Geometric Invariant Digital Image Watermarking Scheme Based on Histogram in DWT Domain

Chi-Man Pun and Xiao-Chen Yuan
Department of Computer and Information Science, University of Macau, Macau SAR, China
Email: {cmpun, ma86558}@umac.mo

Abstract—A robust and geometric invariant digital watermarking scheme for gray-level images is proposed in this paper. The scheme carries out watermark embedding and extraction based on histogram in DWT domain. For watermark embedding, firstly, the original image is decomposed into the approximation and details sub-bands, of which, the approximation sub-band is used for watermark embedding. Pixels of the approximation sub-band are grouped into \( m \) blocks, each of which has the same number of gray-levels, thus the block histogram is generated; with the block histogram, the percentage of number of pixels in each block is calculated. Then some pixels in a block are moved to form a specific pattern in the gray-level histogram distribution, indicating the watermark. Finally, the embedded approximation sub-band and the other three details sub-bands are constructed into a watermarked image. For watermark extraction, firstly, the watermarked image is also decomposed into the approximation and details sub-bands; then the pixels in the approximation sub-band are grouped into blocks in the similar manner. According to the histogram distribution in each block, the watermark is extracted. Experimental results show that the proposed scheme is highly robust against JPEG compression, geometric attacks and some common signal processing, and it outperforms the existing methods in term of robustness.

Index Terms— Robust Watermarking, Geometric Invariant, Histogram, Discrete Wavelet Transform

I. INTRODUCTION

Digital watermarking, a technique of hiding secret information in the images, has emerged as a potentially effective tool for multimedia copyright protection, authentication and tamper proofing. Meanwhile, the robustness against JPEG compression, geometrical attacks, and signal processing, such as RST (rotation, scaling, and translation), flipping, and median filter, especially when some of these are combined together, is one of the key issues for some applications. A lot of significant processes have been made with the proposed schemes in the past years. Tang and Hang [1] proposed a scheme which combines the advantages of feature extraction and image normalization together to resist image geometric distortion and to reduce watermark synchronization problem at the same time. Simulation results show that the scheme can survive low-quality JPEG compression, some geometric attacks and common signal processing. The problem is it depends on the features of the individual images, thus the embedding method may not be compatible for all images. Kang et al. [2] proposed a blind Discrete Wavelet Transform-Discrete Fourier Transform (DWT-DFT) composite image watermarking algorithm. The scheme embeds a message and a training sequence in sub-band LL4 in the DWT domain, and embeds a template in magnitude spectrum in DFT domain. Experimental results demonstrate that the watermark is robust against both affine transformations and JPEG compression. However, the robustness of the informative watermark against median filtering needs to be improved.

Fang and Zhao [3] proposed an image watermarking scheme based on histogram. Different groups of the image histogram bars are chosen, and the message is embedded to sum of every group by using Quantization Index Modulation (QIM). Experimental results show that the scheme is robust to geometrical attacks as well as compression and filtering. Yoo et al. [4] propose a new fragile watermarking algorithm based on localized histogram modification, which can identify the distorted part and recover the rest of image perfectly if the watermarked image is attacked. Experimental results show that the original images can be recovered perfectly except the part of attack. Wu [5] proposed a reversible semi-fragile watermarking scheme, using histogram shifting of integer wavelet coefficients to hide data. Since the histogram shifting operation only modifies the integer wavelet coefficients slightly, the proposed scheme has relatively higher visual quality. Experimental results show that the presented scheme can tolerate JPEG compression at a lower quality factor. Lin et al. [6] proposed a novel approach, based on the properties of histograms, called Histogram-Oriented Watermarking Algorithm (HOWA), to measure the numerous global features of all pixels in a cover image and to construct the three-dimensional feature space. The feature space is dynamically partitioned to identify several blocks used to embed the watermark. One feature of the pixels is modified to form a specially distributed histogram for embedding the watermark in a blind digital watermarking method that can be applied to color images. Experimental results demonstrate the robustness of the proposed
method against common geometric attacks and signal processing.

Tripathi et al. [7] proposed a digital watermarking scheme using the properties of Discrete Cosine Transform (DCT) and Discrete Wavelet Transforms (DWT) to achieve almost zero visible distortion in the watermarked images. The experimental results have shown that the proposed technique will not be threatened by image cropping and JPEG lossy compression as at least 65% extraction is ensured in most cases. Kang et al. [2] proposed a DWT-DFT composite watermarking scheme robust against both affine transformations and JPEG compression. It embeds a message and a training sequence in sub-band in the DWT domain, and embeds a template in magnitude spectrum in DFT domain. Experimental results have demonstrated that the watermark is robust against both affine transformations and JPEG compression when the quality factor is as low as 10. Sun et al. [8] proposed a novel binary image digital watermarking algorithm based on DWT and chaotic encryption. The watermarks are embedded to the low frequency coefficients of deep wavelet domain. The experimental results show that the watermark is robust against the cut attack, salt-and-pepper noise attack, JPEG compression and so on. Zhao [9] proposed a method for watermarking of chaotic sequence and DWT. The original watermark was encrypted by chaotic sequence mapping and embedded into the low frequency sub-band. The coefficients whose energies were larger than the others were selected to hide watermark. Simulation results show that this method can gain high fidelity and high robustness, especially under the typical attack of geometric operations.

While [1] use the features in the images as reference points for watermarks embedding and extraction (feature-based) and in [2], a template is used to detect geometric distortions and to resynchronize the image (template-based); [3-6, 10-14] are some robust watermarking schemes that based on histogram (histogram-based), and [2, 7-9] are the ones based on DWT.

Considering the algorithm HOWA presented by Lin et al. [6], in this paper, we propose a novel technique based on DWT and histogram. Although the standard DWT is not geometric invariant, our proposed watermarking algorithm only embeds the watermark bits into the approximation coefficients which have only little effects on geometric attacks. The scheme embeds the watermark in the approximation sub-band of the wavelet-decomposed image, on base of the properties of histograms. For watermark embedding, firstly, the original image is decomposed into the approximation and details sub-bands, by DWT. Pixels of the approximation sub-band, which is used for watermark embedding, are grouped into m blocks, each of which has the same number of gray-levels, thus the block histogram is generated; with the block histogram, the percentage of number of pixels in each block is calculated and encoded as a key. Then some pixels in a block are moved to form a specific pattern in the gray-level histogram distribution, indicating the watermark. Finally, the embedded approximation sub-band and the other three details sub-bands are constructed by Inverse Discrete Wavelet Transform (IDWT) into a watermarked image. For watermark extraction, firstly, the watermarked image is also decomposed into four sub-bands; then, the key is decoded and the pixels in the LL sub-band are grouped into blocks with the key. According to the histogram distribution in each block, the watermark is extracted.

The paper is organized as follows. Section 2 introduces the standard 2-D Discrete Wavelet Transform. Section 3 shows the procedure of watermark embedding in DWT domain. Then the watermark extraction process of the proposed scheme is explained in section 4. After that, the experimental results are illustrated in section 5. And finally the conclusions are drawn in section 6.

II. STANDARD 2-D DISCRETE WAVELET TRANSFORM

When digital image are to be viewed or processed at multiple resolutions, DWT is the mathematical tool of choice [8]. DWT is a new signal analytic theory that can localize the signal in spatiotemporal and it has already been widely used. The basic idea of applying DWT on the image processing is using discrete wavelet transform, by which, the original image can be decomposed into lower and higher frequency sub-bands. Especially, higher frequency components of images have the self similarity between each frequency components, horizontally, vertically and diagonally. This self-similarity can be derived from the correlation coefficients between lower frequency components and their similar high frequency components [15]. DWT gives a new way to the local analyses of the image. In 2D-DWT analysis, an image is split into an approximation and three detail images. The approximation image is then itself split into a second-level approximation and detail images, and the process is recursively repeated. So there are n+1 possible ways to decompose or encode the image for the n-level decomposition. The standard 2D-DWT can be described by a pair of quadrature mirror filters (QMF) H and G. The filter H is a low-pass filter with a finite impulse response denoted by $h(n)$. And the high-pass G with a finite impulse response is defined by:

$$g(n) = (-1)^{n} h(1 - n), \text{ for all } n$$

(1)

The 2D discrete wavelet decomposition of a $N \times M$ discrete image x up to level $p+1$ ($p \leq \min(\log_2(N), \log_2(M)), p \in \mathbb{N}$) is recursively defined in terms of the coefficients in the approximation image only of level $p$ as follows:
\[ C_{4k+1}^{p+1} = \sum_{m} \sum_{n} h(m)h(n)C_{k,m+2i,n+2j}^p \]  \hspace{1cm} (2)

\[ C_{4k+1}^{p+1} = \sum_{m} \sum_{n} h(m)g(n)C_{k,m+2i,n+2j}^p \]  \hspace{1cm} (3)

\[ C_{4k+2}^{p+1} = \sum_{m} \sum_{n} g(m)h(n)C_{k,m+2i,n+2j}^p \]  \hspace{1cm} (4)

\[ C_{4k+3}^{p+1} = \sum_{m} \sum_{n} g(m)g(n)C_{k,m+2i,n+2j}^p \]  \hspace{1cm} (5)

where \( C_{0,(i,j)}^0 = x_{(i,j)} \) is given by the intensity levels of the image \( x \).

In practice, the image \( x \) has only a finite number of pixels. Different methods such as symmetric, periodic or zero padding could be used for the boundary handling. At each step, we decompose the approximation image \( C_k^p \) into four quarter-size images, \( C_{4k+1}^{p+1} \), \( C_{4k+1}^{p+1} \), \( C_{4k+2}^{p+1} \) and \( C_{4k+3}^{p+1} \). This decomposition algorithm can also be illustrated by the block diagram in Fig. 1. The rows of the image \( C_k^p \) are first convolved with a one-dimensional filter and every other row is retained. Then the columns of the resulting images are convolved with another one-dimensional filter and every other column is retained. Fig. 2 shows one-step decomposition and two-step decomposition of 2-D DWT.

III. WATERMARK EMBEDDING BY HISTOGRAM IN DWT DOMAIN

For watermark embedding, firstly, the original image is decomposed into the approximation and details sub-bands, of which, the approximation sub-band is used for watermark embedding. Pixels of the approximation sub-band are grouped into \( m \) blocks, each of which has the same number of gray-levels, thus the block histogram is generated; with the block histogram, the percentage of number of pixels in each block is calculated. Then some pixels in a block are moved to form a specific pattern in the gray-level histogram distribution, indicating the watermark. Finally, the embedded approximation sub-band and the other three details sub-bands are constructed into a watermarked image. Fig. 3 shows the flow chart of the watermark embedding scheme proposed in this paper, and the detailed steps are as following:

**Fig. 1.** Decomposition of a discrete image \( C_k^p \) into four quarter-size images, \( C_{4k+1}^{p+1} \), \( C_{4k+1}^{p+1} \), \( C_{4k+2}^{p+1} \) and \( C_{4k+3}^{p+1} \) by using the conjugate filters \( H \) and \( G \).

**Fig. 2.** (a) Original image; (b) One-step decomposition; (c) Two-step decomposition

**Fig. 3.** Flow chart of watermark embedding scheme

**STEP-1:** Decompose the original image into four components: \( cA, cH, cV, cD \), using 2D-DWT;

**STEP-2:** Apply the watermark embedding algorithm, Histogram Based Algorithm (HBA), into component \( cA \), and get the embedded one, \( cA_{emb} \);

**STEP-3:** Construct the four parts: \( cA_{emb}, cH, cV, cD \), into one image, using Inverse 2D-DWT, thus get the watermarked image.
For HBA, mentioned in STEP-2, pixels of the embedding part are grouped into \( m \) blocks, each of which has the same number of gray-levels; the \( w \) bits of embedding data \( W \) are embedded into \( m \) blocks, respectively. To implement the embedding process, each block is divided into four sub-blocks according to Gray Level (GL) value and Composite Gradient (CG) value of each pixel. Then, some pixels in a block are moved among the sub-blocks to form a specific pattern in the gray-level histogram distribution and thereby a watermarking data bit is embedded. The procedure of the watermark embedding algorithm HBA is shown in Fig. 4.

![Fig. 4. Procedure of the embedding algorithm HBA](image)

The following sub-chapters will respectively illustrate the steps of watermark embedding algorithm HBA in detail: A. Generation of Block Histogram; B. Sub-block Division & Data Preprocessing; and C. Embedding Implementation.

**Generation of Block Histogram**

Block histogram in HBA is generated by grouping the pixels in the embedding parts into \( m \) blocks, each of which has the same number of gray-levels. Besides, the number of blocks should equal or exceed the number of watermarking data bits. The steps of block histogram generation are as following and Fig. 5 shows the flow chart.

**STEP-1:** The original image \( I \) of size \( M \times N \) is loaded, as Fig. 5-(a) shows. \( I(x,y) \) indicates the gray-level of the pixel located at position \( (x,y) \).

**STEP-2:** Reshape the image matrix \( I \) into a one dimensional matrix \( I' \) with \( M \times N \) pixels and sort the pixels according to their gray-level value, as Fig. 5-(b) shows.

**STEP-3:** Calculate the number of gray-levels in \( I' \).

**STEP-4:** Group all pixels into \( m \) dynamic sized blocks, in each block, the number of gray-levels should approximately equal to \( n \), as shown in Fig. 5-(d). Note that if \( (M \times N/m = n) \) is an integer, then, the number of gray-levels in each block is \( n \); else, assuming the remainder is \( r \), then the number of gray-levels of the first \( r \) blocks is \( n+1 \), while the others are \( n \).

![Fig. 5. Procedure of Block Histogram Generation](image)

The block histogram distribution is shown in Fig. 6. Each block has the same number of gray-levels but varied number of pixels; and for each block in Fig. 6-(a), pixels are located in two-dimensional coordinates, defined by features GL and CG, as Fig.6-(b) shows, which will be explained later.

According to the block histogram and the total number of pixels in original image, the percentage of number of pixels in each block is calculated and then encoded as a KEY, which is for watermark extraction of HBA use later. Here the reason for calculating the percentage of number of pixels instead of adopting number of pixels directly is that, number of pixels will need a great many bits to save, which is not convenient for extraction, while the percentage of number of pixels only takes several bits.

**Sub-block Division & Data Preprocessing**

After the block histogram is generated, each block is divided into four sub-blocks, according to GL and CG value of each pixel. Firstly, the pixels in each block are sorted according to their GL and CG value, respectively. Then the \( GL_{m} \) and \( CG_{m} \) in Fig. 6-(b) can be gained. Therefore, four sub-blocks are created, A, B, C, and D in...
It’s easy to find that the number of pixels in sub-block A equals to that in D, and the same for B and C.

The CG(x,y) is obtained using the following equation [6]:

\[ CG(x,y) = \sqrt{g_x(x,y)^2 + g_y(x,y)^2 + g_{xx}(x,y)^2 + g_{yy}(x,y)^2} \]  

In equation (6), \( g_x(x,y), g_y(x,y), g_{xx}(x,y), g_{yy}(x,y) \) are gradients in the four directions for each pixel. The capacity for resisting rotation attacks increases by using the composite gradient.

For watermark embedding of HBA, some pixels are moved among the four sub-blocks in a block to form a specific pattern in the gray-level histogram distribution and thereby a watermarking data bit is embedded. To implement the pixel moving among the sub-blocks, the GL and CG value of the pixels need to be preprocessed to ensure the uniqueness of the GLm and CGm. That is, the GL and CG of pixels in both sides of GLm and CGm should be different from GLm and CGm, respectively. Referring to Fig. 6-(b), completely four cases for this consideration are listed in the following:

**CASE 1**: If the pixel belongs to sub-block C or D, but the CG of the pixel equals to CGm, the CG will be decreased by minus 0.001;

**CASE 2**: If the pixel belongs to sub-block A or B, but the CG of the pixel equals to CGm, the CG will be increased by adding 0.001;

**CASE 3**: If the pixel belongs to sub-block A or C, but the GL of the pixel equals to GLm, the GL will be decreased by minus 0.001;

**CASE 4**: If the pixel belongs to sub-block B or D, but the GL of the pixel equals to GLm, the GL will be increased by adding 0.001

**Embedding Implementation**

For watermark embedding process, some pixels in a block are moved among the sub-blocks to form a specific pattern in the gray-level histogram distribution. Two variables, \( \alpha \) and \( \beta \), are needed. \( \alpha \) indicates the ratio of the number of pixels to be moved to the number of pixels in a single sub-block, \( 0 \leq \alpha \leq 1 \), and it specifies the robustness of watermark; While \( \beta \) specifies the resistance to JPEG attack, \( 0 \leq \beta \leq 1 \).

Let \( NA + NB = 2C \) (\( NA, NB \) means the number of pixels in sub-block A and B, respectively), \( W(l) \), the \( l \)-th watermarking data bit is embedded into Block Bl. If \( W(l) = 1 \), some pixels in B and C are moved into A and D, respectively, to satisfy equation (2); Otherwise, for \( W(l) = 0 \), some pixels in A and D are moved into B and C, respectively, to satisfy equation (3). The watermark is more robust when \( \alpha \) is larger, especially, when \( \alpha = 1 \), in the condition of \( W(l) = 1 \), all pixels in B and C are moved into A and D. However, some of the regions of the watermarked image may be visibly modified as \( \alpha \) increases.

\[
\begin{align*}
NA &\geq ND \geq (1+\alpha)C \\
NB &\geq NC \geq (1-\alpha)C \\
NB &\geq NC \geq (1+\alpha)C \\
NA &\geq ND \leq (1-\alpha)C
\end{align*}
\]  

On the other hand, two margins, \( GL_i \) and \( GL_m \), are set based on equations (7), to cause a pixel to move unambiguously from one sub-block to another in order to resist JPEG compression attack. The boundary among the sub-blocks becomes clearer as \( \beta \) increases. [6] Fig. 7 illustrates the modification of moving pixels.

\[
\begin{align*}
GL_i &= GL_m - \beta(GL_m - GL_l) \\
GL_l &= GL_m + \beta(GL_m - GL_l)
\end{align*}
\]  

**IV. WATERMARK EXTRACTION IN DWT DOMAIN**

For watermark extraction, firstly, the watermarked image is also decomposed into the approximation and details sub-bands; then the pixels in the approximation sub-band are grouped into blocks in the similar manner. According to the histogram distribution in each block, the watermark is extracted. Fig. 8 shows the flow chart of watermark extraction of the proposed scheme, and the detailed process is as following:

**STEP-1**: Decompose the watermarked image into four components: \( cA_{emb}, cH, cV, cD \), using 2D-DWT;

**STEP-2**: Apply the watermark extraction algorithm of HBA into component \( cA_{emb} \);

**STEP-3**: Get the extracted watermark from the image.
The procedure of watermark extraction algorithm of HBA is shown in Fig. 9, and it is detailedly explained in the following:

**STEP-1:** The watermarked image $I_{emb}$ of size $M \times N$ is loaded. $I_{emb}(x,y)$ indicates the gray-level of the pixel located at position $(x,y)$.

**STEP-2:** Reshape the watermarked image matrix $I_{emb}$ into a one dimensional matrix $I_{emb'}$ with $M \times N$ pixels and sort the pixels according to their GL value.

**STEP-3:** Decode the Key and calculate the number of pixels for each block.

**STEP-4:** Group all pixels into $m$ blocks according to the previous result.

**STEP-5:** Calculate the CG value of pixels using equation (1) and get the $CG_m$ values for each block in watermarked image. This process is the same as STEP-1: Sub-block Division in 3.1.

**STEP-6:** Divide each block into four sub-blocks, $A'$, $B'$, $C'$, and $D'$.

**STEP-7:** Count the number of pixels in sub-block $A'$, $B'$, $C'$, and $D'$. If $NA' + ND' > NB' + NC'$, then the embedded data bit is ‘1’, $W(l) = 1$; otherwise, the embedded data bit is ‘0’, $W(l) = 0$. Thus, the embedded data sequence is extracted from the watermarked image.

![Fig. 9. Procedure of the extraction algorithm of HBA](image)

Note that although HBA is proposed based on HOWA algorithm, they are essentially different. While HOWA groups the pixels into blocks with the same number of pixels in each block, for both embedding and extraction procedure; HBA groups the pixels into dynamic sized blocks with the same number of gray-level in each block, thus generates the block histogram. Along with the block histogram, the KEY is generated, which is used for watermark extraction and thus improve the robustness of watermark.

V. EXPERIMENTAL RESULTS

Many standard gray images of size $512 \times 512$ from The USC-SIPI Image Database [16] are used to evaluate the scheme performance. In our experiments, a random number generator is used to generate the watermark $W$, which is a sequence of $m$ bits of “0” and “1” bits. A 16-bit watermark of sequence of ‘0’ and ‘1’ is embedded into the testing images. PSNR (Peak Signal-to-Noise Ratio) is used to evaluate the distortion of the embedded image. The larger PSNR value means less distortion.

However, based on the gray-level histogram distribution, a watermark data bit can be extracted from a block in an image without an embedded watermark, thus a threshold value $T$ needs to be defined to prevent false alarms [6]. The probability of a false alarm in $n$ bits watermarks is given by the following equation [1].

$$P_{false-alarm} = \sum_{i=0}^{n} C_n^i \left(\frac{1}{2}\right)^i = \sum_{i=0}^{n} \frac{n!}{i!(n-i)!} \left(\frac{1}{2}\right)^i$$  \hspace{1cm} (9)

According to equation (9), for $n = 16$, when the threshold $T$ exceeds 12, the false alarm probability will be less than 0.038. Therefore, in this simulation, the threshold is set to 12 to judge whether the watermark are successfully extracted from an image.

As TABLE 1 shows, the PSNR value decreases, which means the watermark invisibility decrease, as the values of $\alpha$ and $\beta$ increase. As mentioned previously, $\alpha$ in (2) and (3) specifies the robustness of watermarking, while $\beta$ in (4) specifies the strength to resist JPEG attack. The simulation results in Table-1 reveal that considering both robustness and invisibility, the appropriate values of $\alpha$ and $\beta$ are: 0.7 and 0.5, respectively. Until now, all the simulation parameters have been gained: 16-bits watermark, $\alpha = 0.7$, $\beta = 0.5$, $T = 12$.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>PSNR</th>
<th>Number of Correct Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.3</td>
<td>44.57/10</td>
<td>37.43/14</td>
</tr>
<tr>
<td>0.3</td>
<td>0.5</td>
<td>41.70/12</td>
<td>33.48/14</td>
</tr>
<tr>
<td>0.5</td>
<td>0.7</td>
<td>37.62/10</td>
<td>30.88/15</td>
</tr>
<tr>
<td>0.7</td>
<td>0.9</td>
<td>33.51/15</td>
<td>28.52/15</td>
</tr>
<tr>
<td>0.9</td>
<td>1.1</td>
<td>31.54/15</td>
<td>27.68/15</td>
</tr>
</tbody>
</table>

**TABLE 1** shows the PSNR values and the numbers of correct bits extracted from the embedded image, in the form of X/Y. X indicates the **PSNR** of the embedded image and Y indicates the number of correct bits extracted from the watermarked image which is attacked by JPEG compression with a quality factor of 50, image Median filtering of $2 \times 2$, and image cropping of 10% together. The bold and italic result indicates the best simulation result, considering both the PSNR and Number of Correct Bits.
Fig. 10. Watermarked images with the corresponding PSNR results. (a1), (b1), (c1), and (d1) shows the original images: Lena, Baboon, Pepper, and Jet; (a2), (b2), (c2), and (d2) shows the corresponding watermarked images: Lena_emb, Baboon_emb, Pepper_emb, Jet_emb.

Fig. 10 shows several watermarked images with their PSNR value. Fig. 10-(a1), (b1), (c1), and (d1) are the original images: Lena, Baboon, Pepper, and Jet; Fig. 10-(a2), (b2), (c2), and (d2) shows their corresponding watermarked images. The PSNR between the original image and its watermarked image are also given.

TABLE 2 - TABLE 6 present the simulation results of numerous attacks, including geometric distortion, JPEG compression and also some common signal processing.

TABLE 2 shows the extraction results after JPEG compression attack. QF indicates the effect of JPEG compression. The bigger the QF is, the better the compression quality will be, and the smaller the compression quality will be. It demonstrates that the proposed scheme robust against JPEG compression with a low quality factor of up to 20.

<table>
<thead>
<tr>
<th>Median Filter</th>
<th>Lena</th>
<th>Baboon</th>
<th>Pepper</th>
<th>Jet</th>
<th>Successful?</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 × 2</td>
<td>16</td>
<td>14</td>
<td>15</td>
<td>15</td>
<td>YES</td>
</tr>
<tr>
<td>4 × 4</td>
<td>14</td>
<td>12</td>
<td>15</td>
<td>14</td>
<td>YES</td>
</tr>
<tr>
<td>6 × 6</td>
<td>16</td>
<td>12</td>
<td>14</td>
<td>14</td>
<td>YES</td>
</tr>
</tbody>
</table>

In TABLE 4, the watermarked images are cropped in different proportions, starting from the left up corner and the cropping zone is square (cropping proportion = the width of cropping zone / the width of watermarked image). The results demonstrate that the proposed scheme robust against image cropping of up to 20%.

TABLE 4 shows the extraction results after median filter. Median filter losses lots of the details of image, but to the wave coefficients, what it destroys most is the high frequency coefficients; meanwhile, the coefficients chosen to embed watermarks are in the low frequency zone, thus, our scheme robust to median filter [17].

<table>
<thead>
<tr>
<th>Median Filter</th>
<th>Lena</th>
<th>Baboon</th>
<th>Pepper</th>
<th>Jet</th>
<th>Successful?</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 × 2</td>
<td>14</td>
<td>16</td>
<td>16</td>
<td>15</td>
<td>YES</td>
</tr>
<tr>
<td>4 × 4</td>
<td>16</td>
<td>16</td>
<td>14</td>
<td>14</td>
<td>YES</td>
</tr>
<tr>
<td>6 × 6</td>
<td>16</td>
<td>12</td>
<td>14</td>
<td>14</td>
<td>YES</td>
</tr>
</tbody>
</table>

In TABLE 5, the watermarked images are cropped in different proportions, starting from the left up corner and the cropping zone is square (cropping proportion = the width of cropping zone / the width of watermarked image). The results demonstrate that the proposed scheme robust against image cropping of up to 20%.

TABLE 5 shows the extraction results after other attacks. It demonstrates that the proposed scheme robust against image flipping, jitter attacks of random removal of rows and columns, and image scaling with a factor bigger than 0.5.

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Lena</th>
<th>Baboon</th>
<th>Pepper</th>
<th>Jet</th>
<th>Successful?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jitter_17_5</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>YES</td>
</tr>
<tr>
<td>Jitter_256</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>YES</td>
</tr>
<tr>
<td>Flip</td>
<td>16</td>
<td>12</td>
<td>15</td>
<td>15</td>
<td>YES</td>
</tr>
<tr>
<td>Scale_0_2</td>
<td>14</td>
<td>9</td>
<td>15</td>
<td>14</td>
<td>NO</td>
</tr>
<tr>
<td>Scale_0_5</td>
<td>14</td>
<td>14</td>
<td>16</td>
<td>14</td>
<td>YES</td>
</tr>
<tr>
<td>Scale_0_8</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>YES</td>
</tr>
<tr>
<td>Scale_1_2</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>YES</td>
</tr>
<tr>
<td>Scale_1_5</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>15</td>
<td>YES</td>
</tr>
<tr>
<td>Scale_2</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>YES</td>
</tr>
<tr>
<td>Scale_4</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>YES</td>
</tr>
</tbody>
</table>

TABLE 6 shows that the scheme is only robust against rotation with the rotation angle no larger that 10 degrees; which needs to be improved in future work.

TABLE 6 shows the extraction results after image rotation. The results indicate the number of correct bits extracted from the watermarked image, of which, the italic data indicates the number of correct extracted bits which is below the threshold, which means the extraction failed. Although this scheme is robust against various attacks, which is illustrated in TABLE 2 - TABLE 5, it is not well robust against image rotation, as the results in TABLE 6 shown. It will be one of the points which we
will work on in future.

With ‘Lena’ as the original test image, Fig. 11 shows the watermarked images under various attacks and the corresponding extracted correct bits.

![Fig. 11. Various attacks and the corresponding extracted correct bits.](image)

The table below compares the proposed scheme with other schemes.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG</td>
<td>10 - 100</td>
<td>40 - 100</td>
<td>20 - 100</td>
<td>20 - 100</td>
</tr>
<tr>
<td>Median Filter</td>
<td>fail</td>
<td>2 × 2</td>
<td>4 × 4</td>
<td>6 × 6</td>
</tr>
<tr>
<td>Cropping</td>
<td>Up to 65%</td>
<td>Up to 10%</td>
<td>Up to 15%</td>
<td>Up to 20%</td>
</tr>
<tr>
<td>Scaling</td>
<td>pass</td>
<td>–</td>
<td>0.75 – 1.5</td>
<td>0.5 – 4 and even more</td>
</tr>
<tr>
<td>Rotation</td>
<td>1° – 5°</td>
<td>1° – 5°</td>
<td>1° – 5°</td>
<td>1° – 5°</td>
</tr>
<tr>
<td>Embedding Image</td>
<td>Gray</td>
<td>Gray</td>
<td>Color</td>
<td>Gray</td>
</tr>
<tr>
<td>Watermark Length</td>
<td>60 bits</td>
<td>16 bits</td>
<td>128 bits</td>
<td>16 bits</td>
</tr>
</tbody>
</table>

In Table 7, the dash ‘–’ indicates the simulation was not recorded in the literature.

As TABLE 7 shows, the proposed scheme is highly robust against both JPEG compression and geometric distortion: JPEG compression with a low quality factor of up to 20%, image cropping of up to 20%, image flipping, jitter attacks of random removal of rows and columns, and image median filtering. Moreover, the scheme seems very robust to image scaling, with the scaling parameter varying from 0.5 up to 4, and even higher.

VI. CONCLUDING REMARKS

This paper proposes a robust and geometric invariant digital image watermarking scheme based on DWT and image histogram. The scheme embeds the watermark in the approximation sub-band of the wavelet-decomposed image, on the base of the properties of histograms. The proposed scheme is highly robust against both JPEG compression and geometric distortion: image cropping of up to 20%, JPEG compression with a low quality factor of up to 20%, image flipping, jitter attacks of random removal of rows and columns, and image median filtering. Moreover, the scheme seems very robust to image scaling, with the scaling parameter larger than 0.5. In future work, the robustness against image rotation will be investigated and also the process of watermark embedding and extraction will be improved and upgraded.

**REFERENCES**


Chi-Man Pun received the B.Sc. and M.Sc. degrees from the University of Macau in 1995 and 1998 respectively, and Ph.D. degree in Computer Science and Engineering from the Chinese University of Hong Kong in 2002. He currently is an associate professor at the Department of Computer and Information Science of the University of Macau. His research interests include Content-Based Image Indexing and Retrieval, Digital Watermarking, Pattern Recognition, and Computer Vision. He is also an IEEE senior member.