Removable visible image watermarking algorithm in the discrete cosine transform domain

Ying Yang
Xingming Sun
Hengfu Yang
Hunan University
School of Computer and Communication
Hunan, 410082, China
E-mail: sunnudt@163.com

Chang-Tsun Li
University of Warwick
Department of Computer Science
Coventry CV4 7AL, United Kingdom

Abstract. A removable visible watermarking scheme, which operates in the discrete cosine transform (DCT) domain, is proposed for combating copyright piracy. First, the original watermark image is divided into 16 × 16 blocks and the preprocessed watermark to be embedded is generated by performing element-by-element matrix multiplication on the DCT coefficient matrix of each block and a key-based matrix. The intention of generating the preprocessed watermark is to guarantee the infeasibility of the illegal removal of the embedded watermark by the unauthorized users. Then, adaptive scaling and embedding factors are computed for each block of the host image and the preprocessed watermark according to the features of the corresponding blocks to better match the human visual system characteristics. Finally, the significant DCT coefficients of the preprocessed watermark are adaptively added to those of the host image to yield the watermarked image. The watermarking system is robust against compression to some extent. The performance of the proposed method is verified, and the test results show that the introduced scheme succeeds in preventing the embedded watermark from illegal removal. Moreover, experimental results demonstrate that legally recovered images can achieve superior visual effects, and peak signal-to-noise ratio values of these images are >50 dB. © 2008 SPIE and IS&T. [DOI: 10.1117/1.2952843]

1 Introduction

The use of visible watermarks is to identify the ownership of works and to prevent the viewers from making unauthorized use beyond their limited copying rights.1-9 Visible watermarking is the easiest ways to identify the originator of the digital content since no special tools are required to extract the ownership information from the watermarked content. Generally speaking, visible watermarking techniques can be divided into two types: irremovable and removable. The former mainly considers the following two factors. One is that the watermark must be visible in the host image or video. That is to say, the watermark should be visible in the watermarked digital content, but must not affect the visual quality of the original art. The other is that the embedded watermark should be strongly resistant to unintended editing and malicious attacks. By contrast, removable visible watermarking techniques provide another efficient solution to copyright protection problems. The original digital content is marked with an erasable pattern, such as a copyright notification, before distribution or release on the Internet for free viewing. As described in Refs. 7-9, only removable visible watermarking techniques are suitable for some applications, such as the following two:

1. Digital images and videos play an important role in people’s life and are gradually taking the places of their classical analog counterparts. Content providers can distribute or share them via computer networks for free preview and download after embedding the removable visible copyright information. Anyone who has interest in their content can gain higher definition versions by purchasing the secret key from the producers for removing the copyright information.

2. In a commercial environment, a company can gain profits by releasing free-trial software to users for a limited period of time, but it would stop running once the trial period ends. For this purpose, a removable visible watermark can be embedded into the software-dependent digital content (e.g., engineering drawing) to affect its normal use. In order to continue working, the user needs to activate the software and remove the watermark from the digital content with the license key.

Although there are many potential applications requiring the removable visible watermarking, most papers available in the literature focus on the development of irremovable techniques. Many reversible (also called invertible or distortion-free) methods (e.g., Refs. 12-14) have been in-
troduced for invisible watermarking, but they can hardly be extended to removable visible watermarking since the two types of watermarking algorithms are designed to meet different requirements of robustness for different application scenarios.\footnote{7} Thus far, as far as the authors’ knowledge goes, only a few visible watermarking schemes\footnote{7,9} are removable. Mohanty et al.\footnote{4} proposed a discrete cosine transform (DCT) domain adaptive watermarking algorithm. The adaptive factors are computed via a mathematical model developed by exploiting the human visual system (HVS) properties. But the texture features of the original watermark are not considered while calculating these factors and the embedded watermark is legally irremovable. Generally, removable visible watermarking may be further classified into the following two categories: lossy\footnote{7} and lossless.\footnote{8,9} For lossy schemes, the irreversibility is mainly due to the inherent problems such as modulation, overflows, underflows, and rounding error occurring during embedding and removing process. Hu et al.\footnote{7} introduced a strategy for constructing a removable visible watermarking system and implemented it in wavelet domain. The DWT coefficients of the watermark used for embedding are selected by a key-controlled template. However, in some circumstances, this approach fails to authenticate the property of the digital content and attackers may remove the marked copyright information without the secret key with a good chance of success. For lossless schemes, the lossless recovery of the original signal is achieved by employing compression technique\footnote{8} or reversible embedding function.\footnote{9} However, despite their merit of lossless image recovery, the two schemes\footnote{8,9} can only work in the spatial domain, making it unsuitable for most Internet applications, where media transactions are carried out in compressed formats, such as JPEG and JPEG using DCT and DWT.

Aiming at maintaining applicability, a DCT-based removable visible watermarking algorithm for images is proposed. We embed a variant of the original copyright pattern as the visible watermark into the host image. The generation of such a watermark is key controlled; thus, without the secret key the same variant of the original watermark cannot be derived from the marked image. Hence, an adversary has no idea which watermark version has been embedded into the original host image so that the visual quality of the resulting image obtained by the unauthorized users without the correct key may be very poor. In addition, we also expect the visual quality of the resulting image attained by an authorized user after removing the embedded visible watermark to be as high as possible. As will be seen later, this objective is arrived at by estimating the original adaptive factors. Generally, the more accurate the estimated factors are, the better the visual quality of the recovered image is. In order to approximately predict the original adaptive factors, some essential data are pseudorandomly hidden in the high-frequency subbands of the marked image during the embedding process. Hence, the peak signal-to-noise ratio (PSNR) values of the legally recovered images are highly increased.

The rest of the paper is organized as follows. In Sec. 2, we present an approach to preprocess the watermark in order to get a variant for embedding and to compute the adaptive scaling and embedding factors in detail. Sec. 3 describes the strategies to embed and remove the watermark. Some experimental results for evaluating the performance of the proposed algorithm are given in Sec. 4. Section 5 draws the conclusions.

\section{Watermark Preprocessing Procedure}

The watermark preprocessing procedure consists of two subprocedures: (i) the design of the watermark preprocessing module and (ii) the determination of the scaling factor for the host image and the embedding factor for the preprocessed watermark. Watermark preprocessing is necessary because key-dependent preprocessing of the original watermark image \( I^w \) for constructing a variant of the original watermark is to ensure the infeasibility of removing the embedded visible watermark by using the incorrect keys (i.e., illegal removal), and the scaling and embedding factors are to insure the conformity with HVS model. In the proposed algorithm, Eq. (1) is employed to embed the visible watermark,

\[ f_m(i,j) = \alpha_m \times f_m^0(i,j) + \beta_m \times f_m^w(i,j), \quad i = 1, 2, \ldots, 8, \]

\[ j = 1, 2, \ldots, 8, \quad m = 1, 2, \ldots, M, \]

where \( f_m^0(i,j) \), \( f_m^0(i,j) \), and \( f_m^w(i,j) \) denote the \((i,j)\)’th DCT coefficient of the \( 8 \times 8 \)-element block \( m \) of the watermarked image \( F \), host image \( I \); and preprocessed watermark image \( P^w \), respectively. \( \alpha_m \) and \( \beta_m \) are the adaptive scaling and embedding factors for the \( m \)’th block of host image \( I \) and preprocessed watermark image \( P^w \) respectively. \( M \) is the total number of blocks. Note in this work, we use the same symbol (e.g., \( I^w \)) to represent the same image in both spatial and DCT transform domains.
2.1 **Design of the Watermark Preprocessing Module**

The preprocessed visible watermark image $I^w$ needs to meet two fundamental requirements:

1. The intensity of pixels of preprocessed watermark image $I^w$ should be highly sensitive to the change of the real-valued secret key, $k \{k_i \in (0, 1)\}$, to be used in the preprocessing. In other words, the value $\partial I^w / \partial k = \partial f(I^w, k) / \partial k$ should be large, where $I^w = f(I^r, k), f$ is a preprocessing function, and operator $\partial / \partial$ performs partial derivative.

2. The preprocessed watermark image $I^w$ needs to preserve the pattern of the original visible watermark image $I^r$.

Obviously, there exist various functions $f$ that meet the two requirements. We propose a preprocessing function as shown in Fig. 1 and described it as follows:

2.2.1 **Watermark preprocessing function**

Step 1: Divide the original visible watermark $I^r$, into non-overlapping $16 \times 16$-pixel blocks and transform these blocks into DCT domain with the $m$'th block denoted as $I^w_m$.

Step 2: For each watermark block $I^w_m$,

Step 2.1: Use a chaotic logistic map

\[
x_{n+1} = ux_n(1 - x_n)
\]

with the secret key $k$ in the range $(0, 1)$ as initial value $x_0$ (i.e., $x_0 = k$) to generate a chaotic set $x = \{x_1, x_2, x_3, \ldots\}, u$ is a positive number determining the characteristic of $x$. The values of the elements of $x$ are also all in the range of $(0, 1)$. Meanwhile, to ensure that the logistic map falls in a chaotic state, $u$ should be selected from $(3.599456, 4)$. The reader is referred to Ref. 15 for more theoretical details.

Step 2.2: Randomly choose $16 \times 16$ elements with values in the range $[p', 1)$ from the set $x$ to constitute a matrix $D_m$ of $16 \times 16$ elements, where $p'$ is a predetermined real number in the range of $(0, 1)$ to ensure that the pattern of the original watermark image, $I^r$, is preserved in the preprocessed watermark image, $I^w$. Thus, the value of $p'$ should deviate much from 0.

Step 2.3: Obtain key-based matrix $I^w_m$ by performing element-by-element multiplication on $m$'th original watermark image block $I^w_m$ and randomly chosen matrix $D_m$.

Step 3: Perform inverse DCT on all key-controlled $I^w_m$'s to produce preprocessed visible watermark image $I^w$.

The chaotic sequence is random and highly sensitive to initial conditions, in this work the secret key; thus, the random matrix $D_m$'s generated by the chaotic logistic map with different initial values helps to produce different preprocessed watermark image versions $I^w$'s, which satisfies the first requirement mentioned above. On the other hand, in order to meet the second requirement, the minimal element value of $p'$ for constituting the random matrix $D_m$ in Step 2.2 is controlled to prevent the preprocessed watermark image $I^w$ from deviating too much from the pattern of the original image $I^r$. Hence, it is believed that the chaotic logistic map is an effective approach to preprocessed the original image $I^r$. Note that it is also worth mentioning that the chaotic logistic map is just one possible scheme for pre-processing the original watermark. Other suitable key-based pseudorandom generators (e.g., SHA-based) and pre-processing techniques may be used provided they serve the same pre-processing purpose. Because the focus of this work is to introduce the removable visible watermarking, we will not discuss other options of watermark preprocessing further.

2.2 **Determination of Scaling Factor $\alpha_m$ and Embedding Factor $\beta_m$**

The scaling and embedding factors, $\alpha_m$ and $\beta_m$, of Eq. (1) are used to determine the weights of the host image and watermark pattern. That is, they determine the visibility of the watermark pattern and robustness in the marked image. Unlike many existing approaches in which only the host image is considered as the side information/parameter when determining the two factors, the proposed algorithm takes the texture features of both the host image $I^h$ and original watermark image $I^r$ into account in order to better conform to HVS characteristics. There are two aspects of the HVS to take into account when formulating scaling factor $\alpha_m$:

1. First, the HVS is more sensitive to changes in midluminance areas Ref. 16. That is, to maintain the quality of the marked image, assigning greater value of the scaling factor $\alpha_m$ for the midluminance areas is desirable. However, we also want the watermark pattern to be visible enough but not too intrusive. There-
fore, the optimal choice would be assigning a greater scaling factor $\alpha_m$ in the midluminance areas and attenuating its value at darker and brighter components. Intuitively, the histogram of the scaling factor $\alpha_m$ is roughly parabola shaped.

2. Second, because the HVS is less sensitive to changes made in highly textured regions, it is therefore helpful to use lower value for the scaling factor in textured regions.

The steps for determining the scaling and embedding factors, $\alpha_m$ and $\beta_m$, are as follows:

Step 1: Transform all the nonoverlapping $8 \times 8$-pixel blocks of the host image $I^h$ and the preprocessed watermark image $I^{pw}$ into the DCT domain.

Step 2: Because most energy is concentrated in low-frequency components, especially the dc coefficients, we compute the scaling factor, $\alpha_m$, according to the dc coefficients of the host image $I^h$ and preprocessed watermark image $I^{pw}$. Reininger and Gibson,\textsuperscript{17} demonstrated that, for many images, the dc coefficients are best approximated by a normal distribution. Here, the distribution models of the dc coefficients of the host image $I^h$ and the preprocessed watermark image $I^{pw}$ are expressed, respectively, as

$$I^h_{m}(0,0):N(\mu_1, \sigma^2_1), \quad m = 1,2,3, \ldots, M$$

and

$$I^{pw}_{m}(0,0):N(\mu_2, \sigma^2_2), \quad m = 1,2,3, \ldots, M,$$

where $\mu_1$ and $\sigma^2_1$ ($\mu_2$ and $\sigma^2_2$) are the mean and variance of the dc coefficients of the host image $I^h$ (the preprocessed watermark image $I^{pw}$), respectively. Let variable $V^h_m$ be $V^h_m=I^h_{m}(0,0)+I^{pw}_{m}(0,0)$, $m=1,2,3,\ldots,M$. $V^h_m$’s also obey normal distribution with the mean equal to $\mu_1+\mu_2$ and the variance equal to $\sigma^2_1+\sigma^2_2$ because the dc coefficients of the host image $I^h$ and the preprocessed watermark image $I^{pw}$ are independent. Now, to create a parabola-shaped scaling factor $\alpha_m$, we can formulate it as

$$\alpha_m = \frac{1}{\sqrt{2\pi(\sigma^2_1 + \sigma^2_2)}} \exp\left(-\frac{(V^h_m - (\mu_1+\mu_2))^2}{2(\sigma^2_1+\sigma^2_2)}\right). \quad (5)$$

Mean value $\mu_1$ and variance value $\sigma^2_1$ of the dc coefficients of the host image $I^h$ are defined, respectively, as

$$\mu_1 = \frac{1}{M} \sum_{m=1}^{M} I^h_{m}(0,0) \quad \text{and}$$

$$\sigma^2_1 = \frac{1}{M} \sum_{m=1}^{M} [I^h_{m}(0,0) - \mu_1]^2. \quad (7)$$

By the same token, the mean value $\mu_2$ and variance value $\sigma^2_2$ of the preprocessed watermark image $I^{pw}$ can be obtained.

This step reflects the first aspect of the HVS we mentioned earlier because we only take the dc components, which convey the luminance of the corresponding blocks, of the images into account.

Step 3: To take the second aspect of the HVS into account in order to improve the performance, the scaling factor, $\alpha_m$, is corrected by involving ac coefficients, which mainly reflect the texture features of the image. More energy should be received from the preprocessed watermark image $I^{pw}$, where the host image $I^h$ is strongly textured because HVS is less sensitive to such regions. Likewise, more information from the highly textured areas of the preprocessed watermark image $I^{pw}$ should be embedded to put up resistance to unauthorized removal and to enhance the visibility of watermark $I^{pw}$ in watermarked image $I^w$. It has been observed that in strongly textured block, energy tends to be more evenly distributed among the ac coefficients; thus, the variance of the ac coefficients tends to be smaller.\textsuperscript{4} According to the above analysis of the HVS characteristics, we therefore consider that the scaling factor, $\alpha_m$, which is used to determine the weight of the host image $I^h$ in the marked image $I^w$, is in proportion to the $m$’th block variances of the ac coefficients of both the host image $I^h$ and the preprocessed watermark image $I^{pw}$.

To describe the direct proportional relationship, we adopt the simple additive rule in this paper, i.e., we assume that the scaling factor $\alpha_m$ is in direct proportion to variance $\chi_m^r$, i.e., $\alpha_m \propto \chi_m^r$, where $\chi_m^r=\chi_m^f+\chi_m^e$.

The two parameters $\chi_m^f$ and $\chi_m^e$ are the variances of the insignificant ac coefficients of the host image $I^h$ and the preprocessed watermark image $I^{pw}$, respectively. A coefficient is deemed as insignificant if its quantized value is equal to 0. After the amendment, Eq. (5) can be rewritten as

$$\alpha_m = \frac{1}{\sqrt{2\pi(\sigma^2_1 + \sigma^2_2)}} \exp\left(-\frac{(V^h_m - (\mu_1+\mu_2))^2}{2(\sigma^2_1+\sigma^2_2)}\right) + \chi_m^r, \quad (8)$$

where $\chi_m^r$ is the normalized version of $\chi_m$ calculated according to Eq. (9), in which $\chi_m^f$ is the nature logarithm of $\chi_m^f$, i.e., $\chi_m^f=\ln(\chi_m^f)$, $\chi_{min}$ and $\chi_{max}$ are, respectively, the minimum and maximum values of all the $\chi_m^f$'s

$$\chi_m^f = \frac{0.8 \times (\chi_m^f - \chi_{min})}{(\chi_{max} - \chi_{min})} + 0.1. \quad (9)$$

$\chi_m^f$ is so defined in Eq. (9) to make the scaling factor $\alpha_m$ controlled in a narrow range to keep the visual quality of the watermarked image $I^w$.

The computational method for the variances $\chi_m^f$ and $\chi_m^e$ is as follows. Let $S_m$ denote the coordinates whose corresponding DCT coefficients of the preprocessed watermark image block $I^{pw}_{m}$ are insignificant and $S_m'$ be a set created by randomly removing one element from $S_m$. Since there is only one element in $S_m-S_m'$, we also use $S_m-S_m'$ to denote its corresponding coordinates. During embedding, the dc coefficient of the $m$'th block of the host image $I^h$ is hidden in the $(S_m-S_m')$’th coefficient of the watermarked image $I^w$, which facilitates the retrieval of the dc coefficients of the host image $I^h$ from $\cup(S_m-S_m')$ (i.e., the union of $S_m-S_m'$, $m=1,2,\ldots,M$) for the estimation of the two factors $\alpha_m$ and $\beta_m$ during the watermark removal process, and also leads to the impossibility of finding out the original $(S_m-S_m')$’th DCT coefficient of the $m$’th block $I^h_m$ of the host image from the watermarked image block $I^w_m$ by the attackers. Hence, the variance $\chi_m^e$ of the $m$’th host image block $I^h_{m}$ is figured out according to Eq. (10) with coeffi-
coefficients in set $S_m^r$ rather than $S_m$, where operator $|\cdot|$ denotes the cardinality of a set and $\eta_m$ denotes the mean value of all the selected $I_m^h(i,j)$’s of the host image block $I_m^h$,

$$
\chi_m^h = \frac{1}{|S_m^r|} \sum_{(i,j) \in S_m^r} (I_m^h(i,j) - \eta_m)^2.
$$

(10)

By the same token, we can obtain the variance $\chi_m^{pw}$ for the preprocessed watermark image block $I_m^{pw}$.

Step 4: Let $\beta_m = 1 - \alpha_m$ and scale $\alpha_m$ and $\beta_m$ to the ranges $[y', y'']$ and $[z', z'']$ to avoid obtrusive embedding, where $y', y'', z', z''$ are predetermined empirical constants.

3 Visible Watermark Embedding and Removal

3.1 Visible Watermark Embedding

The flowchart of the complete watermarking scheme is illustrated in Fig. 2. Given the availability of the preprocessed watermark $I_m^{pw}$, scaling factor $\alpha_m$, and embedding factor $\beta_m$, the steps for embedding the watermark into the host image $I^h$ are as follows:

Step 1: For each host image block $I_m^h$ and the preprocessed watermark block $I_m^{pw}$, generating image block $f_m$ by employing Eq. (1) to add the significant DCT coefficients of the preprocessed watermark image block $I_m^{pw}$ to the corresponding coefficients of the host image block $I_m^h$.

Step 1.2: Hide the value $\beta_m \times [I_m^h(0,0) - I_m^{pw}(0,0)]/10.0$ into the $(S_m - S_m^r)$th coordinate of the watermarked image block $r'_m$ for facilitating the retrieval of the dc coefficient of the host image block $I_m^h$ from the marked image block $r'_m$ during the watermark removal process.

Step 2: Perform inverse DCT on all the marked image blocks $r'_m$’s to produce the watermarked image $F$.

The reason for embedding only the significant coefficients is to make the other (i.e., insignificant) DCT coefficients of the host image $I^h$ approximately unchanged in the watermarked image $F$; thus, when estimating the scaling factor $\alpha_m$ and the embedding factor $\beta_m$ (see Sec. 3.2), the variance $\chi_m^h$ of the host image $I^h$ can be calculated according to these approximate DCT coefficients directly taken from the watermarked image $F$ without referring to host image $I^h$. Because these significant ones hold most of the energy and information of the preprocessed watermark image $I_m^{pw}$, the experimental results prove that these data are sufficient to reveal the details of the visible watermark image $F$ in the marked image $F$. It is due to the importance of the dc coefficients of the host image $I^h$ to reconstruct the two factors $\alpha_m$ and $\beta_m$ for removing the embedded watermark that we hide $\beta_m \times [I_m^h(0,0) - I_m^{pw}(0,0)]/10.0$ instead of original dc coefficient $I_m^h(0,0)$ of the host image to avoid degrading the quality of watermarked image $F$ and to retrieve the original dc coefficients of the host image $I^h$ from the $(S_m - S_m^r)$th coefficient of the watermarked image $F$.

3.2 Visible Watermark Removal

The procedure of removing the embedded watermark is the inverse operation of the embedding process. In order to recover the host image, scaling and embedding factors $\alpha_m$ and $\beta_m$ must be estimated. Experimental results show that the estimated version of $\alpha_m$ and $\beta_m$ is accurate enough to remove the watermark without inflicting noticeable distortion on the recovered image.

Given the availability of the original watermark image $I^w$, private key $k$, and the watermarked image $F$, the steps for estimating $\alpha_m$ and $\beta_m$ are as follows:

Step 1: Produce the preprocessed watermark $I_m^{pw}$ according to the original watermark image $I^w$ and the private key $k$, and then divide the preprocessed watermark $I_m^{pw}$ and the
The watermarking image \( F \) is divided into nonoverlapping blocks of size 8×8 pixels, and transform these blocks into the DCT domain.

Step 2.1: Pick the \((i,j)\)'th DCT coefficient, \( I_m(i,j) \), where \((i,j)\) is the only element of \((S_m - S_m')\) and take \( I_m(i,j) \times 10.0 + I_m(0,0) \) as the approximate value of the DCT coefficient of the block \( I_m(i,j) \) of the original host image \( I^h \). Namely, \( I_m(i,j) = I_m(i,j) \times 10.0 + I_m(0,0) \). This can be derived from Eq. 1.

Step 2.2: Pick the DCT coefficients from the set \( S_m \) and we get \( I_m(i,j) \), \( \forall (i,j) \in S_m \), since no watermark energy is added to such DCT coefficients of the host image \( I^h \) whose corresponding coordinates are in \( S_m \). Thus, the variance of the insignificant coefficients of the host image \( I^h \) can be roughly recalculated according to Eq. 1.

Step 3: Create a parabola-shaped scaling factor \( \alpha_m \) using these approximate dc coefficients \( I_m(i,j) \)'s of the host image \( I^h \) and dc coefficients \( I_m(0,0) \)'s of the preprocessed watermark image \( I^m \) as in Step 2 of Sec. 2.

Step 4: Correct the scaling factor \( \alpha_m \) by plugging the variance \( \chi_m^h \) of the host image block \( I_m(i,j) \) and the variance \( \chi_m^m \) of the preprocessed watermark image block \( I_m(i,j) \) in the equations defined in Sec. 2 to obtain the final versions of \( \alpha_m \) and \( \beta_m \).

Then, the unmarked image can be obtained by using Eq. 1 to remove the significant DCT coefficients of the preprocessed watermark \( I^m \) from the watermarked image \( I^h \),

\[
P_m(i,j) = \frac{I_m(i,j) - \beta_m \times I_m(i,j)}{\alpha_m}, \quad i = 1, 2, \ldots, 8,
\]

Note in the visible watermarking removal procedure, \( S_m \) should be the same as in the embedding procedure. Because the \((8,8)\)'th DCT coefficient is insignificant, for simplicity, we set \( S_m = (8,8) \) in our experiments.

4 Experimental Results

The proposed algorithm has been implemented and tested on a number of images of 256×256 pixels (obtained in part from Ref. 18) and different watermark patterns of 256×256 pixels for evaluating its performance. These images are of different texture characteristics. In the experiments, the parameter of the chaotic map, \( u = 3.698789 \) [see (2)] is arbitrarily chosen from the range \((3.5699456, 4)\) to ensure that the logistic map is in a chaotic state, \( p^* = 0.41 \) (see Step 2.2 of Sec. 2.1) is to avoid the loss of the pattern of the original watermark \( I^w \), and \( y = 0.15, z = 0.20 \) (see Step 4 of Sec. 2) are empirical values, determined by lots of experiments, to strike a balance between the visual quality of the host image \( I^h \) and the visibility of the embedded visible watermark \( I^w \).

4.1 Diversity of the Preprocessed Watermark Image and the Watermarked Image

When the chaotic logistic map receives different keys as its input, different chaotic matrixes \( D_m \)'s will be generated to preprocess the original watermark image \( I^w \), thus the preprocessed watermark versions \( I^m \)'s are different. Moreover, a variety of watermarked image versions \( I^w \)'s will be yielded after embedding these preprocessed watermarks into the host image \( I^h \).

Figure 3 shows the experimental results carried out on host image lena and watermark image Fig. 3(a). From Fig. 3, we know that the two different watermark versions [Figs. 3(b) and 3(c)] are produced under the control of two slightly different keys and that the two watermarked images [Figs. 3(d) and 3(e)] are perceptually equivalent. However, their difference image of the two watermarked images as shown in Fig. 3(f) points out that they differ from each other significantly, with the watermark pattern clearly visible. This again manifests that the two images carry distinct watermark versions.

4.2 Comparison of Legal and Illegal Watermark Removal

In the paper, given the availability of the algorithm, watermarked image, and the original watermark, if the embedded visible watermark is removed by using the correct key, then we call such removal “legal removal” instead of “illegal removal.” Such a definition is compatible with the concept of Ref. 7.

The removal of the embedded visible watermark for high-quality restoration of the original host image \( I^h \) depends on the secret key rather than the unavailability of the algorithm. It is theoretically infeasible for unauthorized users to exactly predict the 16×16-element chaotic matrix \( D_m \) since the secret key \( k \) is taken from an infinite space. Therefore, an unauthorized user without the correct secret key cannot obtain the recovered image with satisfactory quality because it is unrealistic, if not impossible, to estimate which watermark version \( I^m \) is embedded into the
Moreover, we compare the performance of the proposed algorithm with that of the Hu et al. algorithm and the results are also listed in Table 1. From Table 1, we know that the PSNR values of the images (>50 dB) legally recovered with the proposed algorithm, using the correct secret keys, are always higher than those of the Hu et al. algorithm (~44 dB). On the other hand, the PSNR values of the images (~26 dB) illegally reconstructed with the proposed algorithm, using the incorrect private keys, are always lower than those of Hu et al. (~37 dB). This is clear evidence that our algorithm is superior to Hu et al.

4.3 Attack Experiments

In addition to the above illegal recovery scheme (which removes the embedded visible watermark via trying different keys), an adversary may recover the original image through conducting other attack schemes, such as using image processing software, collusion attack, etc. We, therefore, briefly discuss the performance of the proposed watermarking method to be against some vital attacks in this section.

4.3.1 Image processing software attack

With this type of attack, there are two possible attack categories: manual and automatic. The former is using some proper pixel values to replace the watermarked pixels of the covered image. However, this method is quite difficult and time consuming to completely eliminate the embedded watermark so as to obtain a reconstructed version of the host image with satisfactory visual quality, especially for a marked image with a strongly textured watermark embedded.

The latter attack scheme is employing the integrated image-processing operations in the image-processing tools, such as median filtering, image smoothing, and image sharpening, to roughly recover the original host image. Just as expected, many experiments demonstrate that these common processing operations cannot completely remove the embedded visible watermark as well. Figure 5 shows some results obtained by testing the above three operations.

4.3.2 Collusion attack

Another attack scheme is the collusion attack, which recovers the original host image through combining its different watermarked versions. Thus, an attacker may remove the embedded visible watermark by averaging these different

Table 1 PSNR comparison of the proposed algorithm and Hu's (inducible). Legal removal and illegal removal denote that the recovered images are obtained by using the correct and incorrect keys to remove the embedded visible watermark, respectively.

<table>
<thead>
<tr>
<th></th>
<th>Legal removal</th>
<th>Illegal removal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hu's Proposed</td>
<td>Hu's Proposed</td>
</tr>
<tr>
<td>F-16</td>
<td>44.15 50.51</td>
<td>37.89 26.67</td>
</tr>
<tr>
<td>Lena</td>
<td>44.15 54.38</td>
<td>38.25 26.49</td>
</tr>
<tr>
<td>Peppers</td>
<td>44.13 53.14</td>
<td>37.94 26.61</td>
</tr>
<tr>
<td>Girl</td>
<td>43.51 53.76</td>
<td>38.68 26.68</td>
</tr>
<tr>
<td>Sailboat</td>
<td>44.16 53.73</td>
<td>36.83 26.89</td>
</tr>
</tbody>
</table>

watermarked image $F$. On the other hand, an authorized user, holding the correct secret key, could obtain a high-quality recovered version because he/she knows which preprocessed watermark version should be subtracted from the watermarked image. For instance, Fig. 4(a) indicates that we succeeded in removing the embedded watermark by using the correct secret key. However, as demonstrated in Fig. 4(b) obtained by using incorrect user key, much energy residue of the watermark still exists in the illegally recovered image. The reason is that different private keys correspond to different preprocessed watermark versions for embedding so that the watermarked images differ from each other. In addition, more experimental results by using correct and incorrect keys to remove the embedded watermark are summarized in Table 1. Just as expected, legally reconstructed images have excellent visual quality and the PSNR values of these images are over 50 dB, which are far greater than those of illegally recovered ones (<30 dB). This is because the embedded watermark version depends on the private key so that an unauthorized user has no idea about which watermark version should be subtracted from the watermarked image. Thus, the watermark preprocessing technique for the generation of the various preprocessed watermark image versions to modulate the original watermark with the user key can successfully prevent illegal removal.

![Fig. 5 Illustrations of the resulted images obtained by performing some common image-processing operations to the watermarked image. (a), (b), and (c) are obtained by 3×3 median filtering, 3×3 image smoothing, and Laplacian image sharpening, respectively.](Image)
copies. We, therefore, have tested such collusion attacks on a number of images and, from the experimental results, observed that the visible watermark can be clearly seen from the final image that is resulted from averaging. This implies that the watermark component is still in the final image even if these different marked images have experienced averaging. Thus, this malicious attack scheme still fails to remove the embedded watermark. For visualization, we show an illustration of the related experimental results in Fig. 6. The clearly visible watermarks in Figs. 6(a)–6(c) confirm the conclusion mentioned above.

### 4.4 Robustness Against Compression

Generally, images transmitted on computer networks are in compressed formats. Therefore, the degree of tolerance to compression can also be deemed as an indicator for evaluating the algorithms’ performance. To verify the robustness of the proposed algorithm to the compression schemes, both ordinary JPEG compression and new JPEG compression (i.e., JPEG2000) have been carried out.

#### 4.4.1 Ordinary JPEG compression

Figure 7 illustrates the experimental results conducted on
the host image, Lena. The clearly visible watermarks in the compressed images as shown in Figs. 7\textit{a}, 7\textit{d}, and 7\textit{g} demonstrate that the proposed scheme is robust against compression. The images as shown in the second and third columns of Fig. 7, with the watermark legally removed, are still acceptable if the quality factors of the corresponding compressed images are >80. Additional experimental results for demonstrating the robustness of the algorithm are shown in Fig. 8, which again suggests that the proposed watermarking system is indeed robust against ordinary JPEG compression to a certain extent.

4.4.2 JPEG2000 compression

In addition to the ordinary JPEG compression, we also perform a new compression scheme (i.e., JPEG2000) to verify the robustness of the introduced algorithm.

Figure 9 shows the experimental results tested on the host image Lena. From its compressed versions Figs. 9\textit{a}, 9\textit{d}, and 9\textit{g}, it is easy to see the embedded visible watermarks can survive JPEG2000 compression attack. Furthermore, the good visual qualities of the images in the second and third columns of Fig. 9 with the watermarks legally removed demonstrate that the proposed scheme succeeds in resisting JPEG2000 compression. In addition, we apply the JPEG2000 compression to the other four typical images with visible watermarks embedded, using different compression ratios. Then, the embedded visible watermarks are legally removed from these compressed images, thus yielding the nonwatermarked images. The experimental results for measuring the qualities of these nonwatermarked images are illustrated in Fig. 10. From Fig. 10, we can see the PSNR values of these legally recovered images are

![Fig. 8 Robustness to ordinary JPEG compression. The PSNR values are calculated according to the images legally recovered from the compressed marked images and their corresponding original ones.](image)

![Fig. 9 Robustness against JPEG2000 compression. Images shown in (b), (e), and (h) are recovered from compressed images (a), (d), and (g) with compression ratios 2:1, 3:1, and 4:1, respectively. The PSNR values of these recovered images are 47.81, 45.45, and 42.12 dB, respectively. Images (c), (f), and (i) in the third column are the zoom-in versions of the upper-left quarters of (b), (e), and (h), respectively.](image)
Acknowledgment
This work is supported in part by National Basic Research Program 973 (Grant No. 2006CB303000), in part by National Natural Science Foundation of China (Grant No. 60573045), and in part by National Natural Science Foundation of China (Grant No. 60736016).

References

Yang et al.: Removable visible image watermarking algorithm...
Hengfu Yang received his BS in management information system from Xiangtan University, China, in 1996, and MS in computer application from Guizhou University, China, in 2003. He is currently pursuing his PhD in computer application at the School of Computer and Communication from Hunan University, China. His research interests include information hiding and image processing.

Chang-Tsun Li received his BS in electrical engineering from Chung-Cheng Institute of Technology (CCIT), National Defense University, Taiwan, in 1987, MS in computer science from the U.S. Naval Postgraduate School, in 1992, and PhD degree in computer science from the University of Warwick, in the UK, in 1998. He was an associate professor from 1999 to 2002 in the Department of Electrical Engineering at CCIT and a visiting professor in the Department of Computer Science at the U.S. Naval Postgraduate School in the second half of 2001. He is currently an associate professor in the Department of Computer Science at the University of Warwick. His research interests include image processing, pattern recognition, computer vision, multimedia security, and content-based image retrieval.