Adaptive reversible image watermarking scheme

Chien-Chang Chen\textsuperscript{a,∗}, Yao-Hong Tsaib

\textsuperscript{a} Department of Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan
\textsuperscript{b} Department of Information Management, Hsuan Chuang University, Hsinchu 300, Taiwan

1. Introduction

Digital images are easily transmitted, modified, and reproduced. Consequently, determining the ownership of an image is an important issue. Watermarking has recently become important in solving this image authentication problem for protecting important images (Cox et al., 2002). Watermarking techniques adjust an image by embedding user information, and such modification is always irreversible. Therefore, the original image cannot be recovered after watermark extraction. However, some applications, including medical or military images, must acquire the original image from the watermarked image to determine more accurately on image content. Thus, reversible watermarking approaches are required.

Research on reversible watermarking approaches is briefly reviewed. Reversible watermarking approaches are always performed in the spatial domain. Alattar (2004) calculated the difference expansion of the integer transformation to embed watermarks. Kamstra and Heijmans (2005) improved the DE scheme to increase the efficiency of lossless compression. Thodi and Rodríguez (2007) adopted the histogram shifting technique to embed the location map that the DE scheme required. They then proposed a prediction-error expansion method to embed watermarks. Celik et al. (2005) first compressed the remainder of an image and then embedded reversible watermarks into the saved space of the remainder. Tian (2003) embedded one watermark bit into the LSB of the difference of two pixels. The new calculated difference is then added to these two selected pixels as embedding steps. Lee et al. (2008) first segmented a host image to \( m \times n \) blocks and then each block applied Tian’s difference expansion method to embed reversible watermarks.

Vleeschouwer et al. (2003) employed a circular histogram to solve the reversible watermarking problem. They transformed the histogram to a circle, and then modified the histogram clockwise or counterclockwise by embedding a watermark bit. Chen and Kao (2008) proposed a technique for embedding a binary bit into a continuous series of 0 and 1 coefficients to embed reversible watermarks. Li (2006) presented an image-independent reversible watermarking method that yields a near-constant embedding capacity. Hu and Jeon (2006) presented a reversible visible watermarking approach that utilizes two data packets payloads to recover watermark-covered and nonwatermark-covered regions. The payload is hidden in a region that is not covered by a visible watermark, enabling the image to be reversibly recovered. Fridrich et al. (2002) presented the RS scheme to embed watermark bits into the status of groups of pixels. Chang et al. (2006) presented a reversible data hiding technique for lossy compression by the side match vector quantization (SMVQ). Lee et al. (2007) proposed a high-capacity reversible image watermarking scheme that is based on integer-to-integer wavelet transforms.

Histogram modulation is another important technique for reversible watermarking. Ni et al. (2006) embedded watermarks using the histogram of an image. A watermark bit 1 is embedded by adding the peak-pixel value in histogram, while embedding a 0 involves no modification. Lin et al. (2008) extended Ni’s method from positive numbers to absolute numbers and then applied it to neighboring difference of a 4 \( \times \) 4 block to improve greatly the embedded quantity. Tsai et al. (2009) presented a difference calculation in a \( 5 \times 5 \) block and then applied Ni’s method to improve the
embedded quantity. Hong et al. (2009) employed the prediction errors and the histogram modulation method to embed watermarks.

This study improves Lee et al.’s (2008) method in both embedded quantity and watermarked image quality. In Lee et al.’s (2008) method, the host image is segmented to \( m \times n \) blocks and differences between pixels in each block is used to embed watermarks. However, this study further improves the embedded quantity by the proposed adaptive block sized method. Given an image of size \( 2^N \times 2^N \), the proposed method segments it adaptively to blocks of size \( 2^L \times 2^L \), where \( L \) starts from a user-defined number to 1, according to their block structures and then applies these blocks to the conventional difference embedding method.

The rest of this paper is organized as follows. Section 2 reviews the conventional fixed block sized modulation method. Section 3 introduces the proposed adaptive block size modulation method. Section 4 presents the experimental results of the proposed method. Comparisons with the conventional fixed block sized method are also provided. Finally, Section 5 draws a brief conclusion.

2. Review of conventional fixed block sized modulation method

The proposed reversible watermarking method embeds reversible watermarks using the proposed adaptive block sized modulation method, which is an improvement of the conventional fixed block sized modulation method (Lee et al., 2008). Thus, the conventional fixed block sized modulation method is briefly reviewed in this section.

The conventional fixed block sized method first segments the host image into \( m \times n \) blocks. Assume that \( t(i) \), \( 1 \leq i \leq m \times n \) denotes each ordered pixel in an \( m \times n \) block with \( t(i) \leq t(j) \) of \( i \neq j \), and \( 1 \leq i, j \leq m \times n \). Watermarks are embedded into \( t(k) - t(mid) \), where \( mid = \lfloor (m \times n)/2 \rfloor \), \( 1 \leq k \leq m \times n \) and \( k \neq mid \), by difference expansion (DE) method (Tian, 2003). The embedded quantity \( (0, 1, \text{or } 2 \text{ bits}) \) is also calculated and compared with the predefined threshold \( T \) to determine whether the \( 4 \times 4 \) block should be segmented to four \( 2 \times 2 \) blocks or not. The proposed watermark embedding algorithm is described as follows:

First, partition the image into non-overlapping blocks of size \( 2^L \times 2^L \), and the sorted pixels in each block are denoted as \( t(i) \), \( 1 \leq i \leq 2^L \). For each block \( B_i \), apply the following recursive embedding procedure:

Recursive procedure Embedding(\( B_i \), \( L \))

\[
\begin{align*}
\text{if } T \geq 2L & , \text{ then} \{ \\
& \text{Calculate the maximum difference } d, \text{ denoted by} \\
& \max(|t(k) - t(mid)|), 1 \leq k \leq 2^L, \text{ of the } 2^L \times 2^L \text{ block.} \\
& \text{if } d \leq 2T, \text{ then embeds 2 bits to each } t(k) - t(mid), 1 \leq k \leq 2^L, \text{ excepting } 2^{L-1}, \\
& \text{else } \\
& \text{Segment the block to four } 2^{L-1} \times 2^{L-1} \text{ blocks, } B_1, B_2, B_3 \text{ and } B_4, \\
& \text{Embedding } (B_1, L-1), \\
& \text{Embedding } (B_2, L-1), \\
& \text{Embedding } (B_3, L-1), \\
& \text{Embedding } (B_4, L-1). \\
\} \\
\text{else } \{ \\
& \text{Calculate the maximum difference } d, \text{ denoted by} \\
& \max(|t(k) - t(2)|), 1 \leq k \leq 4. \\
& \text{if } d \leq T, \text{ then embeds 2 bits to each } t(k) - t(2), 1 \leq k \leq 4 \text{ excepting 2,} \\
& \text{else if } d \leq 4T, \text{ then embeds 1 bit to each } t(k) - t(2), 1 \leq k \leq 4 \text{ excepting } 2, \\
& \text{else if } d \leq 8T, \text{ then embeds 1 bit to each } t(k) - t(2), 1 \leq k \leq 4 \text{ excepting 2,} \\
& \text{and record the LSB,} \\
& \text{else embeds nothing.} \\
\}
\end{align*}
\]

3. Proposed method

This section illustrates the proposed adaptive block sized reversible image watermarking method. The proposed method segments the host image to blocks adaptively according to their structures. The structure is determined by the largest difference of the middle pixel and all other pixels in the block. The structure also determines the embedding quantity of the block. The proposed embedding and extracting algorithms are introduced in Sections 3.1 and 3.2, respectively. Section 3.3 compares the proposed method with other important literatures. Embedded capacity is discussed in Section 3.4 to demonstrate the improvement of the proposed method.

3.1. Watermark embedding algorithm

This section demonstrates the embedding algorithm of the proposed reversible watermarking approach. Given an image of size \( 2^N \times 2^N \), the proposed method segments it adaptively to blocks of size \( 2^L \times 2^L \), where \( L \) starts from a user-defined number \( L_{\text{max}} \) to 1, according to their block structures, and then embeds watermark by the following strategy. Denote \( t(i), 1 \leq i \leq 2^L \) as ordered pixels in an \( 2^L \times 2^L \) block with \( t(i) \leq t(j) \) of \( i \neq j \), \( 1 \leq i \leq 2^L \) and \( 1 \leq j \leq 2^L \). Watermarks are embedded into \( t(k) - t(2^{L-1}) \), \( 1 \leq k \leq 2^L \) excepting \( 2^{L-1} \), by Tian’s (2003) difference embedding method. The embedded quantity \( (0, 1, \text{or } 2 \text{ bits}) \) is determined by \( \max(|t(k) - t(2^{L-1})|) \) and threshold \( T \).

For the simplicity of explanation, \( L_{\text{max}} \) is set to 3 in the following. The block segmentation scheme to acquire blocks of size \( 8 \times 8, 4 \times 4, \text{ or } 2 \times 2 \) is described as follows. If the maximum difference \( \max(|t(k) - t(32)|) \) in an \( 8 \times 8 \) block is larger than predefined threshold \( T \), then the \( 8 \times 8 \) block is segmented to four \( 4 \times 4 \) blocks. Otherwise, 2 bits are embedded into each pixel difference \( t(k) - t(32) \). Maximum difference in each \( 4 \times 4 \) block denoted as \( \max(|t(k) - t(8)|) \) is also calculated and compared with the predefined threshold \( T \) to determine whether the \( 4 \times 4 \) block should be segmented to four \( 2 \times 2 \) blocks or not. The proposed watermark embedding algorithm is described as follows:

First, partition the image into non-overlapping blocks of size \( 2^{L_{\text{max}}} \times 2^{L_{\text{max}}} \) and the sorted pixels in each block are denoted as \( t(i) \), \( 1 \leq i \leq 2^{L_{\text{max}}} \). For each block \( B_i \), apply the following recursive embedding procedure:

Recursive procedure Embedding(\( B_i \), \( L \))

\[
\begin{align*}
& \text{if } T \geq 2L, \text{ then} \{ \\
& \text{Calculate the maximum difference } d, \text{ denoted by} \\
& \max(|t(k) - t(2^{L-1})|), 1 \leq k \leq 2^L, \text{ of the } 2^L \times 2^L \text{ block.} \\
& \text{if } d \leq 2T, \text{ then embeds 2 bits to each } t(k) - t(2^{L-1}), 1 \leq k \leq 2^L, \text{ excepting } 2^{L-1}, \\
& \text{else } \\
& \text{Segment the block to four } 2^{L-1} \times 2^{L-1} \text{ blocks, } B_1, B_2, B_3 \text{ and } B_4, \\
& \text{Embedding } (B_1, L-1), \\
& \text{Embedding } (B_2, L-1), \\
& \text{Embedding } (B_3, L-1), \\
& \text{Embedding } (B_4, L-1). \\
& \} \\
& \text{else } \{ \\
& \text{Calculate the maximum difference } d, \text{ denoted by} \\
& \max(|t(k) - t(2)|), 1 \leq k \leq 4. \\
& \text{if } d \leq T, \text{ then embeds 2 bits to each } t(k) - t(2), 1 \leq k \leq 4 \text{ excepting } 2, \\
& \text{else if } d \leq 4T, \text{ then embeds 1 bit to each } t(k) - t(2), 1 \leq k \leq 4 \text{ excepting } 2, \\
& \text{else if } d \leq 8T, \text{ then embeds 1 bit to each } t(k) - t(2), 1 \leq k \leq 4 \text{ excepting } 2, \\
& \text{and record the LSB,} \\
& \text{else embeds nothing.} \\
& \} \\
\end{align*}
\]
At last, combine all processed blocks to acquire the watermarked image and record the image segmentation format and the block embedded type of each block. An example of embedding watermarks into an $8 \times 8$ block is depicted in Fig. 1. First, the maximum distance of the $8 \times 8$ block is calculated in step 1 to determine whether the block should be segmented to four $4 \times 4$ blocks or not. If the segmentation is not needed, then watermarks are embedded into the block. If the block need be segmented, then we acquire four $4 \times 4$ blocks, denoted by $b_1$ to $b_4$. The maximum distance of each $4 \times 4$ block is also calculated to determine whether the block should be segmented or not. If the $4 \times 4$ block should not be segmented, then watermark is embedded into the block. Otherwise, the $4 \times 4$ block is segmented to four $2 \times 2$ blocks and then watermarks are embedding into each $2 \times 2$ block.

Fig. 2 depicts the watermark embedding algorithm. Note that image segmentation format and the block embedded type are required to be recorded in step 3. The image segmentation format records the segmentation of an image, which also means the segmented block size. The block embedded type notes the embedded quantity being 2 bits, 1 bit, or 0 bit of each difference in a block. These two kinds of information are needed in watermark extracting.

### 3.2. Watermark extracting algorithm

This section demonstrates the watermark extracting algorithm, in which the original host image is acquired after extracting the embedded watermarks. The image segmentation format and the block embedded type are both needed in watermark extracting. The proposed watermark extracting algorithm is described as follows:

1. Partition the image into non-overlapping blocks according to the image segmentation format.
2. Extract watermarks from each block according to the block embedded type.
3. Combine all watermarks as the extracted watermark and combine all recovery blocks to acquire the original host image.

All the segmentation selections are determined by the image segmentation format, which is obtained in the embedding algorithm. The block embedded type, denoting as embedding 1 or 2 bits in a difference, is needed to extract watermark and the type is also obtained in the embedding algorithm.

### 3.3. Strategy comparison

Comparisons of the proposed adaptive method and other important literatures like Ni et al. (2006) method, Tian method, and Lee et al. (2008) method are illustrated in Table 1. The comparison shows that only Ni et al. (2006) method embeds watermark by histogram shift strategy. The proposed method and Lee et al. (2008) method embed watermarks by centralized difference expansion technique, which is an improvement of conventional difference expansion method. Therefore, other three methods, excepting Ni et al. (2006) method, can be taken as embedding watermarks by difference expansion strategy, which is proposed by Tian (2003). The proposed method improves Lee et al. (2008) method on adaptively selecting block size to increase the embedded capacity.

### 3.4. Capacity discussion

The proposed method embeds watermarks into blocks of sizes $2^i \times 2^i$ with all possible $L$ for $L \geq 1$. Comparing with the Lee et al.’s (2008) method, the capacity improves in usage of blocks of sizes $2^i \times 2^i$ with all possible $L$ for $L \geq 2$. Assume an image contains $b$ blocks of size $2^a \times 2^a$ under the threshold $T$, the embedded quantity in these $2^a \times 2^a$ block is $(2^a \times 2^a - 1) \times 2 \times b$ bits. Relatively, these $b$ blocks of size $2^a \times 2^a$ forms $2^{a-1} \times 2^{a-1} \times b$ blocks of size $2 \times 2$ and the maximum embedded quantity is $6 \times 2^{a-1} \times 2^{a-1} \times b$ because of the maximum embedded quantity in a $2 \times 2$ block being 6 bits. Therefore, the proposed scheme improves the embedded quantity on those $b$ blocks of size $2^a \times 2^a$ over $2b \times (2^{a-2} \cdot 1)$ bits. Consequently, when an image contains $b_1$ blocks of size $16 \times 16$, the embedded quantity improvement is at least $126 b_1$ bits. Also $b_2$ blocks of size $8 \times 8$ improves $30 b_2$ bits and $b_3$ blocks of size

---

**Fig. 1.** Example of embedding watermarks into an $8 \times 8$ block.
4 \times 4 improves 6b_3 bits. Table 2 compares the theoretical improvement and real examples on three different block size 16 \times 16, 8 \times 8, and 4 \times 4. Consequently, the more 2^L \times 2^L (L \geq 2) blocks found in an image, the higher embedded quantity our proposed method improved.

### Table 1
Comparisons of Ni et al. (2006), Tian (2003), and Lee et al. (2008), and the proposed method.

<table>
<thead>
<tr>
<th>Embedding strategy</th>
<th>Ni et al. method</th>
<th>Tian method</th>
<th>Lee et al. method</th>
<th>The proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block processing</td>
<td>Histogram shifting</td>
<td>Difference expansion</td>
<td>Centralized difference expansion</td>
<td>Centralized difference expansion</td>
</tr>
<tr>
<td>Block size</td>
<td>Fixed</td>
<td>Adaptive</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4. Experimental results

This section presents the experimental results of the proposed method. Fig. 3 shows eight test images Jet, Lena, Peppers, Baboon, Barb, Boat, Tiffany, and Zelda, of size 512 \times 512. The embedded
Fig. 3. Eight test images; (a) Jet; (b) Lena; (c) Peppers; (d) Baboon; (e) Barb; (f) Boat; (g) Tiffany; (h) Zelda.

Table 2

<table>
<thead>
<tr>
<th>Block numbers of size $2^a \times 2^b$</th>
<th>Embedded quantity (bits)</th>
<th>Lee's $2 \times 2$ method</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block numbers of size 16 × 16</td>
<td>$b_1$</td>
<td>$2^{a-1} \times 2^{b-1} \times 2 \times 2$ blocks of size 2 × 2</td>
<td>≥126 $b_1$</td>
</tr>
<tr>
<td>Embedded quantity (bits)</td>
<td>$2^{a-1} \times 2^{b-1} \times 2 \times 2$ blocks of size 2 × 2</td>
<td>≥126 $b_1$</td>
<td></td>
</tr>
<tr>
<td>Block numbers of size 8 × 8</td>
<td>$b_2$</td>
<td>$2^{a-1} \times 2^{b-1} \times 2 \times 2$ blocks of size 2 × 2</td>
<td>≥6 $b_2$</td>
</tr>
<tr>
<td>Embedded quantity (bits)</td>
<td>$2^{a-1} \times 2^{b-1} \times 2 \times 2$ blocks of size 2 × 2</td>
<td>≥6 $b_2$</td>
<td></td>
</tr>
<tr>
<td>Block numbers of size 4 × 4</td>
<td>$b_3$</td>
<td>$2^{a-1} \times 2^{b-1} \times 2 \times 2$ blocks of size 2 × 2</td>
<td>≥24 $b_3$</td>
</tr>
<tr>
<td>Embedded quantity (bits)</td>
<td>$2^{a-1} \times 2^{b-1} \times 2 \times 2$ blocks of size 2 × 2</td>
<td>≥24 $b_3$</td>
<td></td>
</tr>
</tbody>
</table>

Watermarks are randomly generated. The proposed method adaptively segments a host image to blocks of sizes from $32 \times 32$ to $2 \times 2$. Largest block size selection is $32 \times 32$ because no partitioned block of size larger than $32 \times 32$ is found in these test images. Strategies comparison is provided in Table 1. The experimental results are compared with Lee et al. (2008) method as segmenting an image to $2 \times 2$ blocks. The threshold $T$ is empirically determined by 4, both in the proposed method and Lee's method.

Tables 3 and 4 show the experimental results comparing with Lee’s method on above eight test images. In these two tables, numbers of different block size from $32 \times 32$ to $2 \times 2$ are listed in the proposed method. On the other hand, only number of $2 \times 2$ blocks are listed in Lee’s method due to their work only considering fixed $m \times n$ block processing. In these eight images, Jet has the maximum smooth areas, which reveals $28$ blocks of size $16 \times 16$ and $(16 \times 16 - 1) \times 2 \times 28 = 14,280$ bits to embed into these blocks. Relatively, Lee’s method only embeds at most $28 \times 8 \times 8 \times 6 = 10,752$ bits at these 28 blocks. The improvement only on these 28 blocks of size $16 \times 16$ is 3528 bits. So is the Jet image has 28 blocks of size $16 \times 16$, 229 blocks of size $8 \times 8$, 2375 blocks of size $4 \times 4$, and other $2 \times 2$ blocks leads to capacity of 308,016 bits, which is also equals to 1.175 bpp. Comparing with Lee’s 282,147 bits, the proposed method improves the capacity of 9.1% as 25,869 bits and the watermarked image quality only reduces about 1%. Six images Lena, Peppers, Barb, Boat, Tiffany and Zelda have fewer smooth areas, thus acquiring no blocks of size larger than $2 \times 2$. Therefore, Baboon has the same segmentation result both in the proposed method and Lee’s method. Consequently, above analysis acquires the conclusion that the embedded capacity depends on the image structure. Complex image, like Baboon, is hard to find image blocks larger than $2 \times 2$. Therefore, the embedded capacity is the same as Lee’s method. An image with more smooth area, like Jet, generates more large blocks and then leads to higher embedded capacity.

Fig. 4 shows the watermarked image quality (PSNR) and embedded capacity (bpp) change under different thresholds. The test...
### Table 3
Comparisons with Lee's method on test images (a)–(d).

<table>
<thead>
<tr>
<th>Test Image</th>
<th>Jet The proposed method</th>
<th>Jet Lee's $2 \times 2$ method</th>
<th>Lena The proposed method</th>
<th>Lena Lee's $2 \times 2$ method</th>
<th>Peppers The proposed method</th>
<th>Peppers Lee's $2 \times 2$ method</th>
<th>Baboon The proposed method</th>
<th>Baboon Lee's $2 \times 2$ method</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 × 32 block numbers</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16 × 16 block numbers</td>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8 × 8 block numbers</td>
<td>229</td>
<td>9</td>
<td>12</td>
<td>443</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4 × 4 block numbers</td>
<td>2375</td>
<td>18,251</td>
<td>18,075</td>
<td>21,389</td>
<td>13,822</td>
<td>15,653</td>
<td>2616</td>
<td>2616</td>
</tr>
<tr>
<td>2 × 2 block numbers, 2 bits embedded</td>
<td>23,136</td>
<td>23,543</td>
<td>35,709</td>
<td>36,027</td>
<td>42,292</td>
<td>42,425</td>
<td>26,599</td>
<td>26,599</td>
</tr>
<tr>
<td>2 × 2 block numbers, 1 bit embedded and record</td>
<td>4906</td>
<td>4906</td>
<td>5707</td>
<td>5707</td>
<td>4835</td>
<td>4835</td>
<td>17,773</td>
<td>17,773</td>
</tr>
<tr>
<td>2 × 2 block numbers, 0 bit embedded</td>
<td>4287</td>
<td>4287</td>
<td>2413</td>
<td>2413</td>
<td>2623</td>
<td>2623</td>
<td>18,548</td>
<td>18,548</td>
</tr>
<tr>
<td>PSNR</td>
<td>34.3308</td>
<td>34.7083</td>
<td>34.1440</td>
<td>34.2579</td>
<td>34.2573</td>
<td>34.2856</td>
<td>35.599</td>
<td>35.599</td>
</tr>
<tr>
<td>PSNR decreasing rate</td>
<td>1.10%</td>
<td>0.33%</td>
<td>0.08%</td>
<td>0.08%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Capacity</td>
<td>308,016</td>
<td>282,147</td>
<td>259,992</td>
<td>253,536</td>
<td>239,115</td>
<td>235,698</td>
<td>148,782</td>
<td>148,782</td>
</tr>
<tr>
<td>Capacity increasing rate</td>
<td>9.17%</td>
<td>2.55%</td>
<td>1.45%</td>
<td>1.45%</td>
<td>1.45%</td>
<td>1.45%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

### Table 4
Comparisons with Lee's method on test images (e)–(h).

<table>
<thead>
<tr>
<th>Test Image</th>
<th>Barb The proposed method</th>
<th>Barb Lee's $2 \times 2$ Method</th>
<th>Boat The proposed method</th>
<th>Boat Lee's $2 \times 2$ Method</th>
<th>Tiffany The proposed method</th>
<th>Tiffany Lee's $2 \times 2$ Method</th>
<th>Zelda The proposed method</th>
<th>Zelda Lee's $2 \times 2$ Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 × 32 block numbers</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16 × 16 block numbers</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8 × 8 block numbers</td>
<td>2</td>
<td>57</td>
<td>14</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4 × 4 block numbers</td>
<td>755</td>
<td>14,929</td>
<td>17,712</td>
<td>15,355</td>
<td>26,604</td>
<td>12,894</td>
<td>40,260</td>
<td>21,341</td>
</tr>
<tr>
<td>2 × 2 block numbers, 2 bits embedded</td>
<td>26,097</td>
<td>26,446</td>
<td>26,755</td>
<td>27,702</td>
<td>39,473</td>
<td>39,537</td>
<td>40,271</td>
<td>40,442</td>
</tr>
<tr>
<td>2 × 2 block numbers, 1 bit embedded and record</td>
<td>9741</td>
<td>9741</td>
<td>8353</td>
<td>8353</td>
<td>7355</td>
<td>7355</td>
<td>3161</td>
<td>3161</td>
</tr>
<tr>
<td>2 × 2 block numbers, 0 bit embedded</td>
<td>11,637</td>
<td>11,637</td>
<td>4877</td>
<td>4877</td>
<td>4338</td>
<td>4338</td>
<td>592</td>
<td>592</td>
</tr>
<tr>
<td>PSNR</td>
<td>34.9534</td>
<td>35.0721</td>
<td>34.0554</td>
<td>34.3063</td>
<td>34.1860</td>
<td>34.2155</td>
<td>34.1179</td>
<td>34.1585</td>
</tr>
<tr>
<td>PSNR decreasing rate</td>
<td>0.13%</td>
<td>0.74%</td>
<td>0.085%</td>
<td>0.085%</td>
<td>0.12%</td>
<td>0.12%</td>
<td>0.12%</td>
<td>0.12%</td>
</tr>
<tr>
<td>Capacity</td>
<td>219,990</td>
<td>214,833</td>
<td>274,206</td>
<td>255,789</td>
<td>229,002</td>
<td>224,697</td>
<td>261,258</td>
<td>257,733</td>
</tr>
<tr>
<td>Capacity increasing rate</td>
<td>2.4%</td>
<td>7.22%</td>
<td>1.92%</td>
<td>1.92%</td>
<td>1.37%</td>
<td>1.37%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>
image is Jet and this figure demonstrates how the Capacity and PSNR change under different thresholds. Experimental results of our proposed method are depicted by dotted line and results of Lee et al.'s method is depicted by solid line. Two significant properties are discovered in this figure. Firstly, the proposed method performs better than Lee’s method both in capacity and PSNR. For example, when \( T = 3.5 \), the proposed method has PSNR 34.6546 bpp and capacity 1.166 bpp (305,589 bits). The similar embedded image quality in Lee’s method locates at \( T = 4 \) and acquires PSNR 34.7083 bpp and capacity 1.076 bpp (282,147 bits). The proposed method has higher capacity than Lee’s method under almost the same embedded image quality. Secondly, the threshold \( T \) determines the capacity vs. PSNR. Fig. 4 shows that both in the proposed method and Lee’s method, \( T = 4.25 \) to \( T = 5 \) performs closely both in capacity vs. PSNR. Also, \( T = 3.25 \) to \( T = 4 \) performs closely and so is \( T = 2.5 \) to \( T = 3 \). Therefore, integer threshold selection like \( T = 3, 4, \) or 5 is suggested to acquire good watermarked image.

5. Conclusion

This study presents an adaptive block sized reversible image watermarking approach. The embedding method fixes the central pixel in an ordered block and increases differences between the fixed pixel and others to embed watermarks. Therefore, blocks of different size are adaptively utilized to embed more watermarks than conventional \( m \times n \) block scheme. Experimental results show that the proposed adaptive reversible image watermarking approach raises embedded capacity from 9.1% to 0.0% according to the image structure and the image distortion only reduces from 1.10% to 0.0%. These results demonstrate that the proposed method increases high embedded capacity with only little image distortion.

Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References


Chien-Chang Chen received the BS degree from Department of Computer and Information Science at Tunghai University, Taichung, in 1991, and the PhD degree in Computer Science from National Tsing Hua University in 1999. From 2001 to 2010, he was an Assistant, and Associate Professor with the Department of Information Management at Hsuan Chuang University. Currently, he is an Associate Professor with the Department of Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan. His research interests include secret image sharing, watermarking, and texture analysis.

Yao-Hong Tsai received the MS and PhD degrees in information management from the National Taiwan University of Science and Technology, Taipei, Taiwan in 1994 and 1999, respectively. He is a Assistant Professor with the Department of Information Management, Institute of Information Management, Hsuan Chuang University, Hsinchu, Taiwan. His current research interests include image processing, computer vision, and pattern recognition.