TWO-LAYER IMAGE WATERMARKING WITHOUT RESORTING TO THE ORIGINAL IMAGE

Jian-Chyn Liu and Shu-Yuan Chen@

Department of Computer Engineering and Science

Yuan-Ze University, Nei-Li, Chung-Li, Taoyuan, Taiwan, 320, R.O.C.

@Email:cschen@cs.yzu.edu.tw

ABSTRACT

A two-layer image watermarking without resorting to original image is proposed in this study. Two layers of watermarking algorithms are employed to hide the same watermarks in the spatial domain of an image simultaneously. More specifically, the watermarking in layer one can resist high-frequency destruction, while that in layer two can resist low-frequency destruction. Although the image is modified through two layers of watermarking, the watermarks are still invisible. In addition to robustness and invisibility, the proposed watermarking has other advantages as listed below. First, the proposed embedding technique is based on intra relative within the original image rather than inter relative between the original and watermarked images such that the original image is not required during detection process. Second, the proposed watermark is composed of fixed and variable parts. The former leads that the location and the existence of watermark can be verified directly without referring to the original watermark. The latter is involved to increase flexibility and variety of watermarks. Third, the watermark is a short serial number such that it can be duplicated. Due to the duplicity, the majority voting strategy can be employed to facilitate watermark detection. Finally, the proposed method is simple and fast. It takes only one to two seconds either in embedding stage or in detection stage. Various experiments have been conducted to prove the advantages of the two-layer watermarking such as robustness, invisibility and practicability.

Key words: data hiding, digital watermark, patchwork, copyright protection, two-layer watermarking.

1. INTRODUCTION

Due to the rapid growth of the digital representation, information is easy to be recorded and backup. However, it is also easy to be transmitted, distributed and duplicated. Especially, the Internet is very popular in the recent year. For example, when we publish image, sound, and video on the World Wide Web, pirates would embezzle the digital information. Moreover, the pirates maybe announce that they are the owners of these data. Therefore, it is very important to protect the intellectual property rights of the digital information. The digital watermarking technique is a good way to solve this problem.

Past research in digital watermarking can be broadly classified into two categories. The first one performs the watermarking in the transform domain (DFT/DCT) [2,3,5,6,7]. The second one performs the watermarking in the time/spatial domain (DFT/DCT) [1,4,8].

In this study, we propose a novel two-layer watermarking on 256-gray-level images. The watermark is a serial number and embedded in spatial domain. Two-layer watermarking means that two kinds of watermarking techniques are employed simultaneously. In other words, there are two embedding and detection algorithms. Actually, the two watermarking techniques can not interfere with each other. To increase the practicability of watermarking, the watermark is designed to be detected neither resorting to the original image nor the original watermark in the proposed method.

The layer-one watermarking is based on the DC component that will not be changed or only mild changed during high-frequency attack. Thus, the layer-one watermarking can resist high-frequency attack such as JPEG, blurring, mean filer, median filter, etc.

The layer-two watermarking is based on the patchwork method [1]. The patchwork method is resistant to low-frequency attack such as image equalization, quantization, contrast enhancement, etc. However, only little information can be hidden through patchwork watermarking. In this study, multiple pseudorandom patterns are proposed for patches to hide more information.

2. TWO-LAYER WATERMARKING

2.1. Anti-high-frequency-attack watermarking

As mentioned above, the layer-one watermarking is based on the DC component. The main principle of the layer-one watermark is to modify the relatives between the means of two nonoverlapping blocks which are of the same sizes and neighbor each other. The mean difference of the two blocks are then modified to satisfy some criteria such that one bit information can be embedded. More specifically, the mean difference between the two blocks must be equal to two specific values H_0 and H_1. The former indicates "0" being embedded, the latter "1". The value H_0 is a pseudo-random number generated by a secret key. While, the value of H_1 is the addition of value H_0 and $R/2$ where value R is the degree of robustness. The relationship between the values of H_0, H_1 and R is shown in Fig. 1.

How to modify the mean difference between two blocks is described as follows. Assume that two blocks A and B have means m_A and m_B, respectively. Compute the mean difference m_{dif} using the following equation

$$m_{dif} = (m_A - m_B + R \times 256) \bmod R \qquad (1)$$

The term of $R \times 256$ is involved in the above equation to guarantee that m_{dif} is always positive. After that, the value s must be added to each pixel of block A and subtracted from each pixel of block B such the mean difference can satisfy embedding rule. The rules to determine the value s are listed below.

If "1" is embedded, the following preprocessing must be performed first, i.e.,

If m_{dif} is smaller than H_0, $m_{dif} = m_{dif} + R$.

After that, s is set as $(H_1 - m_{dif})/2$;

If "0" is embedded, the following preprocessing must be performed first, i.e.,

If m_{dif} is larger than H_1, $m_{dif} = m_{dif} - R$.
After that, s is set as $(H_0 - m_{dif})/2$. (2)

Note that s will bot be over $R/4$. However, if s is not an integer, the error diffusion method must be performed after addition and subtraction operations.

On the other hand, to detect the embedded one bit, we must check the mean difference between the two blocks. If it is close to H_0, "0" is extracted, otherwise "1" is extracted. The details are described below. Like embedding stage, the mean difference m_{dif} must be calculated using Eq. (1). However, if m_{dif} is less than H_0, m_{dif} must be added with R. After that, the following rule is used to extract embedded bit

If ABS($m_{dif} - H_1$) is smaller than $R/4$, "1" is extracted, otherwise "0" is extracted, (3)
where ABS(x) is the absolute value of x.

Finally, an example of layer-one watermarking is illustrated below. Assume that the sizes of blocks A and B are both 4 × 4. Let the values R and H_0 be 8 and 3, respectively. We then have $H_1 = H_0 + R/2 = 7$. Let the contents of blocks A and B be

Block A:

163	164	179	170
172	170	161	159
175	166	156	168
176	158	169	173

Block B:

176	160	148	144
170	161	158	145
166	166	167	168
160	162	166	170

The means of blocks A and B can then be easily obtained as m_A = 167.43 and m_B = 161.68. From Eq. (1), we have the mean difference $m_{dif} = (m_A - m_B + 256 \times 8) \bmod 8 = 5.75$

If we want to embed "0", from the determination rules of s in Eq. (2), we can get $s = (3 - m_{dif}) / 2 = -1.38$. Henceforth, all of the pixels in the block A are added by s, while all of the pixels in the block B are subtracted by s. Because s is not an integer, error diffusion must be performed. The results are listed below.

Block A:

161.62	162.62	177.62	168.62
170.62	168.62	159.62	157.62
173.62	164.62	154.62	166.62
74.62	156.62	167.62	171.62

==>

Block A via error diffusion:

161	163	177	169
171	168	160	158
173	165	154	167
175	156	168	172

Block B:

177.38	161.38	149.38	145.38
171.38	162.38	159.38	146.38
167.38	167.38	168.38	169.38
161.38	163.38	167.38	171.38

==>

Block B via error diffusion:

177	161	150	145
171	163	159	147
167	167	169	169
161	164	168	171

As a result, the new m_A and m_B are 166.06 and 163.06, respectively. $m_{dif} = (m_A - m_B + 256 \times 8) \bmod 8 = 3 = H_0$. Thus, "0" will be extracted in the detection stage.

2.2. Anti-low-frequency-attack watermarking

As mentioned above, the layer-two watermarking is based on the patchwork method. The main principle of the layer-two watermarking is to modify the relative differences between two patches in one image block. The two patches of the same sizes are non-overlapping and can be selected randomly. The relative difference between the two patches is then modified as times of a specific value d. Moreover, there are multiple pseudo-random patterns for the two patches and each random pattern indicates one kind of value hidden in the block. Obviously, the concept of multiple-pattern-patchwork mentioned above is originated from the patchwork method [1]. Thus, the patchwork method is first summarized, followed by the details of multiple-pattern-patchwork.

Encoding algorithm

Choose two points C and D at random in an image and let their corresponding brightness be c and d. If this procedure is repeated n times, two patches of n points, namely P_C and P_D, will be obtained. Let C_i and D_i be the points of C and D randomly selected during the ith iteration with the respective brightness c_i and d_i. The encoding procedure can be performed n times, one for each pair of (c_i, d_i). Raise the brightness c_i in patch P_C by an amount d, while degrade the brightness d_i in patch P_D by this same amount d. The value of d is not necessary the same but typically in the range of 1 to 5 parts in 256 gray levels.

Decoding algorithm

To detect the embedded bit, we have to compute the value of $S'_n = \sum_{i=1}^{n}(c'_i - d'_i)$, where (c'_i, d'_i) are the corresponding brightness of the pair of points (C_i, D_i) in the marked image. If expected value of S'_n closes to $2nd$, there is one bit watermark hidden in the image. The reason is listed below. Assume that the expected value of $S_n = \sum_{i=1}^{n}(c_i - d_i)$ is 0. The expected value of $S'_n = \sum_{i=1}^{n}(c'_i - d'_i)$ is then expected to be $2nd$.

Although the patchwork method is resistant to low-frequency attack only one-bit information can be hidden. Thus, multiple pseudo-random patterns are proposed in this paper to hide more information. The pseudo-random patterns are defined as position patterns to specify patches. More specifically, the position pattern is a binary matrix to determine which points in the block belong to patch P_C, and the others patch P_D. It follows, the patchwork method can be employed in the block with each position pattern to hide one-value information. Obviously, if there are multiple position patterns, more than one value can be hidden in the block because different position patterns represent different hidden values.

The details of the definition of the position pattern are described below. Assume that information is hidden in an image block of size *m*×*m*. The corresponding position pattern *P* will be defined as a binary matrix of size $\frac{m}{2}\times m$ Let any two neighboring points in the horizontal direction in the block *B* be grouped as a pair. The position pattern is used to indicate the two neighboring points paired as (C_i, D_i) or (D_i, C_i) . More specifically, each element of the position pattern *P*, $P(j,k)$, is defined as follows.

$P(j,k) = 1$, $(B(2\times j,k), B(2\times j+1,k))$ is paired as (C_i, D_i) , otherwise (D_i, C_i)

$$j = 0,\cdots,\frac{m}{2}-1, k = 0,\cdots,m-1 \qquad (4)$$

where $B(j,k)$ is the point at the position (j,k) in the block *B*. An example for a 8×8 block is shown in Fig. 2 with (C_i, D_i) and (D_i, C_i) being labeled as (C, D) and (D, C) , respectively.

If there are *n*-bit information to be hidden, there must be 2^n different position patterns P_x to hide the corresponding value *x*, $0 \le x \le 2^n - 1$. However, the position patterns must be generated randomly to increase the degree of security. On the other hand, all the position patterns must be different as much as possible such that it is robust to detect the hidden value. After the position patterns have been defined, the *n*-bit information can be hidden and then detected in the following way. If a value $x, x < 2^n$, is hidden in a block *B* of size *m*×*m*, the position pattern P_x is used to perform patchwork method as described above. First, the respective sums of the brightness of all the points in the two patches, sc_x and sd_x must be computed by

$$sc_x = \sum_{j=0}^{m/2-1}\sum_{k=0}^{m-1} g(2\times j + 1 - P_x(j,k), k)$$
$$sd_x = \sum_{j=0}^{m/2-1}\sum_{k=0}^{m-1} g(2\times j + P_x(j,k), k) \qquad (5)$$

where $g(x, y)$ is the corresponding brightness of the point $B(x, y)$ in the block *B*. The value *x* is then embedded into the block *B* by the following equation

If $sc_x > sd_x$,

$$g(2\times j, k) = g(2\times j, k) + (d)(-1)^{(1-P_x(j,k))}$$
$$g(2\times j+1, k) = g(2\times j+1, k) + (d)(-1)^{P_x(j,k)}$$

Otherwise,

$$g(2\times j, k) = g(2\times j, k) + (d)(-1)^{P_x(j,k)}$$
$$g(2\times j+1, k) = g(2\times j+1, k) + (d)(-1)^{(1-P_x(j,k))}$$
$$j = 0,\cdots,\frac{m}{2}-1;\ \ k = 0,\cdots,m-1 \ . \qquad (6)$$

During the detection stage, we will compute the relative difference between the two patches from the marked image for each position pattern P_x, $0 \le x \le 2^n - 1$ using the following equation

$$S'_x = ABS(sc'_x - sd'_x) \qquad (7)$$

where sc'_x and sd'_x can be computed using Eq. (5) except that $g(x, y)$ is replaced by the corresponding brightness of the point $B(x, y)$ in the marked image. After that, the following rule is used to extract embedded value.

$$n \text{ is extracted if } n = \arg\max_{0\le x\le 2^n-1}(S'_x). \qquad (8)$$

Finally, an example of layer-two watermarking is illustrated in Fig. 3. The assumptions in this example are listed below. The block size is 4×4. Each block is embedded by 2-bit information, thus 2^2=4 position patterns are needed. Pseudorandomly generate 4 position patterns, any two neighboring pixels in the block are paired as (*C*,*D*) or (*D*,*C*) according to position patterns. The value of d is set as 2.

3. COMBINATION OF TWO LAYERS OF WATERMARKING

Two layers of watermarking algorithms are employed to hide the same watermarks in the image simultaneously. In this section, the interaction between the two layers of watermarking is discussed. Moreover, we will explain how to decide whether watermarks exist and further what the watermarks are if any.

As mentioned above, layer-one watermarking modifies the mean difference between two image blocks; while, layer-two watermarking modifies the relative difference between two patches in an image block. Only if the block in the layer two is totally inside in the block of layer one, the modification of relative difference within a block will not change the block mean and thus the mean difference between two blocks. Henceforth, in the embedding stage, the layer-one watermarking will be performed first, followed by the layer-two watermarking.

The proposed watermark is composed of two parts: fixed and variable. The variable part is included to increase the variety of watermarks. The advantages of including the fixed part in the watermark are listed below. First, the fixed part can be regarded as standard reference to rectify the geometric transformation of the image caused by transformation attack such as translation and crop. Second, the standard reference can also be used to check the existence of watermark without resorting to the original watermark. Note that the fixed part can also be generated by key to increase security. On the other hand, the embedded watermark is a short serial number such that it can be duplicated in the image to increase the degree of robustness. Due to the duplicity, the majority voting strategy can be employed to facilitate the watermark detection.

In the detection stage, only if the existence of watermark can be verified either in the layers one or two, the existence of watermark in the image is positive. However, the detection in the layer two is performed before that in the layer one because layer-two watermarking is in general more robust than layer-one watermarking. The reason will be explained later. Nevertheless, the two layers of watermarking can be performed independently either in the embedding or in the detection stage.

3.1. Interaction between two layers of watermarking

The arrangement of the watermarks in the two layers is shown in Fig. 4(a) under the following assumptions. However, these assumptions are not necessary and can be changed if necessary. The image size is 240×240. Both of the image blocks in the two layers are of size 12×12. Thus, there are 20×20 blocks in an image. The basic units to embed a watermark in layers one and two are 10×10 blocks and 4×4 blocks, respectively. Hence, the duplicities of the watermarks in the layers one and two are $d_1 = 4$ and $d_2 = 25$, respectively.

The variable parts of the watermarks in the two layers are the same serial numbers of 32 bits. However, the fixed parts of the watermarks in the two layers are different. In layer one, every two blocks in the vertical direction can be paired to embed one bit as mentioned in Section 2.1. Hence, the number of bits of the fixed part for the watermark in layer one is $f_1 = 18$. Moreover, the 18-bit fixed number is set as "101010101001100110" in this study. Actually, the fixed number can be any bit string or generated by the key. The arrangement of the fixed and variable parts of each watermark in layer one is shown in Fig. 4(b).

In layer two, a four-bit value is embedded in one block. Thus, the number of bits of the fixed part for the watermark in the layer two is $f_2 = 32$. Moreover, the 32-bit fixed part consists of eight same pseudo-values. The pseudo-value means that the value itself is not important but only a symbol to represent a different value from the embedded values 0 to 15. In other words, 17 position patterns must be generated randomly, one for the pseudo-value of the fixed part and the others for the embedded values 0 to 15 of the variable part. The arrangement of the fixed and variable parts of each watermark in the basic unit of layer two is also shown in Fig. 4(b).

3.2. Detection of watermarks in two layers

The watermark detection includes four stages, i.e. the determination of the location, the content, the existence and the validity of the watermark, respectively. In general, no matter in the layers one or two, the location is determined based on the fixed part, while the content and the validity on the variable part. On the other hand, the existence is checked on the fixed part in layer one, but the variable part in layer two. The reason is that the probability of false alarm in layer two is higher than that in layer one, thus stricter criterion is required in layer two.

Determination of watermark location

In the first stage of location determination, the fixed part of watermark must be searched to locate the correct position of the watermark. The reason is that the position of watermark may be changed due to geometric transformation such as translation and crop. Note that the search method employ brute-force strategy thus inefficient for rotation and scaling attacks. The majority voting is employed to find the most possible position of watermark. The details are listed below.

If the position of watermark can be correctly located, the watermarks whose fixed parts are absolutely equal to the real embedded fixed part will be detected most. Let the fixed part of the embedded watermark in layer one be expressed by a binary string as $F'_E = fb'_{E_0}\, fb'_{E_1} \cdots fb'_{E_{17}}$, i.e., "101010101001100110" in this study. When the starting location is at (*r*, *c*), let the fixed part of the *x*th detected watermark be expressed by a binary string as $F'_x(r,c) = fb'_{x_0}(r,c)\, fb'_{x_1}(r,c) \cdots fb'_{x_{17}}(r,c)$. The final located position (fr', fc') for an image of size $M \times N$ can then be determined by the following equation

$$(fr', fc') = \max_{0 \le r \le M-1, 0 \le c \le N-1} fn'(r,c)$$
$$\text{where } fn'(r,c) = \sum_{x=0}^{t'-1} count'_x(r,c) \qquad (9)$$
$$count'_x(r,c) = \begin{cases} 1 & \text{if } fb'_{x_i}(r,c) = fb'_{E_i}, \forall i = 1, \cdots, 31 \\ 0 & \text{otherwise} \end{cases}$$

where $t' \le 4$ is the number of watermarks detected in layer one.

Similarly, the position of watermark in layer two can be correctly located through majority strategy. Let the fixed part of the embedded watermarks in layer two be expressed by a hexadecimal string as $F''_E = fh''_{E_0}\, fh''_{E_1} \cdots fh''_{E_7}$, i.e., "*vvvvvvvv*" with *v* being a pseudo value as specified in Section 3.1. When the starting location is at (*r*, *c*), let the fixed part of the *x*th detected watermark be expressed by $F''_x(r,c) = fh''_{x_0}(r,c)\, fh''_{x_1}(r,c) \cdots fh''_{x_7}(r,c)$ .The final located position (fr'', fc'') can then be determined by the following equation

$$
\begin{aligned}
&(fr'', fc'') = \max_{0\le r\le M-1, 0\le c\le N-1} fn''(r,c) \\
&\text{where } fn''(r,c) = \sum_{x=0}^{t''-1} count''_x(r,c) \\
&count''_x(r,c) = \begin{cases} 1 & \text{if } fh''_{x_i}(r,c) = fh''_{E_i}, \forall i = 1,\cdots,7 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

where $t'' \le 25$ is the number of watermarks detected in layer two.

Determination of the watermark content

Obviously, both in the layers one and two, the watermark-content detection must be performed after location determination has been completed. The detection algorithms in the two layers are the same as described in Sections 2.1 and 2.2, respectively. However, the detected results for the duplicated watermarks may not be the same. The majority voting is also included to induce the final content of the watermark. The details for layer one are first described below, followed by those for layers two.

Let the corresponding t' watermarks be $V'_0, V'_1, \cdots, V'_{t'-1}$. Each watermark V'_x, $x = 0,\cdots,t'-1$, has 32 bits and can be set as $V'_x = vb'_{x_0}\, vb'_{x_1} \cdots vb'_{x_{31}}$ with vb'_{x_i} denoting *i*th binary bit. The real watermark $V'_C = vb'_{C_0}\, vb'_{C_1} \cdots vb'_{C_{31}}$ can then be obtained by

$$
vb'_{C_i} = \begin{cases} 1 & \text{if } \displaystyle\sum_{x=0}^{t'-1} vb'_{x_i} > \frac{t'}{2} \\ 0 & \text{otherwise} \end{cases} \qquad i = 0,\cdots,31
$$

Let the corresponding t'' watermarks be $V''_0, V''_1, \cdots, V''_{d_2-1}$. Each watermark V''_x, $x = 0,\cdots,d_2-1$, has eight 4-bit values and can be set as $V''_x = vh''_{x_0}\, vh''_{x_1} \cdots vh''_{x_7}$ with vh''_{x_i} denoting *i*th hexadecimal code. The real watermark $V''_C = vh''_{C_0}\, vh''_{C_1} \cdots vh''_{C_7}$ can then be obtained by the following equation

$$
\begin{aligned}
&vh''_{C_i} = \arg\max_{0\le i\le 7} vn''_i(h) \\
&\text{where } \mathrm{v}n''_i(h) = \sum_{j=0}^{t''-1} b_{i,j}(h) \\
&b_{i,j}(h) = \begin{cases} 1 & \text{if } vh''_{j_i} = h \\ 0 & \text{otherwise} \end{cases} \\
&i = 0,\cdots,7,\ j = 0,\cdots,t''-1,\ h = 0,\cdots,15.
\end{aligned}
\tag{10}
$$

Determination of the watermark existence

No matter the image is embedded by watermark or not, a watermark will be induced from the image. Thus, the actual existence of watermark must be verified. However, the existence of watermark is verified by majority strategy rather than the crucial similarity measure between the detected and the original watermarks. The details are described below. In layer one, the criteria are based on the fixed part. The watermarks whose fixed parts absolutely equal to the real embedded fixed part must be detected more than a threshold value th_1. As mentioned above, in layer one, the fixed part of embedded watermarks can be expressed by $F'_E = fb'_{E_0}\, fb'_{E_1} \cdots fb'_{E_{18}}$, i.e., "1010101010 01100110" in this study. While the fixed part of the *x*th detected watermark can be expressed by $F'_x = fb'_{x_0}\, fb'_{x_1} \cdots fb'_{x_{18}}$. The criterion can then be derived by the following equation

$$
\begin{aligned}
&fn' > th_1, \\
&\text{where } fn' = \sum_{x=0}^{t'-1} count'_x, \\
&count'_x = \begin{cases} 1 & \text{if } fb'_{x_i} = fb'_{E_i}, \forall i = 0,\cdots,18 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{11}
$$

In this study, the threshold th_1 is determined experimentally by $t'/2$ as described later.

In layer two, the criteria are based on the variable part and listed below. During the determination of the real content of watermark, the number of pros for each hex-code must be more than a threshold value th_2. The criterion can be expressed by the following equation

$$
\forall i = 0,\cdots,7,\ \exists h = 0,\cdots,15,\ \ni \max_{0\le h\le 15} vn''_i(h) > th_2 \tag{12}
$$

where $vn''_i(h)$ is as specified in Eq. (10). In this study, the threshold th_2 is determined experimentally by $t''/5$.

The threshold values th_1 and th_2 determined by the following experiments. We randomly generate 1000 images in which no watermarks are embedded. The proposed detection algorithm is then applied on the images. After that, the histograms of the values of fn' as specified in Eq. (11) and $\min_{0\le i\le 7}\{\max_{0\le h\le 15} vn''_i(h)\}$ with $vn''_i(h)$ as specified in Eq. (10) are then depicted in Fig. 5. From Fig. 5, we find their values are less than 2=4/2 and 5=25/5, respectively. Thus, the threshold values th_1 and th_2 are defined as $t'/2$ and $t''/5$ in layers one and two, respectively.

Determination of the watermark validity

Finally, the parity-check algorithm is adopted to check the validity of the extracted watermark. Only if the validity check is passed, the extracted watermark is regarded as a correct watermark. Moreover, the validity check is required both in the layers one and two. Let the detected watermarks in layers one and two be expressed by $V'_R = vb'_{R_0} vb'_{R_1} \cdots vb'_{R_{31}}$ and $V''_R = vb''_{R_0} vb''_{R_1} \cdots vb''_{R_{31}}$, respectively. The parity check for layers one and two can be derived by Eqs. (13) and (14), respectively.

$$vb'_{R_i} \oplus vb'_{R_{i+4}} \oplus vb'_{R_{i+8}} \oplus vb'_{R_{i+12}} \oplus vb'_{R_{i+16}} \oplus vb'_{R_{i+20}} \oplus vb'_{R_{i+24}} = vb'_{R_{i+28}}, \qquad i = 0,1,2,3 \tag{13}$$

$$vb''_{R_i} \oplus vb''_{R_{i+4}} \oplus vb''_{R_{i+8}} \oplus vb''_{R_{i+12}} \oplus vb''_{R_{i+16}} \oplus vb''_{R_{i+20}} \oplus vb''_{R_{i+24}} = vb''_{R_{i+28}}, \qquad i = 0,1,2,3 \tag{14}$$

4. EXPERIMENTAL RESULTS

4.1. The characteristics of our watermarking

The host image is a 256-gray-level image. The watermark is a 32-bit serial number. The block size is 12×12 in this experiment, so the size of the host image is at least 120×120. The proposed watermarking is very invisible. Before embedding the watermark, *R*, *d*, key and watermark must be decided first. The values of *R* and *d* indicate the degrees of robustness in the layer-one and layer-two watermarking, respectively. In our experiment, the variable part of watermark is set as "0001001000110100010101100 1110000", and the corresponding hexadecimal value is "12345670".

The host image is shown in Fig. 6(a). When *R* and *d* are set as 8 and 2, respectively, the results of applying only layer-one embedding algorithm, only layer-two embedding algorithm and the layer-one followed by layer-two embedding algorithm on the host image of Fig 6(a) are shown in Figs. 6(b), 6(c) and 6(d), respectively. The modification ranges of gray level for each pixel in the host image are not over 3 and 2 in layers one and two, respectively. Consequently, when both two layers of watermarking are performed, the total modification range of gray level is not over 5. Thus, we can conclude that our watermarking is very invisible. Actually, the larger the degree of robustness *R* and *d*, the more robust the watermarking method. However, the invisibility is decreased. PSNR values of embedded images using different values of *R* and *d* are shown in Fig. 7.

4.2. The attacked results

In the following experiments, the robust level in the layers one and two are set as 8 and 2, respectively, except special description. In practice, the image is processed and then stored through JPEG compression in file system. Thus, we adopted JPEG compress as the second attack. The experimental results for different attacks are listed below.

JPEG compression. The algorithm of the JPEG compression is built in the Borland C++ Builder 4.0. The proposed layer-one watermarking can resist this kind of attack. The minimum quality the proposed method can resist for different images of different sizes are listed in Fig. 8. In general, the bigger the block is, the higher the degree of robustness is. Thus, the minimum qualities the proposed method can resist versus block sizes are shown in Fig. 9. On the other hand, the layer-two watermarking can also be resistant to this kind of attack. However, the degree of robustness is lower than that of the layer-one as shown in Fig 10.

Blurring attack. The layer-one watermarking can resist this kind of attack. We applied Adobe Photoshop blur more function on Fig. 6(d) to simulate this kind of attacks. The result is shown in Fig. 6(e). For attacked image of Fig. 6(e), the proposed watermarking can resist the second JPEG attack until quality of 83.

Quantization attack. The layer-two watermarking can resist this kind of attack. Adobe Photoshop index function was adopted to quantize the marked image of Fig. 6(d) into 5 colors as shown in Fig. 6(f). For the attacked image of Fig. 6(f), the proposed watermarking can resist the second JPEG attack until quality of 73.

Cropping attack. The layer-two watermarking can resist this kind of attack. We crop 75% area in the image to test this kind of attack. The result is shown in Fig. 6(g). For the attacked image of Fig. 6(g), the proposed watermarking can resist the second JPEG attack until quality of 90.

Brightness and contrast attack. The layer-two watermarking can resist this kind of attack. We adopted Adobe PhotoShop to decrease 50 brightness and increase 50 contrast to test this kind of attack. The results are shown in Figs. 6(h) and 6(i), respectively. For the attacked images of Figs. 6(h) and 6(i), the proposed watermarking can resist the second JPEG attack until qualities of 90 and 72, respectively.

Histogram equalization attack. The layer-two watermarking can resist this kind of attack. The attacked result is shown in Fig. 6(j). For the attacked image of Fig. 6(j), the proposed watermarking can resist the second JPEG attack until quality of 74.

Sharpen attack (edge enhance). The layer-two watermarking can resist this kind of attack. We adopted Adobe PhotoShop sharpen more function to test this kind of attack. The attacked result is shown in Fig. 6(k). For the attacked image of Fig. 6(k), the proposed watermarking can resist to the second JPEG attack until quality of 46.

Noise attack. The layer-two watermarking can resist this kind of attack. We adopted Adobe PhotoShop uniform increase 22-noise to test this kind of attack. The result is shown in Fig. 6(l). For the attacked image of Fig. 6(l), the proposed watermarking can resist the second JPEG attack until quality of 86.

5. CONCLUSION

In this paper, we propose two layers of watermarking to resist different kinds of attacks. The watermark is embedded in spatial domain. The detection algorithm need not resort to the original image nor the original watermark.

There are many advantages of the proposed method listed below. First, the proposed embedding method only modifies the brightness of most pixels a little. More important, even though the watermark is invisible, the proposed method is resistant to various kinds of attacks. Second, the method of embedding and detection algorithm is very simple. Although there are two layers of watermarks, the speed of the embedded algorithm is still fast. In most cases, only one to two seconds are required. It is easy to be real-time implemented. Third, the original image is not necessary for detection. Moreover, the detected watermark can be verified directly without resorting to the original watermark. Finally, the size of host image need not be very large, the size of 120× 120 is enough to embed duplicated watermarks. Due to the duplicity, the majority voting strategy can be employed to facilitate watermark detection. However, the resistance to the nonlinear attack, combined attacks and the attack of rotation and scale is not good in the current work and should be improved in future.

REFERENCES

[1] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, “Techniques for data hiding,” *IBM Systems Journal*, vol. 35, nos. 3&4, pp. 313-335, 1996.

[2] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon, “Secure spread spectrum watermarking for multimedia,” *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673-1687, 1997.

[3] C.T. Hsu and J.L. Wu, “Hidden digital watermarks in images,” *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 58-68, 1999

[4] M. Kutter, F. Jordan, and F. Bossen, “Digital watermarking of color images using amplitude modulation,” *Journal of Electronic Imaging*, vol. 7, no. 2, pp. 326-332, 1998.

[5] C.S. Lu, H.Y. Liao, S.K. Huang, and C.J. Sze, “Cocktail watermarking on images”, *Proc. International Workshop on Information Hiding*, Germany, pp. 331-345, 1999.

[6] C.S. Lu, H. Y. Liao, S. K. Huang, and C. J. Sze, “Combined watermarking for image authentication and protection”, to appear in *Proc. International Conference on Multimedia and Expo*, U.S.A., 2000.

[7] M. Marvel, G. Boncelet, and T. Retter, “Spread spectrum image steganography,” *IEEE Transactions on Image Processing*, vol. 8, no. 8, pp. 1075-1083, 1999.

[8] G. Voyatzis and I. Pitas, “Digital image watermarking using mixing systems,” *Computer and Graphics*, vol. 22, no. 4, pp.405-416, 1998.

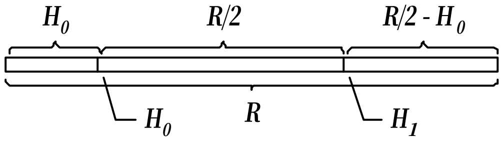
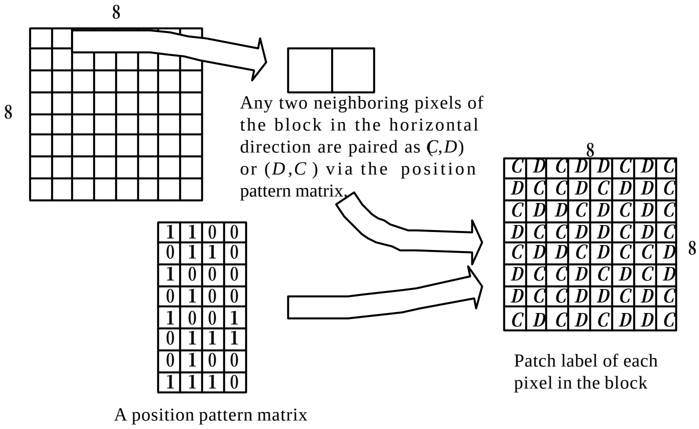


Fig. 1. The relationship between the values of H_0, H_1 and R.

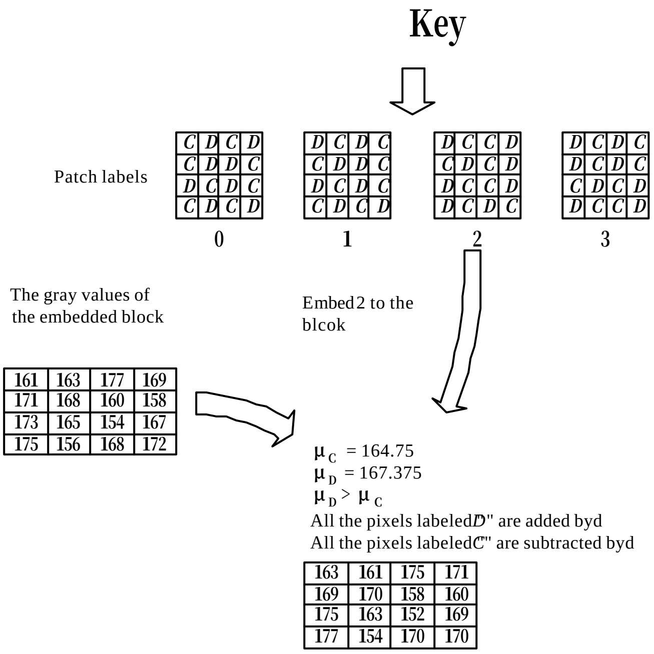


8
8
Any two neighboring pixels of the block in the horizontal direction are paired as (C,D) or (D,C) via the position pattern matrix.
8
8
Patch label of each pixel in the block
A position pattern matrix

1	1	0	0
0	1	1	0
1	0	0	0
0	1	0	0
1	0	0	1
0	1	1	1
0	1	0	0
1	1	1	0

C	*D*	*C*	*D*	*D*	*C*	*D*	*C*
D	*C*	*C*	*D*	*C*	*D*	*D*	*C*
C	*D*	*D*	*C*	*D*	*C*	*D*	*C*
D	*C*	*C*	*D*	*D*	*C*	*D*	*C*
C	*D*	*D*	*C*	*D*	*C*	*C*	*D*
D	*C*	*C*	*D*	*C*	*D*	*C*	*D*
D	*C*	*C*	*D*	*D*	*C*	*D*	*C*
C	*D*	*C*	*D*	*C*	*D*	*D*	*C*

Fig. 2. An example of a position pattern.



C	*D*	*C*	*D*
C	*D*	*D*	*C*
D	*C*	*D*	*C*
C	*D*	*C*	*D*

D	*C*	*D*	*C*
C	*D*	*D*	*C*
D	*C*	*D*	*C*
C	*D*	*C*	*D*

D	*C*	*C*	*D*
C	*D*	*C*	*D*
D	*C*	*C*	*D*
D	*C*	*D*	*C*

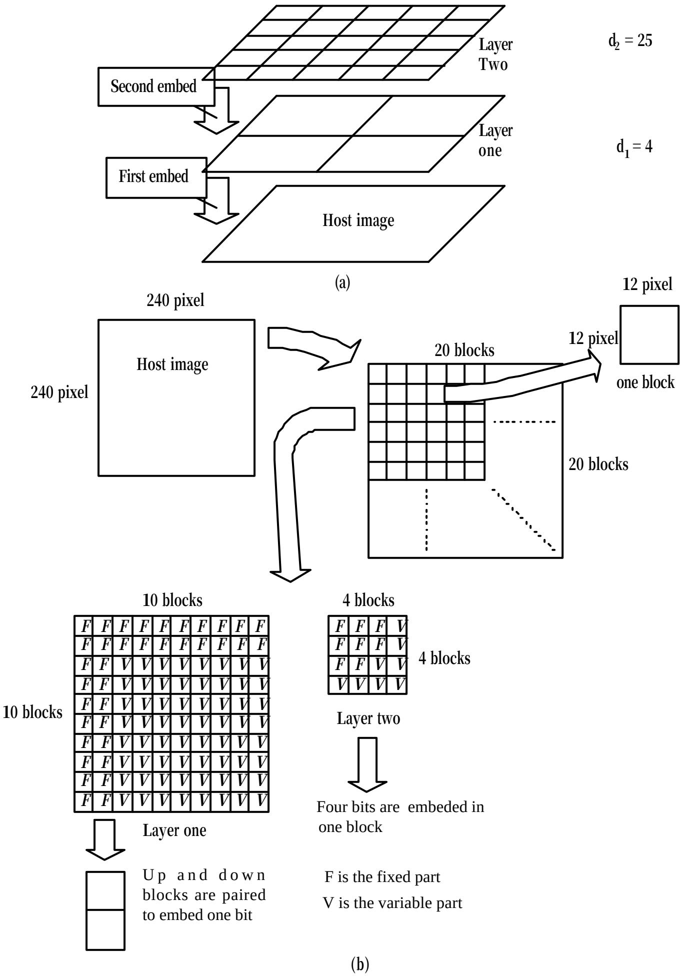
D	*C*	*D*	*C*
D	*C*	*D*	*C*
C	*D*	*C*	*D*
D	*C*	*C*	*D*

161	163	177	169
171	168	160	158
173	165	154	167
175	156	168	172

$\mu_C = 164.75$
$\mu_D = 167.375$
$\mu_D > \mu_C$
All the pixels labeled*D*" are added byd
All the pixels labeled*C*" are subtracted byd

163	161	175	171
169	170	158	160
175	163	152	169
177	154	170	170

Fig. 3. An example of the layer-two -watermark embedding.



F	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*
F	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*
F	*F*	*V*	*V*	*V*	*V*	*V*	*V*	*V*	*V*
F	*F*	*V*	*V*	*V*	*V*	*V*	*V*	*V*	*V*
F	*F*	*V*	*V*	*V*	*V*	*V*	*V*	*V*	*V*
F	*F*	*V*	*V*	*V*	*V*	*V*	*V*	*V*	*V*
F	*F*	*V*	*V*	*V*	*V*	*V*	*V*	*V*	*V*
F	*F*	*V*	*V*	*V*	*V*	*V*	*V*	*V*	*V*
F	*F*	*V*	*V*	*V*	*V*	*V*	*V*	*V*	*V*
F	*F*	*V*	*V*	*V*	*V*	*V*	*V*	*V*	*V*

F	*F*	*F*	*V*
F	*F*	*F*	*V*
F	*F*	*V*	*V*
V	*V*	*V*	*V*

Fig. 4. The arrangement of the watermarks in the two layers. (a) The overall organization; (b) the detailed organization.

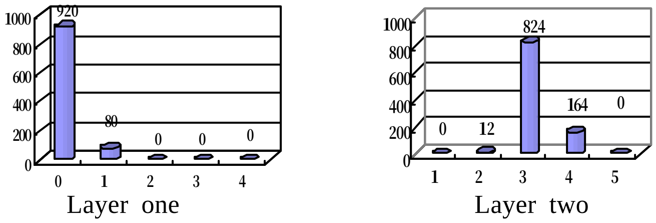


Fig. 5. The number of the detected watermarks in the 1000 images which are not really marked.

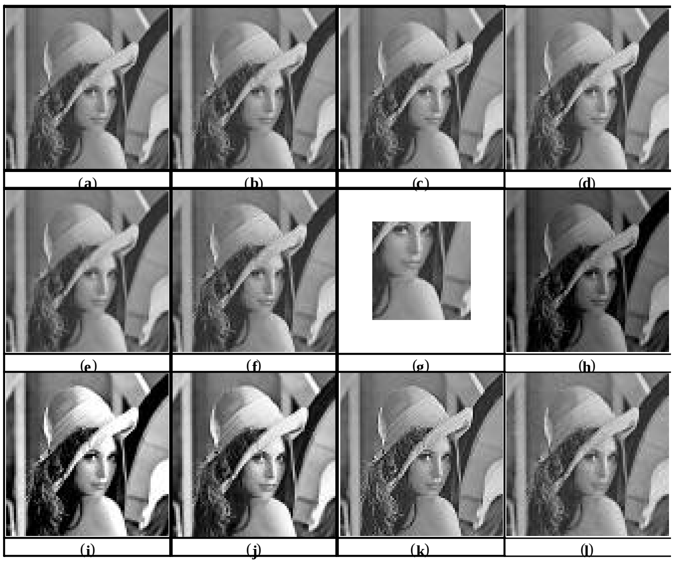


Fig. 6. The experimental results.

Only layer one	Only layer two	Both two layers
R = 8 PSNR = 46.86	*d* = 2 PSNR = 41.16	PSNR = 40.08
R = 9 PSNR = 45.41	*d* = 3 PSNR = 38.08	PSNR = 37.33
R = 10 PSNR = 44.62	*d* = 4 PSNR = 35.79	PSNR = 35.24
R = 11 PSNR = 43.76	*d* = 5 PSNR = 33.96	PSNR = 33.52
R = 12 PSNR = 42.84	*d* = 6 PSNR = 32.44	PSNR = 32.05

Fig. 7. The PSNR values of the marked Lenna image with respect to different values of *R* and *d*.

	The minimum quality (1-100)	The PSNR after JPEG compress	Original file size	Compressed file size
Lenna 256× 256	31	31.53	65536	5993
Lenna 512×512	26	33.02	262144	15365
Baboon 256×256	25	25.61	65536	7660
Baboon 512 × 512	30	26.31	262144	34411
Pepper 256 × 256	25	31.50	65536	5534
Pepper 512 × 512	28	32.61	262144	16130

Fig. 8. The results of layer-one watermarking in terms of minimum JPEG compress quality for different images of different sizes.

Block size	12 × 12	14 × 14	16 × 16
The minimumquality	27	22	20

Fig. 9. The results of layer-one watermarking in terms of minimum JPEG compress quality for the 256×256 Lenna image with respect to different block sizes.

	d = 2	*d* = 3	*d* = 4	*d* = 5
Lenna 256×256	85	79	67	56
Baboon 256×256	79	69	55	41
Pepper 256×256	85	83	76	70

Fig. 10. The results of layer-two watermarking in terms of minimum JPEG compress quality with respect to different levels of robustness *d*.