Signal Processing 164 (2019) 386–401

Contents lists available at ScienceDirect

Signal Processing

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

Content-adaptive reversible visible watermarking in encrypted images

Yuanzhi Yao*, Weiming Zhang, Hui Wang, Hang Zhou, Nenghai Yu

Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China

ARTICLE INFO

Article history:
Received 16 February 2019
Revised 26 May 2019
Accepted 24 June 2019
Available online 25 June 2019

Keywords:
Visible watermarking
Reversible data hiding
Image encryption
Just noticeable difference
Data embedding position

ABSTRACT

The reversible visible watermark which serves as the perceptual ownership identifier can be extracted to losslessly recover the original cover media. This paper presents a novel content-adaptive reversible visible watermarking scheme in encrypted images. To achieve the tradeoff between watermark visibility and marked image quality, data embedding positions for accommodating the watermark are adaptively selected using the visual perceptual model before encryption. Due to weak spatial correlation in encrypted images, the data embedding room is vacated before encryption with a traditional reversible data hiding algorithm to contain pixel bits in data embedding positions. Therefore, it is convenient for the data hider to embed the visible watermark in encrypted images by substituting pixel bits in data embedding positions. If the receiver decrypts the marked encrypted image without extracting the embedded watermark, the visibly marked image can be obtained. In addition, if the receiver decrypts the marked encrypted image and extracts the embedded watermark, the original image can be perfectly recovered. Experimental results demonstrate the merits of the proposed scheme in terms of marked image quality, watermark visibility and watermark robustness.

1. Introduction

Visible watermarking is a technique which perceptibly embeds a watermark in the cover digital media to identify the ownership and deter malicious attempts of copyright violations. Visible watermarks can be company logos, ownership descriptions, and personal digital signatures, etc. Compared with invisible watermarks, visible watermarks can present ownership information directly on the marked media. The key desirable characteristic of visible watermarking is that the embedded watermark should not significantly obscure the marked media details beneath it [1,2].

Watermark embedding degrades the cover media quality in general. Reversible data hiding (RDH) serves as a technique which embeds data into the cover digital media so that the embedded data can be extracted to losslessly recover the original cover [3]. Lossless recovery of the original cover is necessary in some application scenarios (e.g., law forensics, historical art imaging, and medical image analysis) where serious concerns about image quality exist. Fortunately, the reversible visible watermarking scheme can be implemented by combining reversible data hiding and visible watermarking.

Many reversible visible image watermarking schemes are proposed in the past [4,6–10,12–14]. Hu and Jeon [4] proposed a bit plane alteration-based scheme for visible watermark embedding, in which partial one-bit pixels in the watermark embedding region are compressed by the JBIG algorithm [5] and substituted with the to-be-embedded watermark. To achieve the reversibility, additional payload which consists of compressed one-bit pixels should be embedded in the marked image. The watermark embedding capacity is controlled by data compression efficiency in this scheme [4]. Yip et al. [6] presented two reversible visible watermarking algorithms based on pixel value matching and pixel position shift. In [7,8], reversible visible watermarking schemes using one-to-one mapping of image pixels are elaborated. Yang et al. [9] proposed to reveal the reversible visible watermark through adaptively adjusting the pixel values and embed the reconstruction packet to restore the original cover image. Chen et al. [10] proposed to reversibly embed the visible watermark using the conventional difference expansion technique [11]. Yang et al. [12] improved the scheme in [10] to reduce the number of overflow/underflow marked pixels whose values are larger than 255 or less than 0. In the scheme proposed by Mohammad et al. [13], pixel circular shift operation is conducted to reversibly embed the visible watermark in the block truncation coding-compressed image. The watermark can be extracted according to the parity of the bit plane. Lin et al. [14] achieved the reversible visible watermarking scheme in DCT domain. However, the original cover image can be

* Corresponding author.
E-mail addresses: yaoyz@ustc.edu.cn (Y. Yao), zhangwm@ustc.edu.cn (W. Zhang), whglory@mail.ustc.edu.cn (H. Wang), zh2991@mail.ustc.edu.cn (H. Zhou), ynh@ustc.edu.cn (N. Yu).

https://doi.org/10.1016/j.sigpro.2019.06.034
0165-1684/

recovered only in the case that the original watermark image is obtained.

With the prosperity of cloud computing and mobile network, signal processing in the encrypted domain has gained increasing research interests, such as feature extraction [15], image compression [16], and data hiding [17,18]. Consequently, the research on reversible data hiding in encrypted images driven by the needs from cloud platforms and privacy preservation has attracted considerable attention [19–25]. Because image authentication [26–28] serves as the important application scenario of data hiding, the data hider can embed additional authentication data into encrypted images for access control and media annotation without leaking the privacy of the image content owner. Most reversible data hiding schemes in encrypted images aim to invisibly embed additional data. In some application scenarios, reversible visible watermarking in encrypted images which can convey ownership information directly on the marked media is also desirable. However, the open innovation literature has so far witnessed few attempts to explore this subject. In [29], Zhang et al. proposed a reversible visible watermarking scheme in encrypted images using wet paper codes [30]. The exclusive-or operation is applied for image encryption in this scheme. In addition, partial encrypted data corresponding to black pixels of the binary watermark image should be modified to insert the visible watermark and contain some payload for image recovery. The watermark is invisible in the encrypted domain although it is embedded in the encrypted image. After direct decryption operation, the embedded watermark can be visible.

Generally, reversible visible watermarking in encrypted images requires that the embedded watermark is visible yet should not significantly obscure the marked image details beneath it. These two requirements usually conflict with each other. How to obtain the tradeoff between these two conflicting requirements is the key problem of reversible visible watermarking. If the embedded watermark energy is increased to enhance its visibility, image quality degradation will be severer and vice versa. As discussed, many reversible visible image watermarking schemes have been proposed in the past few years. However, the basic issues corresponding to watermark visibility and marked image quality have not been resolved. Hence, new reversible visible watermarking schemes in encrypted images which are capable of achieving the tradeoff between watermark visibility and marked image quality should be sought.

We propose a novel content-adaptive reversible visible watermarking scheme in encrypted images in this paper. In our proposed scheme, we concentrate on adaptively selecting data embedding positions for accommodating the watermark using the visual perceptual model before encryption to achieve the tradeoff between watermark visibility and marked image quality. Considering the weak spatial correlation in encrypted images, the data embedding room is vacated before encryption with a traditional reversible data hiding algorithm to carry pixel bits in data embedding positions. Therefore, the visible watermark can be embedded in encrypted images by substituting pixel bits in data embedding positions.

In conclusion, the highlights of this paper can be summarized as follows.

- Data embedding positions for accommodating the watermark can be adaptively selected to achieve the tradeoff between watermark visibility and marked image quality.
- A novel framework for reversible visible watermarking in encrypted images is presented.
- The original cover image can be losslessly recovered after image decryption and watermark extraction.

The remainder of this paper is organized as follows. The scheme for content-adaptive reversible visible watermarking in encrypted images is elaborated in Section 2 followed by some implementation issues in Section 3. The experimental results and analysis are presented in Section 4. Finally, Section 5 concludes the paper.

2. Proposed scheme for content-adaptive reversible visible watermarking in encrypted images

In this section, the scheme for content-adaptive reversible visible watermarking in encrypted images is elaborated. The framework of the proposed scheme is shown in Fig. 1. The scheme is composed of four main components: data embedding position selection, encrypted image generation, watermark embedding in encrypted image, and watermark extraction and image recovery. To achieve the tradeoff between watermark visibility and marked image quality, the key issue is selecting data embedding positions

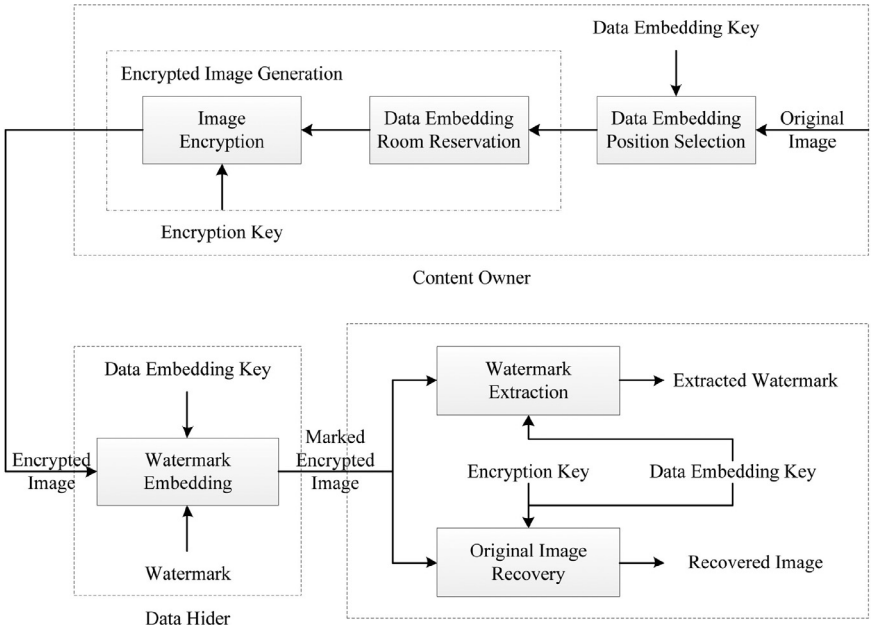


Fig. 1. Framework of proposed content-adaptive reversible visible watermarking scheme.

for accommodating the watermark based on image content. These data embedding positions and corresponding pixel bits need to be saved as the side information. Since losslessly vacating room in the encrypted image is relatively difficult, it is necessary to reserve room prior to image encryption for containing the side information. Traditional reversible data hiding algorithms can be applied for reserving room before image encryption. Therefore, the data hider (e.g., a database manager or a cloud server) can easily embed the visible watermark in the encrypted image by substituting pixel bits in data embedding positions. After image decryption and watermark extraction, the original image can be perfectly recovered.

2.1. Data embedding position selection

In visible watermarking, a to-be-embedded watermark which can be a binary image is inserted perceptibly into a cover image so that the watermark is visible to the human visual system. Visible watermark embedding is essentially substituting pixel bits in different bit planes of the cover image with corresponding watermark information. Generally, the embedded watermark should be visible yet cannot significantly obscure the marked image details beneath it. However, these two requirements usually conflict with each other. To address the above problem, visible watermark embedding can be modeled as data embedding position selection problem. This motivates us to consider the human visual perceptual model and the cover image content to achieve a tradeoff between these conflicting requirements.

2.1.1. Visual perceptual model

The perceptual characteristics of human visual system (HVS) play an important role in many practical applications, such as digital watermarking [9], image quality assessment [31], and image/video coding [32]. Locating the perceptual image region is a key issue for embedding the visible watermark. The human visual system (HVS) can only sense the image content change which is larger than a certain threshold. This threshold is usually termed as just noticeable difference (JND) which describes the visibility of the HVS on visual contents. Some efficient JND models have been proposed during the past decade [33–35].

The HVS is good at summarizing rules of an input scene and is highly adapted to extract the repeated visual contents as the pattern [35]. In the JND model proposed by Wu et al. [35], both pattern complexity and luminance contrast are considered to deduce a novel spatial masking estimation function. Combining with the luminance adaptation, a pixel-wise JND estimation model which performs consistently with the visual perception is described by

$$T_{\mathrm{JND}}(p_{i,j}) = L_A(p_{i,j}) + M_S(p_{i,j}) - \gamma \cdot \min\{L_A(p_{i,j}), M_S(p_{i,j})\} \quad (1)$$

where $p_{i,j}$ is the pixel at (i, j) in a given image and γ is the gain reduction parameter determined by the overlapping between $L_A(p_{i,j})$ and $M_S(p_{i,j})$. In Eq. (1), the visibility threshold of the luminance adaptation $L_A(p_{i,j})$ is modeled as

$$L_A(p_{i,j}) = \begin{cases} 17 \cdot \left(1 - \sqrt{\dfrac{B(p_{i,j})}{127}}\right) & \text{if } B(p_{i,j}) < 127 \\ \dfrac{3}{128} \cdot (B(p_{i,j}) - 127) + 3 & \text{if } B(p_{i,j}) \geq 127 \end{cases} \quad (2)$$

where $B(p_{i,j})$ is the background luminance which is calculated as the mean luminance value of a surrounding region. In Eq. (1), the total spatial masking effect $M_S(p_{i,j})$ which combines the pattern masking effect $M_P(p_{i,j})$ and the contrast masking effect $M_C(p_{i,j})$ is calculated as

$$M_S(p_{i,j}) = \max\{M_P(p_{i,j}), M_C(p_{i,j})\} \quad (3)$$

In view of space limitation, refer to Wu et al. [35] for the definitions of $B(p_{i,j})$, $M_P(p_{i,j})$, and $M_C(p_{i,j})$. To help understand the above visual perceptual model, the just noticeable difference (JND) demonstration using the JND estimation model in Eq. (1) is shown in Fig. 2, where the standard test image Lena [36] with size of 512×512 is considered.

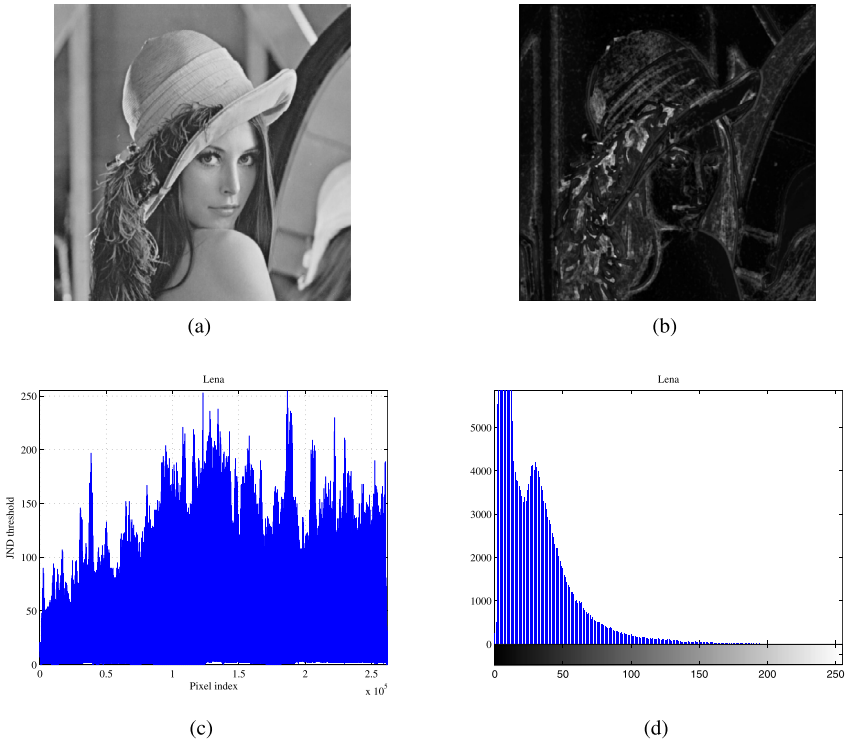


Fig. 2. Just noticeable difference (JND) demonstration for image Lena. (a) Image Lena, (b) JND map, (c) JND threshold versus pixel index, and (d) JND threshold histogram.

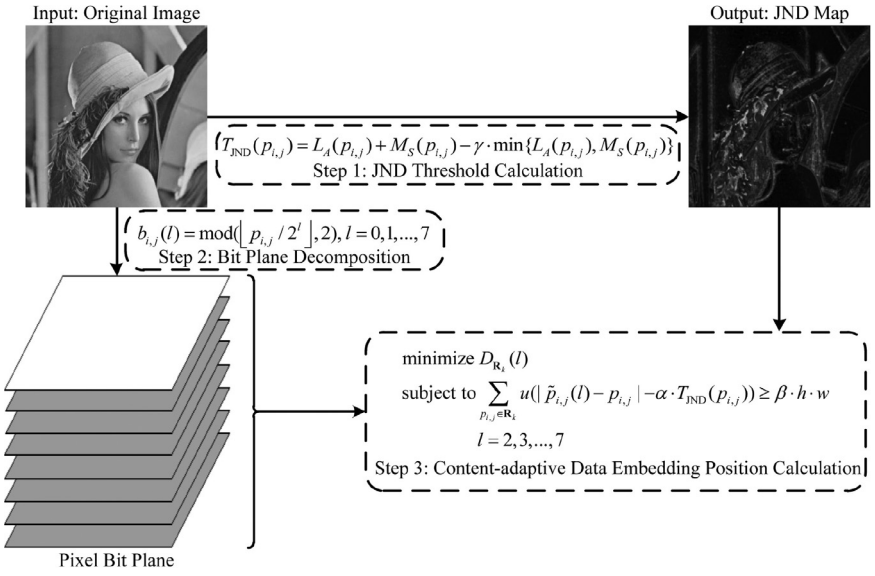


Fig. 3. Illustration of data embedding position calculation strategy.

2.1.2. Data embedding position calculation strategy

Data embedding positions denote bit planes of the original cover image **C** which need to be substituted with corresponding watermark information. Data embedding positions directly determine the change of pixels and affect the watermark visibility. In most existing visible image watermarking schemes, data embedding positions are not adapted to the image content. It means that the same data embedding position is applied to all blocks in a cover image. Constant data embedding position is not optimal because the differences of JND thresholds among the blocks are not considered. Moreover, how to balance watermark visibility and marked image quality is another important issue which needs to be taken into account. To target these two problems, we propose the content-adaptive data embedding position calculation strategy by incorporating the visual perceptual model. Fig. 3 illustrates the data embedding position calculation strategy.

Considering a cover image with the size of $H \times W$, we divide it into non-overlapping blocks which are denoted by

$$\Omega_{\mathbf{R}} = \left\{ \mathbf{R}_k \middle| k = 1, 2, \ldots, \left\lfloor \frac{H}{h} \right\rfloor \cdot \left\lfloor \frac{W}{w} \right\rfloor \right\} \tag{4}$$

where $h \times w$ is the block size and $\lfloor \cdot \rfloor$ is the flooring operator. The watermark embedding region which often has a much smaller size than the cover image is determined by the content owner. We denote (o_y, o_x) as the top left coordinate of the watermark embedding region and assume that the size of the binary watermark image **W** is $M \times N$. Therefore, the watermark embedding region can be denoted by $\Omega_{\mathbf{W}} = \{(i, j) | o_y \leq i \leq o_y + M - 1, o_x \leq j \leq o_x + N - 1\}$. However, the to-be-embedded binary watermark image may not be accessible for the content owner. Assuming that the pixel distribution in the binary watermark image **W** is uniform, the data pre-embedding operation should be conducted to calculate the data embedding position. Given the pixel $p_{i,j}$ at (i, j) in the original cover image **C**, we can estimate the marked pixel $\tilde{p}_{i,j}(l)$ after embedding $\delta_{y,x} \in \{0, 1\}$ into $p_{i,j}$ using Eq. (5).[1]

$$\tilde{p}_{i,j}(l) = \begin{cases} p_{i,j} - b_{i,j}(l) \cdot 2^l + \delta_{y,x} \cdot 2^l & \text{if } r_{i,j} = 1 \\ p_{i,j} - b_{i,j}(l-1) \cdot 2^{l-1} + \delta_{y,x} \cdot 2^{l-1} & \text{if } r_{i,j} = 0 \end{cases} \tag{5}$$

[1] The coordinates of two points (i, j) and (y, x) in Eq. (5) satisfy $i = o_y + y - 1$, $j = o_x + x - 1$.

where

$$b_{i,j}(l) = \mathrm{mod}(\lfloor p_{i,j} / 2^l \rfloor, 2), \quad l = 2, 3, \ldots, 7$$

In Eq. (5), l is the data embedding position. Total $\lceil \log_2 7 \rceil = 3$ bits are needed to record each certain data embedding position. The pixel bit $b_{i,j}(l)$ which is corresponding to the data embedding position l should be saved. In order to enhance security, a pseudo random number generator which is controlled by the data embedding key is introduced to generate $r_{i,j} \in \{0, 1\}$ which follows uniform distribution. Considering the k-th block $\mathbf{R}_k$, its visual distortion caused by watermark embedding can be estimated by

$$D_{\mathbf{R}_k}(l) = \frac{1}{h \times w} \sum_{p_{i,j} \in \mathbf{R}_k} \left(\tilde{p}_{i,j}(l) - p_{i,j} \right)^2 \tag{6}$$

Data embedding positions affect the watermark visibility and the marked image quality. To balance these two conflicting factors, we introduce the content-adaptive data embedding position calculation strategy to optimize the data embedding position l_k of the k-th cover image block $\mathbf{R}_k$ in Eq. (7).

$$\begin{aligned} &\text{minimize} && D_{\mathbf{R}_k}(l) \\ &\text{subject to} && \textstyle\sum_{p_{i,j} \in \mathbf{R}_k} u(|\tilde{p}_{i,j}(l) - p_{i,j}| - \alpha \cdot T_{\mathrm{JND}}(p_{i,j})) \geq \beta \cdot h \cdot w \\ & && l = 2, 3, \ldots, 7 \end{aligned} \tag{7}$$

where

$$u(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

In Eq. (7), α and β are tuning parameters. The optimal data embedding position set $L = \{l_k | k = 1, 2, \ldots, \lceil M/h \rceil \cdot \lceil N/w \rceil\}$ can be constructed using Eq. (7). In order to obtain the side information, optimal data embedding positions l_k and corresponding pixel bits $b_{i,j}(l_k)$ should be saved as the side information sequence S_1. The length of S_1 which equals $(3 \cdot \lceil M/h \rceil \cdot \lceil N/w \rceil + M \cdot N)$ bits can be determined as follows.

- The optimal data embedding positions l_k which are represented by $3 \cdot \lceil M/h \rceil \cdot \lceil N/w \rceil$ bits.
- The pixel bits $b_{i,j}(l_k)$ corresponding to optimal data embedding positions which are represented by $M \cdot N$ bits.

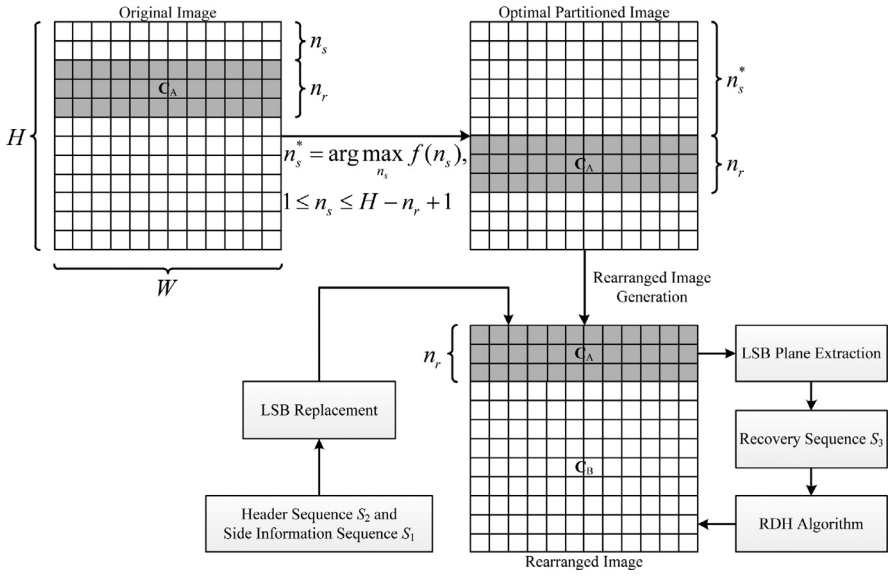


Fig. 4. Illustration of data embedding room reservation.

2.2. Encrypted image generation

After data embedding position selection, optimal data embedding positions and corresponding pixel bits have been saved as the side information. It is necessary to reserve room prior to image encryption for containing the side information because losslessly vacating room in the encrypted image is relatively difficult. Generating the encrypted image can be divided into two steps which are data embedding room reservation and image encryption respectively.

2.2.1. Data embedding room reservation

Fig. 4 illustrates the process of data embedding room reservation. In order to reserve room prior to image encryption, the original cover image $\mathbf{C}$ should be partitioned into two parts $\mathbf{C}_A$ and $\mathbf{C}_B$. The least significant bits of $\mathbf{C}_A$ are self-embedded into $\mathbf{C}_B$ with classical reversible data hiding (RDH) algorithms like [37,38]. $\mathbf{C}_B$ should be smoother for improving data self-embedding performance. Suppose that the original cover image $\mathbf{C}$ is in uncompressed format and each pixel grayscale value whose range is [0,255] is represented by 8 bits. Firstly, the content owner iteratively constructs $\mathbf{C}_A$ by choosing several overlapping slices along rows from the original cover image $\mathbf{C}$. Each candidate slice which is composed of pixels is overlapped by pervious and/or subsequent slices along the rows. Each slice consists of n_r rows and n_r is determined by the side information sequence S_1 and the header sequence S_2. n_r can be calculated by

$$n_r = \left\lceil \frac{|S_1| + |S_2|}{W} \right\rceil = \left\lceil \frac{3 \cdot \lceil M/h \rceil \cdot \lceil N/w \rceil + M \cdot N + 4 \cdot \lceil \log_2 H \rceil + 2 \cdot \lceil \log_2 W \rceil}{W} \right\rceil \tag{8}$$

where $\lceil \cdot \rceil$ is the ceiling operator and $|\cdot|$ is the cardinality of a set. The header sequence S_2 tells the data hider the number of rows in $\mathbf{C}_A$ and the starting row of $\mathbf{C}_A$. The length of S_2 which equals $(4 \cdot \lceil \log_2 H \rceil + 2 \cdot \lceil \log_2 W \rceil)$ bits can be determined as follows.

- The number of rows n_r and the starting row n_s of $\mathbf{C}_A$ which are represented by $2 \cdot \lceil \log_2 H \rceil$ bits.
- The top left coordinate (o_y, o_x) of the watermark embedding region which is represented by $(\lceil \log_2 H \rceil + \lceil \log_2 W \rceil)$ bits.
- The size of the binary watermark image $\mathbf{W}$ which is represented by $(\lceil \log_2 H \rceil + \lceil \log_2 W \rceil)$ bits.

The starting row of $\mathbf{C}_A$ is denoted by n_s. The number of candidate slices can be computed by $n_c = H - n_r + 1$. We adopt the function $f(n_s)$ which is depicted in Eq. (9) to measure the texture complexity of each candidate slice.

$$f(n_s) = \sum_{i=n_s+1}^{n_s+n_r-2} \sum_{j=2}^{W-1} \left| p_{i,j} - \frac{p_{i,j-1} + p_{i+1,j} + p_{i,j+1} + p_{i-1,j}}{4} \right| \tag{9}$$

It is obvious that the slice with higher $f(n_s)$ contains relatively more complex texture. Therefore, the image content owner can iteratively construct $\mathbf{C}_A$ by selecting slices until the slice texture complexity reaches the highest $f(n_s^*)$ where the optimal n_s^* can be determined by

$$n_s^* = \arg\max_{n_s} f(n_s), \quad 1 \le n_s \le H - n_r + 1 \tag{10}$$

Once $\mathbf{C}_A$ is constructed, $\mathbf{C}_A$ should be concatenated by the other part $\mathbf{C}_B$ with smoother areas to generate the rearranged image $\mathbf{C}'$. Least significant bits of $\mathbf{C}_A$ are saved as the recovery sequence S_3 whose length equals $n_r \cdot W$ bits and self-embedded into $\mathbf{C}_B$ with a reversible data hiding algorithm. This process does not depend on any specific reversible data hiding algorithm. We simplify the method in [37] to depict the process of data self-embedding.

Pixels in $\mathbf{C}_B$ are categorized into two sets: cross pixel $u_{i,j}$ with its indices satisfying $\mathrm{mod}((i+j), 2) = 0$ and dot pixel $v_{i,j}$ with its indices satisfying $\mathrm{mod}((i+j), 2) = 1$. The process of data self-embedding consists of two rounds. In the first round, cross pixels are used for data self-embedding and dot pixels are used for prediction, while in the second round, dot pixels are used for data self-embedding and cross pixels are used for prediction. We can take the first round for illustration. For each cross pixel $u_{i,j}$, its predicted value $u'_{i,j}$ is computed by averaging its four nearest dot pixels as shown in Eq. (11).

$$u'_{i,j} = \left\lfloor \frac{v_{i,j-1} + v_{i+1,j} + v_{i,j+1} + v_{i-1,j}}{4} \right\rfloor \tag{11}$$

We can obtain the prediction errors $d_{i,j}$ by subtracting the predicted value $u'_{i,j}$ from the original pixel $u_{i,j}$ as follows.

$$d_{i,j} = u_{i,j} - u'_{i,j} \tag{12}$$

Afterwards, least significant bits b_k of $\mathbf{C}_A$ are embedded into the prediction errors $d_{i,j}$ through prediction error expansion and histogram shifting techniques which are illustrated in Eq. (13).

$$D_{i,j} = \begin{cases} 2d_{i,j} + b_k & \text{if } d_{i,j} \in [T_n, T_p] \\ d_{i,j} + T_p + 1 & \text{if } d_{i,j} > T_p \text{ and } T_p \geq 0 \\ d_{i,j} + T_n & \text{if } d_{i,j} < T_n \text{ and } T_n < 0 \end{cases} \tag{13}$$

where T_p and T_n are positive threshold and negative threshold respectively for controlling prediction error expansion. After data self-embedding, the original pixel $u_{i,j}$ is modified to $U_{i,j}$ as follows.

$$U_{i,j} = D_{i,j} + u'_{i,j} \tag{14}$$

Data self-embedding in dot pixels is similar to data self-embedding in cross pixels. After two rounds are completed, we can obtain the image $\mathbf{C}''$ which contains the least significant bits of $\mathbf{C}_A$. Details of data embedding and extraction procedures can be seen in [37].

2.2.2. Image encryption

The bits of each pixel $P_{i,j}$ in the image $\mathbf{C}''$ are denoted as $b_{i,j,7}, b_{i,j,6}, \ldots, b_{i,j,0}$ where (i, j) indicates the pixel coordinate so that

$$b_{i,j,k} = \text{mod}(\lfloor P_{i,j}/2^k \rfloor, 2), \quad k = 0, 1, \ldots, 7 \tag{15}$$

and

$$P_{i,j} = \sum_{k=0}^{7} b_{i,j,k} \cdot 2^k \tag{16}$$

In the encryption phase, the encrypted bit $\hat{b}_{i,j,k}$ can be obtained after the exclusive-or operation

$$\hat{b}_{i,j,k} = b_{i,j,k} \oplus r_{i,j,k} \tag{17}$$

where the pseudo random number $r_{i,j,k} \in \{0, 1\}$ is generated with the encryption key using a standard stream cipher. Finally, we replace the least significant bits of first $(3 \cdot \lceil M/h \rceil \cdot \lceil N/w \rceil + M \cdot N + 4 \cdot \lceil \log_2 H \rceil + 2 \cdot \lceil \log_2 W \rceil)$ pixels in the encrypted version of $\mathbf{C}_A$ with the header sequence S_2 and the side information sequence S_1.[2] After the encrypted image $\mathbf{E}$ is generated, the data hider or the third party cannot access the image content without the encryption key so that the privacy of the image content owner is preserved. The flow chart of image encryption is summarized in Fig. 5.

2.3. Watermark embedding in encrypted image

Once the data hider acquires the encrypted image $\mathbf{E}$, he/she can adaptively embed the watermark into it without getting access to the original image. The watermark embedding process starts with locating the encrypted version of $\mathbf{C}_A$ which is denoted by $\mathbf{E}_A$. The data hider firstly extracts the header sequence S_2 from $\mathbf{E}_A$ to obtain the number of rows n_r. Secondly, the data hider extracts the side information sequence S_1 according to n_r in $\mathbf{E}_A$ for obtaining the optimal data embedding positions l_k where $k = 1, 2, \ldots, \lceil \frac{M}{h} \rceil \cdot \lceil \frac{N}{w} \rceil$. Finally, given the pixel $E_{i,j}$ at (i, j) in the encrypted image $\mathbf{E}$ and the watermark pixel $\omega_{y,x} \in \{0, 1\}$ at (y, x) in the binary watermark image $\mathbf{W}$, we can generate the marked encrypted pixel $\tilde{E}_{i,j}(l_k)$ after embedding $\omega_{y,x}$ into $p_{i,j}$ using Eq. (18).

$$\tilde{E}_{i,j}(l_k) = \begin{cases} E_{i,j} - b_{i,j}(l_k) \cdot 2^{l_k} + \omega_{y,x} \cdot 2^{l_k} & \text{if } r_{i,j} = 1 \\ E_{i,j} - b_{i,j}(l_k - 1) \cdot 2^{l_k - 1} + \omega_{y,x} \cdot 2^{l_k - 1} & \text{if } r_{i,j} = 0 \end{cases} \tag{18}$$

[2] To deter malicious watermark removal, the header sequence S_2 and the side information sequence S_1 can be encrypted using the data embedding key.

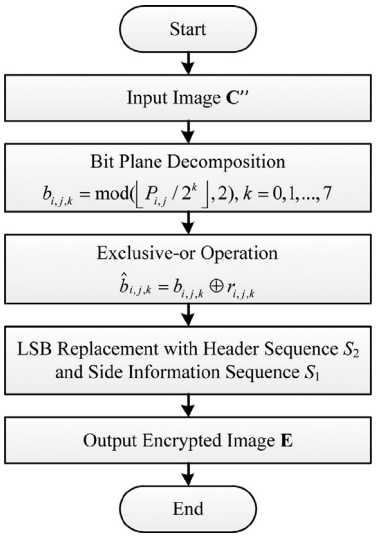


Fig. 5. Flow chart of image encryption.

where

$$b_{i,j}(l_k) = \text{mod}(\lfloor E_{i,j}/2^{l_k} \rfloor, 2), \quad l_k = 2, 3, \ldots, 7$$

In Eq. (18), $r_{i,j} \in \{0, 1\}$ is the pseudo random number generated by the data embedding key. The marked encrypted image $\tilde{\mathbf{E}}$ can be obtained using Eq. (18). Since the optimal data embedding positions l_k have been obtained through the data embedding position calculation strategy, the tradeoff between watermark visibility and marked image quality can be achieved after image decryption.

2.4. Watermark extraction and image recovery

The receiver may download the image from the cloud server and view the decrypted image using the encryption key. The decrypted image still contains the embedded visible watermark. The following steps should be performed to form the marked decrypted image $\tilde{\mathbf{C}}$ which is composed of $\tilde{\mathbf{C}}_A$ and $\tilde{\mathbf{C}}_B$.

- *Step 1*: Extract the header sequence S_2 and the side information sequence S_1 from $\tilde{\mathbf{E}}_A$ to obtain the number of rows n_r and the starting row n_s of $\tilde{\mathbf{E}}_A$.
- *Step 2*: Decrypt the marked encrypted image $\tilde{\mathbf{E}}$ with the encryption key except the least significant bits of $\tilde{\mathbf{E}}_A$ and the pixel bits corresponding to optimal data embedding positions l_k. Due to the symmetry of the bitwise exclusive-or operation, the decryption operation is symmetric to the encryption operation.
- *Step 3*: Rearrange $\tilde{\mathbf{C}}_A$ and $\tilde{\mathbf{C}}_B$ according to the header sequence S_2 to obtain the marked decrypted image $\tilde{\mathbf{C}}$.

Compared with the original cover image $\mathbf{C}$, the distortion of the marked decrypted image $\tilde{\mathbf{C}}$ is introduced by data self-embedding, modifying the least significant bits of $\mathbf{E}_A$ and watermark embedding. The data self-embedding distortion $\rho_1(\lambda, S_3)$ is determined by the adopted reversible data hiding algorithm λ and the recovery sequence S_3. Assuming that the distribution of the least significant bits of $\mathbf{E}_A$ is uniform, the distortion $\rho_2(S_1, S_2)$ introduced by modifying the least significant bits of $\mathbf{E}_A$ can be estimated by $\frac{n_r \cdot W}{2}$. Moreover, according to Eq. (18), the watermark embedding distortion $\rho_3(L, \mathbf{W})$ can be estimated by

$$\rho_3(L, \mathbf{W}) \approx \frac{1}{2} \cdot \sum_{i=o_y}^{o_y+M-1} \sum_{j=o_x}^{o_x+N-1} \left[(\omega_{y,x} - b_{i,j}(l_k))^2 \cdot 4^{l_k} \right.$$

$$+ (\omega_{y,x} - b_{i,j}(l_k - 1))^2 \cdot 4^{l_k - 1}\Big] \le 2^{14} \cdot M \cdot N \quad (19)$$

where L is the optimal data embedding position set. The content owner can recover the marked decrypted image $\tilde{\mathbf{C}}$ to generate the original cover image $\mathbf{C}$ as follows.

- *Step 1*: Extract the header sequence S_2 and the side information sequence S_1 from $\tilde{\mathbf{C}}$ to obtain the number of rows n_r of $\tilde{\mathbf{C}}_{\mathrm{A}}$, the starting row n_s of $\tilde{\mathbf{C}}_{\mathrm{A}}$, the optimal data embedding positions l_k, and corresponding pixel bits $b_{i,j}(l_k)$.
- *Step 2*: Extract the recovery sequence S_3 from $\tilde{\mathbf{C}}_{\mathrm{B}}$ with the reversible data hiding algorithm [37].
- *Step 3*: Replace the least significant bits of $\tilde{\mathbf{C}}_{\mathrm{A}}$ with the recovery sequence S_3.
- *Step 4*: Extract the embedded visible watermark using the optimal data embedding positions l_k, the corresponding pixel bits $b_{i,j}(l_k)$, and the data embedding key in Eq. (20) to obtain the original cover image $\mathbf{C}$.

Given the marked pixel $\tilde{p}_{i,j}$ at (i, j) in the marked decrypted image $\tilde{\mathbf{C}}$, the optimal data embedding positions l_k, and corresponding pixel bits $b_{i,j}(l_k)$, we can extract the binary watermark image $\mathbf{W}$ and recover the original cover image $\mathbf{C}$ with the data embedding key as depicted in Eq. (20), where $r_{i,j}$ is the pseudo random number generated by the data embedding key.

$$p_{i,j}(l_k) = \begin{cases} \tilde{p}_{i,j} - \omega_{y,x} \cdot 2^{l_k} + b_{i,j}(l_k) \cdot 2^{l_k} & \text{if } r_{i,j} = 1 \\ \tilde{p}_{i,j} - \omega_{y,x} \cdot 2^{l_k - 1} + b_{i,j}(l_k - 1) \cdot 2^{l_k - 1} & \text{if } r_{i,j} = 0 \end{cases} \quad (20)$$

where

$$\omega_{y,x} = \begin{cases} \mathrm{mod}(\lfloor \tilde{p}_{i,j}/2^{l_k} \rfloor, 2) & \text{if } r_{i,j} = 1 \\ \mathrm{mod}(\lfloor \tilde{p}_{i,j}/2^{l_k - 1} \rfloor, 2) & \text{if } r_{i,j} = 0 \end{cases}$$

3. Implementation issues

As the scheme for content-adaptive reversible visible watermarking in encrypted images is elaborated in Section 2, we discuss the practical implementation issues for the proposed scheme in this section. The proposed scheme can be divided to two main stages which are watermark embedding stage and watermark extraction stage.

The watermark embedding stage is constituted of data embedding position selection, data embedding room reservation, image encryption, and watermark embedding in encrypted image. The watermark embedding stage is outlined as Algorithm 1.

Algorithm 1 Watermark embedding.

Input The cover image $\mathbf{C}$ and the watermark image $\mathbf{W}$.
Output The marked encrypted image $\tilde{\mathbf{E}}$.

1: Calculate the optimal data embedding positions l_k and corresponding pixel bits $b_{i,j}(l_k)$ using Eqs. (5)–(7).
2: Generate the rearranged image $\mathbf{C}'$ after partitioning the cover image $\mathbf{C}$ into two parts $\mathbf{C}_{\mathrm{A}}$ and $\mathbf{C}_{\mathrm{B}}$ using Eqs. (8)–(10) to obtain the side information sequence S_1 and the header sequence S_2.
3: Embed the recovery sequence S_3 which is represented by least significant bits of $\mathbf{C}_{\mathrm{A}}$ into $\mathbf{C}_B$ with the reversible data hiding algorithm [37] to generate the image $\mathbf{C}''$.
4: Encrypt the image $\mathbf{C}''$ using Eqs. (15)–(17) and embed the header sequence S_2 and the side information sequence S_1 to generate the encrypted image $\mathbf{E}$.
5: Embed the watermark image $\mathbf{W}$ into the encrypted image $\mathbf{E}$ using Eq. (18) to obtain the marked encrypted image $\tilde{\mathbf{E}}$.

The watermark extraction stage is the inverse process of the watermark embedding stage. The original cover image can be recovered after image decryption and watermark extraction. The watermark extraction stage is outlined as Algorithm 2.

Algorithm 2 Watermark extraction.

Input The marked encrypted image $\tilde{\mathbf{E}}$.
Output The original cover image $\mathbf{C}$ and the watermark image $\mathbf{W}$.

1: Extract the header sequence S_2 and the side information sequence S_1 from $\tilde{\mathbf{E}}_{\mathrm{A}}$ to obtain the number of rows n_r and the starting row n_s of $\tilde{\mathbf{E}}_{\mathrm{A}}$.
2: Decrypt the marked encrypted image $\tilde{\mathbf{E}}$ with the encryption key except the least significant bits of $\tilde{\mathbf{E}}_{\mathrm{A}}$ and the pixel bits corresponding to optimal data embedding positions l_k.
3: Rearrange $\tilde{\mathbf{C}}_{\mathrm{A}}$ and $\tilde{\mathbf{C}}_{\mathrm{B}}$ according to the header sequence S_2 to obtain the marked decrypted image $\tilde{\mathbf{C}}$.
4: Extract the recovery sequence S_3 from $\tilde{\mathbf{C}}_{\mathrm{B}}$ with the reversible data hiding algorithm [37] and replace the least significant bits of $\tilde{\mathbf{C}}_{\mathrm{A}}$ with the recovery sequence S_3.
5: Extract the watermark image $\mathbf{W}$ using Eq. (20) to obtain the original cover image $\mathbf{C}$.

4. Experimental results and analysis

In this section, in order to demonstrate the effectiveness of the proposed scheme for content-adaptive reversible visible watermarking in encrypted images, the schemes proposed by Hu et al. [4], Zhang et al. [29], Chen et al. [10], and Mohammad et al. [13] are implemented for performance comparison. In our proposed scheme, the block size $h \times w$ is fixed at 8×8. Marked image quality, watermark visibility and watermark robustness are utilized to evaluate the performance of reversible visible image watermarking schemes. We use commonly adopted measurements PSNR and SSIM [39] to assess marked image quality. PSNR and SSIM are calculated by comparing the marked decrypted image with the original cover image. As shown in Figs. 6 and 7, four standard test images (i.e., Lena, Living room, Mandril, and Woman) with the size of 512×512 are used as cover images [36] and two binary images (i.e., Butterfly and Horse) are used as watermark images [40] in the experiments. The detailed description of all test images is given in Table 1. The original binary images Butterfly and Horse can be resized with different resolutions (i.e., 256×256, 128×128, 64×64, and 32×32) as watermark images in our experiments. Because the image center is usually the region of interest which attracts more attention in most practical application scenarios, the watermark embedding region is located in the center of each given cover image for performance comparison in the experiments.

4.1. Parameter analysis

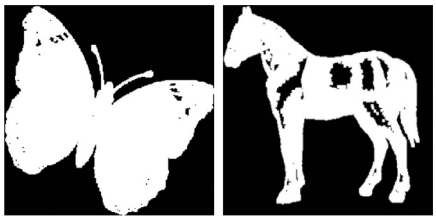
As discussed in Section 2, tuning parameters α and β in Eq. (7) optimize the data embedding positions and thus affect the watermark visibility and the marked image quality. In this subsection, we will investigate the effect of tuning parameters α

Table 1
Description of test images in detail.

Image	Type	Resolution	Format
Lena	Grayscale	512×512	TIF
Living room	Grayscale	512×512	TIF
Mandril	Grayscale	512×512	TIF
Woman	Grayscale	512×512	TIF
Butterfly	Binary	392×359	GIF
Horse	Binary	315×266	GIF

(a) (b) (c) (d)

Fig. 6. Cover images used in the experiments. (a) Lena, (b) Living room, (c) Mandril, and (d) Woman.



(a) (b)

Fig. 7. Watermark images used in the experiments. (a) Butterfly, (b) Horse.

and β on the proposed scheme performance. In our experiments, α ranges from 0.5 to 1.5 with the step of 0.1 and β ranges from 0.1 to 0.5 with the step of 0.1. The used watermark size in our experiments is 256×256. Fig. 8 illustrates PSNR values of marked decrypted images embedded with watermark Butterfly corresponding to different tuning parameter combinations. It can be seen that PSNR values of marked decrypted images get decreasingly lower with the increase of tuning parameters α and β. Tuning parameters α and β actually control data embedding positions for accommodating the watermark. We can infer that watermark visibility is improved with the increase of tuning parameters α and β because more watermark bits can be allocated in higher bit planes.

To subjectively explore the effect of tuning parameters α and β on watermark visibility, Figs. 9 and 10 show marked decrypted images[3] embedded with watermark Butterfly using different tuning parameters β with $\alpha = 1.0$. According to the above results in Fig. 8–10, we can conclude that the watermark visibility increases with β becoming ever larger while the marked image quality decreases at the same time. Therefore, it is of great importance to make a tradeoff between watermark visibility and marked image quality in practical applications.

4.2. Marked image quality

To objectively assess the marked image quality using Hu et al.'s scheme [4], Zhang et al.'s scheme [29], Chen et al.'s scheme [10], Mohammad et al.'s scheme [13], and our proposed scheme, PSNR and SSIM values of marked decrypted images embedded with watermarks Butterfly and Horse respectively are listed in Tables 2 and 3, where tuning parameters $\alpha = 1.0$ and $\beta = 0.3$ are used in our proposed scheme. The bold digits in Tables 2 and 3 mean that the corresponding scheme can achieve the best marked image quality in terms of PSNR and SSIM. Given the cover image and the watermark image, the marked image quality using certain reversible visible image watermarking scheme degrades with the increase of watermark size. In most cases, the best marked image quality can be obtained using our proposed scheme. For example, our proposed scheme outperforms Hu et al.'s scheme, Zhang et al.'s scheme, Chen et al.'s scheme, and Mohammad et al.'s scheme in terms of PSNR by 12.823 dB, 3.356 dB, 2.838 dB and 10.175 dB using cover Living room and watermark Horse with size of 256×256. In some cases, SSIM values of marked decrypted images using our proposed scheme are not highest but the deficiency is minor. For cover Lena and watermark Butterfly with size of 32×32, the SSIM value difference between Zhang et al.'s scheme and our proposed scheme is just 0.0006. Through the comparisons in Tables 2 and 3, we can infer that our proposed content-adaptive data embedding position calculation strategy is effective for obtaining optimal data embedding positions.

4.3. Watermark visibility

Watermark visibility is another important performance for the reversible visible image watermarking scheme. To compare the watermark visibility using Hu et al.'s scheme [4], Zhang et al.'s scheme [29], Chen et al.'s scheme [10], Mohammad et al.'s scheme [13], and our proposed scheme, Figs. 11 and 12 depict marked decrypted images embedded with watermarks Butterfly and Horse respectively, where tuning parameter $\alpha = 1.0$ is used in our proposed scheme. In the experiments, the used watermark size is 256×256. As shown in Figs. 11 and 12, it is easy to see that the watermark visibility is improved when tuning parameter β is modified from 0.3 to 0.5 in our proposed scheme. The subjective visual quality of marked decrypted images beneath the watermarks greatly degrades in Hu et al.'s scheme and Mohammad et al.'s scheme. As a result, it is almost unable to perceive the image content after watermark embedding. Zhang et al.'s scheme, Chen et al.'s scheme, and our proposed scheme can achieve almost the same level of watermark visibility. Our proposed scheme not only maintains the acceptable watermark visibility, but also introduces less distortion on the cover image.

It should be noted that Hu and Jeon's scheme [4], Chen et al.'s scheme [10], and Mohammad et al.'s scheme [13] cannot embed the watermark in the encrypted image. To compare the watermark visibility in the encrypted domain, Figs. 13 and 14 show marked encrypted images embedded with watermarks Butterfly and Horse respectively using Zhang et al.'s scheme [29] and our proposed scheme. Because Zhang et al.'s scheme embeds the watermark in the encrypted image by flipping some chosen pixel bits, the embedded watermark cannot be perceived in the encrypted domain. However, our proposed scheme embeds the visible watermark in the encrypted image by substituting pixel bits in data embedding positions. Therefore, the embedded watermarks Butterfly and Horse can be easily perceived as shown in Figs. 13 and 14 respectively.

[3] Due to space limitation, marked decrypted images can be zoomed to the original resolution for better visual perception.

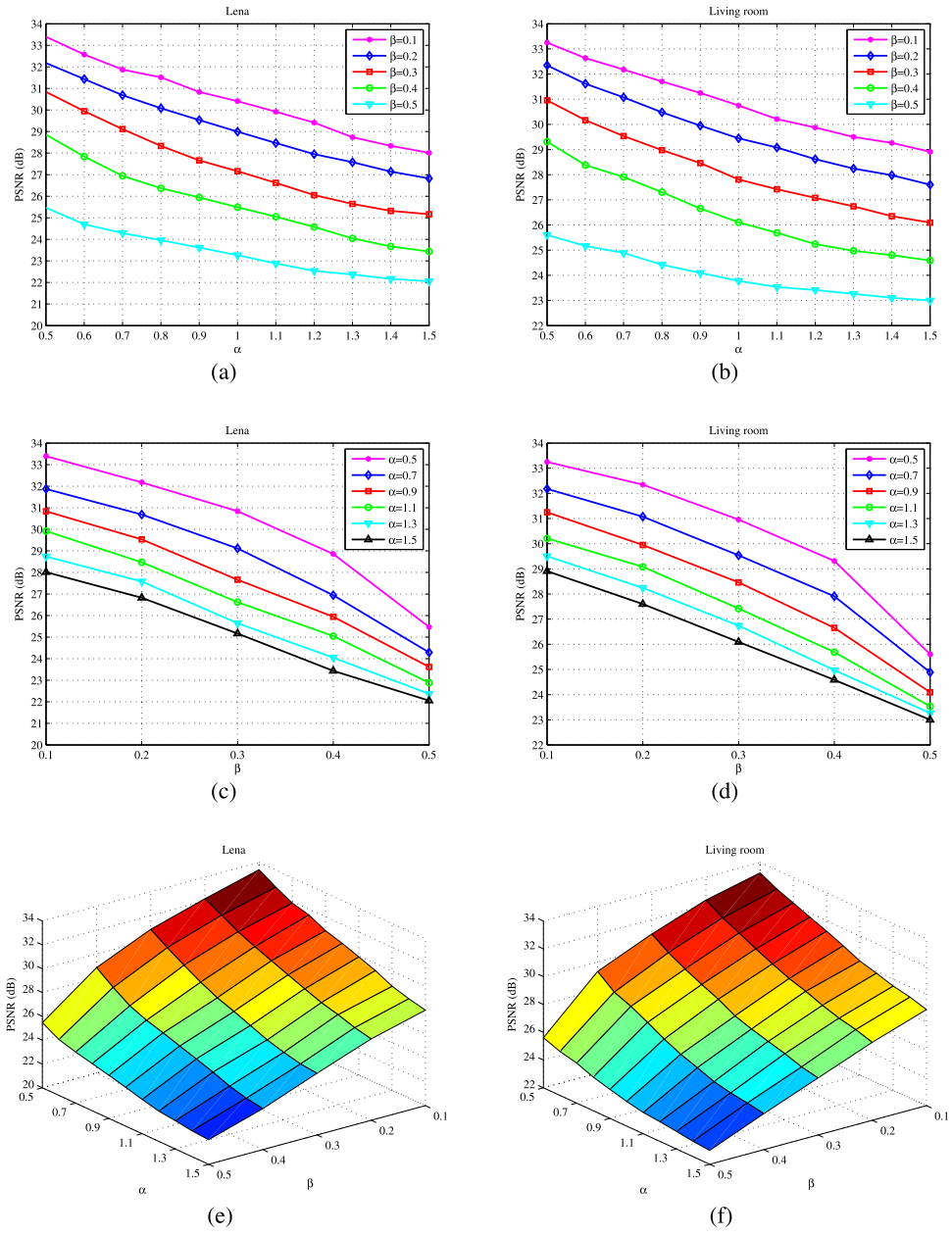


Fig. 8. PSNR values of marked decrypted images embedded with watermark Butterfly corresponding to different tuning parameter combinations. (a) PSNR value versus α using cover Lena, (b) PSNR value versus α using cover Living room, (c) PSNR value versus β using cover Lena, (d) PSNR value versus β using cover Living room, (e) PSNR value versus α and β using cover Lena, and (f) PSNR value versus α and β using cover Living room.

Fig. 9. Marked decrypted images using cover Lena for different tuning parameters β with $\alpha = 1.0$. (a) $\beta = 0.1$, (b) $\beta = 0.2$, (c) $\beta = 0.3$, (d) $\beta = 0.4$, and (e) $\beta = 0.5$.

(a) (b) (c) (d) (e)

Fig. 10. Marked decrypted images using cover Living room for different tuning parameters β with $\alpha = 1.0$. (a) $\beta = 0.1$, (b) $\beta = 0.2$, (c) $\beta = 0.3$, (d) $\beta = 0.4$, and (e) $\beta = 0.5$.

(a) (b) (c)

(d) (e) (f)

Fig. 11. Marked decrypted images embedded with watermark Butterfly using different reversible visible image watermarking schemes. (a) Hu and Jeon [4], (b) Zhang et al. [29], (c) Chen et al. [10], (d) Mohammad et al. [13], (e) Proposed scheme with $\beta = 0.3$, and (f) Proposed scheme with $\beta = 0.5$.

(a) (b) (c)

(d) (e) (f)

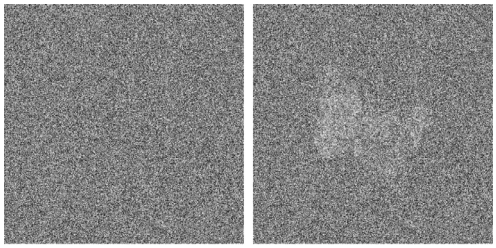
Fig. 12. Marked decrypted images embedded with watermark Horse using different reversible visible image watermarking schemes. (a) Hu and Jeon [4], (b) Zhang et al. [29], (c) Chen et al. [10], (d) Mohammad et al. [13], (e) Proposed scheme with $\beta = 0.3$, and (f) Proposed scheme with $\beta = 0.5$.

Table 2
PSNR and SSIM values of marked decrypted images embedded with watermark Butterfly using different reversible visible image watermarking schemes.

Cover image	Watermark size	PSNR					SSIM				
		Hu and Jeon [4]	Zhang et al. [29]	Chen et al. [10]	Mohammad et al. [13]	Proposed	Hu and Jeon [4]	Zhang et al. [29]	Chen et al. [10]	Mohammad et al. [13]	Proposed
Lena	32 × 32	33.450	**46.416**	44.453	36.857	44.226	0.9498	**0.9989**	0.9988	0.9974	0.9983
	64 × 64	27.852	38.653	38.432	30.587	**41.011**	0.8938	0.9957	0.9947	0.9906	**0.9958**
	128 × 128	22.364	32.654	32.789	24.478	**33.804**	0.8037	**0.9793**	0.9774	0.9647	0.9775
	256 × 256	16.695	25.044	26.700	17.977	**27.162**	0.5955	**0.9121**	0.9032	0.8673	0.9076
Living room	32 × 32	31.244	40.936	45.155	35.029	**45.174**	0.8949	0.9979	**0.9986**	0.9966	0.9981
	64 × 64	28.240	35.557	38.203	29.275	**40.034**	0.8821	0.9936	0.9937	0.9890	**0.9941**
	128 × 128	22.677	30.134	32.353	23.473	**34.098**	0.7828	0.9760	0.9763	0.9627	**0.9770**
	256 × 256	16.207	24.700	26.067	17.773	**27.811**	0.5677	0.9117	0.9031	0.8656	**0.9119**
Mandril	32 × 32	29.122	42.842	46.072	35.843	**50.146**	0.8761	0.9977	0.9984	0.9966	**0.9989**
	64 × 64	25.172	34.870	**40.682**	29.204	39.073	0.8034	0.9912	**0.9950**	0.9888	0.9924
	128 × 128	21.410	30.041	**34.716**	23.539	34.256	0.7625	0.9735	**0.9781**	0.9617	0.9744
	256 × 256	16.866	25.618	28.217	17.964	**28.469**	0.5737	0.9089	**0.9332**	0.8621	0.9099
Woman	32 × 32	32.672	43.187	46.702	35.492	**49.345**	0.9166	0.9969	**0.9974**	0.9963	0.9973
	64 × 64	28.449	37.032	39.820	29.613	**43.486**	0.9044	0.9903	0.9915	0.9890	**0.9920**
	128 × 128	22.942	29.922	32.299	23.444	**35.507**	0.8058	0.9682	0.9711	0.9637	**0.9718**
	256 × 256	16.628	24.622	26.479	17.732	**27.812**	0.5919	0.8951	0.8916	0.8676	**0.8999**

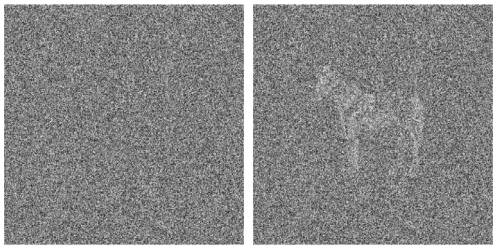
Table 3
PSNR and SSIM values of marked decrypted images embedded with watermark Horse using different reversible visible image watermarking schemes.

Cover image	Watermark size	PSNR					SSIM				
		Hu and Jeon [4]	Zhang et al. [29]	Chen et al. [10]	Mohammad et al. [13]	Proposed	Hu and Jeon [4]	Zhang et al. [29]	Chen et al. [10]	Mohammad et al. [13]	Proposed
Lena	32 × 32	33.872	46.394	45.375	38.686	**46.882**	0.9499	**0.9993**	0.9988	0.9979	0.9987
	64 × 64	27.915	39.757	39.136	32.020	**42.456**	0.8938	**0.9966**	0.9948	0.9918	**0.9966**
	128 × 128	22.561	33.649	33.725	26.056	**35.930**	0.8047	**0.9837**	0.9788	0.9725	0.9816
	256 × 256	16.774	28.105	27.570	20.179	**28.708**	0.5963	**0.9456**	0.9095	0.9031	0.9299
Living room	32 × 32	30.913	43.459	**46.316**	37.246	46.208	0.8948	0.9984	**0.9986**	0.9972	0.9983
	64 × 64	28.257	36.783	38.841	30.542	**41.415**	0.8823	0.9947	0.9939	0.9903	**0.9953**
	128 × 128	22.448	31.427	33.107	25.118	**35.584**	0.7808	0.9814	0.9770	0.9705	**0.9816**
	256 × 256	16.857	26.324	26.842	19.505	**29.680**	0.5720	**0.9408**	0.9044	0.8991	0.9375
Mandril	32 × 32	28.882	44.500	47.341	37.587	**51.958**	0.8759	0.9984	0.9985	0.9973	**0.9991**
	64 × 64	25.153	36.221	**41.940**	30.767	41.215	0.8031	0.9926	**0.9947**	0.9904	0.9942
	128 × 128	21.277	32.115	36.117	25.310	**36.563**	0.7618	0.9795	0.9801	0.9698	**0.9808**
	256 × 256	16.341	27.461	29.507	19.750	**30.345**	0.5699	**0.9368**	0.9337	0.8962	0.9331
Woman	32 × 32	32.171	45.472	48.260	37.374	**51.633**	0.9165	0.9976	0.9975	0.9970	**0.9979**
	64 × 64	27.765	38.468	40.874	31.222	**45.130**	0.9042	0.9918	0.9910	0.9902	**0.9935**
	128 × 128	22.489	31.318	33.111	24.954	**36.911**	0.8054	0.9753	0.9711	0.9700	**0.9780**
	256 × 256	16.326	26.382	27.315	19.558	**29.827**	0.5903	**0.9266**	0.8944	0.8998	0.9265



(a) (b)

Fig. 13. Marked encrypted images embedded with watermark Butterfly using different reversible visible image watermarking schemes. (a) Zhang et al. [29], (b) Proposed.



(a) (b)

Fig. 14. Marked encrypted images embedded with watermark Horse using different reversible visible image watermarking schemes. (a) Zhang et al. [29], (b) Proposed.

4.4. Watermark robustness

Reversible visible watermarking can convey ownership information directly on the marked image. Watermark robustness is a beneficial property for reversible visible watermarking. In some application scenarios, marked images tend to be transmitted through heterogeneous networks where the network bandwidth is often changed so that image compression is frequently performed. JPEG compression is one of the most efficient techniques for attacking data hiding schemes including reversible visible watermarking. To test the watermark robustness of our proposed scheme against JPEG compression, we compress marked decrypted images where tuning parameters $\alpha = 1.0$ and $\beta = 0.5$ using different quality factors (QF). Figs. 15 and 16 present marked decrypted images embedded with watermarks Butterfly and Horse respectively against JPEG compression, where the used watermark size is 256×256. The influence of JPEG compression on watermark visibility is weak so that we can perceive the embedded watermarks in marked decrypted images after JPEG compression. It can be concluded that the embedded watermark can survive against JPEG compression using our proposed scheme.

4.5. Discussion of the security

In the proposed scheme for content-adaptive reversible visible watermarking in encrypted images, the security includes the image content security and the visible watermark security.

4.5.1. Image content security

The content owner does not allow the data hider or the unauthorized third party to access the original image without the encryption key. In the proposed scheme, the secure stream cipher is used to encrypt the image $\mathbf{C}'$. For the image $\mathbf{C}'$ with size of $H \times W$, there are $256^{H \cdot W}$ possible bit sequences to change the pixel values. Thus, the possibility of breaking the encrypted results without the encryption key is as small as $\frac{1}{256^{H \cdot W}}$. Moreover, our proposed scheme can keep the encrypted image unintelligible. As shown in Figs. 13 and 14, PSNR values of marked encrypted images embedded with watermark Butterfly and Horse are 9.182 dB and 9.198 dB respectively. It is difficult to recognize the contents of marked encrypted images except for the embedded visible watermarks.

4.5.2. Visible watermark security

The data hider (e.g., a database manager or a cloud server) does not allow the unauthorized third party to maliciously remove the embedded visible watermark. According to the widely accepted viewpoint, the simple replacement of the pixel bit $b_{i,j}(l_k)$ with the watermark pixel $\omega_{y,x}$ does not satisfy the requirement of security. To enhance the visible watermark security, the data embedding key-controlled embedding mechanism is built as depicted in Eq. (18). The data embedding key generates the pseudo random number $r_{i,j} \in \{0, 1\}$. If $r_{i,j}$ equals 1, the current watermark pixel $\omega_{y,x}$ is used to replace the pixel bit $b_{i,j}(l_k)$ corresponding to the optimal data embedding position l_k. Otherwise, the current watermark pixel $\omega_{y,x}$ is used to replace the pixel bit $b_{i,j}(l_k - 1)$. Although this mechanism breaks the tradeoff between watermark visibility and marked image quality to some extent, the embedded visible watermark cannot be removed without the data embedding key. Therefore, the visible watermark security can be improved.

4.6. Performance on color images

Color images are more common media in practical applications. It is conventionally assumed that watermark schemes for grayscale images can be directly applied to color images by embedding the watermark independently in different color channels. However, the correlation among color channels may be ignored. To consider the correlation among color channels, we convert the color cover image from RGB color space to YCbCr color space for obtaining the luminance component Y as follows [41].

$$Y = \text{round}(0.299R + 0.587G + 0.114B) \tag{21}$$

where R, G, and B are scales from three color channels respectively. After the above color space conversion, we can apply our proposed scheme on the luminance component for watermark embedding and watermark extraction. Due to rounding errors in color space conversion, the rounding errors should be embedded in the color image as the auxiliary information to achieve reversibility. The adopted measurement PSNR_c for assessing the color image quality is defined as

$$\text{PSNR}_c = 10\log_{10}\left(\frac{255^2}{\text{MSE}}\right) \tag{22}$$

where

$$\text{MSE} = \frac{1}{3 \cdot H \cdot W}\sum_{k=1}^{3}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(p_{i,j,k} - \tilde{p}_{i,j,k}\right)^2$$

In Eq. (22), $p_{i,j,k}$ is the scale value at (i, j) in the kth color channel of the original color image and $\tilde{p}_{i,j,k}$ is the corresponding marked scale value. To test the performance of our proposed scheme on color images, the famous UCID database [42] which consists of 1338 uncompressed color images is used. For fair comparison, color space conversion should also be applied in Hu and Jeon's scheme [4], Zhang et al.'s scheme [29], Chen et al.'s scheme [10], and Mohammad et al.'s scheme [13] for watermark embedding and watermark extraction.

To objectively assess the marked color image quality using Hu and Jeon's scheme [4], Zhang et al.'s scheme [29], Chen

Fig. 15. Marked decrypted images embedded with watermark Butterfly against JPEG compression. (a) Without compression, (b) QF = 95, (c) QF = 85, (d) QF = 75, (e) QF = 65, and (f) QF = 55.

Fig. 16. Marked decrypted images embedded with watermark Horse against JPEG compression. (a) Without compression, (b) QF = 95, (c) QF = 85, (d) QF = 75, (e) QF = 65, and (f) QF = 55.

et al.'s scheme [10], Mohammad et al.'s scheme [13], and our proposed scheme, $PSNR_c$ values of marked decrypted color images embedded with watermark Butterfly on the UCID database are described in Fig. 17, where tuning parameters $\alpha = 1.0$ and $\beta = 0.3$ are used in our proposed scheme. The used watermark size is 128×128 in the experiments. It can be seen that our proposed scheme can achieve the best marked color image quality for most test images in the UCID database. The average $PSNR_c$ values of marked decrypted color images using Hu and Jeon's scheme, Zhang et al.'s scheme, Chen et al.'s scheme, Mohammad et al.'s scheme, and our proposed scheme are 20.514 dB, 26.991 dB, 29.681 dB, 22.418 dB, and 31.159 dB respectively in Fig. 17.

To compare the watermark visibility using Hu and Jeon's scheme [4], Zhang et al.'s scheme [29], Chen et al.'s scheme [10], Mohammad et al.'s scheme [13], and our proposed scheme, Fig. 18 depicts marked decrypted color images embedded with watermark Butterfly, where tuning parameters $\alpha = 1.0$ and $\beta = 0.3$ are used in our proposed scheme. The used watermark size is 256×256 and the test image is ucid00038 in the experiments. The $PSNR_c$ values of the marked decrypted color image using Hu and Jeon's scheme, Zhang et al.'s scheme, Chen et al.'s scheme, Mohammad et al.'s scheme, and our proposed scheme are 14.089 dB, 24.260 dB, 24.231 dB, 15.677 dB, and 24.757 dB in Fig. 18. It is almost unable to perceive the image content beneath the watermark in Hu et al.'s scheme and Mohammad et al.'s scheme. The embedded watermark does not significantly obscure the marked image details beneath it using our proposed scheme and the best marked color image quality can be obtained.

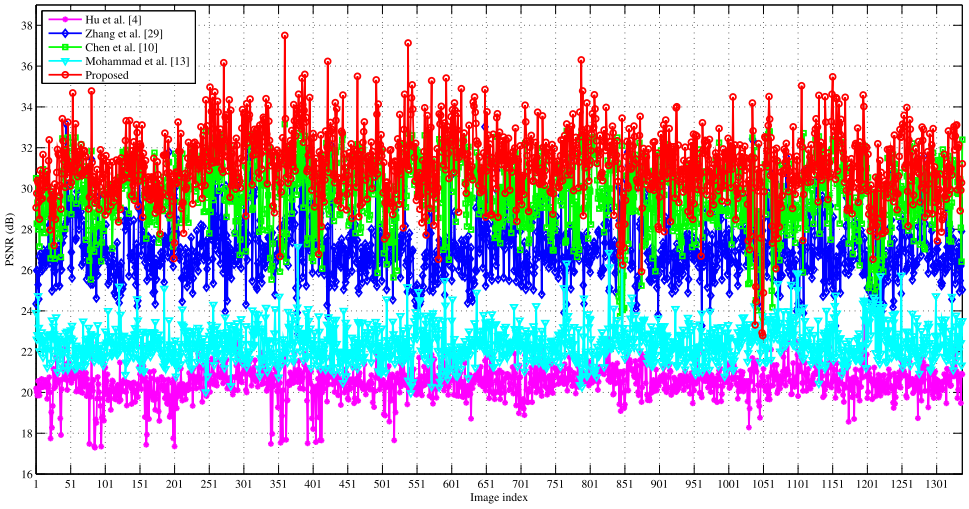


Fig. 17. $PSNR_c$ values of marked decrypted color images embedded with watermark Butterfly using different reversible visible image watermarking schemes on the UCID database.

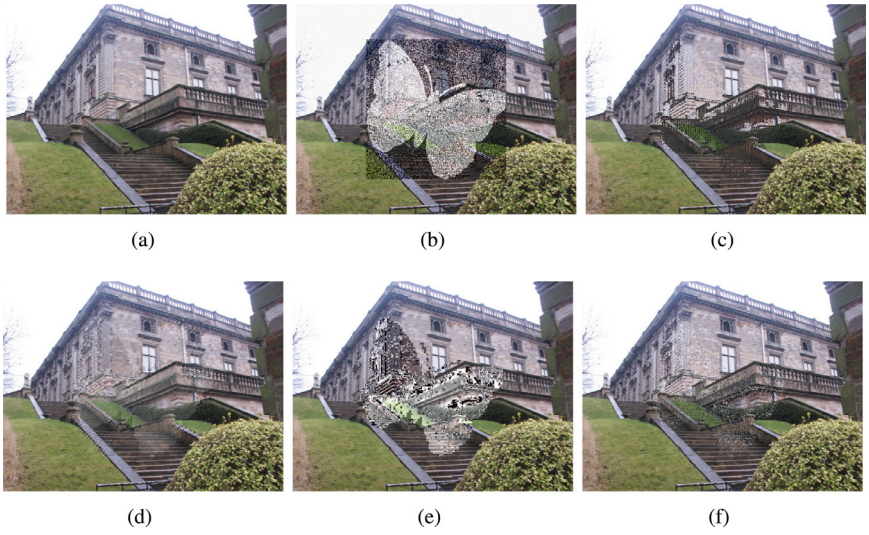


Fig. 18. Marked decrypted color images embedded with watermark Butterfly using different reversible visible image watermarking schemes. (a) Original color image, (b) Hu and Jeon [4], (c) Zhang et al. [29], (d) Chen et al. [10], (e) Mohammad et al. [13], and (f) Proposed scheme with $\alpha = 1.0$ and $\beta = 0.3$.

5. Conclusion and future work

With regard to the basic issues corresponding to watermark visibility and marked image quality, a reversible visible watermarking scheme in encrypted images which is capable of achieving the tradeoff between watermark visibility and marked image quality is proposed in this paper. The data embedding position calculation strategy is presented to select optimal data embedding positions for accommodating the watermark with the visual perceptual model. To target the problem of limited embedding capacity in encrypted images for reversible visible watermarking, the data embedding room is vacated before encryption with a traditional reversible data hiding algorithm to carry pixel bits in data embedding positions. This novel framework for reversible visible watermarking in encrypted images is suitable for different reversible data hiding algorithms. Experimental results demonstrate the merits of the proposed scheme in terms of marked image quality, watermark visibility and watermark robustness. Specifically, the proposed scheme can obtain average $PSNR_c = 31.159$ dB on the UCID database and resist JPEG compression with quality factors ranging from 95 to 55.

In the proposed scheme, optimal data embedding positions which serve as the side information balance watermark visibility and marked image quality. However, the side information sequence embedding will affect the marked image quality. In the future, reducing the side information size and designing a general metric for evaluating the watermark visibility deserves further investigation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 4
Important notations used in this paper.

Notation	Description	Notation	Description
$\mathbf{C}$	Cover image	$\mathbf{W}$	Watermark image
$H \times W$	Cover image size	$M \times N$	Watermark image size
$p_{i,j}$	Cover image pixel	$\omega_{y,x}$	Watermark image pixel
(i, j)	Cover image pixel coordinate	(y, x)	Watermark image pixel coordinate
$h \times w$	Block size	$T_{\mathrm{JND}}(p_{i,j})$	JND threshold of $p_{i,j}$
L	Optimal data embedding position set	(o_y, o_x)	Top left coordinate of watermark embedding region
S_1	Side information sequence	S_2	Header sequence
S_3	Recovery sequence	$\tilde{\mathbf{C}}$	Marked decrypted image
$\mathbf{C}_{\mathrm{A}}$	Textured partition of $\mathbf{C}$	$\mathbf{C}_{\mathrm{B}}$	Smoother partition of $\mathbf{C}$
$\mathbf{E}$	Encrypted image	$\tilde{\mathbf{E}}$	Marked encrypted image
T_p	Positive threshold for prediction error expansion	T_n	Negative threshold for prediction error expansion

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61802357, Grant U1636201, and Grant 61572452, and in part by the Anhui Initiative in Quantum Information Technologies under Grant AHY150400. The authors would like to sincerely thank the editors and the anonymous reviewers for their valuable comments.

Appendix

This appendix is to list important notations used in this paper as depicted in Table 4.

References

[1] M.S. Kankanhalli, Rajmohan, K.R. Ramakrishnan, Adaptive visible watermarking of images, in: Proceedings of International Conference on Multimedia Computing and Systems, 1999, pp. 568–573.
[2] S.P. Mohanty, K.R. Ramakrishnan, M.S. Kankanhalli, A DCT domain visible watermarking technique for images, in: Proceedings of International Conference on Multimedia and Expo, 2000, pp. 1029–1032.
[3] Y.Q. Shi, X. Li, X. Zhang, H.T. Wu, B. Ma, Reversible data hiding: advances in the past two decades, IEEE Access 4 (2016) 3210–3237.
[4] Y. Hu, B. Jeon, Reversible visible watermarking and lossless recovery of original images, IEEE Trans. Circ. Syst. Video Technol. 16 (11) (2006) 1423–1429.
[5] JBIG-KIT, lossless Image Compression Library (Online), 2018. Available: < https://www.cl.cam.ac.uk/~mgk25/jbigkit/>.
[6] S.K. Yip, O.C. Au, C.W. Ho, H.M. Wong, Lossless visible watermarking, in: Proceedings of International Conference on Multimedia and Expo, 2006, pp. 853–856.
[7] H.M. Tsai, L.W. Chang, A high secure reversible visible watermarking scheme, in: Proceedings of International Conference on Multimedia and Expo, 2007, pp. 2106–2109.
[8] T.Y. Liu, W.H. Tsai, Generic lossless visible watermarking–a new approach, IEEE Trans. Image Process 19 (5) (2010) 1224–1235.
[9] Y. Yang, X. Sun, H. Yang, C.T. Li, R. Xiao, A contrast-sensitive reversible visible image watermarking technique, IEEE Trans. Circ. Syst. Video Technol. 19 (5) (2009) 656–667.
[10] C.C. Chen, Y.H. Tsai, H.C. Yeh, Difference-expansion based reversible and visible image watermarking scheme, Multimed. Tools Appl. 76 (6) (2017) 8497–8516.
[11] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circ. Syst. Video Technol. 13 (8) (2003) 890–896.
[12] G. Yang, W. Qi, X. Li, Z. Guo, Improved reversible visible watermarking based on adaptive block partition, in: Proceedings of International Workshop on Digital Forensics and Watermarking, 2017, pp. 303–317.
[13] N. Mohammad, X. Sun, H. Yang, J. Yin, G. Yang, M. Jiang, Lossless visible watermarking based on adaptive circular shift operation for BTC-compressed images, Multimed. Tools Appl. 76 (11) (2017) 13301–13313.
[14] Y.K. Lin, C.H. Yang, J.T. Tsai, More secure lossless visible watermarking by DCT, Multimed. Tools Appl. 77 (7) (2018) 8579–8601.
[15] C.Y. Hsu, C.S. Lu, S.C. Pei, Image feature extraction in encrypted domain with privacy-preserving SIFT, IEEE Trans. Image Process. 21 (11) (2012) 4593–4607.
[16] C. Qin, Q. Zhou, F. Cao, J. Dong, X. Zhang, Flexible lossy compression for selective encrypted image with image inpainting, IEEE Trans. Circ. Syst. Video Technol. (2018), doi:10.1109/TCSVT.2018.2878026.
[17] X. Cao, L. Du, X. Wei, D. Meng, X. Guo, High capacity reversible data hiding in encrypted images by patch-level sparse representation, IEEE Trans. Cybern. 46 (5) (2016) 1132–1143.
[18] C. Qin, Z. He, X. Luo, J. Dong, Reversible data hiding in encrypted image with separable capability and high embedding capacity, Info. Sci. 465 (2018) 285–304.
[19] K. Ma, W. Zhang, X. Zhao, N. Yu, F. Li, Reversible data hiding in encrypted images by reserving room before encryption, IEEE Trans. Inf. Forensics Secur. 8 (3) (2013) 553–562.
[20] F. Huang, J. Huang, Y.Q. Shi, New framework for reversible data hiding in encrypted domain, IEEE Trans. Inf. Forensics Secur. 11 (12) (2016) 2777–2789.
[21] S. Yi, Y. Zhou, Binary-block embedding for reversible data hiding in encrypted images, Signal Process. 133 (2017) 40–51.
[22] X. Wu, J. Weng, W. Yan, Adopting secret sharing for reversible data hiding in encrypted images, Signal Process. 143 (2018) 269–281.
[23] C. Qin, W. Zhang, F. Cao, X. Zhang, C.C. Chang, Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection, Signal Process. 153 (2018) 109–122.
[24] S. Yi, Y. Zhou, Separable and reversible data hiding in encrypted images using parametric binary tree labeling, IEEE Trans. Multimed. 21 (1) (2019) 51–64.
[25] Z. Qian, H. Xu, X. Luo, X. Zhang, New framework of reversible data hiding in encrypted JPEG bitstreams, IEEE Trans. Circ. Syst. Video Technol. 29 (2) (2019) 351–362.
[26] T. Tuncer, D. Avci, E. Avci, A new data hiding algorithm based on minesweeper game for binary images, J. Fac. Eng.Arch. Gazi Univ. 31 (4) (2016) 951–959.
[27] T. Tuncer, A novel image authentication method based on singular value decomposition, J. Fac. Eng. Arch. Gazi Univ. 32 (3) (2017) 877–886.
[28] T. Tuncer, A probabilistic image authentication method based on chaos, Multimed. Tools Appl. 77 (16) (2018) 21463–21480.
[29] X. Zhang, Z. Wang, J. Yu, Z. Qian, Reversible visible watermark embedded in encrypted domain, in: Proceedings of China Summit and International Conference on Signal and Information Processing, 2015, pp. 826–830.
[30] J. Fridrich, M. Goljan, P. Lisonek, D. Soukal, Writing on wet paper, IEEE Trans. Signal Process. 53 (10) (2005) 3923–3935.
[31] X. Gao, W. Lu, D. Tao, X. Li, Image quality assessment based on multiscale geometric analysis, IEEE Trans. Image Process. 18 (7) (2009) 1409–1423.
[32] H.R. Wu, A.R. Reibman, W. Lin, F. Pereira, S.S. Hemami, Perceptual visual signal compression and transmission, Proc. IEEE 101 (9) (2013) 2025–2043.
[33] A. Liu, W. Lin, M. Paul, C. Deng, F. Zhang, Just noticeable difference for images with decomposition model for separating edge and textured regions, IEEE Trans. Circ. Syst. Video Technol. 20 (11) (2010) 1648–1652.
[34] J. Wu, W. Lin, G. Shi, X. Wang, F. Li, Pattern masking estimation in image with structural uncertainty, IEEE Trans. Image Process. 22 (12) (2013) 4892–4904.
[35] J. Wu, L. Li, W. Dong, G. Shi, W. Lin, C.C.J. Kuo, Enhanced just noticeable difference model for images with pattern complexity, IEEE Trans. Image Process. 26 (6) (2017) 2682–2693.
[36] Standard Test Images (Online), 2004. Available: <http://www.imageprocessingplace.com/root_files_V3/image_databases.htm>.
[37] V. Sachnev, H.J. Kim, J. Nam, S. Suresh, Y.Q. Shi, Reversible watermarking algorithm using sorting and prediction, IEEE Trans. Circ. Syst. Video Technol. 19 (7) (2009) 989–999.
[38] X. Li, J. Li, B. Li, B. Yang, High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion, Signal Process. 93 (2013) 198–205.
[39] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[40] L.J. Latecki, R. Lakamper, T. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, 2000, pp. 424–429.
[41] I.-R. Recommendation, Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios, 2011.
[42] G. Schaefer, M. Stich, UCID: an uncompressed color image database, in: Proceedings of SPIE Storage and Retrieval Methods and Applications for Multimedia, 2004, pp. 472–480.