

# Improved Algorithms for Robust Histogram Shape-Based Image Watermarking

Bingwen Feng<sup>1</sup>, Jian Weng<sup>1(✉)</sup>, and Wei Lu<sup>2</sup>

<sup>1</sup> College of Information Science and Technology,  
Jinan University, Guangzhou 510632, China  
bingwfeng@gmail.com, cryptjweng@gmail.com

<sup>2</sup> School of Data and Computer Science,  
Guangdong Key Laboratory of Information Security Technology,  
Sun Yat-sen University, Guangzhou 510006, China  
luwei3@mail.sysu.edu.cn

**Abstract.** In histogram shape-based watermarking schemes, watermark bits are embedded by altering the shape of a histogram extracted from the host image. Exited embedding algorithms use a group of histogram bins to embed only one watermark bit, which results in a rather low watermark capacity. In this paper, we improve the embedding algorithm in two new ways. The first proposed algorithm performs multi-round embedding to carry more watermark bits. In each round of embedding, a specified histogram is extracted so that the embedding operation does not affect watermark bits embedded in previous rounds. The second proposed algorithm uses a group of histogram bins to embed more than one watermark bits, where the coefficient transferring is optimized to minimize the embedding distortion. These algorithms can effectively enlarge the capacity. Furthermore, a histogram preadjustment method, together with a refined coefficient transferring method, is introduced. As a result, reasonable performances on robustness and watermarked image quality are available. The proposed algorithms provide various tradeoff among capacity, robustness, and perceptibility, which supports a wide range of applications.

**Keywords:** Blind watermarking · Robustness · Multilevel histogram · Multiple histogram adjustment

## 1 Introduction

Digital image watermarking is a technique about covertly embedding messages into host images for certain security purposes. Robustness is an important property for a watermarking scheme. It calls for that the embedded messages should resist various attacks, such as common signal processing and geometric attacks. Compared with common signal processing, it is more difficult to develop effective watermarking algorithms robust to geometric attacks, since they will cause challenging synchronization problem.

Many watermarking schemes have been suggested to tackle geometric attacks. By assuming the possible space of attack parameters, some schemes embed negotiated templates and exhaustively resynchronize them in the watermarked image [1–3]. Geometrically invariant image features can also help synchronize watermark bits. Invariant feature detectors such as multiscale Harris detector [4], Harris-Laplace detector [5], scale-invariant feature transform (SIFT) [3], etc., have been used to extract local embedding areas. A concern related to these approaches is that they may suffer from expensive computational cost and high false alarm probability [6]. Some schemes exploit the geometrically invariant domain to gain the robustness against the corresponded geometric attacks. A pioneering work is the Fourier-Mellin transformation designed to be invariant to global rotation, translation, and scaling [7]. Other transformation methods such as moment invariants [8], uniform log-polar mapping [9], polar harmonic transformation [10] are also developed to create geometrically invariant domain. The complexity of these schemes is still high, which is not suitable for real-time applications.

In 2008, Xiang et al. proposed a robust watermarking scheme by utilizing the histogram shape [11]. The histogram constructed from an entire image is independent of pixel locations and thus robust to various geometric distortions. Furthermore, the corresponded watermark extraction does not require local image information, which reduces the risk of desynchronization. This histogram-based watermarking method has been further extended in [5, 12, 13], etc. They improve robustness by combining other robust watermarking techniques or compensating the drawback in the original design. However, the watermark embedding algorithm remains almost unchanged. In these schemes, several bins are employed to embed one watermark bit, which results in a rather low capacity. It is not satisfactory for many applications.

In this paper, two embedding algorithms are proposed to enlarge the capacity of histogram shape-based watermarking. The first one employs multilevel histogram. It embeds watermark bits via several rounds, in each of which a histogram at a specified level is extracted to carry one watermark sequence. The second one considers multiple bin adjustment. It divides the histogram into segments, into each of which a number of watermark bits can be embedded simultaneously. In the embedding procedure, a histogram preadjustment method is introduced to make the histogram extracted more suitable for embedding watermark bits. The coefficient transferring is also refined to minimize the embedding distortion. These proposed algorithms present various tradeoff between robustness and perceptibility, which enriches the application of histogram shape-based watermarking.

## 2 Previous Works

This section briefly describes the histogram-based watermarking method suggested in [11]. It starts by pre-filtering the host image with a Gaussian low-pass filter to gain the robustness against common signal processing. Denote the low-pass filtered image as  $I_{\text{low}}$ , which is of size  $m_I \times n_I$ . Then a global histogram

with  $\#(H)$  bins,  $H = \{H(i)|1 \leq i \leq \#(H)\}$ , is extracted from  $I_{\text{low}}$  by

$$H(i) = \sum_{x=1}^{m_I} \sum_{y=1}^{n_I} \delta \left( \left\lceil \frac{I_{\text{low}}(x, y) - b_1}{t} \right\rceil = i \right) \tag{1}$$

where  $\delta(\cdot) = 1$  if and only if its argument is satisfied, otherwise  $\delta(\cdot) = 0$ .  $t$  denotes the histogram bin width and can be obtained as  $t = (b_2 - b_1)/\#(H)$ .  $b_1$  and  $b_2$  define the range of coefficient values used to extract the histogram. That is,  $H$  only involves  $I_{\text{low}}(x, y) \in [b_1, b_2]$ . This range is modeled by the mean of  $I_{\text{low}}$ .

Then each two neighboring bins form a group to embed one watermark bit. Suppose two bins  $H(2i - 1)$  and  $H(2i)$ ,  $1 \leq i \leq \#(H)/2$ , are used to embed watermark bit  $M(i)$ . Their population is then adjusted in order to satisfy

$$\begin{cases} \frac{H(2i-1)+n}{H(2i)-n} \geq \alpha & \text{if } M(i) = 1 \\ \frac{H(2i)-n}{H(2i-1)+n} \geq \alpha & \text{if } M(i) = 0 \end{cases} \tag{2}$$

where threshold  $\alpha$  controls the population gap between bins, and  $n$  represents the number of coefficients that are transferred from  $H(2i)$  to  $H(2i - 1)$ .  $n$  is negative when coefficients need to be transferred from  $H(2i - 1)$  to  $H(2i)$ . Its value can be calculated as

$$n = \begin{cases} \max \left\{ \frac{\alpha \times H(2i) - H(2i-1)}{1+\alpha}, 0 \right\} & \text{if } M(i) = 1 \\ \min \left\{ \frac{\alpha \times H(2i-1) - H(2i)}{1+\alpha}, 0 \right\} & \text{if } M(i) = 0 \end{cases} \tag{3}$$

to minimize the number of transferred coefficients.

At the extraction phase, the histogram  $\tilde{H}$  is extracted from the received image. Watermark bits are extracted according to

$$\tilde{M}(i) = \begin{cases} 1, & \text{if } \tilde{H}(2i - 1) \geq \tilde{H}(2i) \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Mean value and Gaussian kernel are searched in addition to increase the probability of watermark matching. Note that these searchings require the embedded message to be known at the receiver.

It can be observed that the maximum payload of the embedding algorithm given in Eq. (2) is  $\#(M) = \lfloor (b_2 - b_1)/(2t) \rfloor$ . This scheme has been extended in many approaches, e.g., [5, 12, 13]. However, the embedding algorithm remains similar. As a result, they suffer from the same payload limitation. In the next sections, we present two improved embedding algorithms, which can enlarge the payload effectively.

### 3 Improved Algorithm 1

#### 3.1 Coefficient Transferring

The histogram shape-based embedding method requires transferring a certain number of low-frequency coefficients from one histogram bin to another.

Denote the operation of transferring  $n$  coefficients from  $H(i)$  to  $H(j)$  as  $H(i) \xrightarrow{n} H(j)$ . We propose a new transferring method to minimize the coefficient modification with respect to the Peak Signal to Noise Ratio (PSNR) index.

Take the situation  $H(i) \xrightarrow{n} H(j)$ ,  $i < j$ , as an example. Since the value range of the coefficients in  $H(j)$  is  $[b_1 + (j - 1) \times t, b_1 + j \times t)$ , these coefficients can be changed to  $b_1 + (j - 1) \times t + 0.1$  to restrict the embedding distortion. Let  $\mathcal{B} = \{(x_k, y_k) | 1 \leq k \leq n\}$  denote the set of the best  $n$  coefficients. The  $k$ -th element of  $\mathcal{B}$  can be obtained as

$$(x_k, y_k) = \arg \min_{(x_p, y_p) \in \mathcal{H}'(i)} (b_1 + (j - 1) \times t + 0.1 - I_{\text{low}}(x_p, y_p))^2 \quad (5)$$

where

$$\mathcal{H}'(i) = \mathcal{H}(i) - \{(x_p, y_p) | 1 \leq p \leq k - 1\} \quad (6)$$

where  $(x_k, y_k) \in \mathcal{H}(i)$  holds if  $I_{\text{low}}(x_k, y_k)$  belongs to  $H(i)$ . The situation when  $i > j$  is similar, except the objective function defined in Eq. (5) now becomes

$$(x_k, y_k) = \arg \min_{(x_p, y_p) \in \mathcal{H}'(i)} (b_1 + j \times t - 0.1 - I_{\text{low}}(x_p, y_p))^2 \quad (7)$$

### 3.2 Histogram Preadjustment

Occasionally some bins of the histogram extracted from the host image are thinly populated and thus not suitable to carry watermark bits. Herein we introduce a histogram preadjustment method to guarantee good population for each bin. It transfers coefficients from the other bins to those whose population is less than a threshold  $\beta$ . This preadjustment is detailed in Algorithm 1. The selection of  $\beta$  will be discussed in Sect. 5.1.

---

#### Algorithm 1. Histogram Adjustment

---

**Require:** original histogram  $H$ , threshold  $\beta$ .

- 1: **for**  $i = 1$  to  $\#(H) - 1$  **do**
  - 2:    $n \leftarrow \left( \sum_{j=i}^{\#(H)} H(j) \right) - (\#(H) - i) \times \beta$ .
  - 3:   **if**  $H(i) > n$  **then**
  - 4:      $H(i) \xrightarrow{H(i)-n} H(i+1)$ .
  - 5:   **end if**
  - 6:    $n \leftarrow \beta - H(i)$ ,  $j \leftarrow i + 1$ .
  - 7:   **while**  $n > 0$  **do**
  - 8:      $n' \leftarrow \min\{H(j), n\}$ .
  - 9:      $H(j) \xrightarrow{n'} H(i)$ .
  - 10:     $j \leftarrow j + 1$ ,  $n \leftarrow n - n'$ .
  - 11:   **end while**
  - 12: **end for**
  - 13: **return** adjusted histogram  $H$ .
-

### 3.3 Embedding Algorithm Based on Multilevel Histogram

The first embedding algorithm embeds watermark bits via several rounds. Suppose in the first embedding round, a histogram  $H^{(1)}$  with  $\#(H^{(1)})$  bins, which are of width  $t^{(1)}$ , is extracted, and the  $\#(M^{(1)})$  watermark bits to be embedded are  $M^{(1)} = \{M^{(1)}(i) \mid 1 \leq i \leq \#(M^{(1)})\}$ . Then each two neighboring bins  $H^{(1)}(2i-1)$  and  $H^{(1)}(2i)$ ,  $1 \leq i \leq \#(H^{(1)})/2$ , are employed to embed watermark bit  $M^{(1)}(i)$  by using Eq. (2).

In the  $u$ -th embedding round, a finer histogram  $H^{(u)}$  is extracted by dividing each bin in  $H^{(u-1)}$ , namely  $H^{(u-1)}(i)$ , into two neighboring bins  $H^{(u)}(2i-1)$  and  $H^{(u)}(2i)$  of equal width. It can be implemented equivalently by performing Eq. (1) with bin width  $t^{(u)} = t^{(u-1)}/2$ . There are  $\#(H^{(u)}) = 2\#(H^{(u-1)})$  bins in  $H^{(u)}$ . Consequently, another watermark sequence of length  $\#(M^{(u)}) = 2\#(M^{(u-1)})$  can be embedded. The embedding processing is as same as that in the first round. Since it satisfies that

$$H^{(u)}(2i-1) + H^{(u)}(2i) = H^{(u-1)}(i) \tag{8}$$

reassigning coefficients between  $H^{(u)}(2i-1)$  and  $H^{(u)}(2i)$  does not alter the shape of  $H^{(u-1)}$ .

### 3.4 The Embedding Procedure

This section presents a watermarking scheme by using the first embedding algorithm. Suppose the number of embedding round is  $\#(u)$ . The procedure of embedding watermark bits with the first algorithm consists of the following steps.

1. Low-pass filter the host image  $I$  with a Gaussian filter similar to that in [11] to obtain the low-frequency component  $I_{\text{low}}$  and the high-frequency residual  $I_{\text{high}} = I - I_{\text{low}}$ .
2. Initialize the embedding round as  $u = 1$  and the intermediate watermarked low-frequency component as  $\bar{I}_{\text{low}} = I_{\text{low}}$ .
3. Extract the  $u$ -th level histogram  $H^{(u)}$  from  $\bar{I}_{\text{low}}$  via Eq. (1) with bin width  $t^{(u)}$ .
4. If  $u = 1$ , adjust the histogram by Algorithm 1. Otherwise, adjust the histogram by using

$$\begin{cases} H(2i-1) \xrightarrow{\gamma^{(u)} - H(2i)} H(2i), & \text{if } H(2i) < \gamma^{(u)} \\ H(2i) \xrightarrow{\gamma^{(u)} - H(2i-1)} H(2i-1), & \text{if } H(2i-1) < \gamma^{(u)} \\ \text{No modification,} & \text{otherwise} \end{cases} \tag{9}$$

where  $\gamma^{(u)}$  denotes the lower bound of bin population for the  $u$ -th round of embedding.

5. Embed the  $i$ -th watermark bit  $M^{(u)}(i)$  into  $H^{(u)}(2i-1)$  and  $H^{(u)}(2i)$  by Eqs. (2) and (3) with threshold  $\alpha^{(u)}$ . Note that all coefficient modifications are carried out on  $\bar{I}_{\text{low}}$ .

6. Repeat Step 5 until all the  $\#(M^{(u)})$  watermark bits have been embedded, which gives a new  $\bar{I}_{\text{low}}$ .
7. Increase the embedding round as  $u = u + 1$ . If  $u \leq \#(u)$ , set  $t^{(u)} = t^{(u-1)}/2$  and redo the embedding procedure from Step 3 to perform the next round of embedding. Go to Step 8 otherwise.
8. Post-process the  $\bar{I}_{\text{low}}$  obtained after the last round of embedding in a way similar to that in [11]. That is, for the  $(x, y)$ -th coefficient that belongs to  $H^{(\#(u))}(i)$ ,

$$\bar{I}_{\text{low}}(x, y) = \begin{cases} (i-1) \times t^{(\#(u))} + b_1 + 0.75, & \text{if } \bar{I}_{\text{low}}(x, y) < (i-1) \times t^{(\#(u))} + b_1 + 0.75 \\ i \times t^{(\#(u))} + b_1 - 0.75, & \text{if } \bar{I}_{\text{low}}(x, y) > i \times t^{(\#(u))} + b_1 - 0.75 \\ \bar{I}_{\text{low}}(x, y), & \text{otherwise} \end{cases} \quad (10)$$

9. Reconstruct the watermarked image  $\bar{I}$  by  $\bar{I} = \bar{I}_{\text{low}} + I_{\text{high}}$ .

In the above procedure,  $\gamma^{(u)}$  and  $\alpha^{(u)}$  for each round of embedding should be carefully set so that there are enough coefficients for each coefficient transferring. They will be experimentally discussed in Sect. 4.1. The embedding parameters,  $b_1, b_2, t^{(1)}, \gamma^{(u)}$ , and  $\alpha^{(u)}$ , should be prefixed. At the extraction phase, parameters  $b_1, b_2$ , and  $t^{(1)}$  are required. The extraction procedure can be derived accordingly and omitted here due to limited space.

## 4 Improved Algorithm 2

### 4.1 Embedding Algorithm Based on Multiple Adjustment

The second embedding algorithm uses a group of histogram bins to embed more than one watermark bit. The histogram extracted from the host image is first divided into segments containing  $\pi$  neighboring bins. Then, each segment is used to embed  $\pi - 1$  watermark bits.

Take the first histogram and watermark segments, denoted as  $[H(1), \dots, H(\pi)]$  and  $[M(1), \dots, M(\pi - 1)]$ , respectively, as an example. Each two neighboring histogram bins in the segment,  $H(j)$  and  $H(j+1)$ , embed one watermark bit,  $M(j)$ , by using the rule

$$\begin{cases} H(j) \geq \alpha H(j+1) & \text{for } M(j) = 1 \\ H(j+1) \geq \alpha H(j) & \text{for } M(j) = 0 \end{cases} \quad (11)$$

The desired coefficient transferring should minimize the total embedding distortion while guaranteeing the population gaps between neighboring bins and the

lower bound of bin population. Still consider PSNR as the perceptual measurement. Then the best coefficient transferring can be obtained by solving

$$\arg \min_N \sum_{j,k} ((j - k - 1) \times t + \delta)^2 |N(j, k)| \tag{12}$$

s.t. for  $1 \leq \forall j, \forall k \leq \pi$  :

$$\begin{aligned} & H(j) - \sum_p N(j, p) + \sum_q N(q, j) \geq \gamma \\ & \begin{cases} H(j) - \sum_p N(j, p) + \sum_q N(q, j) \geq \\ \alpha \left( H(j+1) - \sum_p N(j+1, p) + \sum_q N(q, j+1) \right), \text{ if } M(j) = 1 \\ H(j+1) - \sum_p N(j+1, p) + \sum_q N(q, j+1) \geq \\ \alpha \left( H(j) - \sum_p N(j, p) + \sum_q N(q, j) \right), \text{ if } M(j) = 0 \end{cases} \\ & N(j, k) = 0, \text{ if } j > k \end{aligned}$$

where  $\gamma$  represents the allowable thinnest population for each bin, and  $N$  denotes the  $\pi \times \pi$  sized transferring number matrix.  $N(j, k) > 0$  means we should perform  $H(j) \xrightarrow{N(j,k)} H(k)$ , and  $N(j, k) < 0$  calls for the operation  $H(k) \xrightarrow{-N(j,k)} H(j)$ .

By rewriting the above  $l_1$ -norm problem to a linear program, and using, for example, the dual-simplex algorithm, we can obtain a solution, say  $N'$ . Then the histogram segment is modified according to  $N'$  by starting from the transferring with the largest distance, namely  $N'(j', k')$  with

$$(j', k') = \arg \min_{j,k} |j - k| \quad \text{s.t. } N'(j, k) \neq 0 \tag{13}$$

The processing of embedding the  $i$ -th watermark segment into the  $i$ -th histogram segment is similar. Since the influences of modifying coefficients on PSNR are independent of each other and strictly convex, the above embedding algorithm can achieve the minimum embedding distortion with respect to PSNR.

### 4.2 The Embedding Procedure

A watermarking scheme using the second embedding algorithm is developed here. Note that all the coefficient transferring involved in this scheme is still performed via the method introduced in Sect. 3.1. The embedding parameters,  $b_1, b_2, t, \pi, \gamma$ , and  $\alpha$  need to be prefixed. The procedure of embedding watermark bits with the second algorithm is as follows.

1. Obtain the low-frequency component  $I_{low}$  and the high-frequency residual  $I_{high}$  in the same way as described in Step 1 in Sect. 3.4.
2. Extract histogram  $H$  from  $I_{low}$  via Eq. (1) with bin width  $t$ . Then adjust the histogram by Algorithm 1.
3. Divide  $H$  into histogram segments of length  $\pi$ , and divide  $M$  into watermark segments of length  $\pi - 1$ .

4. Use the  $i$ -th histogram segment to embed the  $i$ -th watermark segment according to Eq. (12). Repeat this step until all the watermark segments have been embedded, which gives the watermarked low-frequency component  $\bar{I}_{\text{low}}$ .
5. Post-process  $\bar{I}_{\text{low}}$  and reconstruct the watermarked image  $\bar{I}$  via the same ways as described in Steps 8 and 9 in Sect. 3.4.

At the watermark extraction phase, parameters  $b_1$ ,  $b_2$ ,  $t$ , and  $\pi$  should be known in advance. Note that the extraction rule slightly differs from Eq. (4). Still take the the first histogram segment extracted at the receiver as an example. Suppose it is  $[\tilde{H}(1), \dots, \tilde{H}(\pi)]$ . Then each two neighboring bins  $\tilde{H}(j)$  and  $\tilde{H}(j+1)$  are used to extract the  $j$ -th watermark bit according to

$$\tilde{M}(j) = \begin{cases} 1, & \text{if } \tilde{H}(j) \geq \tilde{H}(j+1) \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

The extraction procedure can be obtained easily and still omitted here because of limited space.

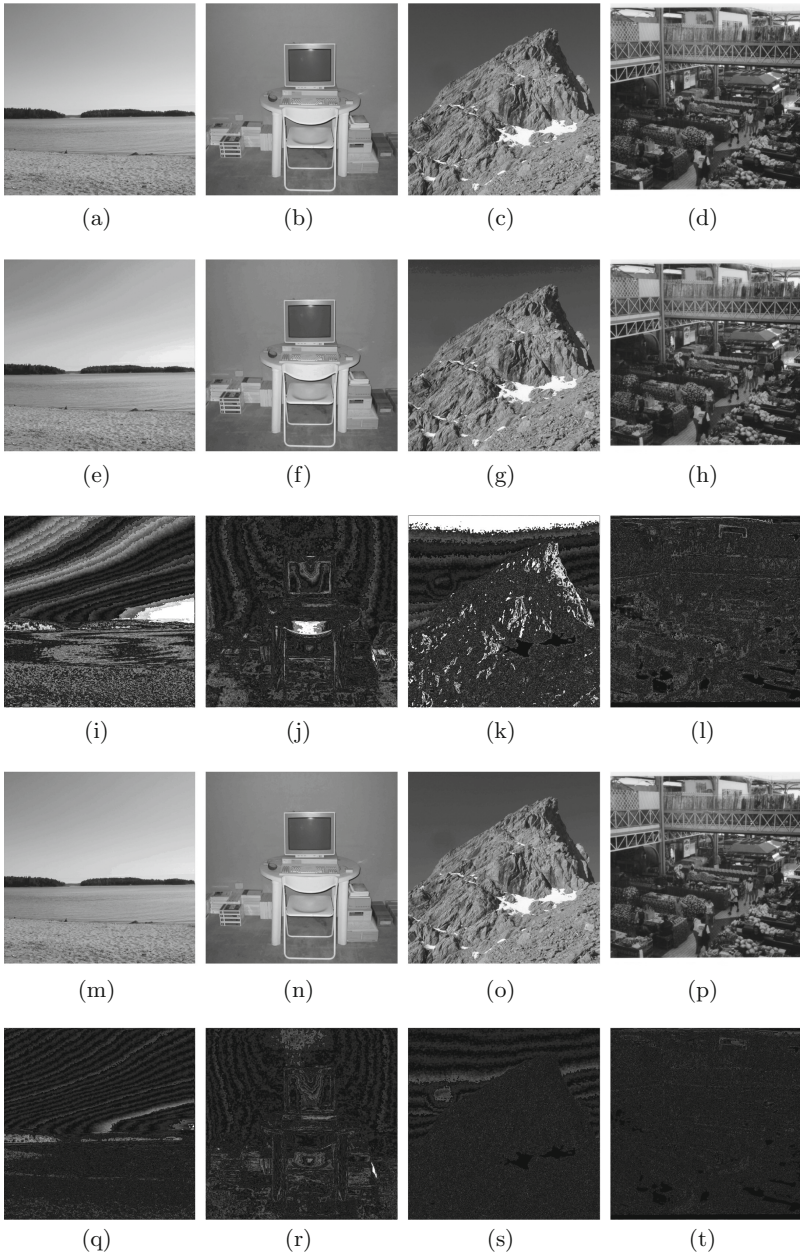
## 5 Experimental Results

The two embedding algorithms are evaluated by testing the corresponded schemes on natural images. 50 images of size  $512 \times 512$  randomly selected from the BOWS2 database [14] are employed as the test images. Some of them are illustrated in Fig. 1. The perceptual quality of watermarked images is measured by PSNR, while the robustness is measured by bit error rate (BER). The considered attacks comprise of common image processing (including JPEG compression and additive white Gaussian noise (AWGN)) and geometric attacks (including rotation, cropping, warping and random bending). These attacks are simulated by Checkmark [15].

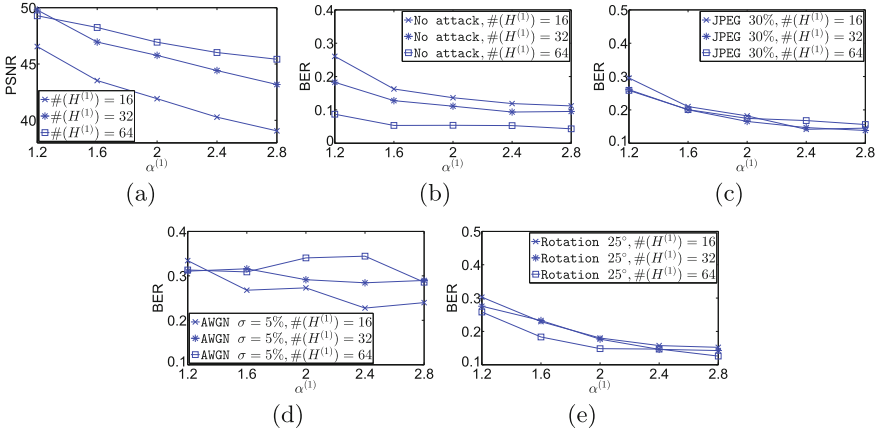
### 5.1 Parameter Setting

Our experiments suggest that a suitable range of coefficient values for the histogram extraction is  $[15, 240]$ , that is,  $b_1 = 15$  and  $b_2 = 240$ . This setting can effectively remove the thinly populated bins at the first and last of the histogram. In the first scheme,  $\gamma^{(u)}$  can be set larger than 10 to compensate the detection error caused by thinly populated bins. However, further increasing  $\gamma^{(u)}$  cannot improve robustness obviously. Note that  $\beta$  used in the histogram preadjustment and  $\gamma^{(u)}$  used in coarser level histograms should be amplified accordingly so that  $\gamma^{(u)}$  can be reached for each level histogram. The case in the second scheme is similar. The settings of  $\beta$ ,  $\gamma^{(u)}$ , and  $\gamma$  are listed in Table 1. There are two parameters left in the first scheme,  $\alpha^{(u)}$  and  $\#(H^{(u)})$ , and three parameters left in the second scheme,  $\alpha$ ,  $\#(H)$ , and  $\pi$ , that require to be prefixed. Herein we experimentally discuss their settings by using the test images mentioned above.





**Fig. 1.** Examples of host and watermarked images. In the first row are the test images: (a) “8137.pgm”, (b) “7088.pgm”, (c) “452.pgm”, and (d) “5149.pgm”. (e)–(h) are the watermarked versions obtained by Scheme I with  $\#(M) = 56$ . (i)–(l) are the corresponded embedding modifications, i.e.,  $|\bar{I} - I|$ . (m)–(p) are the watermarked images obtained by Scheme II with  $\#(M) = 56$ , and (q)–(t) are the corresponded embedding modifications. Parameters of Schemes I and II in this case are listed in Table 2. The display range of  $[0, 255]$  is employed to represent a modification value varying from 0 to 10 in (i)–(l) and (q)–(t).



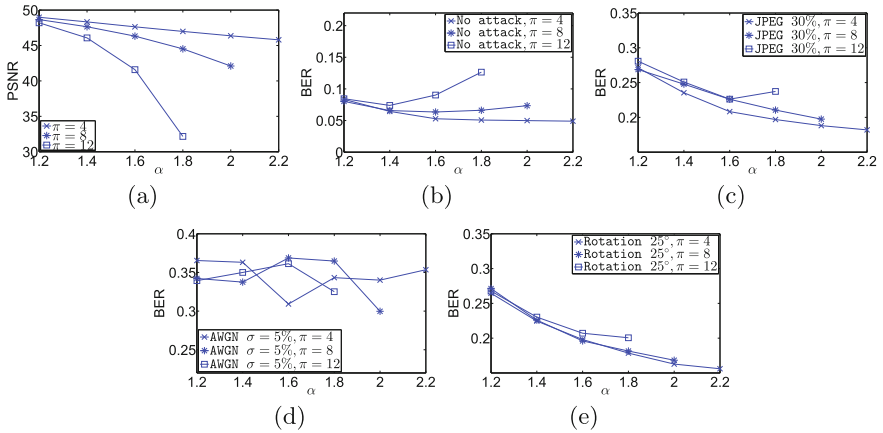
**Fig. 2.** Demonstration of the influences of  $\alpha^{(1)}$  in the first scheme. (a) shows the influence on image quality, while (b) to (e) are on the robustness.  $\gamma^{(1)} = 10$ .  $\beta$  is set according to Table 1.

$\alpha^{(u)}$  should be set according to current  $\#(H^{(u)})$  in the first scheme, since the less the bin number, the more the coefficients contained in each bin and thus in each coefficient transferring. We test their influences on image quality and robustness by using a single level histogram, that is,  $\#(u) = 1$ . The robustness is evaluated in the presences of no attacks, JPEG compression with compression rate 30%, AWGN with standard deviation  $\sigma = 5$ , and rotation with angle  $25^\circ$ . Figure 2 demonstrates this influence by considering PSNR and BER as functions of  $\alpha^{(1)}$ . It can be observed that coarser level histograms perform slightly worse than finer level histograms on both image quality and robustness when fixing  $\alpha^{(1)}$ . It may be because changing the shape of coarser level histograms will cause more coefficient modifications, which will aggravate the side effect of the Gaussian filtering. In addition, robustness turns to rise slowly when  $\alpha^{(1)}$  becomes larger. In view of these, we set  $\alpha^{(u)}$  as given in Table 1, which experimentally presents good tradeoff between robustness and perceptibility.

**Table 1.** Partial parameter settings in proposed schemes.

	$\alpha^{(u)}$ (or $\alpha$ )	$\gamma^{(u)}$ (or $\gamma$ )	$\beta$	Total payload
Scheme I	$1 + \frac{\#(H^{(u)})}{64}$	if $u < \#(u)$ : $\gamma^{(u+1)} \times \left( \frac{1 - (\alpha^{(u)})^2}{1 - \alpha^{(u)}} \right)$ if $u = \#(u)$ : 10	$\frac{1 - (\alpha^{(1)})^2}{2 - 2\alpha^{(1)}} \times \gamma^{(1)}$	$\#(H^{(\#(u))}) \times \left( 1 - \left( \frac{1}{2} \right)^{\#(u)} \right)$
Scheme II	$1 + \frac{\#(H)}{64} \times \sqrt{\frac{2}{\pi}}$	10	$\frac{1 - \alpha^\pi}{(1 - \alpha) \times \pi} \times \gamma$	$\#(H) - \left\lceil \frac{\#(H)}{\pi} \right\rceil$

In the second scheme, both  $\#(H)$  and  $\pi$  will affect the choice of  $\alpha$ . Their influences are assessed on single level histogram with scenarios similar to those in the first scheme. Figure 3 shows the influence of varying  $\alpha$ . It can be observed that increasing  $\pi$  makes watermarked image quality more sensitive to  $\alpha$ , but affects robustness marginally. This is because increasing  $\pi$  will pressure some coefficients to be transferred among histogram bins with larger distance, which impairs watermarked image quality. However, disturbing one histogram bin only affects two embedded watermark bits. Therefore robustness keeps unchanged when varying  $\pi$ . Note that  $\beta$  used in the histogram preadjustment rises exponentially with  $\pi$ . This may also incur the rising of the BER- $\alpha$  curve in the case of  $\pi = 12$ . Therefore,  $\alpha$  should be small enough to guarantee the success of embedding watermark bits. As a result,  $\alpha$  is set as given in Table 1.



**Fig. 3.** Demonstration of the influences of  $\alpha$  in the second scheme. (a) shows the influence on image quality, while (b) to (e) are on the robustness.  $\#(H) = 64$ .  $\gamma = 10$ .  $\beta$  is set according to Table 1.

Table 1 also lists the payloads provided by the proposed schemes. Note that the residual  $\#(H) - \lfloor \#(H)/\pi \rfloor \times \pi$  bins can form an additional histogram segment to embed more watermark bits in the second scheme. It can be observed that the numbers of watermark bits that can be embedded by both schemes tend to be as same as the number of histogram bins, which is much larger than existed histogram-based embedding algorithms.

## 5.2 Performance Comparison

We evaluate the proposed schemes by comparing them with those suggested in [11, 13]. The number of histogram bins  $\#(H)$  in the compared two schemes can be increased to enlarge the payload. However, their supported payloads are still rather small. In view of this, we firstly compare all the schemes in the cases of

**Table 2.** Parameter settings for all the schemes.

Payload ( $\#(M)$ )		32	48	56	63
Scheme I	Coef. range	[15, 240]	[15, 240]	[15, 240]	[15, 240]
	$\#(u)$	1	2	3	2
	$\#(H^{(1)})$	64	32	16	42
	The other parameters are given in Table 1.				
Scheme II	Coef. range	[15, 240]	[15, 240]	[15, 240]	[15, 240]
	$\pi$	2	4	8	4
	$\#(H)$	64	64	64	84
	The other parameters are given in Table 1.				
Xiang et al. [11]	Coef. range	[0.3A, 1.7A]	[0.3A, 1.7A]	–	–
	$\alpha$	2	6	–	–
	$\#(H)$	64	96	–	–
Zong et al. [13]	Coef. range	[0, 255]	[0, 255]	–	–
	$\alpha$	2	4	–	–
	$t$	3	2	–	–
	$\#(H)$	64	96	–	–

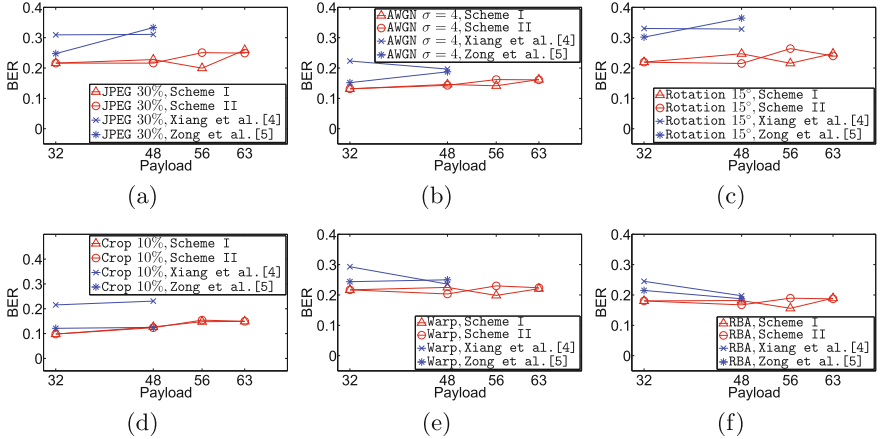
A denotes the mean of a histogram.

$\#(M) = 32$  and  $\#(M) = 48$ , then the two proposed schemes (denoted as Scheme I and Scheme II) are further compared with respect to higher payloads. The embedding threshold  $\alpha$  in [11, 13] is set to keep the PSNR scores of watermarked images similar, which ensures a fair comparison. Parameter settings of these schemes are listed in Table 2.

The watermarked images obtained by Schemes I and II with watermark bit length  $\#(M) = 56$  are demonstrated in Figs. 1(e)–(h) and (m)–(p), respectively, and their embedding modifications are depicted in Figs. 1(q)–(t) and (i)–(l), respectively. The corresponded PSNR scores are compared in Table 3. It can be observed that Scheme I with 3 embedding rounds gives watermarked images with the worst quality. This is because embedding watermark bits in a rather coarse level histogram will cause severe distortions. Nevertheless, as shown in Figs. 1(e)–(h), this quality still seems acceptable in practice.

**Table 3.** Comparison of PSNRs among Different Schemes.

Payload ( $\#(M)$ )		32	48	56	64
Averaged PSNR	Scheme I	46.92	44.07	38.99	42.37
	Scheme II	46.92	43.69	43.40	41.67
	Xiang et al. [11]	46.85	43.97	–	–
	Zong et al. [13]	46.12	43.46	–	–



**Fig. 4.** Robustness performance of different schemes under: (a) JPEG compression, (b) AWGN, (c) rotation, (d) cropping, (e) warping, and (f) random bending.

We test the robustness of these schemes under common signal processing and geometric attacks. The testing results are reported in Fig. 4. It can be observed that the proposed schemes outperform the compared even in the case of small payload, which can be attributed to the coefficient transferring and histogram preadjustment methods proposed. It also shows that increasing the payload affects robustness marginally. Furthermore, the proposed two schemes perform similarly, except that Scheme I presents better robustness in the case of  $\#(M) = 56$ , which is at the cost of watermarked image quality. This suggests that we can choose the embedding algorithm according to practical requirements.

## 6 Discussion and Conclusion

In this paper, we propose two improved embedding algorithms to enlarge the capacity provided by histogram shape-based image watermarking methods. In existed approaches originally suggested in [11], each histogram bin can only embed 0.5 watermark bits at most. This value rises to almost 1 in the proposed algorithms via exploiting multilevel histogram and multiple histogram adjustment. Two new operations, namely histogram preadjustment and coefficient transferring, are developed to further enhance robustness. In the embedding procedure, we alter the embedding algorithm while retaining the other operations as given in [11], such as the low-pass filtering and post-processing, to compare different embedding algorithms. The comparison results show that the proposed algorithms can achieve good performances on both watermarked image quality and robustness. Furthermore, our algorithms present different tradeoff between robustness and perceptibility, which can support various applications. Experimentally we find that the side effect of the Gaussian filtering seriously

degrades the performance of the proposed algorithms. Designing more effective compensative method is our future research.

**Acknowledgements.** This work was supported by National Science Foundation of China (Grant Nos. 61472165 and 61373158), Guangdong Provincial Engineering Technology Research Center on Network Security Detection and Defence (Grant No. 2014B090904067), Guangdong Provincial Special Funds for Applied Technology Research and development and Transformation of Important Scientific and Technological Achieve (Grant No. 2016B010124009), the Zhuhai Top Discipline–Information Security, Guangzhou Key Laboratory of Data Security and Privacy Preserving, Guangdong Key Laboratory of Data Security and Privacy Preserving.

## References

1. Ni, J., Zhang, R., Huang, J., Wang, C., Li, Q.: A rotation-invariant secure image watermarking algorithm incorporating steerable pyramid transform. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 446–460. Springer, Heidelberg (2006). doi:[10.1007/11922841\\_36](https://doi.org/10.1007/11922841_36)
2. Dugelay, J.L., Roche, S., Rey, C., Doërr, G.: Still-image watermarking robust to local geometric distortions. *IEEE Trans. Image Process.* **15**(9), 2831–2842 (2006)
3. Su, P.C., Chang, Y.C., Wu, C.Y.: Geometrically resilient digital image watermarking by using interest point extraction and extended pilot signals. *IEEE Trans. Inf. Forensics Secur.* **8**(12), 1897–1908 (2013)
4. Gao, X., Deng, C., Li, X., Tao, D.: Geometric distortion insensitive image watermarking in affine covariant regions. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **40**(3), 278–286 (2010)
5. Deng, C., Gao, X., Li, X., Tao, D.: Local histogram based geometric invariant image watermarking. *Sig. Process.* **90**(12), 3256–3264 (2010)
6. Licks, V., Jordan, R.: Geometric attacks on image watermarking systems. *IEEE Multimed.* **3**, 68–78 (2005)
7. Ruanaidh, J.J.O., Pun, T.: Rotation, scale and translation invariant spread spectrum digital image watermarking. *Sig. Process.* **66**(3), 303–317 (1998)
8. Zhang, H., Shu, H., Coatrieux, G., Zhu, J., Wu, Q.J., Zhang, Y., Zhu, H., Luo, L.: Affine legendre moment invariants for image watermarking robust to geometric distortions. *IEEE Trans. Image Process.* **20**(8), 2189–2199 (2011)
9. Kang, X., Huang, J., Zeng, W.: Efficient general print-scanning resilient data hiding based on uniform log-polar mapping. *IEEE Trans. Inf. Forensics Secur.* **5**(1), 1–12 (2010)
10. Li, L., Li, S., Abraham, A., Pan, J.S.: Geometrically invariant image watermarking using polar harmonic transforms. *Inf. Sci.* **199**, 1–19 (2012)
11. Xiang, S., Kim, H.J., Huang, J.: Invariant image watermarking based on statistical features in the low-frequency domain. *IEEE Trans. Circuits Syst. Video Technol.* **18**(6), 777–790 (2008)
12. He, X., Zhu, T., Yang, G.: A geometrical attack resistant image watermarking algorithm based on histogram modification. *Multidimension. Syst. Sig. Process.* **26**(1), 291–306 (2015)
13. 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.* **25**(5), 717–729 (2015)

14. Bas, P., Furon, T.: BOWS-2 (2007). <http://bows2.gipsa-lab.inpg.fr/>
15. Pereira, S., Voloshynovskiy, S., Madueno, M., Marchand-Maillet, S., Pun, T.: Second generation benchmarking and application oriented evaluation. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 340–353. Springer, Heidelberg (2001). doi:[10.1007/3-540-45496-9\\_25](https://doi.org/10.1007/3-540-45496-9_25)