applied sciences

MDPI

Article

A Large Capacity Histogram-Based Watermarking Algorithm for Three Consecutive Bins

Zhen Yue [1,*], Zichen Li [2,*], Hua Ren [3] and Yixian Yang [1]

[1] Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China; yxyang@bupt.edu.cn
[2] School of Information Engineering, Beijing Institute of Graphic Communication, Beijing 102600, China
[3] College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China; renhuahtu@163.com
* Correspondence: yue_zhen@foxmail.com (Z.Y.); lizc2020@163.com (Z.L.); Tel.: +86-188-1071-1660 (Z.Y.); +86-135-8170-5330 (Z.L.)

Received: 20 November 2018; Accepted: 11 December 2018; Published: 14 December 2018

Abstract: The histogram watermark, which performs watermark embedding by slightly modifying the histogram of the original image, has been a hot research topic in information hiding technology due to the superiority of its pixel modification during the watermark embedding process, which is independent of the pixel position. This property makes the histogram-based watermark strong resistant to geometric attacks, such as cropping attack, crossed attack, rotation attack, etc. In this paper, we propose a large capacity histogram-based robust watermarking algorithm based on three consecutive bins for the first time. In our scheme, we divide the shape of three consecutive bins into eight cases. According to these cases, we embed Information Number 0, 1, 2, 3, 4, 5, 6, or 7, respectively. The embedded information capacity reaches one bit per bin (bpb), and the amount of embedded information is equal to 200% of the previous existing algorithms. Experimental results show that the new algorithm not only has a large capacity of embedding information, but also has strong robustness to geometric attacks, as well as common image processing operations.

Keywords: robust watermarking; histogram; three consecutive bins; large capacity

1. Introduction

With the rapid growth of popular and low-cost access to image editing applications, some intentional or unintentional manipulations of digital media during transmission are becoming available [1,2]. This will bring a troublesome issue for copyright protection and authentication since these illegally-manipulated copies of digital media may be easily redistributed and transmitted. Robust watermarking [3] is the technique that embeds information in the image to provide authentication, copyright protection, copy control, etc. The objective of earlier research works of robust watermarking was primarily to realize the capability to resist common image processing operations [4], while recent research works have been prone to emphasizing the functions of resisting geometric attacks [5–12]. Geometric attacks primarily introduce some synchronization errors between the encoder and the decoder. Under geometric attacks, the watermark is still present, whereas the detector is no longer able to extract it. Therefore, how to design a watermarking algorithm robust to geometric attacks has become indispensable and vital in the digital watermarking field.

A variety of watermarking-related methods has been proposed so far, and they can be classified into three categories: inverse transformation based [5–7], geometric invariance domain based [8–10], and feature area embedding based [11,12]. Inverse transformation-based methods mainly exploit exhaustive search [5] and the embedding template, in addition to the watermark [6] to resist geometric attacks. The implementations of geometric invariance domain-based methods need to use

the mathematical transform on the input image before embedding and extracting procedures, like the image normalization method [9] and Zernike moments [10]. Feature area embedding-based methods extract feature points [11] or salient points such as the eyes and the mouth [12] firstly and then embed the watermark in them. Among various categories, geometric invariance domain-based methods are usually vulnerable to cropping attacks; in contrast, feature area-based methods can provide better robustness.

However, all the above schemes, both the earlier and recent schemes robust to geometric attacks, are not fully satisfied. On the one hand, these schemes suffer from computational efficiency problems in the watermark embedding and extracting procedure. On the other hand, these schemes cannot efficiently implement watermark robustness to some geometric attacks, such as random bending attacks (RBAs) and random cropping. It is well known that in an image that has undergone geometric attacks, there may occur three different changes: pixel position, pixel value, and pixel number. In this case, the positions or values of some pixels may be modified; even the number of pixels may be decreased or increased linearly (scaling the size of the image). Based on the analysis, Xiang et al. [13] developed an invariant watermarking solution to geometric attacks by using the histogram and the mean. In their method, the histogram shape was used to embed the watermark based on the property that the histogram shape is only related to the pixel count of each grayscale, not the position of pixels.

Nevertheless, there exist some issues in Xiang et al.'s method [13]: (1) randomly selecting pixels to modify may bring a relatively weak robustness to JPEG compression; (2) the visual quality of some smooth areas in an image will seriously reduce with the increase of the bins' size; (3) bad bins may exist (the occurrence frequencies of pixels in these bins are zero or few). To solve the first problem, Hu et al. [14] used a novel pixel modification method, i.e., mean squared error (MSE), to choose some blocks to modify preferably. The second issue can be resolved by the methods of [14–16]. All of them are based on this strategy, which partitions every three consecutive bins into a group for embedding a one-bit watermark, which guarantees that if a larger bin size is chosen, the image quality of the smooth area will not decrease obviously. Meanwhile, the authors of [15] have proposed to use a predefined threshold to select a large population of pixels in the hope that this will suffice to eliminate possible bad bins. Besides, Zong et al. [17] proposed a histogram-shape-related index method to form and select the most suitable pixel groups for embedding. With their method, some possible bad bins can also be eliminated to some degree. Recently, other possible drawbacks, such as the lower embedding capacity existing in Xiang et al.'s method [13], were also resolved by Feng et al. [18]. Although these improved schemes [14–18] are effective to some extent, they pay no attention to a drawback existing in the original embedding procedure. That is, the embedding procedure of [13] failed to take the approximately equal number of adjacent histogram bins into account, and thus resulted in a direct decline of embedding capacity. Obviously, this restricts many applications, especially limited to those with a strict requirement for watermark capacity.

In this paper, we propose a large capacity histogram-based watermarking algorithm for three consecutive bins. To the best of our knowledge, this is the first scheme to expand from two bins to three bins. The improvement that we propose for watermark embedding differs from those proposed by recent state-of-the-art publications. This is because the proposed watermark embedding algorithm can remedy the drawback in the original design well, but other improved works remain almost invariant in the watermark embedding algorithm. A predefined threshold, together with a block-based pixel modification and new watermark embedding procedures, also makes the proposed scheme a viable alternative to the histogram-based methods. Experiments demonstrate that the maximum embedding capacity of using the proposed histogram-based watermarking algorithm outperforms all the existing histogram shape-based publications and can truly achieve one bit per bin (bpb). Meanwhile, the image quality of the stego-image, which is generated by modifying a part of the pixel values of the original image to carry the watermark information, together with the capability to resist some geometric attacks, can also be ensured.

The rest of this paper is organized as follows. Section 2 describes the invariance of the histogram shape and existing histogram-based watermark embedding. Detailed embedding and extraction processes of the proposed scheme are described in Section 3. Experimental results and related analyses are given in Section 4, followed by a conclusion in Section 5.

2. Preliminaries

2.1. Invariance of the Histogram Shape

In earlier works [13–18], a great number of researchers utilized the invariance of the histogram shape to design robust image watermarking. The motivation behind using this histogram-based property can be categorized into two points: (1) modifying a portion of the pixels to embed the watermark is only related to the modification count of each grayscale, not to the position of pixels; (2) the histogram of the host image that has suffered some attacks is immune to those changed pixels in this image, including pixel position, value, and count. Therefore, the robustness of these schemes actually originates from the histogram invariance. It is well known that a histogram is a display of statistical information, the grayscale occurrence frequency of which can be calculated by splitting the pixel values into equally-sized bins. Given an image $I = \{I(x,y)|x = 1,\cdots,R, y = 1,\cdots,C\}$ (here, R and C are the number of rows and columns in the image, respectively), the histogram can be described by:

$$H = \{h(i)|i = 1,\cdots,L\}, \tag{1}$$

where H and L denote the grey-level histogram of the image and the total number of equally-sized bins, respectively, and $h(i)$ means the number of pixels in the i-th bin, and it satisfies $\sum_{i=1}^{L} h(i) = R \times C$.

2.2. Existing Histogram-Based Watermark Embedding

In Xiang et al.'s method [13], the histogram-based watermarking method primarily utilized the histogram H and mean $\overline{A}$ to determine the embedding range $B = [(1-\lambda)\overline{A}, (1+\lambda)\overline{A}](0 < \lambda < 1)$ firstly; then they divided these bins into a series of groups where each of them contained two neighboring bins; lastly, they embedded watermark bits into these groups. For clarity and the ease explanation, the concrete rule of watermark embedding can be formulated as follows:

$$\begin{cases} \frac{a}{b} \geq T, & W(i) = 1 \\ \frac{b}{a} \geq T, & W(i) = 0 \end{cases}, \tag{2}$$

where a and b refer to the number of the two consecutive bins and T is a threshold controlling the number of modified pixels, the value of which has a direct influence on both robustness and image quality. If the numbers a and b in the adjacent bins satisfy the conditions in Equation (2), no operation is needed. Otherwise, some modifications will be made to let some pixels in one bin jump into the other. Here, a possible case is presented in Figure 1. When the watermark to be embedded is $W(i) = 1$, no change is made to the two bins since the numbers a and b in the adjacent bins have already met Equation (2). When the watermark to be embedded is $W(i) = 0$, the number of pixels in the two adjacent bins will be adjusted until satisfying the condition $\frac{b}{a} \geq T$, as shown in Figure 1.

However, the watermark embedding design of Xiang et al. [13] fails to take the approximately equal number of adjacent bins into consideration, i.e., $W(i) = 2$, as shown in the marked part with red dashed line. This will have a direct influence on the watermark embedding capacity; in turn, the watermark capacity, lowered greatly, restricts many application scenarios. To solve this issue, we design a novel histogram shape-based watermark scheme in the next section.

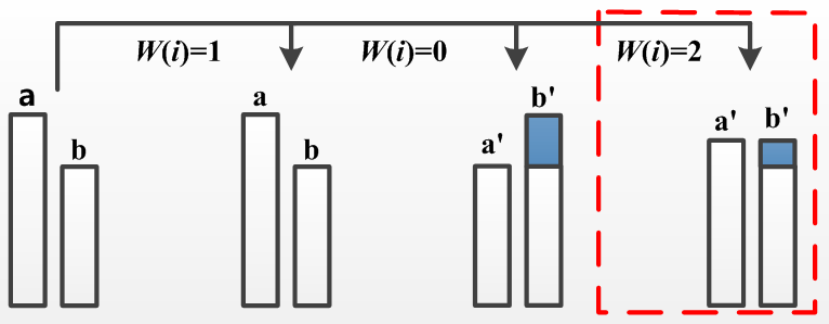


Figure 1. The issue existing in the original watermark embedding design.

3. The Proposed Method

In this section, a novel high-capacity octal histogram-based watermarking algorithm for three consecutive bins is proposed. The proposed watermarking algorithm comprises three parts: the selection of the embedding range, the position selection of the pixel to be modified, and the embedding and extraction processes of the watermark. Next, we will present them in detail.

3.1. Embedding Range Selection

In order to remove possible bad bins, a better histogram embedding range selection method presented by Deng et al. [15] is employed. In their method, the embedding range selection is determined by the following thresholds T_1 and T_2, as shown in Equations (3) and (4).

$$T_1 = \frac{M_0 \times N_0}{n \times n} \times \alpha_1 \tag{3}$$

$$T_2 = \frac{M_0 \times N_0}{n \times n} \times \alpha_2 \tag{4}$$

where M_0 and N_0 are the height and width of an image, respectively, n represents the size of the block and α_1 and α_2 are a constant controlled by a secret key to adjust the embedding watermark, the values of which are in the range of $[0, 1]$.

For clarity, Figure 2 illustrates the comparison of two different embedding range selection methods. Figure 2a shows the embedding range selection method presented by Xiang et al., and Figure 2b presents a better embedding range selection method initially proposed by Deng et al. [15]. Notably, some possible bad bins (the occurrence frequencies of pixels in these bins are zero or few) are removed successfully in Figure 2b. Based on this point, the robustness of the embedded watermark can be enhanced dramatically.

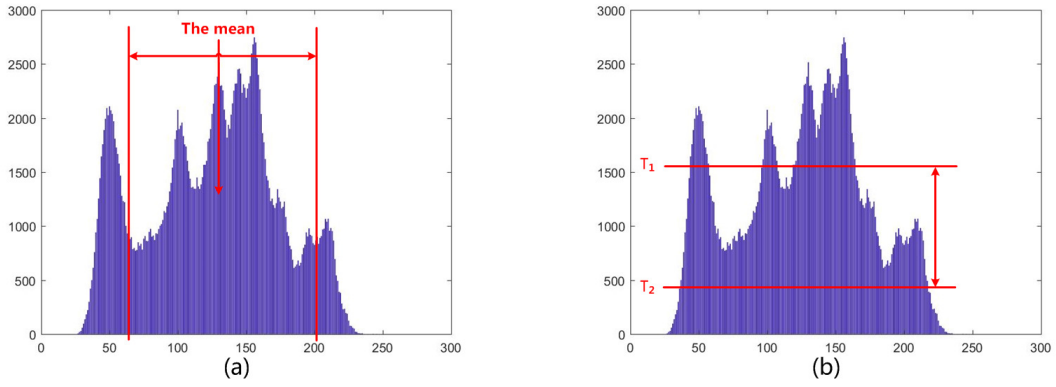


Figure 2. Comparison of two embedding range selection methods: (**a**) Xiang et al.'s method and (**b**) Deng et al.'s method.

3.2. Block-Based Pixel Modification

Xiang et al. performed the modification of the pixel value by randomly selecting the pixel position, the defect of which easily causes very limited pixel modifications in certain smooth areas. This also makes the comparison between the modified block and the surrounding unmodified blocks quite obvious; in other words, the visual quality with their method will be decreased (block effect). In order to avoid this, Hu et al. [14] initially proposed the block-based method after learning the human visual system (HVS) [19] and applying Xiang et al.'s idea. In this method, the average variance of each sub-block is calculated to reflect its smoothness. When one current pixel is modified, it is sorted according to the mean squared error (MSE) of the block, and then, the non-smooth region will be preferentially modified according to the feature of the current block. Given an image I with the size of $M_0 \times N_0$, if the block size in the image is set as $n \times n$, there will exist $\frac{M_0}{n} \times \frac{N_0}{n}$ blocks. The mean and mean squared error of these blocks are calculated, respectively, as shown in Equations (5) and (6).

$$mean(p,q) = \frac{1}{n \times n} \sum_{i=1}^{n} \sum_{j=1}^{n} I(i,j) \tag{5}$$

$$MSE(p,q) = \sqrt{\frac{1}{n \times n} \sum_{i=1}^{n} \sum_{j=1}^{n} (I(i,j) - mean(p,q))^2} \tag{6}$$

where $mean(p,q)$ and $MSE(p,q)$ are the two functions calculating the average value and mean squared error located at the block (p,q), respectively, and both p and q satisfy the conditions $p = 1, 2, \cdots, \frac{M_0}{n}, q = 1, 2, \cdots, \frac{N_0}{n}$.

3.3. Embedding and Extraction of the Watermark

To resolve the watermark design drawbacks mentioned above, this paper takes the approximately equal number of adjacent bins into consideration without compromising the modification degree of each pixel. This section will present the watermark embedding rules, embedding steps, as well as extraction steps.

3.3.1. Watermark Embedding Rules

Given the octal watermark information $W \in (0,1,2,3,4,5,6,7)$ with the length of L_W, it can be converted into the binary watermark information $W_B \in (0,1)$ with the length of $3 \times L_W$. The obtained binary watermark information was further transformed into a key-based PNsequence to enhance the security of the watermark. With respect to the private key utilized during watermark encryption, it can be shared with specific authorized receivers if one needs to detect the presence of the watermark [13]. For an easy explanation, Table 1 presents the various embedding rules under different watermark information.

Table 1. Pixel modification rules for various watermark information (where $\Delta_1, \Delta_2, \Delta_3 \in [0,1]$).

Case	Original Watermark	Its Binary Type	Pixel Modification Rules
1	0	000	$a' = \lfloor (a+b+c)/3 \rfloor + \Delta_1$, $b' = \lfloor (a+b+c)/3 \rfloor + \Delta_2$, $c' = \lfloor (a+b+c)/3 \rfloor + \Delta_3$
2	1	001	$a' = \lfloor (a+b+c)/4 \rfloor + \Delta_1$, $b' = \lfloor (a+b+c)/4 \rfloor + \Delta_2$, $c' = \lfloor (a+b+c)/2 \rfloor + \Delta_3$
3	2	010	$a' = \lfloor (a+b+c)/4 \rfloor + \Delta_1$, $b' = \lfloor (a+b+c)/2 \rfloor + \Delta_2$, $c' = \lfloor (a+b+c)/4 \rfloor + \Delta_3$
4	3	011	$a' = \lfloor (a+b+c)/5 \rfloor + \Delta_1$, $b' = \lfloor 2 \times (a+b+c)/5 \rfloor + \Delta_2$, $c' = \lfloor 2 \times (a+b+c)/5 \rfloor + \Delta_3$
5	4	100	$a' = \lfloor (a+b+c)/2 \rfloor + \Delta_1$, $b' = \lfloor (a+b+c)/4 \rfloor + \Delta_2$, $c' = \lfloor (a+b+c)/4 \rfloor + \Delta_3$
6	5	101	$a' = \lfloor 2 \times (a+b+c)/5 \rfloor + \Delta_1$, $b' = \lfloor (a+b+c)/5 \rfloor + \Delta_2$, $c' = \lfloor 2 \times (a+b+c)/5 \rfloor + \Delta_3$
7	6	110	$a' = \lfloor 2 \times (a+b+c)/5 \rfloor + \Delta_1$, $b' = \lfloor 2 \times (a+b+c)/5 \rfloor + \Delta_2$, $c' = \lfloor (a+b+c)/5 \rfloor + \Delta_3$
8	7	111	$a' = \lfloor (a+b+c)/6 \rfloor + \Delta_1$, $b' = \lfloor (a+b+c)/3 \rfloor + \Delta_2$, $c' = \lfloor (a+b+c)/2 \rfloor + \Delta_3$

First, we transform the original watermark into its binary type. For various watermark data, pixel modification rules are slightly different. Note that having eight types of situations is mainly attributed to two factors: one is that the approximately equal pixel number of adjacent bins is considered here. The other is that every three consecutive bins is partitioned into a group to perform watermark embedding. Unlike other histogram-based watermark methods [14–16], these methods do not consider this case in their watermark designs; thus, only one-bit watermark data can be embedded into each of the three consecutive bins. In the proposed methods, every 3 bits of watermark information can be simultaneously embedded into each of the three consecutive bins. This is also why pixel modification rules for various watermark data are required.

For clarity, we give a simple example to explain the relationship between binary watermark information and its corresponding formula. Suppose a, b, and c denote the original three consecutive bins before watermark embedding and a', b', and c' denote the corresponding modified bins after watermark embedding, respectively. Let us define the proportional relationship of binary watermark information "0" and "1" as 1:2 to distinguish various types of watermark. For instance, given the watermark "000", one should modify the original bins a, b, and c using the formula in the second line of Table 1, so that the pixel number of the modified bins a', b', and c' closes to 1:1:1. In a similar modification/adjustment manner, the proportional relationship of the modified bins a', b', and c' after watermark embedding will close to 1:1:2 if the given watermark is "001". Others are all similar to the above, except for a special case, i.e., the watermark "111". It can be distinguished by predefining the proportional relationship of the modified bins a', b', and c' as 1:2:3 roughly. With the assistance of this predefinition, one can accomplish the corresponding modification/adjustment before and after watermark embedding by making use of the formula, as shown in the last line of Table 1. Based on the above analysis, the extraction accuracy of the embedded watermark can be ensured.

After that, we can embed the 3-bit watermark into each group containing three consecutive bins. To analyze the embedding rules explicitly, an illustration of embedding the 3-bit watermark in eight cases is also shown in Figure 3. Assume that there remains an original group containing three consecutive bins, as shown in Figure 3a; let us analyze the case $W(i) = 2$. Obviously, the original group in Figure 3a does not satisfy the condition in Case 3, shown in the fourth line in Table 1; a part of the pixels in both bins a and c, shown in Figure 3a, will be modified to b, so that the modified result satisfies $a' = \lfloor (a+b+c)/4 \rfloor + \Delta_1$, $b' = \lfloor (a+b+c)/2 \rfloor + \Delta_2$, and $c' = \lfloor (a+b+c)/4 \rfloor + \Delta_3$; here, $\lfloor \bullet \rfloor$ represents a down integer function, $\Delta_1, \Delta_2, \Delta_3 \in [0,1]$ and $a+b+c = a'+b'+c'+\Delta_1+\Delta_2+\Delta_3$, as shown in Figure 3d. For the watermark embedding process of other cases, the modification process is similar to the above.

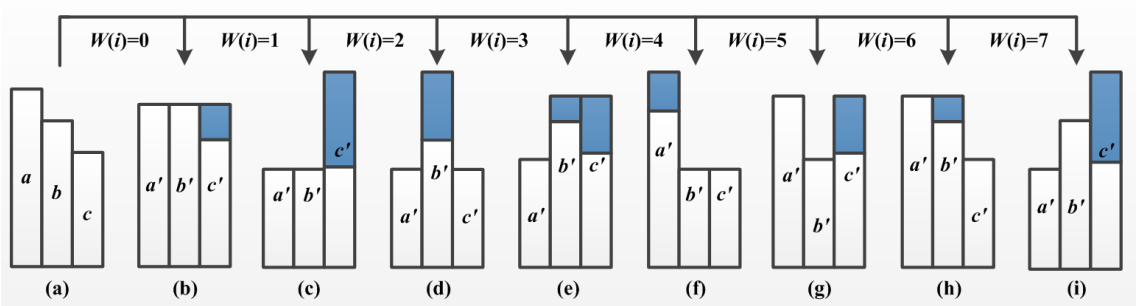


Figure 3. Illustration of embedding the 3-bit watermark in eight cases. (**a**) The original histogram; (**b**–**i**) the modified histograms for the cases $W(i) = 0, 1, 2, 3, 4, 5, 6, 7$, respectively.

It is worth noting that for the approximately equal adjacent bins, it may decrease the watermark extraction accuracy when the histogram is fully distorted, but a large number of embedded watermarks can still be extracted if the histogram is slightly changed. In the specific experimental operation, the default value (the case that does not satisfy certain proportional relationships is set as an approximately equal treatment) is used to weaken the fragile relationship for the adjacent bins, which are approximately equal.

3.3.2. Watermark Embedding Steps

Having described the rules of watermark embedding in detail in the previous part, we will present the watermark embedding steps in this subsection. The procedure of watermark embedding consists of the following steps.

Step 1: Divide the input image I_0 into non-overlapping blocks with the size of $n \times n$, and calculate the $MSE(p, q)$ of each block.

Step 2: Extract histogram H from the input image I_0 and generate the embedding range H_T using Equations (3) and (4).

Step 3: In view of the histogram bin's width M, determine the number of the bins from the selected embedding range firstly and then take each of three continuous bins to form a group. Suppose the height (pixel's numbers) in Bin 1, Bin 2, and Bin 3 is a, b, and c, respectively; the embedding rules are as follows:

Case 1: If it satisfies the condition $|a-b| < (a+b+c)/3 \quad \& \quad |a-c| < (a+b+c)/3 \quad \& \quad |b-c| < (a+b+c)/3$, no pixels will be adjusted; else, the other operations are as follows:

- If it meets $(a-b) > (a+b+c)/3$, the number of modified pixels is $transfer_{ab} = \lfloor (a-b)/2 \rfloor$ ($transfer_{ab}$ refers to the selected pixels, which will be moved from Bin 1 to Bin 2), these selected pixels of Bin 1 will be added to M;
- If it meets the condition $(a-c) > (a+b+c)/3$, firstly, choose all $transfer_{ab} = \lfloor (a-c)/2 \rfloor$ pixels which will be moved from Bin 1 to Bin 2 by adding to M, then all $transfer_{bc} = \lfloor (a-c)/2 \rfloor$ pixels of Bin 2 are chosen to be modified so as to make them fall in Bin 3. To ensure the modification/adjustment degree M of each pixel, the order in the above steps cannot be changed because these pixels chosen from Bin 2 should not contain those that have been adjusted in Bin 1. The corresponding modification/adjustment degree of some pixels may be $2 \times M$ if the order is exchanged.
- The others are similar to the above.

Case 2: If it meets the condition $b \geq T \times a \quad \& \quad b \geq T \times c \quad \& \quad |a-c| < (a+b+c)/4$, no operation is needed. Otherwise, if it meets the condition $a > (a+b+c)/4$, move all $transfer_{ab} = a - (a+b+c)/4$ pixels from Bin 1 to Bin 2 by adding to M; if $c > (a+b+c)/4$, move all $transfer_{cb} = c - (a+b+c)/4$ pixels from Bin 3 to Bin 2 by reducing M.

The other cases are similar to the two cases mentioned above.

Step 4: According to the above histogram modification/adjustment procedure, to modify the histogram shape of each group, when adjusting the pixel value, choose blocks with larger MSE.

Step 5: These above steps are repeated until the whole PN sequence is embedded. The final modified image is the watermarked image.

For the ease of understanding, we display the partial pseudo-code of the embedding algorithm. This is shown in Algorithm 1.

Algorithm 1. Watermark embedding.

Input:
  $I_0 = []_{M_0 \times N_0}, W_{L_W}, n, T, T_1, T_2$
Output:
  I_W
Process:
  $I_0 \rightarrow H, allblock_{MSE[max \rightarrow min]}$
  $H \rightarrow H_T(T_2, T_1)$
  $H_T \rightarrow [Bin1, Bin2, Bin3]_{3 \times L_W} \rightarrow [a, b, c]_{3 \times L_W}$
  for $i \rightarrow 1$ *to* L_W **do**
    $W_{L_W}(i) \rightarrow 010$
    if $b \geq T \times a$ & $b \geq T \times c$ & $|a - c| < (a + b + c)/4$
      Do nothing;
    else
      if $a > (a + b + c)/4$
        $Transfer_{ab} = \lfloor a - (a + b + c)/4 \rfloor$;
        $I_W = I_0(ii \rightarrow Transfer_{ab})_{allblock_{MSE}} + M$;
      end
      if $c > (a + b + c)/4$
        $Transfer_{cb} = \lfloor c - (a + b + c)/4 \rfloor$;
        $I_W = I_0(ii \rightarrow Transfer_{cb})_{allblock_{MSE}} - M$;
      end
    end
  end
  Other $W_{L_W}(i)$, do similarly to the above.
  $I_0 \rightarrow W_{L_W} \rightarrow I_W$

3.3.3. Watermark Extraction Steps

When the watermarked images are illegally manipulated during transmission, an effective way to protect the owner's copyright is to examine the suspicious images with proper watermark extraction algorithms. At the watermark extraction phase, parameters α_1, α_2, and T should be known in advance. The detailed steps are as follows.

Step 1: Calculate the histogram H' from the received watermarked image I'_W.

Step 2: Extract H'_T in the same way as was done in the embedding rule. In each of the detection ranges, each of the three consecutive equally-sized bins is divided as a group.

Step 3: Set the pixel numbers of Bin 1, Bin 2, and Bin 3 in the current group as a'', b'', and c'', respectively; according to their proportional relationship, extract the watermark information $W'(i)$.

Without loss of generality, if it meets the condition $b'' \geq T \times a''$ & $b'' \geq T \times c''$ & $|a'' - c''| < (a'' + b'' + c'')/4$, we can extract watermark information as $W'(i) = 2$; if it meets the relationship $b'' \geq T \times a''$ & $c'' \geq T \times a''$ & $|b'' - c''| < (a'' + b'' + c'')/5$, the watermark information is extracted as $W'(i) = 3$. In a similar way, the proposed method can extract all of the other watermark information.

Step 4: Repeat Step 3 until all of the watermark is extracted.

Similarly, we also give a partial version of the pseudo-code of the watermark extraction process, as shown in Algorithm 2.

Algorithm 2. Watermark extraction.

Input:
$I'_w = []_{M_0 \times N_0}, T, T_1, T_2$
Output:
W'
Process:
$I'_W \rightarrow H'$
$H' \rightarrow H'_T(T_2, T_1)$
$H'_T \rightarrow [Bin1, Bin2, Bin3]_{3 \times L_W} \rightarrow [a'', b'', c'']_{3 \times L_W}$
for $i \rightarrow 1$ *to* L_W **do**
  if $b'' \geq T \times a''$ & $b'' \geq T \times c''$ & $|a'' - c''| < (a'' + b'' + c'')/4$
    $W'_{L_W}(i) \rightarrow 2$;
  end
  if $b'' \geq T \times a''$ & $c'' \geq T \times a''$ & $|b'' - c''| < (a'' + b'' + c'')/5$
    $W'_{L_W}(i) \rightarrow 3$;
  end
end
Other $W'_{L_W}(i)$, do similarly to the above.
$I'_w \rightarrow W'$

4. Experimental Results and Analysis

In this section, we offer experimental results to confirm the effectiveness of the improved watermarking scheme with a standard test database [20]. The most typical representatives with the size of 512×512, including Baboon, Barbara, Lena, and Peppers, were tested with a series of experiments. The performances of the proposed method were also compared with the state-of-the-art methods [13,17,18].

4.1. Embedding Capacity and Perceptual Similarity

In Xiang et al.'s method [13], two consecutive bins were combined into a group, and the bin width was set as $M = 2$; thus each of four bins carried a one-bit watermark. The total capacity in their method was $256/(2 \times M)$. If M is larger ($M > 2$), the capacity will decrease obviously. However, in the proposed method, each of three consecutive bins was combined into a group to embed a 3-bit watermark. For an image of 8 bit in depth, the maximum embedding capacity of the watermarking algorithm was mathematically calculated as $256/M$; in other words, this method will implement a 256-bit watermark insertion if the bin width is set as $M = 1$. Therefore, the proposed method is superior in terms of embedding capacity.

4.1.1. Embedding Capacity versus Perceptual Similarity

In order to further test the visual quality of watermarked images, the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) were used for judging the distortion degree between the original and watermarked images. In general, higher values of PSNR and SSIM lead to a better visual quality of the watermarked image.

For different test images, including Baboon, Barbara, Lena, and Peppers, we have shown the perceptual similarity (under different bin widths, thresholds, and capacities) in Table 2. As shown in Table 2, both PSNR and SSIM in all images were extremely high. For instance, the PSNR values in all images were still more than 50 dB when the embedding capacity was 63 bits. In terms of SSIM, the minimum of SSIM in all images could reach 0.9955. Furthermore, there was only a very small change in all of the SSIM values. Note that given the conditions T and M, the PSNR values of a given image will decrease with the increase of payload length. This is because there are many more pixels

that need to be modified to embed more payload bits. If we only consider a concrete embedding capacity (termed as one column in Table 2), PSNR values will decrease when the parameter M increases. This can be explained by the analyses mentioned above (refer to the Introduction): in short, the bin size used has a direct influence on the visual quality of the resulting images. Based on the above analysis, the visual quality of the proposed scheme was superior.

Table 2. Perceptual similarity for different test images (including Baboon, Barbara, Lena, and Peppers) at different bin widths, thresholds, and capacities.

Payload (bit)		**32**		**48**		**56**		**63**	
Perceptual Similarity		**PSNR (dB)**	**SSIM**	**PSNR (dB)**	**SSIM**	**PSNR (dB)**	**SSIM**	**PSNR (dB)**	**SSIM**
Baboon	$T = 2, M = 1$	69.09	1.0000	66.65	1.0000	65.55	1.0000	64.96	1.0000
	$T = 2, M = 2$	59.08	0.9999	55.85	0.9997	54.88	0.9996	54.39	0.9996
	$T = 2, M = 3$	51.58	0.9992	49.85	0.9986	-	-	-	-
Barbara	$T = 2, M = 1$	66.44	0.9999	65.27	0.9999	64.69	0.9999	64.29	0.9999
	$T = 2, M = 2$	57.25	0.9993	55.53	0.9991	54.73	0.9990	54.39	0.9990
	$T = 2, M = 3$	51.72	0.9980	49.97	0.9973	49.44	0.9969	-	-
Lena	$T = 2, M = 1$	66.43	0.9999	65.26	0.9999	64.32	0.9999	63.73	0.9999
	$T = 2, M = 2$	56.84	0.9992	55.04	0.9988	53.89	0.9984	53.51	0.9982
	$T = 2, M = 3$	51.16	0.9971	49.49	0.9955	-	-	-	-
Peppers	$T = 2, M = 1$	67.52	0.9999	64.98	0.9999	63.49	0.9998	62.99	0.9998
	$T = 2, M = 2$	57.22	0.9993	56.05	0.9992	55.13	0.9990	54.59	0.9988
	$T = 2, M = 3$	52.62	0.9982	50.76	0.9970	-	-	-	-

Besides, Figure 4 also shows the original images, the watermarked images, and their embedding modifications for Barbara and Lena, in which the bin's width M and the threshold T were both set as two, and the embedding capacity C reached 63 bits. As shown in Figure 4c,f, the modification of the original image caused by the payload embedding was acceptable. Compared with Figure 4a,b together with Figure 4d,e, the visual quality of watermarked images was satisfactory. In summary, the image quality of watermarked images obtained by the proposed algorithm could fully meet the requirements of practical application scenarios.

4.1.2. Comparison with the State-of-the-Art Methods

In this section, we further evaluate the effectiveness of the proposed method by comparing it with those suggested in [13,17,18]. All of the parameter settings for different algorithms are shown in Table 3. The corresponding PSNRs are compared in Table 4. It can be observed that our method provided a preferable visual quality of the watermarked image; meanwhile, it ensured more information could be embedded into the pixels located at the same bins. Our algorithm, with respect to the existing ones, improved the embedding capacity by at least 100%.

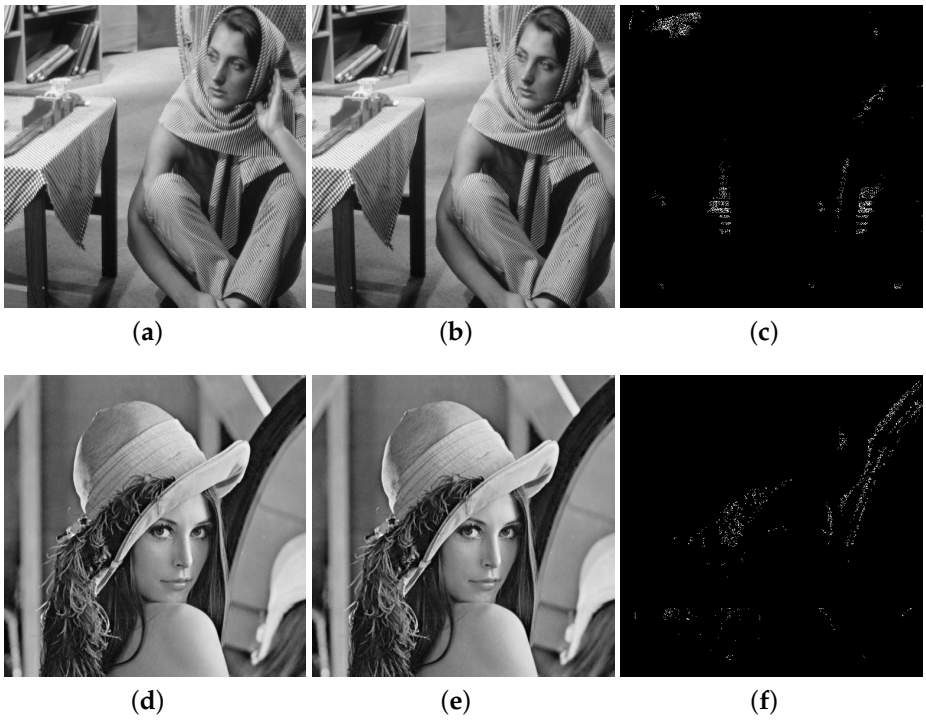


Figure 4. The original images, the watermarked images, and their difference for Barbara and Lena ($M = 2$, $T = 2$, and $C = 63$). (**a**) Barbara (original); (**b**) Barbara (watermarked); (**c**) Barbara (residual); (**d**) Lena (original); (**e**) Lena (watermarked); (**f**) Lena (residual).

Table 3. Parameter settings for different algorithms.

Payload (bit)		**32**	**48**	**56**	**63**
Xiang et al. [13]	embedding range	[0.3A,1.7A]	[0.3A,1.7A]	-	-
	T	2	6	-	-
	M	2	2	-	-
	bins' number	64	96	-	-
Zong et al. [17]	embedding range	[0,255]	[0,255]	-	-
	T	2	4	-	-
	M	3	2	-	-
	bins' number	64	96	-	-
Scheme 1 [18]	embedding range	[15,244]	[15,244]	[15,244]	[15,244]
	T	2	1.5	1.25	1.5
	M	2	2	2	2
	bins' number	64	96	112	126
Scheme 2 [18]	embedding range	[15,244]	[15,244]	[15,244]	[15,244]
	T	3	2+sqrt(0.5)	2.5	37/16+sqrt(0.5)
	M	2	4	8	4
	bins' number	64	64	64	84
The proposed	embedding range	**[0,255]**	**[0,255]**	**[0,255]**	**[0,255]**
	T	**2**	**2**	**2**	**2**
	M	**2**	**2**	**2**	**2**
	bins' number	**32**	**48**	**56**	**63**

Table 4. Comparison of PSNRs among different algorithms.

Payload (bit)		32	48	56	63
Averaged PSNR (dB)	Xiang et al. [13]	46.85	43.97	-	-
	Zong et al. [17]	46.12	43.46	-	-
	Scheme 1 [18]	46.92	44.07	38.99	42.37
	Scheme 2 [18]	46.92	43.69	43.40	41.67
	The proposed	**56.84**	**55.04**	**53.89**	**53.51**

4.2. Watermark Robustness

Some common attacks, such as cropping, crossed attack, rotation distortions, AWGN (additive white Gaussian noise) attack, and JPEG compression attack, were used to examine the robustness of the watermark with our method. We compared this with those suggested in [13–15]. The main image editing and attacking tool used in our experiments was MATLAB. For the four test images, we set the experimental parameters as the following: (1) the bin's width was set as two; (2) the threshold was set as five; (3) the capacity was set as 24 bits. Without loss of generality, we take Barbara as an example of the attacked image. Corresponding to the simulation results, we have the following observations:

1. Cropping: The cropping ranges from 5–50% with the interval of 5%. Figure 5 shows the watermarked image with the cropping of 5%, 25%, and 50%, respectively. Figure 6a,b presents the accuracy of watermark extraction when the watermarked image suffered from different levels of cropping attack.
2. Crossed attack: The number of lines increased from 1–8. This was set as 1, 4, and 8, respectively, as shown in Figure 7. The corresponding extraction results after crossed attack are given in Figure 6c,d.
3. Rotation distortions: The rotation angles ranged from 3°–30° with the interval of 3°. From Figure 8, one can understand the watermarked image with rotation angles of 3°, 18°, and 30°, respectively. Figure 6e,f presents the results after rotation attack.
4. AWGN attack: The standard deviation of AWGN was set from 1–10. Similarly, we can see from Figure 6g,h that the watermarked image could resist AWGN attack well.
5. JPEG compression: The quality factor of JPEG compression was changed from 80–100. Figure 6i,j provides a significant indication that the proposed watermark method was equipped with the capability of resisting JPEG compression. Here, it should be emphasized that the BER (bit error rate) of our method was a little higher than the other methods; this is because approximately equal adjacent bins were considered in our method, and thus, this may have increased the error rate of watermark extraction.

(**a**) cropping 5% (**b**) cropping 25% (**c**) cropping 50%

Figure 5. Examples of local cropping with different percentages for Barbara.

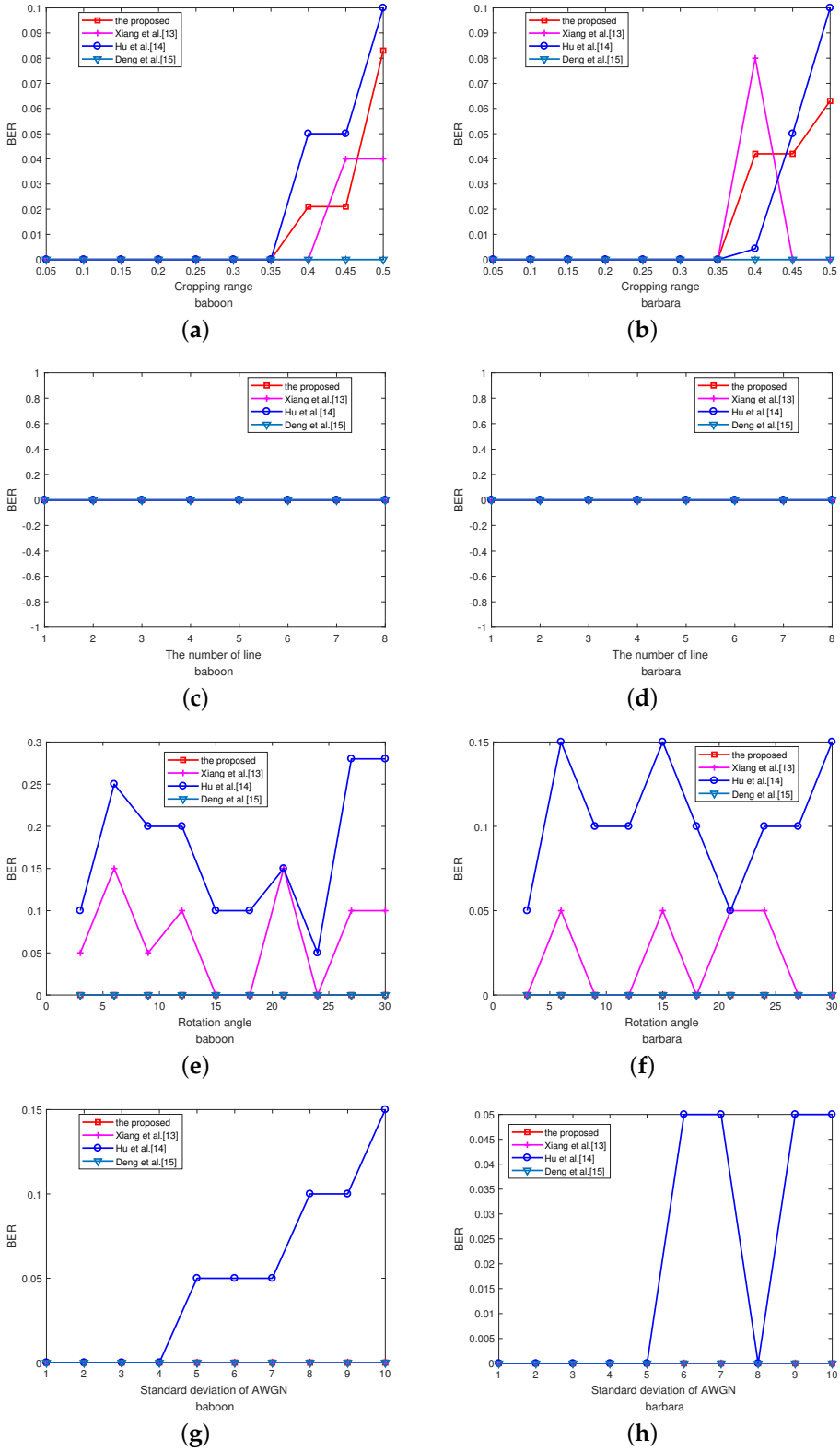
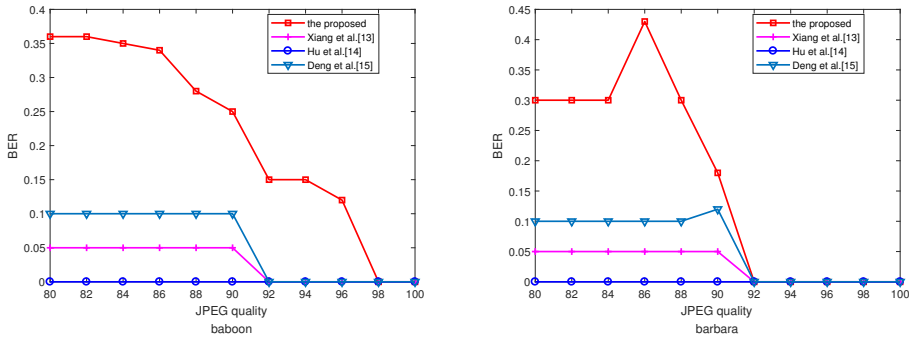
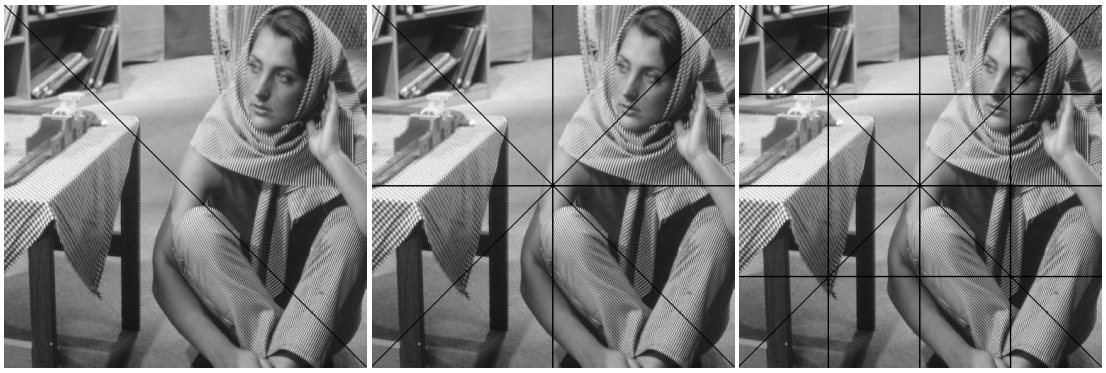


Figure 6. *Cont.*



(**i**) (**j**)

Figure 6. Robustness to cropping, crossed, and rotation attack for Baboon and Barbara ($M = 2$, $T = 5$, and $C = 24$). (**a**,**b**) The cropped watermarked images of Baboon and Barbara; (**c**,**d**) the crossed watermarked images of Baboon and Barbara; (**e**,**f**) the rotation watermarked images of Baboon and Barbara; (**g**,**h**) the AWGN watermarked images of Baboon and Barbara; (**i**,**j**) the JPEG watermarked images of Baboon and Barbara.



(**a**) the crossed line 1 (**b**) the crossed line 4 (**c**) the crossed line 8

Figure 7. Examples of crossed attack with different numbers of lines for Barbara.

(**a**) rotation angles 3° (**b**) rotation angles 18° (**c**) rotation angles 30°

Figure 8. Examples of different rotation angles for Barbara.

From the above observations, the proposed method was powerful in robustness against geometric attacks and common image processing operations. Because the histogram of the modified image incurred during the cropping and crossed attacks kept approximately the same as that of the original image without suffering from attack, the watermark algorithm had higher robustness to cropping and crossed attacks. When suffering from AWGN and JPEG compression attacks, the embedded watermark information could still be effectively extracted, indicating that the functionality of resisting

AWGN and JPEG compression could be ensured. Finally, the embedding of watermark information mainly depended on the statistical property of the histogram and was independent of the size of the image; thus, this proposed watermark method could be applied to images of any size.

5. Conclusions

In this paper, a large capacity histogram-based watermark algorithm is proposed. The watermark design of the proposed method has satisfactory functionalities in terms of robustness to geometric attacks and common image processing operations. In the embedding range selection, possible bad bins will be eliminated successfully with the assistance of a predefined threshold. In the watermark embedding design, by taking the approximately equal number of adjacent histogram bins into account, the proposed method makes every three consecutive bins carry a 3-bit watermark; or rather, it realizes the maximum embedding rate (1 bpb) for the first time. The experiment results demonstrate that the proposed watermark algorithm can provide a preferable tradeoff between embedding capacity and robustness, especially suitable for applications where robustness and embedding capacity are essential, such as copyright protection requirement for enough watermark information and higher robustness. However, a possible limitation for the proposed histogram-based watermarking is the situation where certain operations may make the histogram shape distorted too much. In this case, the watermark extraction will be affected directly. In our future research, one consideration is to improve the newly-designed appropriately equal condition so that when the histogram shape is distorted seriously, the accuracy of watermark extraction is still acceptable.

Author Contributions: All authors contributed equally to this work.

Acknowledgments: We would like to take the opportunity to thank the reviewers for their thoughtful and meaningful comments. This work is supported by the Projects of the National Natural Science Foundation (61370188); the Scientific Research Common Program of the Beijing Municipal Commission of Education (*KM*201610015002, *KM*201510015009); the Beijing City Board of Education Science and the technology key project (*KZ*201510015015, *KZ*201710015010); the Project of Beijing Municipal College Improvement Plan (*PXM*2017_014223_000063); the cooperative education project between the Ministry of Education and Qualcomm Corp (201602034017).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Castiglione, A.; Pizzolante, R.; Palmieri, F.; Masucci, B.; Carpentieri, B.; Santis, A.; Castiglione, A. On-Board Format-Independent Security of Functional Magnetic Resonance Images. *ACM Trans. Embed. Comput. Syst.* **2017**, *16*, 56. [CrossRef]
2. Li, M.; Ren, H.; Zhang, E.; Wang, W.; Sun, L.; Xiao, D. A VQ-Based Joint Fingerprinting and Decryption Scheme for Secure and Efficient Image Distribution. *Secur. Commun. Netw.* **2018**, *2018*, 4313769. [CrossRef]
3. Yang, C. Robust high-capacity watermarking scheme based on Euclidean norms and quick coefficient alignment. *Multimed. Tools Appl.* **2017**, *76*, 1455–1477. [CrossRef]
4. Kang, X.; Huang, J.; Shi, Y.; Lin, Y. A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 776–786. [CrossRef]
5. Barni, M. Effectiveness of exhaustive search and template matching against watermark desynchronization. *IEEE Signal Process. Lett.* **2005**, *12*, 158–161. [CrossRef]
6. Pereira, S.; Pun, T. Robust template matching for affine resistant image watermarks. *IEEE Trans. Image Process.* **2000**, *9*, 1123–1129. [CrossRef] [PubMed]
7. Dugelay, J.; Roche, S.; Rey, C.; Doerr, G. Still-image watermarking robust to local geometric distortions. *IEEE Trans. Image Process.* **2006**, *15*, 2831–2842. [CrossRef]
8. Zheng, D.; Zhao, J.; Saddik, A.E. RST-invariant digital image watermarking based on log-polar mapping and phase correlation. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 753–765. [CrossRef]
9. Dong, P.; Brankov, J.; Galatsanos, N.; Yang, Y.; Davoine, F. Digital watermarking robust to geometric distortions. *IEEE Trans. Image Process.* **2005**, *14*, 2140–2150. [CrossRef] [PubMed]

10. Yuan, C.; Pun, C.; Chen, C. Geometric invariant watermarking by local Zernike moments of binary image patches. *Signal Process.* **2013**, *93*, 2087–2095. [CrossRef]
11. Bas, P.; Chassery, J.; Macq, B. Geometrically invariant watermarking using feature points. *IEEE Trans. Image Process.* **2002**, *11*, 1014–1028. [CrossRef] [PubMed]
12. Nikolaidis, A.; Pitas, I. Robust watermarking of facial images based on salient geometric pattern matching. *IEEE Trans. Multimed.* **2000**, *2*, 172–184. [CrossRef]
13. Xiang, S.; Kim, H.; Huang, J. Invariant Image Watermarking Based on Statistical Features in the Low-Frequency Domain. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 777–790. [CrossRef]
14. Hu, X.; Wang, D. A Histogram Based Watermarking Algorithm Robust to Geometric Distortions. In Proceedings of the International Conference on Electrical, Computer Engineering and Electronics, Jinan, China, 29–31 May 2015.
15. Deng, C.; Gao, X.; Peng, H.; An, L.; Ji, F. Histogram modification based robust image watermarking approach. *Int. J. Multimed. Intell. Secur.* **2010**, *1*, 153–168. [CrossRef]
16. He, X.; Zhu, T.; Yang, G. A geometrical attack resistant image watermarking algorithm based on histogram modification. *Multidimens. Syst. Signal Process.* **2015**, *26*, 291–306. [CrossRef]
17. Zong, T.; Xiang, Y.; Natgunanathan, I.; Guo, S.; Zhou, W.; Beliakov, G. Robust histogram shape-based method for image watermarking. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 717–729. [CrossRef]
18. Feng, B.; Weng, J.; Lu, W. Improved Algorithms for Robust Histogram Shape-Based Image Watermarking. *Int. Workshop Digit. Watermark.* **2017**, *10431*, 275–289.
19. Qi, H.; Zheng, D.; Zhao, J. Human visual system based adaptive digital image watermarking. *Signal Process.* **2008**, *88*, 174–188. [CrossRef]
20. Miscelaneous Gray Level Images. Available online: http://decsai.ugr.es/cvg/dbimagenes/g512.php (accessed on 13 March 2014).