NEW DIGITAL WATERMARKING FOR FEW-COLOR IMAGES

Shu-Kei Yip, Oscar C. Au, Chi-Wang Ho, Hoi-Ming Wong, Richard Y.M. Li

Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
Email: {sukiyip, eeaau, jodyho, hoimingw, ymli}@ust.hk

ABSTRACT

In this paper, we propose a new digital watermarking algorithm named Spatial Unified Key Insertion (SUKI) for few-color images. By performing the adaptive threshold halftoning and contour shaping and modification, we can embed a binary logo into a digital image with superior JPEG compression resistance ability. The Bit Error Rate (BER) of JPEG attack is equal to zero for all testing images with Quality Factor \( \geq 80 \). The BER is much lower than the traditional watermarking algorithms, such as Spread Spectrum Watermarking and Quantization Index Modulation Watermarking. The payload is considerably high and there are no “salt-and-peppers” artifacts in the watermarked images. No additional color is introduced and the palette keeps unchanged after watermark embedding. The proposed algorithm can be applied to natural images with distinct boundaries by segmenting the images into different parts.

Index Terms — Copyright protection, Robustness

1. INTRODUCTION

Digital watermarking is one of solutions to prove the ownership and the authenticity of the media. In many existing algorithms, watermark, itself, actually is a random sequence. Recently, there are some outliers that a real image, such as a logo, should be embedded and watermarked into the media in order to declare the ownership. The major advantage is that the logo is still readable by human eyes although the watermarked image is corrupted.

In order to make the watermark detectable after some common attacks, the watermark should be perceptually transparency and robustness [1] [2]. Perceptually transparency means the difference between watermarked media with the original one should not be observable. Robustness means the watermark should be difficult to remove if some common signal processing operations applied, such as JPEG compression.

Most of the existing watermarking algorithms mainly focus on natural images. Data hiding or watermarking in few-color images, such as cartoon images, is more challenging than that in natural images. The main difference between few-color images and natural images is that few-color images do not consist of complicated texture or color variation. Although there are some watermark algorithms for halftone images [3] [4], the algorithms usually do not perform for the images with more than 2 colors. In 2002, G.Pan et al. [5] proposed a data hiding method for few-color images. Information is embedded by using “subblock and supblock” and pattern matching. Although the payload of this method is quite high (around 0.0034 Bit Per Pixel), “salt-and-peppers” artifacts may appear in watermarked images due to the improper selection of the pattern sets. Moreover, this method cannot withstand attacks, such as JPEG compression.

In this paper, we propose a new watermarking algorithm for few-color images named Spatial Unified Key Insertion (SUKI). By utilizing the relationship between the halftone image, which is formed by adaptive threshold halftoning, and the original image, boundary blocks detection and contour shaping can be carried out with higher degree of accuracy. No “salt-and-peppers” artifacts are introduced during watermarking. The proposed algorithm has considerably high payload and excellent JPEG attack resistance ability. The Bit Error Rate of the extracted logos of all the testing images with JPEG attack can be as low as zero for Quality Factor \( \geq 80 \). No additional color is introduced and the palette keeps unchanged after watermark embedding.

This paper is organized as follows. Section 2 describes the proposed watermarking algorithm named Spatial Unified Key Insertion (SUKI). Section 3 shows the experimental results, such as Peak Signal-to-Noise Ratio (PSNR) between the watermarked images and the original images, the payload and the Bit Error Rate of the extracted logos. In section 4, there is a conclusion.

2. THE PROPOSED ALGORITHM

Few-color images mean the images usually with 2 to 20 colors. These images generally have three properties (P1-3).

P1) These images consist of many big homochromatic regions.

P2) There are distinct boundaries between homochromatic regions.

P3) All the edges in the images are with the same color, which is usually black in color.

By using the (P1-3) above, the watermark embedding is performed in spatial domain. For an image, \( S \), with the size of \( S_X \times S_Y \), segmentation may perform for an image with multiple objects. Each segment can be treated as an independent image. Segmentation is to ensure each segment satisfied (P1-3). A binary logo, \( L \), is embedded into \( S \) to form a watermarked image, \( W \). Section 2.1 presents the details of adaptive threshold halftoning. By using halftoning method, the image approximates to a binary image first. The halftoning process is the main procedure to resist the JPEG attack. Section 2.2 describes the requirements of boundary blocks and candidate blocks selection. As the halftone image only consists of binary color, boundary blocks detection can be carried out with higher degree of accuracy. Some boundary blocks are selected as possible candidate blocks for embedding under certain requirements. The selection process is to ensure the perceptual transparency. Lastly, the candidate blocks perform the contour shaping and watermarking based on \( L \). One information bit is embedded in each candidate block. The contour shaping and
modification are described in section 2.3. Figure 1 shows the flow diagram of the watermark embedding.

2.1. Adaptive Threshold Halftoning

Before performing the halftoning process, we need to define the threshold for halftoning, \( T_H \), by analyzing the histogram of the image and the edge color. The edge color can be easily detected by simple edge detection technique. In this paper, assuming the edges are black in color (it can be easily extended to the edges with multiply colors by dividing the image into different segments). For demonstration purpose, a histogram is shown in figure 2.

From figure 2, there are altogether four peaks for content colors (Content colors mean the colors apart from the edge color). We first detect the peak for each content color. In this paper, a simple method is proposed. There is a peak when the occurrence is greater than 0.1 percent of the total samples. The threshold for detecting a peak, \( \lambda \), is:

\[
\lambda = 0.001 \times S_x \times S_y
\]  

(1)

Let us denote the pixel value of edge color as \( P_E \) and the minimum pixel value of the content colors as \( P_M \). After locating the peaks, \( T_H \) is one of the values in between the region of the \( P_E \) and \( P_M \), which means \( T_H \in \{P_E,...,P_M\} \). From figure 2, the green region is the possible values of \( T_H \). \( T_H \) is selected based on two conditions (C1-2):

C1) Within the region, if there are some pixel values with zero occurrence, those pixel values belong to the “Zero Occurrence Set”, \( ZSet \). \( T_H \) is set to be the median of \( ZSet \).

C2) If \( ZSet \) is empty, \( T_H \) is set to be the weighted average of \( P_E \) and \( P_M \). In this paper, the weighting of \( P_E \) and \( P_M \) are set to be equal, as a result, \( T_H \) is set to be the average of \( P_E \) and \( P_M \).

\( T_H \) is not a predefined threshold, it is adaptively chosen based on the image histogram. For no attack, all the pixel values in between \( P_E \) and \( P_M \) usually belong to \( ZSet \). However, when there is JPEG attack, the pixel values belong to \( ZSet \) reduce. The main reason is due to the “Color Diffusion” in JPEG compression. The histogram of JPEG attack of Lion with Quality Factor (QF) = 60 is shown in figure 3.

When comparing figure 3 with figure 2, the sharp peaks become triangle in shape. The new color is mainly due to the quantization and the finite precision in DCT transform. This effect is called “Color Diffusion”. For the image with larger difference between \( P_E \) and \( P_M \), it has greater capacity for handling “Color Diffusion”. As a result, there is more chance to have a non-empty \( ZSet \). We can reduce or eliminate the “Color Diffusion” effect by selecting a suitable \( T_H \), especially when \( ZSet \) is a non-empty set. After the selection of \( T_H \), the adaptive threshold halftoning is performed by using the following equation, and the halftone image of \( S \), \( S_H \), is formed.

\[
S_H(x,y) = \begin{cases} 1 & \text{if } S(x,y) \geq T_H \\ 0 & \text{if } S(x,y) < T_H \end{cases}
\]  

(2)

2.2. Boundary Blocks Detection and Possible Candidates Identification

\( S_H \) is the halftone image, and \( S_H(x,y) \in \{0,1\} \). We can find out the blocks with edge or the boundary blocks easily. The \( S_H \) is first divided into \( N \times N \) blocks, \( N \) can be any odd numbers greater than 1. The blocks belong to the “Boundary Blocks Set”, \( BSet \), if total block intensity is not equal to zero or \( N^2 \).

After identification of \( BSet \), we need to look for the blocks which are suitable for embedding in \( BSet \). In order to be the possible candidates for embedding, the blocks in \( BSet \) should satisfy three requirements (R1-3) by investigating the pixels of \( S_H \) except the center pixel of the block. The total intensity of the block without the center pixel is denoted as \( Int_{B/C} \).
R1) $\text{ceiling}(N/2) \times N-1 \geq \text{Int}_B|C \geq N$
R2) The group of black pixels should be connected together.
R3) For $\text{Int}_B|C = N$, the shape of black pixels should not be a horizontal line or a vertical line.

(R1) is used to ensure the number of black pixels and the white pixels in a block are approximately the same, and (R2) is used to ensure there are only two clusters in a block. Without the constraints of (R1-2), “salt-and-peppers” artifacts may appear after embedding. Lastly, human visual system is more sensitive to the strong vertical or horizontal edge, so (R3) is used to preserve those blocks with strong vertical and horizontal edge in order to maintain high visual quality. Figure 4 shows some examples whose blocks belong to $BSet$, but not suitable for embedding with $N=3$. The “X” means the center pixel is not care in verifying (R1-3).

![Violation of (R2)](image1)
![Violation of (R1)](image2)
![Violation of (R3)](image3)
![Violation of (R2)](image4)

Figure 4. The $BSet$ Blocks which are Not Suitable for Embedding

For those blocks which satisfy (R1-3), they are the candidates for embedding and they belong to the “Embedding Blocks Set”, $ESet$. One information bit is embedded in each block of $ESet$.

### 2.3. Contour Shaping and Modification

The contour shaping is performed for the blocks in $ESet$. The blocks in $ESet$ are divided into three groups according to their contours. They are “Concave Blocks”, “Convex Blocks” and “Planar Blocks”. The two outermost black pixels with the largest distance are used to form the Principal Axis. For a $N \times N$ block, the contour is formed by connecting the outermost black pixels with at least $N$ black pixels. According to binary logo, $L$, the contours of the blocks in $ESet$ are modified. For example, the “Convex Blocks” and the “Planar Blocks” are used to represent an information bit of “1” and the “Concave Blocks” are used to represent an information bit of “0”. Figure 5 shows the contour pair of different blocks in $ESet$.

![Concave ↔ Convex](image5)
![Concave ↔ Planar](image6)
![Planar ↔ Convex](image7)
![Planar ↔ Convex](image8)

Figure 5. The Contour Pair of Blocks for $N=3$

From figure 5, the blue line represents the Principal Axis and the red line represents the shape of contour. We can see that contour modification can be interrupted as changing the center pixel value to the content color of the block or the edge color (As in $S_{ij}(x,y)$, black pixels mean edge color and white pixels mean content colors). By identifying the corresponding blocks in $S$, we can change the center pixel of the blocks to content color of the blocks or edge color according to $L$.

In watermark extraction, it is similar to watermark embedding. $T_H$ is selected according to the received image histogram and (C1-2). The blocks of $BSet$ and $ESet$ are identified and located using (R1-3). As (R1) is a subset of conditions for boundary blocks detection and the center pixels are not used for (R1-3), we can find out the $ESet$ which is the same as watermark embedding using proper $T_H$ or under no attack. The contour shaping is followed and the Principal Axis is found out. The hidden information is extracted by identifying whether the blocks are concave, convex or planar.

### 3. EXPERIMENTAL RESULTS

We have tested the proposed algorithm using five few-color images from Microsoft Clip Art. They are Dog, House, Lion, Robot and Sheep, and only the luminance component is used for watermarking. A binary logo with equal probability of “1” and “0” is embedded into five testing images. The results of Peak Signal-to-Noise Ratio (PSNR) between the watermarked images and the original images, the payload and the Bit Error Rate (BER) of the extracted logos with JPEG attack are shown in this section. Figure 6 shows the testing images and the binary logo.

![Testing Images](image9)
![Logo](image10)

Figure 6. (a) Dog; (b) House; (c) Lion; (d) Robot; (e) Sheep; (f) Logo

Using Lion as an example, the original image and the watermarked image with payload of 0.00495 Bit Per Pixel (which is 634 Bits) are shown in figure 7 and 8 respectively.

![Original Image and Watermarked Image](image11)

Figure 7. The Original Image (Left), Watermarked Image (right)

Figure 8. The Original Image (Left), Watermarked Image (right)

The performances of the testing images are shown in table 1. The payload is in Bit Per Pixel (bpp) and the PSNR is in dB.
Table 1. The Performance of Testing Images

<table>
<thead>
<tr>
<th>Image</th>
<th>Size</th>
<th>Payload (bpp)</th>
<th>PSNR (dB)</th>
<th>BER*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dog</td>
<td>320 x 420</td>
<td>0.00331</td>
<td>30.1488</td>
<td>0</td>
</tr>
<tr>
<td>House</td>
<td>560 x 400</td>
<td>0.00956</td>
<td>28.2542</td>
<td>0</td>
</tr>
<tr>
<td>Lion</td>
<td>320 x 400</td>
<td>0.00495</td>
<td>28.5179</td>
<td>0</td>
</tr>
<tr>
<td>Robot</td>
<td>400 x 400</td>
<td>0.00417</td>
<td>30.6055</td>
<td>0</td>
</tr>
<tr>
<td>Sheep</td>
<td>400 x 400</td>
<td>0.00265</td>
<td>30.2152</td>
<td>0</td>
</tr>
</tbody>
</table>

* Under No Attack

From table 1, the average payload is 0.004929 bpp, which is larger than that of [5]. Moreover, our proposed algorithm can withstand JPEG attack with a great extent. The Bit Error Rate (BER) of testing images against JPEG attack is shown in figure 9.

Figure 9. The BER of the Extracted Logos against JPEG Attack

From figure 9, for Quality Factor (QF) ≥ 80, the BER for all testing images are equal to zero. For QF = 70, the BER for most of the testing images are equal to zero (For House and Robot, the BER is 0.0144 and 0.0464 respectively). For the testing image of Sheep, the BER is equal to zero for QF ≥ 40. The superior JPEG resistance ability is mainly due to the adaptive threshold halftoning. The larger the difference between P_E and P_M is, the more resistance the JPEG attack is. For the case of Sheep, the BER is so low because the region between P_E and P_M is the largest amongst the testing images.

The performance in JPEG attack of the proposed algorithm outperforms the traditional watermarking techniques for natural images, such as Spread Spectrum Watermarking (SS) [6] and Quantization Index Modulation Watermarking (QIM) [7]. For SS, the BER is not zero even there is no attack. It is because the image and the codewords are not independent. For QIM, it cannot withstand simple scaling attack as the centers of the quantization cells will be changed. Figure 10 shows the watermarked image using SS and QIM. Figure 11 shows the comparison of average BER for the proposed algorithm, SS and QIM with same payload. For SS, the block size, the codeword size and the step size is set to 5, 16 and 12 respectively, codewords are with Walsh Code structure.

\[ t(x) = t'(x) + \alpha \times c(x) \]  \hspace{1cm} (3)

\[ t'(x) \] is the watermarked DCT coefficient, \( t(x) \) is original DCT coefficients and \( c(x) \) is the codeword. For QIM, the quantization step size is set to 5.

Figure 10. Watermarked Image Using SS (Left) and QIM (Right)

Figure 11. The Comparison of the Average BER of the Extracted Logos Using the Proposed Algorithm with SS and QIM

From figure 10 and 11, although the visual quality of QIM is similar to the proposed algorithm, it is not robust to JPEG attack (red line). For SS, the visual quality is bad as some middle or high frequency components are introduced during watermarking, and the BER of SS (green line) is not as low as the proposed algorithm (blue line).

4. CONCLUSION

In this paper, a new watermarking algorithm named Spatial Unified Key Insertion is proposed. Using adaptive threshold halftoning, boundary blocks and candidate blocks selection, and contour shaping and modification, we can achieve an excellent performance against JPEG attack for few-color images. The BER of the extracted logos can be as low as zero for JPEG Quality Factor ≥ 80. The BER is much lower than the traditional watermarking algorithms, such as SS and QIM. The payload is considerably high and there are no “salt-and-peppers” artifacts in the watermarked images. No additional color is introduced and the palette keeps unchanged after watermark embedding.

5. ACKNOWLEDGEMENT

This work has been supported in part by the Innovation and Technology Commission (projects no. ITS/122/03 and GHP/033/05) and the Research Grant Council (DAG04/05.EG34) of the Hong Kong Special Administrative Region, China.

6. REFERENCES