J Supercomput (2014) 70:247–268
DOI 10.1007/s11227-014-1258-y

# Visible watermarking with reversibility of multimedia images for ownership declarations

**Fu-Hau Hsu · Min-Hao Wu · Cheng-Hsing Yang · Shiuh-Jeng Wang**

Published online: 24 July 2014


**Abstract** Digital watermarking technology is primarily the joining of the rightful owner of the protected media. Once the media are suspected to be illegally used, an open algorithm can be used to extract the digital watermark for the purpose of showing the media's ownership. From a hidden watermark in the media from the appearance point of view, general digital watermarking technologies can be divided into two categories: visible watermark technology and invisible watermark technology. Visual watermark technology embeds a watermark into the protected media to declare ownership and deter pirate behavior. In this paper, we propose a reversible visible watermark method, which embeds a binary-imaged watermark into gray-scale images to create a visible watermark. Not using complex calculations, this paper tries to simply change the pixel value to achieve the digital watermark, where our scheme is also against the possible detections with LSB-based manners in use. Furthermore, a reversible steganographic method is used to embed the watermarking information into the watermarking images. The watermark information can then be used to recover the original images.

**Keywords** Visible · Ownership · Reversibility · Watermarking · Histogram

F.-H. Hsu · M.-H. Wu
Department of Computer Science and Information Engineering,
National Central University, Taoyuan 320, Taiwan

C.-H. Yang
Department of Computer Science, National Pingtung University of Education,
Pingtung 900, Taiwan

S.-J. Wang (✉)
Department of Information Management, Central Police University, Taoyuan 333, Taiwan
e-mail: dopwang@gmail.com

## 1 Introduction

The popularity of the Internet offers great convenience in transmitting large amounts of data via the network. With the rapid development of network technologies and the coming of the digital era, the Internet has become indispensable for many people, as well as enterprises, which have expanded their traditional business activities on the Internet. Although encrypting a message before transmitting it on the Internet may provide a safe way for secret communication, encryption systems, such as data encryption standard (DES) and RSA, encrypt a message by transforming it into a meaningless form, which may alert interceptors. Therefore, transmitting digital data secretly and securely have increasingly become an issue of concern. Information hiding, which hides secret data into cover media, is an important technique for resolving this problem.

Many approaches to information hiding have been proposed for different attributes, such as capacity, imperceptibility, being undetectable, robustness, and reversibility. These attributes are used for various applications, such as secret communication, copyright protection, tampering detection, and other human-centered approaches. Although traditional data hiding techniques have been able to make secret data imperceptible to attackers, distortions of cover media, caused by the hiding process, have often been inevitable and irreversible. For this reason, researchers have focused their attention on developing reversible data hiding methods. Reversibility allows the original media to be completely recovered from stego-media after the embedded message is extracted. Many reversible data hiding approaches have been proposed [1–25]. According to where the data are embedded, these approaches can be classified into three categories: the spatial domain [1,3–14,16–18,24], the frequency domain [35,36], and other compression types, such as vector quantization (VQ) [21–25].

In those afore-mentioned developed reversible data hiding methods, four main technologies have been widely applied: compression-based technology [1–7], difference expansion-based technology [8–12], histogram-based technology [13–20] and visible watermarking technology [26–34]. For the compression-based technology, in 2002, Fridrich et al. [1,2] compressed the least significant bit (LSB) plane to obtain additional space for embedding secret data. In 2002, Celik et al. [3–5] improved the method of Fridrich et al. and proposed the generalized-LSB (G-LSB) scheme by compressing the quantization residuals of pixels to yield additional space to embed a message. In 2004, Awrangjeb and Kankanhalli [6,7] presented a scheme that detects the textured blocks, extracts the LSBs of the pixel values from these textured blocks based on the human visual system (HVS), and concatenates the authentication information with the compressed bit-string. For the difference expansion-based technology, in 2003, Tian [8] presented a method that expands the difference between two neighboring pixels to obtain redundant space for embedding a message. In 2004, Alattar [9] used the difference expansion of vectors of adjacent pixels to obtain more embedding space. Using Tian's scheme of difference expansion, in 2006, Chang and Lu [10] calculated the difference between a pixel and the mean value of its neighboring pixels to embed a message. In 2007, Weng et al. [11,12] used the correlations among four pixels in a quad and embedded data by expanding the differences between one pixel and each of its three neighboring pixels. For the histogram-based technology, in 2006, Ni

et al. [13] presented a reversible data hiding method based on the histogram. Their method guarantees that the change of each pixel in the stego image remains within $\pm 1$. Therefore, the PSNR value of the stego image is at least 48 dB. However, their method used the pixel values in the original image to create the histogram, and the peak points of the histogram are not high enough. Some methods used predictive concepts to increase the peak highs [14–20]. In 2008, Lin et al. [16] proposed a multilevel reversible data hiding scheme based on the histogram of the difference image. The scheme utilized a multilevel data hiding strategy to achieve large hiding capacity and keep the distortion comparatively low. In 2009, Tsai et al. [17] proposed a reversible image hiding scheme based on histogram shifting for medical images. To enlarge the embedding capacity, the similarity of neighboring pixels was explored. In addition, the prediction technique and the residual histogram of the predicted errors on the cover image were used to hide the secret messages. In 2009, Li et al. [19] proposed a reversible data hiding scheme based on the similarity between neighboring pixels. The difference value between a pair of adjacent pixels has a high probability close to zero. Therefore, transforming the cover image into an intermediary difference sequence can increase the heights of the peak points. They employed the histogram of the pixel difference sequence to increase the embedding capacity. For the vector quantization technology, in 2005, the modified fast correlation VQ (MFCVQ) was proposed by Yang et al. [21]. Chen and Huang [22] proposed a hybrid method combining the dynamic tree-coding scheme (DTCS) and the modified search order coding scheme (MSOC) to re-encode the output index map efficiently without causing any extra coding distortion. In 2009, Lu et al. [23] provided a VQ reversible data hiding method with an MFCVQ-based approach. There was also a new study in [24] related to VQ-based data hiding, where the side-matching background is incorporated to achieve the improvements in terms of image quality and secret capacity. Besides, Zhang and Wang extended Mielikainen's method to propose an EMD (exploiting modification direction) method, where $n$ pixels are used as an embedding unit [35]. In 1997, Wu and Memon proposed a method of GAP (gradient adjusted prediction [36]. It is more robust than the manners in linear predictors at the areas of strong edge-skeleton.

For visible-watermarking-based technology, the existence of the watermark can be judged by human eyes [25–33]. Embedding watermarks visibly will of course degrade the quality of the original images. However, reversible techniques can be applied to allow legitimate users to remove the embedded visible watermark and restore the original images. Several reversible visible watermarking techniques have been proposed [26,30]. One common kind of approach is to compress a portion of the original image and then embed the compressed data together with the intended payload into the image [26]. Another kind of approach is based on the use of deterministic one-to-one compound mappings of image pixel values for overlaying a variety of visible watermarks of arbitrary sizes on cover images [30]. In 2010, Yang et al. proposed a reversible visible watermarking scheme to satisfy the application, where the application of the visible watermark is expected to combat copyright piracy [31]. In this paper, a new reversible visible watermarking method is proposed. Binary images are used as the watermark, and the embedding approach for the visible watermark is based on pixel modifications. After embedded, the watermark

can be clearly distinguished by the human eye. Moreover, some information, including the watermark, is imperceptibly embedded into the stego image by a reversible steganographic approach. The information can be extracted and used to completely recover the original image. The remainder of this paper is organized as follows. Some related works are reviewed in Sect. 2. Our proposed approach is shown in Sect. 3. Section 4 gives some experimental results. Finally, we draw some conclusions in Sect. 5.

## 2 Related work

In this section, we review Hu and Jeon's [26] data hiding and visible watermark method and Yang and Tsai's [20] histogram-based reversible data hiding method.

### 2.1 Hu and Jeon's method

In this section, Hu and Jeon [26] first proposed a reversible visible watermarking scheme by modifying one significant bit plane of the pixels of the original image. They achieved reversibility via hidden the compressed version of the altered bit plane without loss in the non-watermarked image region. However, the embedded visible watermark with this method appears to be somewhat blurred, and the visual quality of the original image is significantly distorted. There are two procedures in this algorithm: visible watermark embedding and data hiding. Figure 1 illustrates the framework of the proposed algorithm. $W$ and $R$ denote the visible watermark and the image region to be protected, respectively. $W$ has the same size as $R$. To achieve lossless recovery of the image $I$, the bit plane of $R$ must be preserved in the non-watermarked image area $I - R$ before $W$ is embedded into a bit plane of $R$. In Stage 1, the bit plane of $R$ constitutes the pixel set $D$. A bit-plane's data usually have a statistical structure, so $D$ is further compressed into $D_c$ in Stage 2 using the open C code of JBIG-KIT.

In Stages 3 and 4, a payload-adaptive scheme is performed to constitute a subset $S$ of the lowest bit plane in $I - R$. $S$ satisfies the constraint $|D_c| = |S| - |S_c|$, where $S_c$ is the compressed result of $S$ and the operator $|\cdot|$ represents the length of a sequence or the cardinality of a set. In Stages 5 and 6, $H = S_c \cup D_c$ is hidden into image area

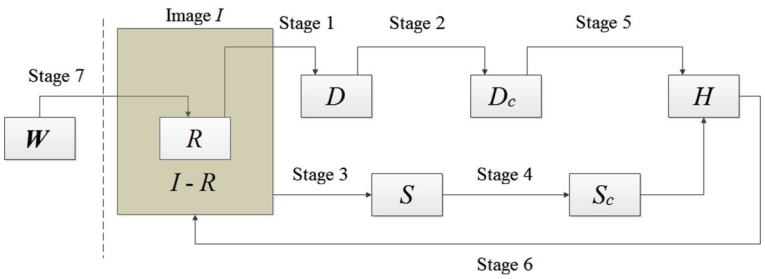


**Fig. 1** The framework of Hu and Jeon's algorithm

$I - R$ by directly replacing $S$ with $H$. Finally, in Stage 7, $W$ is embedded into $R$ by replacing the bit plane directly. Similarly, following the embedding procedure, the $W$ could be output as the extraction procedure. The data hiding process is briefly shown as the following steps.

Input: cover image $I$, watermarking $W$.

| | |
|---|---|
| Step 1. | Determine the watermarking region $R$ and denote $R_{D,}$ which is the bit plane of $R$ to be altered by $W$. |
| Step 2. | Compress $D$ by JBIG to generate $D_{C,}$ which will be embedded in $I - R$. |
| Step 3. | Choose the compression tools version of the CACM. Determine the key bit plane $I_K$ of each bit plane from the LSB to the MSB plane of the $I - R$. |
| Step 4. | Find $S$ via Stem, which is composed of the one-bit pixel on the $I_K$. The constitution of $S_{tem}$ starts at $\mid S_{tem} \mid = 1$. |
| Step 5. | If $S$ is found, then compress $S_{tem}$ before each addition and stratify $\mid DC \mid = \mid S_{tem} \mid - \mid S_{tem,\,c} \mid$.<br>Otherwise, repeat Step 5. |
| Step 6. | Construct the payload bit stream as $H = S_C \cup D_C$. Replace $S$ with $H$ to create $(I - R)_m$. |

## 2.2 Yang and Tsai's method

In 2010, Yang and Tsai [20] provided interleaving prediction methods to enhance the histogram-based reversible hiding approach on gray images. The predictive difference values are transformed into histograms to create higher peak values. For one level data hiding, the change of each pixel between the original image and the stego image remains within $\pm 1$. Figure 2a shows their column interleaving prediction. Pixels in odd columns will be predicted by pixels in even columns, and pixels in even columns are predicted by pixels in odd columns. In the embedding process, predictive difference values of odd columns are used to generate a histogram to embed secret data. Then, predictive difference values of even column are used. In the extracting and reversing the process, predictive difference values of even columns are processed first; then predictive difference values of odd columns are processed. Figure 2b shows their

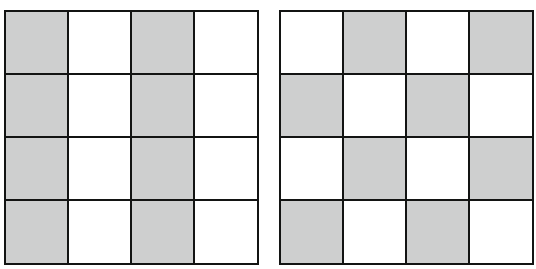


**Fig. 2** Two interleaving prediction strategies: **a** column based; **b** chessboard based

chessboard interleaving prediction. An image is considered a chessboard containing white pixels and black pixels. In the first stage, white pixels are processed and predicted by their neighboring black pixels. Then, in the second stage, black pixels are processed and predicted by their neighboring white pixels. The predictive difference value is calculated by the difference between a pixel value and the averaged pixel value of its neighboring pixels.

Suppose that the original image is a 512 × 512 gray image. $I_{i,j}$ and $I'_{i,j}$ are the pixel values at location $(i, j)$ before and after being processed, respectively. $D_{i,j}$ is the prediction error value at location $(i, j)$, and $D'_{i,j}$ is the embedded predictive error value at location $(i, j)$. The embedding algorithm is as follows.

Input: Original image, secret message.

Output: Stego-image, two pairs of peak and zero points.

Step 1. Input the original image *I*.

Step 2. The predictive difference value $D_{i,j}$ will use pixels in odd columns will be predicted by pixels in even columns, and then pixels in even columns are predicted by pixels in odd columns.

if $j = 1$ then, $D_{i,j} = I_{i,j} - I_{i,j} + 1$

if $j$ mod 2 = 1, then $D_{i,j} = I_{i,j} - \left\lfloor \frac{I_{i,j-1} + I_{i,j+1}}{2} \right\rfloor$

Step 3. Create the histogram $H(x)$ with $x \in [-255, 255]$ from all prediction values.

Step 4. Find two pairs of peak and zero points.

(a) Select the two highest peak points from $H(x)$, where $P_1 > P_2$.

(b) Select two zero points $Z_1$ and $Z_2$ from $H(x)$ with $x \in [P_1+1, 255]$ and $x \in [-255, P_2 - 1]$.

Step 5. Shift the histogram.

(c) if $x > P_1$ and $x < Z_1$ then move the whole part of the histogram $H(x)$ to the right by one unit.

(d) if $x < P_2$ and $x > Z_2$ then move the whole part of the histogram $H(x)$ to the left by one unit.

Step 6. Embed a secret bit if the predictive difference value $D_{i,j}$ is equal to $P_1$ and $P_2$ and then put the embedded bit in $D'_{i,j}$.

Step 7. To convert the predictive difference value $D_{i,j}$, use pixels in even columns in the odd columns.

if $j$=1, then $I'_{i,j} = D'_{i,j} + I_{i,j}$

if $j$ mod 2 = 0, then $I'_{i,j} = D_{i,j} + \left\lfloor \frac{I_{i,j-1} + I_{i,j+1}}{2} \right\rfloor$

Step 8. Similar to Steps 2 − 6, embed the message by the predictive difference values in even columns.

Step 9. Output the stego-image and two pairs of peak and zero points.

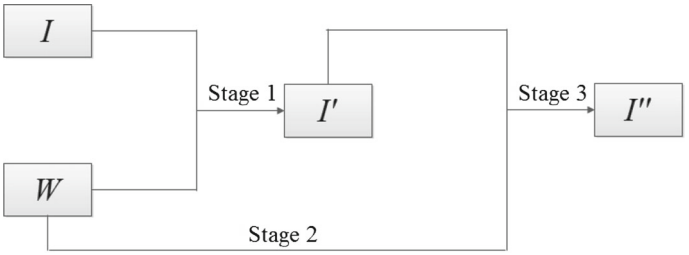


**Fig. 3** The flowchart of our embedding algorithm

## 3 Our proposed visual watermark

This study proposes a visual watermark method that embeds black-and-white binary images into the target grayscale images. The embedding method changes the pixel values by adding positive random values to them, such that the changed results are visible. After that, the watermark information is hidden in the gray image with invisibly reversible steganographic method. The watermark information can be extracted to completely recover the watermarked images. Our watermark embedding and removing procedures are introduced in the following subsections.

### 3.1 Visible watermarks embedding

The original image and digital watermark are defined as follows:

$$I = \{I(i,j) \mid 0 \le i < M_1, 0 \le j < M_2\},$$
$$W = \{W(i,j) \mid 0 \le i < N_1, 0 \le j < N_2\},$$

where $I$ denotes the original image, $W$ denotes the watermark, and $(i, j)$ denotes the pixel coordinate. The original image $I$ is an $M_1 \times M_2$ grayscale image, the watermark image $W$ is an $N_1 \times N_2$ binary image with white background pixels denoted as 1 and black logo pixels denoted as 0, and $N_i \le M_i$ for $i = 1, 2$. Figure 3 shows the flowchart of our embedding algorithm. In Stage 1, the embedding region is selected. It is represented by a coordinate where watermark $W$ begins to be embedded. In Stage 2, the watermark $W$ is visibly embedded into the image $I$ to form a watermarked image $I'$. Next, Stage 3 is given, which is to solve the underflow and overflow problems in our procedures. Finally, Stage 4, it is basically inspired by [20]. The watermark $W$ and the region information are invisibly embedded into image $I'$ to form a reversible watermarked image $I''$. The detailed embedding algorithm is described as follows:

Input: Original image $I$, binary watermark $W$, random seed $K$, maximum random value $MR$, and distance $D$.

Output: Reversible watermarked image $I''$

Step 1. 1.1. Use random seed $K$ to create a random number $R$, where $0 \le R \le MR$.

1.2. Select a coordinate $C$ of images $I$ where watermark $W$ begins to be embedded.

Step 2. Read the pixel value $W(i, j)$ of digital watermark $W$. Let $I(k, l)$ be the pixel which watermark bit $W(i, j)$ will be embedded into. If $W(i, j) = 1$, preserve the original pixel $I(k, l)$, i.e. $I'(k, l) = I(k, l)$;if $W(i, j) = 0$, and proceed with the following instructions:

$$\begin{cases} I'(k,l) = I(k,l) + D + R, \\ R = R + Rb \end{cases} \tag{1}$$

where $Rb$ is a random value of $\{-1, 0, 1\}$ satisfying $0 \le R \le MR$.

Step 3.

```
if I'(k, l) > 255 – (D + MR)
        If I'(k, l) > 255
                I'(k, l) = I'(k, l) – (D + MR)
                Overflow-flag string F = F || 1
        else
                Overflow-flag string F = F || 0
        endif
endif
```

Step 4. For $W$, $F$, and $C$, using invisibly reversible steganographic method to embed them into the image $I'$ to construct image $I''$.

Step 5. Output image $I''$.

In the algorithm, distance $D$ is used to control the depth or the visibility of the watermark. The random number $R$ [0, $MR$] is for security. $R$ is changed by a value $Rb$ in Eq. (1). $Rb$ values are small, such that the difference between two neighboring pixels' watermark depths is small. Binary string $F$ is used to process the overflow conditions. A watermarked pixel $I'(k, l)$ is overflow if $I'(k, l)$ is larger than 255. If $I'(k, l)$ is overflowing, the pixel value is adjusted to $x'(k, l) - (D + MR)$, where $255 - (D + MR) < I'(k, l) - (D + MR) < 255$. A bit of $F$ is created to record whether $I'(k, l)$ is an overflow or not when $I'(k, l) > 255 - (D + MR)$.

## 3.2 Watermark removing and image reconstructing

Figure 4 shows the flowchart of our watermark removing algorithm. On Stage 1, the embedded information $W$, $F$, and $C$ in the image $I''$ is extracted and image $I''$ is recovered into the image $I'$. Then, in Stage 2, the information $W$, $F$, and $C$ is used to recover image $I$ from an image $I'$. The detailed algorithm is as follows.

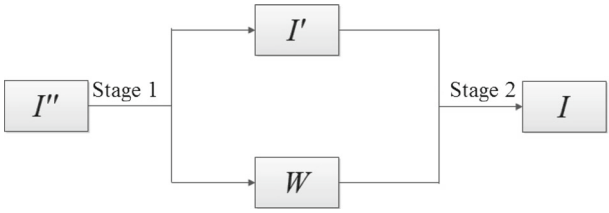


**Fig. 4** The flowchart of our watermark removing algorithm

Input: Image $I''$, random seed $K$, maximum random value $MR$, and distance $D$.

Output: Original image $I$ and watermark $W$.

Step 1. Use random seed $K$ to create a random number $R$, where $0 \le R \le MR$.

Step 2. Use the invisibly reversible steganographic method to extract digital watermark $W$, overflow-flag string $F$, and the beginning coordinate $C$ and recover image $I'$ from the image $I''$.

Step 3. Remove each watermark bit $W(i, j)$ from image $I'$ to get image $I$ as follows:

Let $I'(k, l)$ be the pixel that watermark bit $W(i, j)$ has been embedded into.

```
if W(i, j) = 1
        if I'(k, l) > 255 − (D + MR)
                Remove bit b from the beginning of F.
                if b = 1
                        I'(k, l) = I'(k, l) + (D + MR)
                end if
        I(k, l) = I'(k, l) − D − R
        R = R + Rb
        endif
else
        I(k, l) = I'(k, l)
end if
```

Step 4. Output watermark $W$ and original image $I$.

## 3.3 Watermark compression

In our watermark embedding algorithm, the watermark $W$ is embedded into watermarked image $I'$ for the purpose of reversibility. However, before $W$ is embedded, it is better to compress $W$ into $W_c$ to reduce the information size. To efficiently compress the watermark $W$, we apply the side-match prediction concept to change $W$ into $W'$. Then, $W'$ uses the open C code of JBIG-KIT to be compressed into $W'_c$. Figure 5 shows some adjacent pixels of pixel $I(k, l)$. We used the average of pixels $I(k, l-1)$ and $I(k-1, l)$ to predict the value of watermarked pixel $I'(k, l)$. If $I'(k, l)$ is sufficiently

**Fig. 5** The locations of some adjacent pixels

| | | |
|---|---|---|
| $I(k-1, l-1)$ | $I(k-1, l)$ | $I(k-1, l+1)$ |
| $I(k, l-1)$ | $I(k, l)$ | |

larger than the average, it is a high probability that a watermark bit 0 is embedded into $I(k, l)$. Suppose that a watermark bit $W(i, j)$ will be embedded into pixel $I(k, l)$.

The following steps are used to change each watermark bit $W(i, j)$ into $W'(i, j)$.

Step 1. $d = I'(k,l) - \frac{I(k,l-1)+I(k-1,l)}{2}$

Step 2. If $d$ is greater than $D/2$, $W'(i, j)$ is set to the inverted value of $W(i, j)$. Otherwise, $W'(i, j)$ is set to the value of $W(i, j)$.

Figure 6a shows the decision tree which changes $W$ into $W'$. Similarly, Fig. 4b shows the decision tree which recovers $W$ from $W'$. If this strategy is operated, most bits of $W'$ would have a value of 1.

## 4 Experiments

In this section, we perform some experiments for our method. Figure 7 shows the experimental binary watermark of size 100 × 300. Six grayscale images of size 512 × 512 are shown in Fig. 8. The image quality of the stego image is evaluated by the

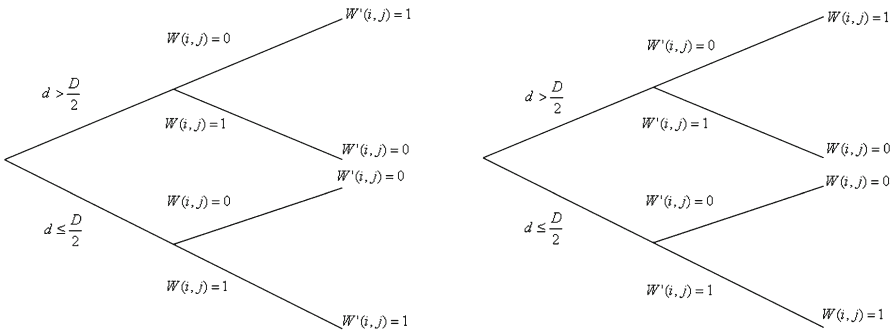


**Fig. 6** The decision trees of exchanging $W$ and $W'$: **a** Changing $W$ into $W'$; **b** Changing $W'$ into $W$

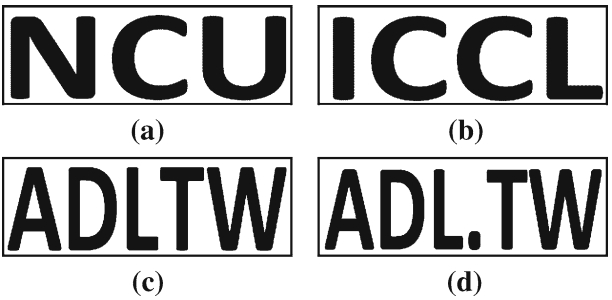


**Fig. 7** Binary watermark

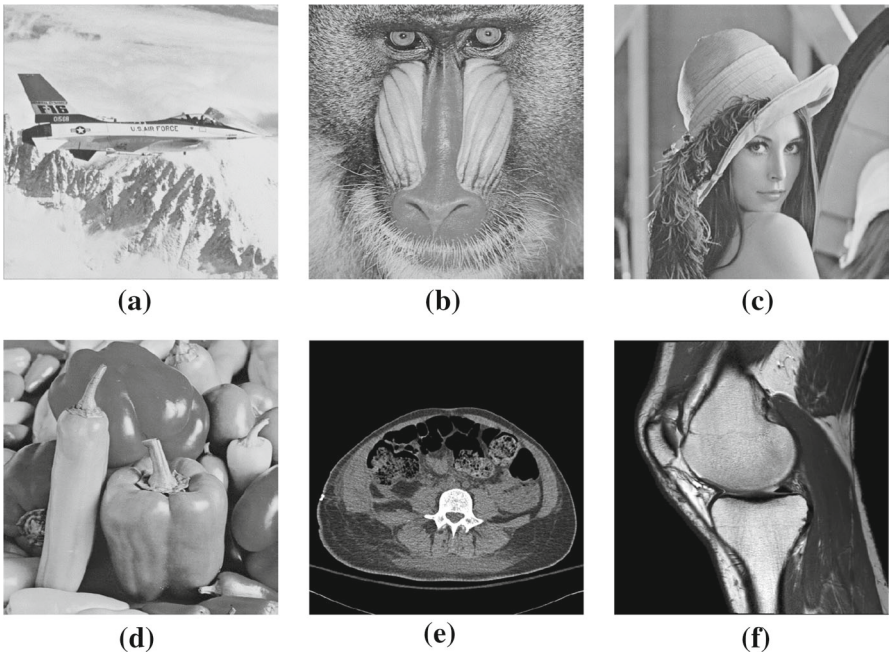


**Fig. 8** Six experimental cover images: **a** Airplane; **b** Baboon; **c** Lena; **d** Pepper; **e** CT0001; **f** MR0001

peak signal to noise ratio (PSNR) which is defined as:

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255^2}{\mathrm{MSE}} (\mathrm{dB}), \tag{2}$$

where MSE is the mean square error between the original image and the stego image. For a $512 \times 512$ gray image, the MSE is defined as:

$$\mathrm{MSE} = \frac{1}{512 \times 512} \sum_{i=0}^{511} \sum_{j=0}^{511} (I_{i,j} - I'_{i,j})^2, \tag{3}$$

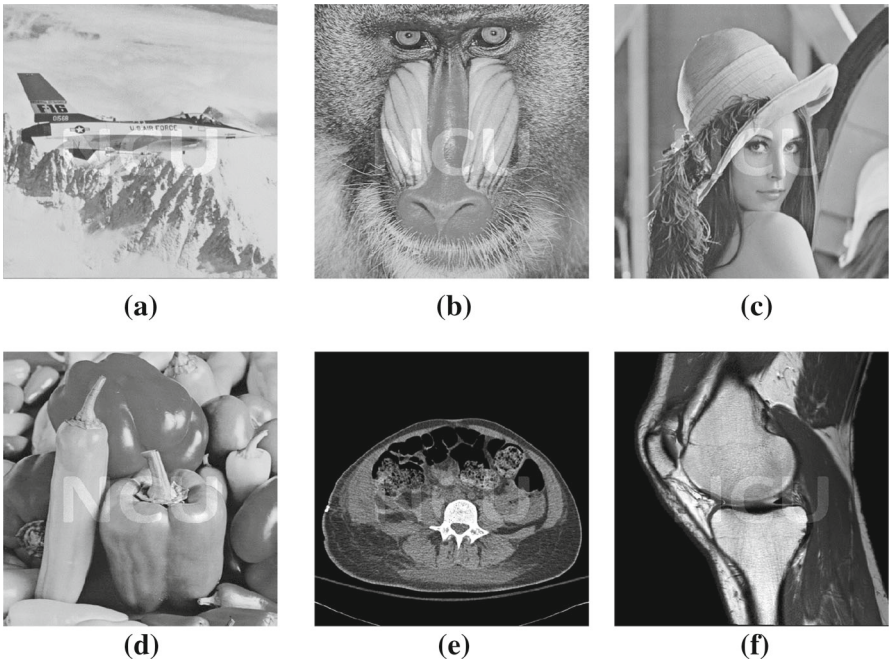


**Fig. 9** Watermarked images: **a** Airplane; **b** Baboon; **c** Lena; **d** Peppers; **e** CT0001; **f** MR0001

where $I_{i,j}$ and $I'_{i,j}$ are the pixel values of the original image and the stego image, respectively.

Figure 9 shows the reversible watermarked results for $D = 24$ and $MR = 16$. The results show that the watermarks are clearly visible. Yang and Tsai's [20] reversible data hiding method is used to embed the information of $W$, $F$, and $C$. In Table 1, the NCU watermark for peak signal to noise rations (PSNR) values between $I, I'$, and $I''$ of the six images is used to assess the similarity between them. In Table 2, the ICCL watermark for PSNR values between $I, I'$, and $I''$ of the six images is used to assess the similarity between them. In Table 3, the ADL.TW watermark for PSNR values between $I, I'$, and $I''$ of the six images is used to assess the similarity between them. In Table 4, the ADL.TW watermark for PSNR values between $I, I'$, and $I''$ of the six images is used to assess the similarity between them. In our experimental results, the PSNR from $I''$ to $I$ had the least changed effect on the PSNR value for the size of the watermark, and the watermark within the letters of the alphabet amount will not affect the PSNR. When we used the information of $W$, $F$, and $C$ to embed, the effect on the PSNR value is the least because that amount of information data was too small to affect the $I''$ PSNR value.

One of simpler embedding methods is to replace Eq. (1) with the following equation:

$$I'(k,l) = (I(k,l) \pm R(k,l)) \mod 256 \tag{4}$$

**Table 1** The NCU watermarks for PSNR values between $I$, $I'$, and $I''$ of the six images

| Watermark sizes | $D/MR$ | Images | | | | | |
|---|---|---|---|---|---|---|---|
| | | Airplane | Baboon | Lena | Pepper | CT0001 | MR0001 |
| 128 × 128 | 16/16 | | | | | | |
| | $I'$ to $I$ | 36.52 | 36.52 | 36.52 | 36.52 | 36.92 | 36.52 |
| | $I''$ to $I$ | 36.26 | 36.18 | 36.26 | 36.24 | 35.88 | 36.22 |
| | 24/24 | | | | | | |
| | $I'$ to $I$ | 32.95 | 32.95 | 32.97 | 32.96 | 33.42 | 32.95 |
| | $I''$ to $I$ | 32.81 | 32.77 | 32.83 | 32.81 | 32.87 | 32.79 |
| | 36/16 | | | | | | |
| | $I'$ to $I$ | 31.39 | 31.37 | 31.42 | 31.43 | 31.93 | 31.37 |
| | $I''$ to $I$ | 31.29 | 31.24 | 31.32 | 31.32 | 31.56 | 31.26 |
| | 36/36 | | | | | | |
| | $I'$ to $I$ | 30.26 | 29.72 | 29.72 | 29.66 | 30.03 | 29.51 |
| | $I''$ to $I$ | 30.16 | 29.62 | 29.63 | 29.57 | 29.79 | 29.42 |
| 300 × 100 | 16/16 | | | | | | |
| | $I'$ to $I$ | 33.82 | 33.82 | 33.82 | 33.82 | 34.00 | 33.90 |
| | $I''$ to $I$ | 33.66 | 33.63 | 33.66 | 33.65 | 33.50 | 33.69 |
| | 24/24 | | | | | | |
| | $I'$ to $I$ | 30.27 | 30.27 | 30.28 | 30.29 | 30.47 | 30.42 |
| | $I''$ to $I$ | 30.18 | 30.17 | 30.19 | 30.20 | 30.29 | 30.28 |
| | 36/16 | | | | | | |
| | $I'$ to $I$ | 28.72 | 28.68 | 28.69 | 28.72 | 28.90 | 28.87 |
| | $I''$ to $I$ | 28.66 | 28.61 | 28.63 | 28.66 | 28.73 | 28.80 |
| | 36/36 | | | | | | |
| | $I'$ to $I$ | 27.91 | 27.03 | 26.93 | 26.98 | 27.00 | 27.10 |
| | $I''$ to $I$ | 27.82 | 26.97 | 26.87 | 26.92 | 26.91 | 27.02 |

where $R(k, l)$ is a random value between 0 and $MR$. The plus and minus are randomly used during the embedding process. Figure 10 shows the watermarked images with $MR = 30$. The watermarks in Fig. 10 do not look smooth for many spots appear in them. The spots are due to the use of both plus and minus and the happening of overflow or underflow. Therefore, in our embedding algorithm, only the plus sign is used and the overflow is handled. Sometimes, if two neighboring random values, say $R(k, l)$ and $R(k, l + 1)$, are apparently different, spots can happen. Figure 11 shows the watermarked images when $D$= 24 and $MR = 16$, the plus sign is used, and the overflow is handled. Some spots happen in the watermarks of Fig. 11. Thus, we use $Rb$ to control the difference between two neighboring random values. If the set of $Rb$ values is larger, the difference between two neighboring random values will also become larger.

Distance $D$ is used to control the depth or the visibility of the watermark. A larger $D$ value will mark the watermark more distinctly. Embedding the watermark into a high-frequency region especially needs a large $D$ value for the visibility. Figure 12 shows the watermarked images with different $D$ values and a fixed $MR$ value. Figure 12a shows a larger $D$ value than that of Fig. 12b. Therefore, the watermark in Fig. 12a is more obvious than that in Fig. 12b. The $MR$ value can influence the security. Figure 13 shows two watermarked results with different $MR$ values

**Table 2** The ICCL watermarks for PSNR values between $I$, $I'$,and $I''$ of the six images

| Watermark sizes | $D/MR$ | Images | | | | | |
|---|---|---|---|---|---|---|---|
| | | Airplane | Baboon | Lena | Pepper | CT0001 | MR0001 |
| 128 × 128 | 16/16 | | | | | | |
| | $I'$ to $I$ | 37.40 | 37.40 | 37.40 | 37.40 | 37.80 | 37.40 |
| | $I''$ to $I$ | 37.11 | 37.02 | 37.10 | 37.09 | 36.88 | 37.07 |
| | 24/24 | | | | | | |
| | $I'$ to $I$ | 33.83 | 33.83 | 33.84 | 33.84 | 34.32 | 33.84 |
| | $I''$ to $I$ | 33.67 | 33.63 | 33.69 | 33.67 | 33.78 | 33.66 |
| | 36/16 | | | | | | |
| | $I'$ to $I$ | 32.26 | 32.25 | 32.27 | 32.29 | 32.82 | 32.26 |
| | $I''$ to $I$ | 32.15 | 32.11 | 32.15 | 32.18 | 32.49 | 32.13 |
| | 36/36 | | | | | | |
| | $I'$ to $I$ | 31.10 | 30.55 | 30.45 | 30.50 | 30.94 | 30.42 |
| | $I''$ to $I$ | 30.99 | 30.44 | 30.36 | 30.40 | 30.74 | 30.32 |
| 300 × 100 | 16/16 | | | | | | |
| | $I'$ to $I$ | 34.42 | 34.42 | 34.42 | 34.42 | 34.70 | 34.53 |
| | $I''$ to $I$ | 34.25 | 34.21 | 34.25 | 34.23 | 33.95 | 34.29 |
| | 24/24 | | | | | | |
| | $I'$ to $I$ | 30.87 | 30.87 | 30.88 | 30.91 | 31.20 | 31.05 |
| | $I''$ to $I$ | 30.78 | 30.75 | 30.79 | 30.8 | 30.85 | 30.93 |
| | 36/16 | | | | | | |
| | $I'$ to $I$ | 29.33 | 29.27 | 29.3 | 29.35 | 29.65 | 29.52 |
| | $I''$ to $I$ | 29.26 | 29.20 | 29.23 | 29.28 | 29.39 | 29.43 |
| | 36/36 | | | | | | |
| | $I'$ to $I$ | 28.48 | 27.58 | 27.47 | 27.63 | 27.75 | 27.81 |
| | $I''$ to $I$ | 28.40 | 27.51 | 27.41 | 27.57 | 27.64 | 27.74 |

and Fig. 14 shows their attacked results for different $MR$ values under the condition that only the random seed $K$ is unknown to the attacker. A larger $MR$ value will have better security. To improve the compression efficiency, watermark $W$ is changed into $W'$ by the side-match prediction concept. Figure 15 shows the watermark $W$ and its changed results for Lena and Baboon when $D = 24$ and $MR = 24$.

Table 5 shows a comparison PSNR with the watermark image size of 128 × 128 between the scheme of Hu et al. in [26] at $N_{RD} = 6$ and the scheme of Yang et al. in [31] at either $L$-bit-length = 3 or 4, and our scheme with $D = 16$, $MR = 16$. We have ever requested the same watermark to [26] when we asked for the offer of the image. But the authors in [26] have never given us the image for the requests in academic use. Alternatively, more watermark images with the same size and more words in a watermark image are next taken into account to fulfill our algorithms. In image quality, it turns out that our scheme is superior to the compared one for PSNR measure as long as the same length of 4-word is given in the watermark image. Furthermore, the PSNR measure is either the same or higher than the compared one if more words are shown in a tested watermark image. It is significantly seen that the image quality in PSNR-representation is superior to the scheme of Yang et al. [31] when the same watermarking logos are embedded into.

**Table 3** The ADLTW watermarks for PSNR values between $I$, $I'$, and $I''$ of the six images

| Watermark sizes | $D/MR$ | Images | | | | | |
|---|---|---|---|---|---|---|---|
| | | Airplane | Baboon | Lena | Pepper | CT0001 | MR0001 |
| 128 × 128 | 16/16 | | | | | | |
| | $I'$ to $I$ | 37.17 | 37.17 | 37.17 | 37.17 | 37.45 | 37.17 |
| | $I''$ to $I$ | 36.87 | 36.77 | 36.86 | 36.84 | 36.74 | 36.82 |
| | 24/24 | | | | | | |
| | $I'$ to $I$ | 33.60 | 33.60 | 33.60 | 33.60 | 33.93 | 33.60 |
| | $I''$ to $I$ | 33.43 | 33.39 | 33.43 | 33.42 | 33.35 | 33.41 |
| | 36/16 | | | | | | |
| | $I'$ to $I$ | 32.03 | 32.02 | 32.02 | 32.04 | 32.41 | 32.03 |
| | $I''$ to $I$ | 31.90 | 31.86 | 31.89 | 31.90 | 32.15 | 31.88 |
| | 36/36 | | | | | | |
| | $I'$ to $I$ | 30.94 | 30.36 | 30.21 | 30.29 | 30.50 | 30.15 |
| | $I''$ to $I$ | 30.82 | 30.23 | 30.11 | 30.18 | 30.31 | 30.04 |
| 300 × 100 | 16/16 | | | | | | |
| | $I'$ to $I$ | 34.09 | 34.09 | 34.09 | 34.09 | 34.29 | 34.24 |
| | $I''$ to $I$ | 33.91 | 33.88 | 33.91 | 33.90 | 33.83 | 33.95 |
| | 24/24 | | | | | | |
| | $I'$ to $I$ | 30.54 | 30.53 | 30.53 | 30.55 | 30.76 | 30.74 |
| | $I''$ to $I$ | 30.44 | 30.42 | 30.44 | 30.44 | 30.54 | 30.57 |
| | 36/16 | | | | | | |
| | $I'$ to $I$ | 28.99 | 28.94 | 28.95 | 28.98 | 29.20 | 29.22 |
| | $I''$ to $I$ | 28.92 | 28.86 | 28.88 | 28.91 | 29.04 | 29.11 |
| | 36/36 | | | | | | |
| | $I'$ to $I$ | 28.27 | 27.26 | 27.17 | 27.26 | 27.28 | 27.46 |
| | $I''$ to $I$ | 28.18 | 27.19 | 27.11 | 27.19 | 27.20 | 27.39 |

Table 6 shows the comparisons with our proposed scheme, Yongjian–Byeungwoo scheme [26], the scheme of Shu-Kei et al. [32], and Tsai–Chang scheme [33] in terms of the visibility of watermarking, transparency of watermarking with adjustment, step-complexity of algorithms, and quality-good-looking in quality. They are shown in the report of the sophisticated experiments and summarized in Table 6.

Security is a vital concept for developing a data hiding scheme, but it was not considered in the study of [26]. A statistical stego image analyzer, called Chi-square testing, is used to demonstrate that the proposed approach can satisfy the security requirements. Figure 16 shows the test results of the cover image, the stego image embedded by the 1-bit LSB substitution method. Three curves as displayed in Fig. 16, including the red one, the green one, and the blue one, are depicted in the results of the Chi-square testing. In this analysis, the red curve indicates the outcome of the Chi-square test. In general, the red curve is reclined on the line of zero when a natural image is used as the test image. In other words, an image has a high probability of embedding random messages if the red curve is close to one. Namely, secret data have been embedded in the image. The second output is the blue curve, in which each vertical blue line represents 1Kbytes of embedded data. The third output is the green curve that remains around 0.5 if a random message is embedded. Figure 16a–c shows the undetected results since the red curves are close to zero. This result indicates that in the watermark $W$ for Lena when $D = 24$ and $MR = 24$, the watermarked

**Table 4** The ADL.TW watermark for PSNR values between $I$, $I'$, and $I''$ of the six images

| Watermark sizes | $D/MR$ | Images | | | | | |
|---|---|---|---|---|---|---|---|
| | | Airplane | Baboon | Lena | Pepper | CT0001 | MR0001 |
| 128 × 128 | 16/16 | | | | | | |
| | $I'$ to $I$ | 37.45 | 37.45 | 37.45 | 37.45 | 37.78 | 37.45 |
| | $I''$ to $I$ | 37.13 | 37.03 | 37.12 | 37.10 | 36.77 | 37.08 |
| | 24/24 | | | | | | |
| | $I'$ to $I$ | 33.88 | 33.88 | 33.89 | 33.89 | 34.24 | 33.89 |
| | $I''$ to $I$ | 33.71 | 33.66 | 33.70 | 33.70 | 33.77 | 33.68 |
| | 36/16 | | | | | | |
| | $I'$ to $I$ | 32.32 | 32.30 | 32.31 | 32.34 | 32.73 | 32.31 |
| | $I''$ to $I$ | 32.18 | 32.13 | 32.17 | 32.19 | 32.36 | 32.16 |
| | 36/36 | | | | | | |
| | $I'$ to $I$ | 31.18 | 30.66 | 30.58 | 30.63 | 30.80 | 30.46 |
| | $I''$ to $I$ | 31.06 | 30.52 | 30.47 | 30.51 | 30.66 | 30.34 |
| 300 × 100 | 16/16 | | | | | | |
| | $I'$ to $I$ | 34.61 | 34.61 | 34.61 | 34.61 | 34.77 | 34.76 |
| | $I''$ to $I$ | 34.41 | 34.37 | 34.42 | 34.40 | 34.13 | 34.49 |
| | 24/24 | | | | | | |
| | $I'$ to $I$ | 31.06 | 31.06 | 31.06 | 31.08 | 31.22 | 31.29 |
| | $I''$ to $I$ | 30.96 | 30.94 | 30.95 | 30.96 | 30.95 | 31.17 |
| | 36/16 | | | | | | |
| | $I'$ to $I$ | 29.50 | 29.46 | 29.47 | 29.51 | 29.66 | 29.77 |
| | $I''$ to $I$ | 29.42 | 29.37 | 29.40 | 29.43 | 29.44 | 29.66 |
| | 36/36 | | | | | | |
| | $I'$ to $I$ | 28.74 | 27.82 | 27.70 | 27.79 | 27.75 | 28.00 |
| | $I''$ to $I$ | 28.65 | 27.74 | 27.63 | 27.72 | 27.59 | 27.92 |

image embedded results of our method would be undetected and have the highest secrecy.

We further apply the LSB steganalysis developed in [37] to our scheme. The result is shown in Fig. 17. Next, there is yet another testing method [38] of Asymptotically Uniformly Most Powerful (AUMP) that is experimented for the detection to the embedding secret in our scheme, which is shown in Fig. 18. Fortunately, in the two LSB-based testing mentioned above for stego image, the results are the same as that of the original image (cover image). It means that the secret embedded in our scheme is robust against the popular LSB steganalysis.

## 5 Conclusions

In this study, a new visible watermarking technology is proposed to embed binary watermark images into grayscale images. The embedding method simply increases or decreases a random value on a pixel. Parameters $D$ and $R$ are used for visibility and security, respectively. In addition, embedding the information of the watermark into the watermarked image by a reversible steganographic method can successfully recover the original image. Our studies in this paper, the concerns of overflow and underflow

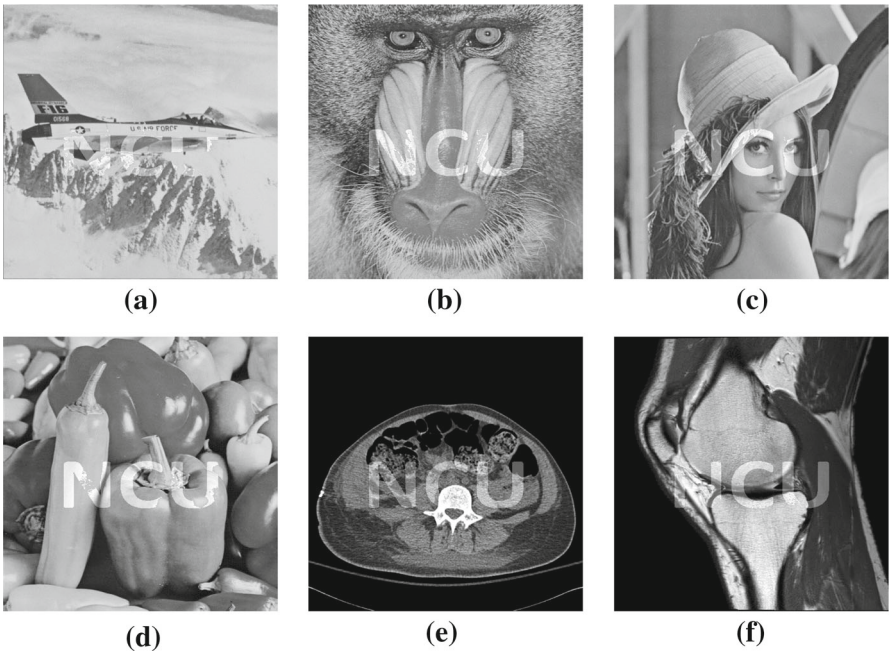


**Fig. 10** Watermarked images with Eq. (2): **a** Airplane; **b** Baboon; **c** Lena; **d** Peppers; **e** CT0001; **f** MR0001

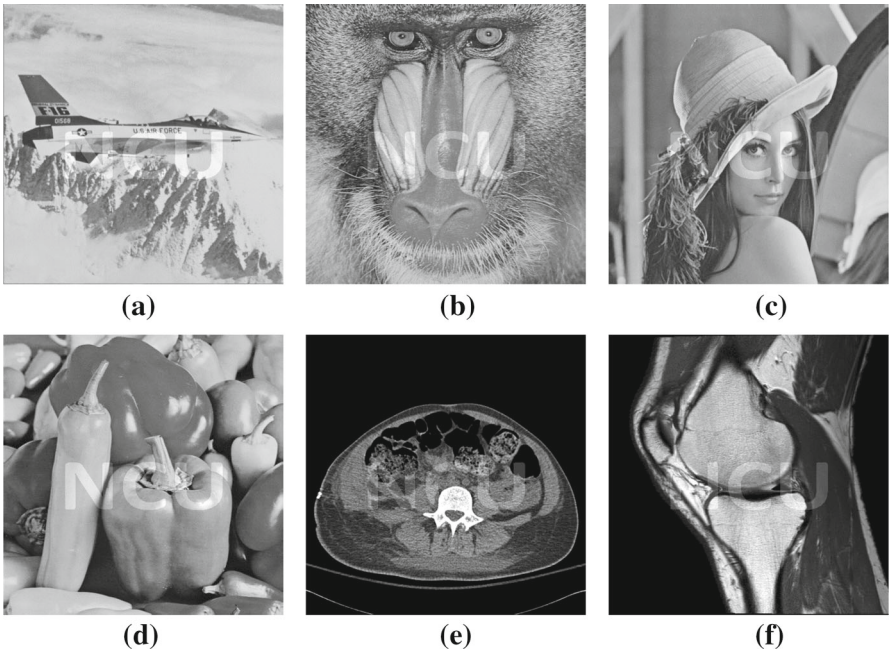


**Fig. 11** Watermarked images without using *Rb*: **a** Airplane; **b** Baboon; **c** Lena; **d** Peppers; **e** CT0001; **f** MR0001


**Fig. 12** Watermarked images of different $D$ value: **a** $D = 48$, $MR = 16$; **b** $D = 24$, $MR = 16$


**Fig. 13** Watermarked images before attacked: **a** $D = 36$, $MR = 16$; **b** $D = 36$, $MR = 36$


**Fig. 14** Watermarked images after attacked: **a** $D = 36$, $MR = 16$; **b** $D = 36$, $MR = 36$

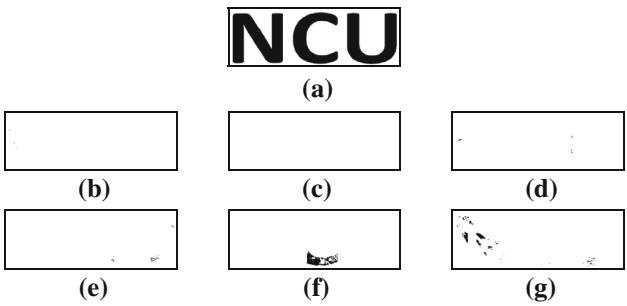


**Fig. 15** Watermarks $W$ and its changed results $W'$: **a** NCU of binary watermark; **b** Airplane; **c** Baboon; **d** Lena; **e** Pepper; **f** CT0001; **g** MR0001

**Table 5** The comparison PSNR among the scheme of Hu et al., the scheme of Yang et al., and our proposed scheme

| Images | Watermark image of 4-word in [26] | The scheme of Yang et al. [31] | NCU | ICCL | ADLTW | ADL.TW |
|---|---|---|---|---|---|---|
| Lena | 37.00 | 24.95 | 36.26 | 37.10 | 33.91 | 34.42 |
| Airplane | 39.00 | 24.30 | 36.26 | 37.11 | 33.91 | 34.41 |
| Pepper | 34.00 | N/A | 36.24 | 37.09 | 33.90 | 34.4 |
| Baboon | N/A | 23.48 | 36.18 | 37.02 | 33.88 | 34.37 |
| Average | 36.67 | 24.24 | 36.24 | 37.08 | 33.90 | 34.40 |

**Table 6** The function comparisons among visibility, transparency, step-complexity, and quality-looking view

| | Visibility | Transparency | Step-complexity | Quality-looking-view |
|---|---|---|---|---|
| Yongjian–Byeungwoo scheme [26] | Better | N/A | Simple | Inferior |
| The scheme of Shu–Kei et al. [32] | Inferior | N/A | Simple | Inferior |
| Tsai–Chang [33] | Inferior | Yes | Hard in reading | Better |
| Our proposed scheme | Better | Yes | Simple | Better |

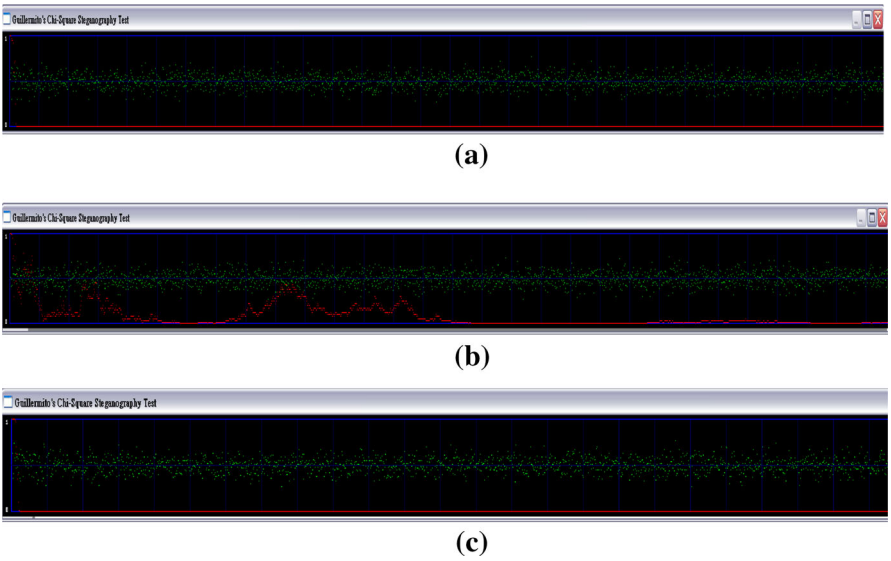


**Fig. 16** The Chi-square test results of image Lena for various approaches: **a** the cover image of Lena; **b** 1-bit LSB; **c** the watermarked image $I''$

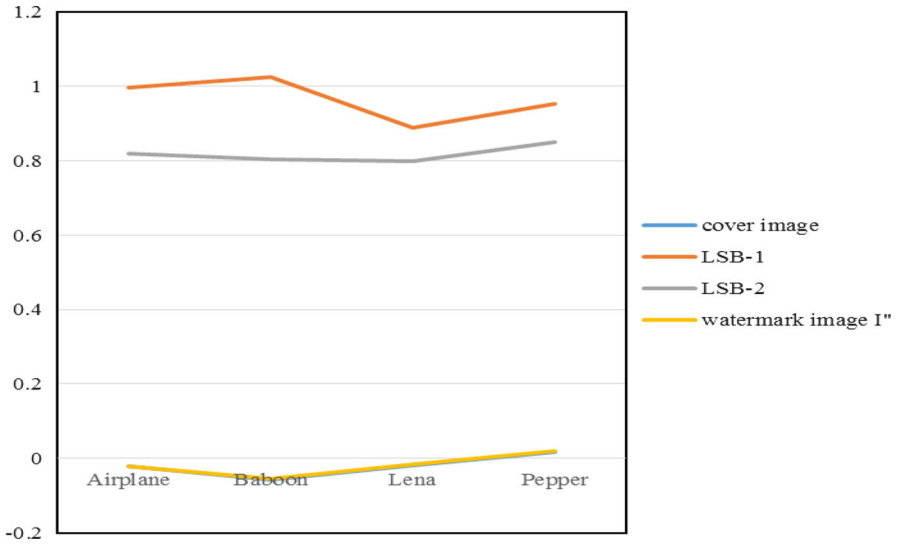


**Fig. 17** Testing result of LSB steganalysis using [37]

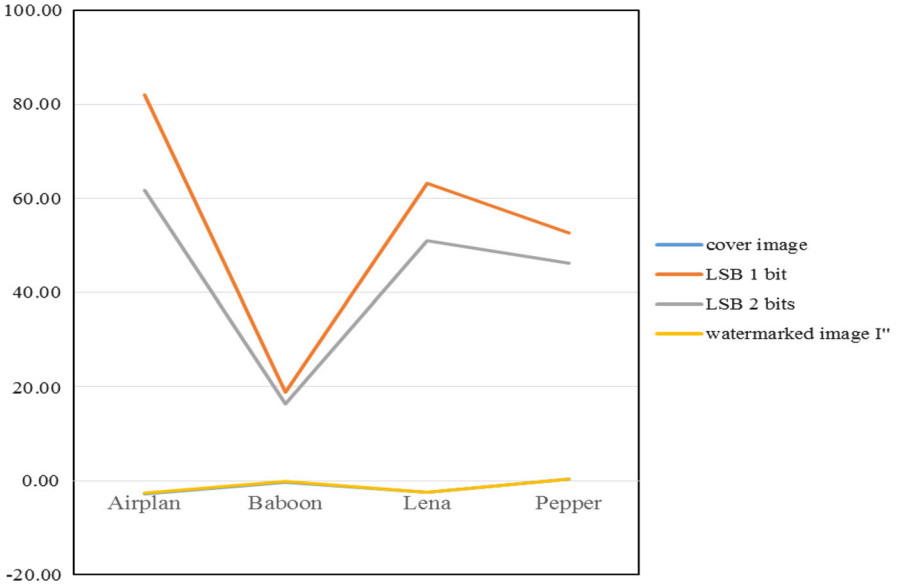


**Fig. 18** Testing result of LSB steganalysis using [38]

are both solved to make sure the reversibility feasible. The secret detections are also discussed to get rid of LSB-based testing to guarantee the robust secret embedding in the images, for example, Chi-square testing, and etc. The image quality of the reversible watermarked image is similar to the watermarked image. Experimental

results show that our method can clearly display a watermark and can completely remove the watermark, as well.

**Acknowledgments** This research was partially supported by the National Science Council of the Republic of China under the Grant NSC 100-2221-E-015-001-MY2-, NSC 102-2221-E-015-001-, 101-2221-E-008-028 -MY2, 100-2218-E-008 -013 -MY3, and NSC 101-2221-E-153-002-MY2.

## References

1. Fridrich J, Goljan M, Du R (2001) Invertible authentication. In: Proceedings of SPIE security and watermarking of multimedia contents, pp 197–208
2. Fridrich J, Goljan M, Du R (2002) Lossless data embedding-new paradigm in digital watermarking. EURASIP J Appl Signal Process 2:185–196
3. Celik MU, Sharma G, Tekalp AM, Saber E (2002) Reversible data hiding. In: Proceedings of IEEE international conference on image processing, pp 157–160
4. Celik MU, Sharma G, Tekalp AM, Saber E (2005) Lossless generalized-LSB data embedding. IEEE Trans Image Process 14:253–266
5. Celik MU, Sharma G, Tekalp AM (2006) Lossless watermarking for image authentication: a new framework and an implementation. IEEE Trans Image Process 15:1042–1049
6. Awrangjeb M, Kankanhalli MS (2004) Lossless watermarking considering the human visual system. Lect Notes Comput Sci 2939:329–336
7. Awrangjeb M, Kankanhalli MS (2005) Reversible watermarking using a perceptual model. J Electron Imaging 14:1–8
8. Tian J (2003) Reversible data embedding using a difference expansion. IEEE Trans Circuits Syst Video Technol 13:890–896
9. Alattar M (2004) Reversible watermark using the difference expansion of a generalized integer transform. IEEE Trans Image Process 13:1147–1156
10. Chang C, Lu TC (2006) A difference expansion oriented data hiding scheme for restoring the original host images. J Syst Softw 79:1754–1766
11. Weng S, Zhao Y, Pan JS, Ni R (2007) A novel reversible watermarking based on an integer transform. In: Proceedings of IEEE international conference on image processing, pp 241–244
12. Weng S, Zhao Y, Pan JS, Ni R (2008) Reversible watermarking based on invariability and adjustment on pixel pairs. IEEE Signal Process Lett 15:721–724
13. Ni Z, Shi YQ, Ansari N, Su W (2006) Reversible data hiding. IEEE Trans Circuits Syst Video Technol 16:354–362
14. Li YC, Yeh CM, Chang CC (2010) Data hiding based on the similarity between neighboring pixels with reversibility. Digit Signal Process 20:1116–1128
15. Lin CC, Hsueh NL (2008) A lossless data hiding scheme based on three-pixel block differences. Pattern Recognit 41:1415–1425
16. Lin CC, Tai WL, Chang CC (2008) Multilevel reversible data hiding based on histogram modification of difference images. Pattern Recognit 41(12):3582–3591
17. Tsai P, Hu YC, Yeh HL (2009) Reversible image hiding scheme using predictive coding and histogram shifting. Signal Process 89(6):1129–1143
18. Tseng HW, Hsieh CP (2009) Prediction-based reversible data hiding. Inf Sci 179(14):2460–2469
19. Zeng XT, Ping L, Li Z (2009) Lossless data hiding scheme using adjacent pixel difference based on scan path. J Multimed 4:145–152
20. Yang CH, Tsai MH (2010) Improving histogram-based reversible data hiding by interleaving predictions. IET Image Process 4(4):223–234
21. Yang B, Lu ZM, Sun SH (2005) Reversible watermarking in the VQ-compressed domain. In: Proceedings of the Fifth IASTED international conference on visualization, imaging, and image processing (VIIP'2005), pp 298–303
22. Chen WJ, Huang WT (2009) VQ indexes compression and information hiding using hybrid lossless index coding. Digital Signal Process 19:433–443
23. Lu ZM, Wang JX, Liu BB (2009) An improved lossless data hiding scheme based on image VQ-index residual value coding. J Syst Softw 82:1016–1024

24. Lee CF, Chen HL, Lai SH (2010) An adaptive data hiding scheme with high embedding capacity and visual image quality based on SMVQ prediction through classification codebooks. Image Vis Comput 28:1293–1302
25. Hu YJ, Kwong S, Huang J (2006) An algorithm for removable visible watermarking. IEEE Trans Circuits Syst Video Technol 16:129–133
26. Hu Y, Jeon B (2006) Reversible visible watermarking and lossless recovery of original images. IEEE Trans Circuits Syst Video Technol 16(11):1423–1429
27. Lin SD, Shie SC (2004) Improving robustness of visible watermarking schemes for images. In: The 2004 IEEE international symposium on consumer electronics, pp 11–14
28. Tsai HM, Chang LW (2007) A high secure reversible visible watermarking scheme. IEEE international conference on multimedia and expo, pp 2106–2109
29. Hu Y, Kwong S (2001) Wavelet domain adaptive visible watermarking. Electron. Lett. 37(20):1219–1220
30. Liu TY, Tsai WH (2010) Generic lossless visible watermarking—a new approach. IEEE Trans Image Process 19(5):1224–1235
31. Yang Y, Sun X, Yang H, Li C-T, Xiao R (2009) A contrast-sensitive reversible visible image watermarking technique. IEEE Trans Circuits Syst Video Technol 19(5):656–667
32. Yip SK, Au OC, Ho CW, Wong HM (2006) Lossless visible watermarking. IEEE international conference on multimedia and expo, pp 853–856
33. Tsai HM, Chang LW (2010) Secure reversible visible image watermarking with authentication. Signal Process Image Commun 25(1):10–17
34. Tsai MJ (2009) A visible watermarking algorithm based on the content and contrast aware (COCOA) technique. J Vis Commun Image Represent 20:323–338
35. Zhang X, Wang S (2006) Efficient steganographic embedding by exploiting modification direction. IEEE Commun Lett 10:781–783
36. Wu X, Memon N (1997) Context-based, adaptive, lossless image coding. IEEE Trans Commun 45(4):437–444
37. Dumitrescu S, Wu X, Memon N (2002) On steganalysis of random LSB embedding in continuous-tone images. International conference on proceedings in image processing, pp 641–644
38. Fillatre L (2012) Adaptive steganalysis of least significant bit replacement in grayscale natural images. IEEE Trans Signal Process 60(2):556–569