indicating that the threshold determination is quite effective. Similar results appear in the DTDI and LP mosaic images.

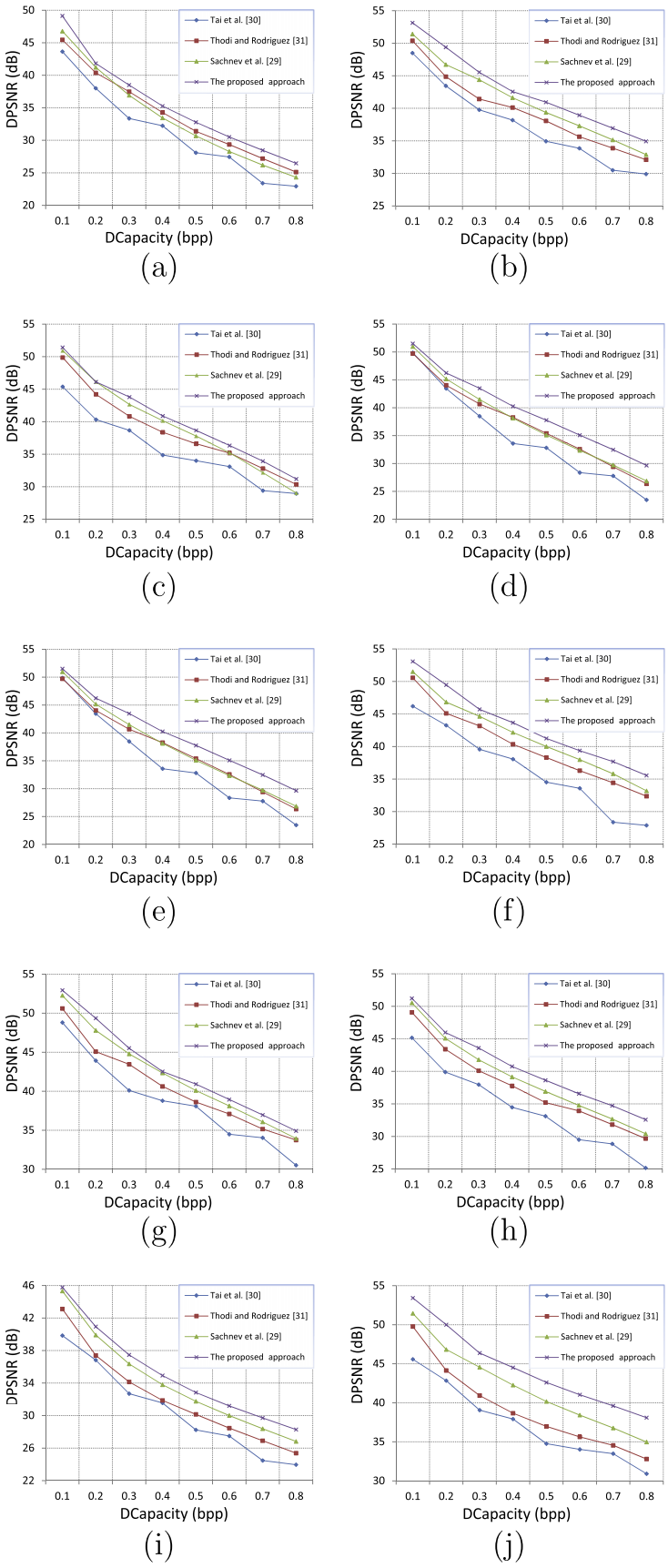

Fig. 18. The DPSNR for different values of DCapacity for DTDI images: (a) wall, (b) girl, (c) lighthouse, (d) house, (e) boat, (f) carving, (g) door, (h) village, (i) satellite-map, and (j) brick-texture.

Table 1
Average PSNR (dB) (over 10 LP mosaic images) for different values of embedding capacity.

Capacity (bpp)							
0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Tai et al.'s algorithm [30]							
46.24	41.65	38.02	35.79	33.29	31.52	28.90	27.22
Thodi and Rodriguez's algorithm [31]							
48.66	43.09	39.65	37.29	35.08	32.97	30.99	28.90
Sachnev et al.'s algorithm [29]							
50.07	44.86	41.47	38.81	36.35	34.09	31.86	29.51
The proposed algorithm							
50.45	45.43	41.84	39.17	36.73	34.53	32.37	30.10

Table 2
Number of iterations to determine the threshold for different values of embedding capacity.

	Capacity (bpp)							
Image	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Wall	0	2	3	5	7	10	14	20
Girl	0	0	1	2	2	3	5	7
Lighthouse	0	1	1	2	3	5	7	12
House	0	1	1	2	4	6	10	16
Boat	0	1	1	2	3	5	7	11
Carving	0	0	1	1	2	3	4	7
Door	0	0	1	2	2	3	5	7
Village	0	1	1	2	3	5	6	10
Satellite-map	1	2	3	5	7	9	11	15
Brick-texture	0	0	1	1	2	2	3	4

Table 3
The effect on the average PSNR (dB) when using sorting strategy to determine the embedding order.

Capacity (bpp)							
0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Without sorting strategy							
50.90	46.23	43.00	40.29	38.32	36.28	34.36	32.30
With sorting strategy							
51.28	46.60	43.40	40.63	38.51	36.46	34.47	32.39

The embedding order of the proposed algorithm is determined by sorting the local variances and embed hidden data first in the pixels with small variances. Table 3 demonstrates the effect on the average PSNR of the marked mosaic images when using the sorting strategy to determine the embedding order. It is clear that under the same embedding capacity, the algorithm with the sorting strategy yielded higher average PSNR than the one without the sorting strategy, especially in the case of low capacity. Since a few pixels are required to embed hidden data in the case of low capacity, only the pixels with very small local variance would be selected, implying that embedding hidden data into the selected pixels would only cause little distortion. Therefore, the effect of sorting strategy on the quality of marked images is clear in this case. However, for the case of high embedding capacity, the effect of sorting strategy becomes less significant since most pixels will be used for embedding the hidden data, indicating that there exists not much room for improvement by the embedding order.

7. Concluding remarks

We have proposed a reversible data hiding algorithm for CFA, DTDI and LP mosaic images. The algorithm differs from existing approaches in that it uses prediction errors in the color difference domain to embed hidden data. Since such errors tend to follow a Laplacian distribution with significantly small variance, the algorithm's embedding capacity is high and it has good quality on the marked images. To the best of our knowledge, this is the first reversible data hiding algorithm designed specifically for CFA, DTDI, and LP mosaic images. The results of experiments on typical CFA, DTDI, and LP test mosaic images demonstrate that under the same embedding capacity, the proposed data hiding algorithm produces better quality of the marked images than the three compared data hiding algorithms [29–31]. The proposed reversible data hiding algorithm is particularly useful for the field of sensitive images, such as military, medical, and artwork images, where the total reconstruction of the original images is imperative.

References

[1] J.M. Barton, Method and Apparatus for Embedding Authentication Information Within Digital Data, US Patent# 5 646 997, 1997.
[2] B.E. Bayer, Color imaging array, US Patent# 3 971 065, 1976.
[3] E. Bodenstorfer, J. Furtler, J. Brodersen, K.J. Mayer, C. Eckel, K. Gravogel, H. Nachtnebel, High speed line-scan camera with digital time delay integration, in: Proceedings of the Electronic Imaging Conference on Real Time Imaging, San Jose, CA, 2007, pp. 64960I-1–64960I-10.
[4] C.C. Chang, W.L. Tai, C.C. Lin, A reversible data hiding scheme based on side match vector quantization, IEEE Trans. Circuits Syst. Video Technol. 16 (2006) 1301–1308.
[5] C.C. Chang, C.C. Lin, C.S. Tseng, W.L. Tai, Reversible hiding in DCT-based compressed images, Inform. Sci. 177 (2007) 2768–2786.
[6] C.C. Chang, T.D. Kieu, W.C. Wu, A lossless data embedding technique by joint neighboring coding, Pattern Recognition 42 (2009) 1597–1603.
[7] C.C. Chang, T.D. Kieu, A reversible data hiding scheme using complementary embedding strategy, Inform. Sci. 180 (2010) 3045–3058.
[8] C.C. Chang, P.Y. Pai, C.M. Yeh, Y.K. Chan, A high payload frequency-based reversible image hiding method, Inform. Sci. 180 (2010) 2286–2298.
[9] K.L. Chung, C.H. Shen, L.C. Chang, A novel SVD- and VQ-based image hiding scheme, Pattern Recognition Lett. 22 (2001) 1051–1058.
[10] K.L. Chung, W.N. Yang, Y.H. Huang, S.T. Wu, Y.C. Hsu, On SVD-based watermarking algorithm, Appl. Math. Comput. 188 (2007) 54–57.
[11] K.L. Chung, W.J. Yang, W.M. Yan, C.C. Wang, Demosaicing of color filter array captured images using gradient edge detection masks and adaptive heterogeneity-projection, IEEE Trans. Image Process. 17 (2008) 2356–2367.
[12] I.J. Cox, M. Miller, J. Bloom, Digital Watermarking, Morgan Kaufman, San Francisco, CA, 2001.
[13] M. Fallahpour, Reversible image data hiding based on gradient adjusted prediction, IEICE Electron. Express 5 (2008) 870–876.
[14] X. Gao, L. An, Y. Yuan, D. Tao, X. Li, Lossless data embedding using generalized statistical quantity histogram, IEEE Trans. Circuits Syst. Video Technol. 21 (2011) 1061–1070.
[15] C.W. Honsinger, P. Jones, M. Rabbani, J.C. Stoffel, Lossless recovery of an original image containing embedded data, US Patent# 6 278 791, 2001.
[16] C.T. Hsu, J.L. Wu, Hidden digital watermarks in images, IEEE Trans. Image Process. 8 (1999) 58–68.
[17] L.H.J. Kamstra, A.M. Heijmans, Reversible data embedding into images using wavelet techniques and sorting, IEEE Trans. Image Process. 14 (2005) 2082–2090.
[18] H.J. Kim, V. Sachnev, Y.Q. Shi, J. Nam, H.G. Choo, A novel difference expansion transform for reversible data embedding, IEEE Trans. Inform. Foren. Security 3 (2008) 456–465.
[19] C.C. Lin, W.L. Tai, C.C. Chang, Multilevel reversible data hiding based on histogram modification of difference images, Pattern Recognition 41 (2008) 3582–3591.
[20] C.S. Lu, H.Y. Mark Liao, Multipurpose watermarking for image authentication and protection, IEEE Trans. Image Process. 10 (2001) 1579–1592.
[21] W. Lu, Y.P. Tan, Color filter array demosaicking: new method and performance measures, IEEE Trans. Image Process. 12 (2003) 1194–1210.
[22] R. Lukac, K.N. Plataniotis, Color filter arrays: design and performance analysis, IEEE Trans. Consumer Electron. 51 (2005) 1260–1267.
[23] R. Lukac, K.N. Plataniotis, D. Hatzinakos, M. Aleksic, A new CFA interpolation framework, Signal Process. 86 (2006) 1559–1579.
[24] R. Lukac, K.N. Plataniotis, Single-sensor camera image compression, IEEE Trans. Consumer Electron. 52 (2006) 299–307.
[25] H. Luo, F.X. Yu, H. Chen, Z.L. Huang, H. Li, P.H. Wang, Reversible data hiding based on block median preservation, Inform. Sci. 181 (2011) 308–328.
[26] S. Mallat, A Wavelet Tour of Signal Processing, second ed., Academic Press, Orlando, FL, 1999.
[27] Y.Q. Ni, N.A. Shi, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol. 16 (2006) 354–362.
[28] S.C. Pei, I.K. Tam, Effective color interpolation in CCD color filter arrays using signal correlation, IEEE Trans. Circuits Syst. Video Technol. 13 (2003) 503–513.
[29] V. Sachnev, H.J. Kim, J. Nam, S. Suresh, Y.Q. Shi, Reversible watermarking algorithm using sorting and prediction, IEEE Trans. Circuits Syst. Video Technol. 19 (2009) 989–999.
[30] W.L. Tai, C.M. Yeh, C.C. Chang, Reversible data hiding based on histogram modification of pixel differences, IEEE Trans. Circuits Syst. Video Technol. 19 (2009) 906–910.
[31] D.M. Thodi, J.J. Rodriguez, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process. 16 (2007) 721–730.
[32] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol. 13 (2003) 90–896.
[33] H.W. Tseng, C.P. Hsieh, Prediction-based reversible data hiding, Inform. Sci. 179 (2009) 2460–2469.
[34] J.X. Wang, Z.M. Lu, A path optional lossless data hiding scheme based on VQ joint neighboring coding, Inform. Sci. 179 (2009) 3332–3348.
[35] M. Weinberger, G. Seroussi, S. Sapiro, LOCO-I: A low complexity, context-based, lossless image compression algorithm, in: Proceedings of the IEEE Data Compression Conference, 1996, pp. 140–149.
[36] X. Wu, L. Zhang, Improvement of color video demosaicking in temporal domain, IEEE Trans. Image Process. 15 (2006) 3138–3151.
[37] L. Zhang, X. Wu, Color demosaicking via directional linear minimum mean square-error interpolation, IEEE Trans. Image Process. 14 (2005) 2167–2178.
[38] [Online]. Available: <http://www.site.uottawa.ca/~edubois/demosaicking/>.
[39] [Online]. Available: <http://140.118.175.164/WJYang/paper/CFARDH/>.