

文章编号:1007-5321(2021)01-0001-06

DOI:10.13190/j.jbupt.2020-107

# 广义直方图多 Bin 多进制水印算法

岳 桢<sup>1</sup>, 李子臣<sup>2</sup>, 杨义先<sup>1</sup>

(1. 北京邮电大学 信息安全中心, 北京 100876; 2. 北京印刷学院 信息工程学院, 北京 102600)

**摘要:** 基于直方图的数字水印算法,因具有较强的抗几何攻击和图像处理攻击能力,一直是信息隐藏领域研究的热点. 为了进一步扩大容量和解决比例关系相等问题,提出了多 Bin 多进制直方图数字水印算法概念,引入图像直方图多 Bin 全局调整策略,设计了一种广义的多 Bin 多进制图像数字水印算法. 实验结果表明,设计的算法比现有算法至少提高 2 倍的嵌入容量;同时,在满足载体图像视觉质量的基础上,还能抵抗一定程度的几何攻击和图像处理攻击.

**关键词:** 直方图修改; 多 Bin 多进制; 全局调整

**中图分类号:** TP391

**文献标志码:** A

## A Generalized Multi-Bin and M-ary Histogram-Based Image Watermarking Algorithm

YUE Zhen<sup>1</sup>, LI Zi-chen<sup>2</sup>, YANG Yi-xian<sup>1</sup>

(1. Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China;

2. School of Information Engineering, Beijing Institute of Graphic Communication, Beijing 102600, China)

**Abstract:** The existing histogram-based digital watermarking algorithms, on account of its strong capability to resist geometric attack and image processing attack, have been the focus and hotspot of data hiding domain. In order to further expand capacity and resolve the relationship issue of equal proportions, the author initially proposes a concept of multi-Bin and M-ary histogram digital watermarking algorithm by introducing a new multi-Bin adjustment strategy of image histogram, and designs a generalized multi-Bin and M-ary image digital watermarking algorithm. Experiments show that the proposed algorithm achieves at least 2 times embedding capacity than the existing related works, and can resist geometric attacks and image processing attacks to some extent on the basis of satisfying the image quality.

**Key words:** histogram modification; multi-Bin and M-ary; multi-Bin adjustment strategy

目前,基于图像直方图的数据嵌入已经成为信息隐藏领域重要的研究热点. 现有的基于图像直方图的数据嵌入主要分为 2 类:一是致力于秘密通信的直方图平移技术<sup>[1-2]</sup>;二是致力于版权保护等功能的直方图修改技术<sup>[3-10]</sup>. 直方图修改技术特指对原

始宿主图像的直方图修改,该技术不再强调原始宿主图像是否能够完美恢复,而更多关注于版权保护功能的水印鲁棒性.

Xiang 等<sup>[3]</sup>最早提出对原始宿主图像的直方图进行直接修改以实现水印嵌入,使用高斯滤波

收稿日期: 2020-08-02

基金项目: 国家自然科学基金项目(61370188); 北京市教委科研计划一般项目(KM201610015002, KM201510015009); 北京市教委科研计划重点项目(KZ201510015015, KZ201710015010)

作者简介: 岳 桢(1988—), 男, 博士生.

通信作者: 李子臣(1965—), 男, 教授, 博士生导师, E-mail: lizc2020@163.com.

对宿主图像进行预处理,水印嵌入采用基于嵌入阈值的直方图修改,根据提取阈值实现水印信息的直接提取;该技术主要解决了在水印嵌入过程中实施与像素位置相关联的宿主内容修改而导致抗几何攻击能力不足等问题. 该技术虽然在鲁棒性上取得了一定的效果,然而其所存在的 Bin 大小设置、随机像素修改以及不合适 Bin 选择等问题仍需亟待解决. 为了解决上述算法中由于 Bin 大小设置而导致的容量受限问题, Deng 等<sup>[4-5]</sup>提出由原始的连续 2 个 Bin 扩展为连续 3 个 Bin 来修改像素值,在保证最大修改量恒定不变的前提下提高了水印的鲁棒性. 为了选择更优的像素修改策略, Hu 等<sup>[7]</sup>对整幅宿主图像分成等大小的无重叠图像块,并在水印嵌入过程中引入均方误差方法来优先选择一些图像块进行修改,在一定程度上提高了鲁棒性. 为了解决不合适的 Bin 选择问题, Zong 等<sup>[8]</sup>在对 Bin 大小设置前使用选定的阈值对宿主图像直方图进行划分,只有满足该阈值的像素才符合修改范围,从而在一定程度上淘汰掉不符合候选条件的 Bin. Feng 等<sup>[9-12]</sup>从不可感知性或从鲁棒性上对基于直方图修改的鲁棒水印方法进行了改进,这些算法增强水印鲁棒性,解决了原始算法设计中的某种缺陷,然而所嵌入的水印容量不理想. 最近, Yue 等<sup>[13]</sup>提出一种新的基于 3 个 Bin 八进制直方图修改水印算法,该算法在保证鲁棒性的同时显著提高了嵌入容量,但并没有提供具体的比例相等关系的解决算法,而仅仅是人工的附加额外比例关系来实现水印嵌入和提取.

针对大多数基于直方图修改水印研究还普遍存在低容量问题<sup>[3-10]</sup>和比例相等关系的问题<sup>[13]</sup>,提出了一种广义的直方图多 Bin 多进制水印算法. 提出的全局思想不仅解决了连续 Bin 间比例相等关系现象导致提取正确性受干扰的问题,而且还避免了嵌入过程中可能出现的溢出像素导致比例关系更改的问题. 具体来说,在嵌入过程中首先判断待嵌入水印信息情况,分为比例关系不相等和比例关系相等 2 种情况. 对于比例关系不相等的情况,根据所含的水印信息计算对应的阈值  $T_u$  以及所需要的总像素数目,根据其阈值完成比例不相等水印信息的嵌入,然后把多出的像素数目移至比例关系相等的 Bin 组. 对于比例关系相等的情况,根据所含比例关系相等水印信息计算相应的阈值  $T_e$ ,根据阈值完成比例相等水印信息的嵌入. 实验结果表明,该算法

不仅具有良好的不可视性和较高的嵌入容量,而且能抵抗一定程度的几何攻击和图像处理攻击. 所提算法的主要贡献如下:

1) 提出了一种新的多进制水印概念. 原始的方法仅利用相邻 Bin 间比例关系表示二进制水印信息,而所提方法利用单个 Bin 表示  $\{0, 1, \dots, r-1\}$  的  $r(r \geq 2)$  种可能情况,根据相邻 Bin 的个数  $m(m \geq 2)$  确定  $r^m$  进制水印概念.

2) 提出一种新的直方图区间选择策略. 通过一个预先设定的阈值对原始直方图进行划分,选取合适的直方图嵌入区间,消除不合适 Bin 的同时也避免了提取水印时直方图上溢的问题.

3) 详细阐述了比例相等关系的解决算法.

4) 所提算法极大地提高了基于直方图修改算法的嵌入容量. 传统的算法每个 Bin 嵌入 0.5 bit 信息,所提算法每个 Bin 至少嵌入 1 bit 信息,对于不同进制的水印信息,容量更高.

## 1 多 Bin 多进制水印算法的原理与分析

### 1.1 多 Bin 多进制水印算法概念

**定义** 在基于直方图的数字水印算法中,假设每个 Bin 可表示  $\{0, 1, \dots, r-1\}$  的  $r(r \geq 2)$  种情况,那么相邻  $m$  个 Bin 可表示  $r^m$  种情况,嵌入的水印信息可以表示为  $r^m$  进制,称这样的水印算法为  $m$  个 Bin 的  $r^m$  进制水印算法. 通常也称为多 Bin 多进制水印算法.

若  $m=3, r=3$ ,水印信息编码为  $\{0, 1, \dots, 26\}$  的二十七进制序列,称这样的基于直方图修改水印算法为 3 个 Bin 二十七进制水印算法.

### 1.2 多 Bin 多进制水印算法

假设长度为  $L$  的待嵌入多进制水印信息为  $W = \{w_l | l=1, 2, \dots, L\}$ ,  $w_l \in \{0, 1, \dots, r^m-1\}$ , 每个灰度级组成一个 bin, 每 2 个相邻 bin 组成一个 Bin, 每个 Bin 所含像素的数目等于所包含 2 个 bin 的像素之和. 每相邻  $m$  个 Bin 嵌入一个  $w_l$  信息, 共需  $L$  个 Bin 组. 载体原始 Bin 组所含像素的数目分别为  $H(l)$ , 那么针对不同嵌入水印信息的分布情况如表 1 所示, 其中,  $i \in \{1, 2, \dots, m\}$ .

当  $r_1, r_2, \dots, r_m$  同时表示相同数值时, 相邻  $m$  个 Bin 所满足的比例关系会出现相等的情况. 此时, 引入全局处理策略, 首先计算比例不相等情况和比例相等情况所对应的阈值  $T_u$  和  $T_e$ , 并记录比例相等情况所对应的位置信息  $P(P = \{P_{l_e} | l_e = 1, 2, \dots,$

$L_e\}$ ,  $L_e$  为比例相等水印的总个数).

比例不相等情况所对应阈值  $T_u$  的计算如下:

$$T_u = \left\lfloor \frac{\sum_{l_u=1}^{L_u} H_u(l_u)}{\sum_{l_u=1}^{L_u} \left[ m + \sum_{i=1}^m r_i^u(l_u) \right]} \right\rfloor \quad (1)$$

其中:  $L_u$  为比例不相等水印的总个数,  $H_u(l_u)$  为第  $l_u$  ( $l_u \in \{1, 2, \dots, L_u\}$ ) 个比例不相等水印信息所对应的原始 Bin 组所含像素数目,  $r_i^u(l_u)$  为第  $l_u$  个比例不相等 Bin 组中第  $i$  个 Bin 情况,  $m + \sum_{i=1}^m r_i^u(l_u)$

为第  $l_u$  个比例不相等水印信息所对应的单位数,  $\sum_{l_u=1}^{L_u} \left[ m + \sum_{i=1}^m r_i^u(l_u) \right]$  为所有比例不相等水印信息所对应的单位总数.

由式(1)可知, 阈值  $T_u$  是向下取整, 因此有可能多出部分像素  $\Omega_u$  ( $\Omega_u \geq 0$ ) 没有分配, 提出将  $\Omega_u$  分配给比例相等所对应的 Bin 组.

$$\Omega_u = \sum_{l_u=1}^{L_u} H_u(l_u) - T_u \sum_{l_u=1}^{L_u} \left[ m + \sum_{i=1}^m r_i^u(l_u) \right] \quad (2)$$

表 1 不同嵌入水印信息下相邻  $m$  个 Bin 所对应的多 Bin 多进制水印分布情况

待嵌入水印信息 $w_l$	相邻 $m$ 个 Bin 表示情况	相邻 $m$ 个 Bin 满足比例关系	相邻 $m$ 个 Bin 所需单位情况
'0'	$\overbrace{00 \cdots 0}^m$	$\overbrace{1:1: \cdots: 1}^m$	$m$
'1'	$\overbrace{00 \cdots 01}^{m-1}$	$\overbrace{1:1: \cdots: 1:2}^{m-1}$	$m+1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
' $r-1$ '	$\overbrace{00 \cdots 0(r-1)}^{m-1}$	$\overbrace{1:1: \cdots: 1:r}^{m-1}$	$m-1+r$
' $r$ '	$\overbrace{00 \cdots 010}^{m-2}$	$\overbrace{1:1: \cdots: 1:2:1}^{m-2}$	$m+1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
' $r_m + r_{m-1}r + \cdots r_2r^{m-2} + r_1r^{m-1} =$ $\sum_{i=1}^m r_i \times r^{m-i}, r_i \in \{0, 1, \dots, r-1\}$ '	$\overbrace{r_1 r_2 \cdots r_m}^m$	$\overbrace{(r_1+1):(r_2+1): \cdots: (r_m+1)}^m$	$m + \sum_{i=1}^m r_i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
' $r^m - 1$ '	$\overbrace{(r-1)(r-1) \cdots (r-1)}^m$	$\overbrace{r:r: \cdots: r}^m$	$m \times r$

比例相等情况所对应阈值  $T_e$  的计算如下:

$$T_e = \left\lfloor \frac{\Omega_u + \sum_{l_e=1}^{L_e} H_e(l_e)}{\sum_{l_e=1}^{L_e} \left[ m + \sum_{i=1}^m r_i^e(l_e) \right]} \right\rfloor \quad (3)$$

其中:  $H_e(l_e)$  为第  $l_e$  ( $l_e \in \{1, 2, \dots, L_e\}$ ) 个比例相等水印信息所对应的原始 Bin 组所含像素数目,  $r_i^e(l_e)$  为第  $l_e$  个比例相等 Bin 组中第  $i$  个 Bin 情况,  $m + \sum_{i=1}^m r_i^e(l_e)$  为第  $l_e$  个比例相等水印信息所对应的单位数,  $\sum_{l_e=1}^{L_e} \left[ m + \sum_{i=1}^m r_i^e(l_e) \right]$  为所有比例相等水印信息所对应的单位总数. 同时,  $L_u + L_e = L$ .

由式(3)可知, 如同式(1)一样, 会出现像素溢出现象, 此时, 将溢出部分像素  $\Omega_e$  ( $\Omega_e \geq 0$ ) 分配至最后一个比例相等 Bin 组的第  $m$  个 Bin 上, 在提取最

后一个比例相等水印时, 只需比较前  $m-1$  个 Bin 各自所含像素数目相等即可.

$$\Omega_e = \Omega_u + \sum_{l_e=1}^{L_e} H_e(l_e) - T_e \sum_{l_e=1}^{L_e} \left[ m + \sum_{i=1}^m r_i^e(l_e) \right] \quad (4)$$

以比例相等  $r^m$  进制水印信息为例, 嵌入水印信息后相邻  $m$  个 Bin 表示情况如图 1 所示, 其中,  $H'_e(l_e)_k, H'_e(l_e)_k^2, \dots, H'_e(l_e)_k^m, k \in \{0, 1, \dots, r-1\}$  分别表示第  $l_e$  个 Bin 组中,  $m$  个 Bin 同时表示  $k$  时每个 Bin 所含像素数目, 且  $H'_e(l_e)_k = H'_e(l)_k^1 + H'_e(l_e)_k^2 + \cdots + H'_e(l_e)_k^m$ . 接下来将分 2 种情况进行解释.

1) 对应 Bin 表示为  $\overbrace{kk \cdots k}^m, k \in \{0, 1, \dots, r-1\}$ , 即比例相等. 首先, 比较当前第  $l$  个 Bin 组所含像素

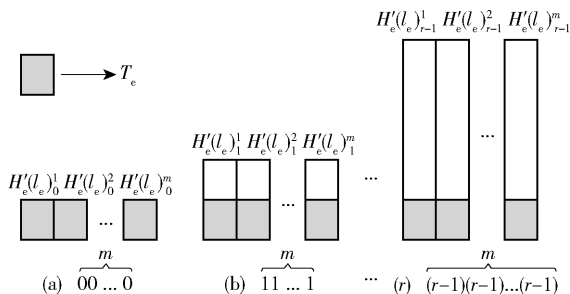


图1 嵌入比例相等水印信息后,相邻  $m$  个 Bin 表示情况

个数  $H(l)$  与  $(k+1)T_e m$  的大小关系,若  $H(l) > (k+1)T_e m$ ,则移出量为  $\Delta_1 = H(l) - (k+1)T_e m$ ,然后从当前第  $l$  个 Bin 组移出  $\Delta_1$  个像素至第  $l+1$  个 Bin 组,最后使当前任意 Bin 所含像素个数满足  $(k+1)T_e$ ,完成水印嵌入;若  $H(l) < (k+1)T_e m$ ,则移入量为  $\Delta_2 = (k+1)T_e m - H(l)$ ,然后从第  $l+1$  个 Bin 组移出  $\Delta_2$  个像素至第  $l$  个 Bin 组,最后使当前任意 Bin 所含像素个数满足  $(k+1)T_e$ ,完成水印嵌入;否则,对当前 Bin 组不进行移入/移出操作,只需使当前任意 Bin 所含像素个数满足  $(k+1)T_e$ ,即完成水印嵌入。

2) 对应 Bin 表示比例不相等,即  $\overbrace{r_1^u r_2^u \cdots r_m^u}^m$ . 首先,比较当前第  $l$  个 Bin 组所含像素数目  $H(l)$  与  $\left[ m + \sum_{i=1}^m r_i^u(l) \right] T_u$  的大小关系,若  $H(l) > \left[ m + \sum_{i=1}^m r_i^u(l) \right] T_u$ ,则移出量为  $\Delta_1 = H(l) - \left[ m + \sum_{i=1}^m r_i^u(l) \right] T_u$ ,然后从当前第  $l$  个 Bin 组移出  $\Delta_1$  个像素至第  $l+1$  个 Bin 组,最后使当前  $m$  个 Bin 所含像素个数分别满足  $\{(r_i^u + 1)T_u \mid i = 1, 2, \dots, m\}$ ,完成水印嵌入;若  $H(l) < \left[ m + \sum_{i=1}^m r_i^u(l) \right] T_u$ ,

则移入量为  $\Delta_2 = \left[ m + \sum_{i=1}^m r_i^u(l) \right] T_u - H(l)$ ,然后从第  $l+1$  个 Bin 组移出  $\Delta_2$  个像素至第  $l$  个 Bin 组,最后使当前  $m$  个 Bin 所含像素个数分别满足  $\{(r_i^u + 1)T_u \mid i = 1, 2, \dots, m\}$ ,完成水印嵌入;否则,对当前 Bin 组不进行移入/移出操作,只需使当前  $m$  个 Bin 所含像素个数分别满足  $\{(r_i^u + 1)T_u \mid i = 1, 2, \dots, m\}$ ,即完成水印嵌入。

最后,给出水印嵌入流程,如图2所示。

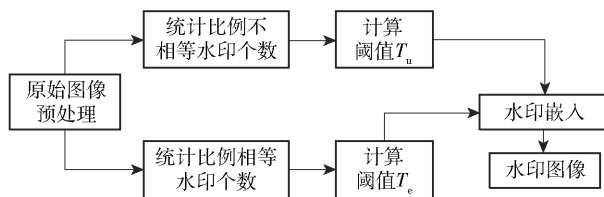


图2 水印嵌入流程

水印提取过程则为水印嵌入逆过程,这里不再赘述。

## 2 实验结果及分析

选用最具有代表性和常用的标准图像系统地测试水印的性能。测试图像是大小为  $512 \times 512$  的标准8位灰度图像<sup>[14]</sup>。实验所用参数为  $m=3, r=3$ 。

### 2.1 嵌入容量分析

嵌入容量是指载体图像能够嵌入的最大水印信息量,其定义为

$$C = Llbr^m \quad (5)$$

根据式(5),嵌入容量取决于  $r$  和  $m$  的取值大小,较大的  $r$  和  $m$  会提供较高的嵌入容量,但视觉质量难免会下降,这是因为需要修改更多的像素来嵌入水印信息。

表2所示为所提算法与文献[3,8,9,11]的算法分别在 Bin 数目为40,60,80的情况下,嵌入容量的对比。在具体的 Bin\_num 设置下,相邻 Bin 数目

表2 所提算法与其他算法嵌入容量比较

算法		Bin_num 为 40		Bin_num 为 60		Bin_num 为 80	
		$m = 2$	$m = 3$	$m = 2$	$m = 3$	$m = 2$	$m = 3$
文献[3]		20	13	30	20	40	26
文献[8]		20	13	30	20	40	26
文献[9]	方法 1	20	13	30	20	40	26
	方法 2	20	13	30	20	40	26
文献[11]		–	13	–	20	–	26
所提算法	$r = 2$	40	26	60	40	80	52
	$r = 3$	$40 \times \text{lb}3$	$26 \times \text{lb}3$	$60 \times \text{lb}3$	$40 \times \text{lb}3$	$80 \times \text{lb}3$	$52 \times \text{lb}3$

$m$  分别设置为 2 和 3. 从表 2 的每一列对比结果中可以看出,即所提算法具有最大的嵌入容量,因为所提算法的嵌入容量与参数  $r$  和  $m$  呈对数正相关关系,即所提算法每 Bin 可以嵌入  $\text{lbr}$  比特信息,而其他算法<sup>[3,8,9,11]</sup> 在每 Bin 中最多嵌入 0.5 bit 信息. 最不理想的嵌入容量下,即  $r = 2$  时,  $\text{lbr} = 1$ , 是 0.5 的 2 倍,因此所提算法显著增大了嵌入容量.

2.2 图像质量分析

实验验证了所提算法在不同嵌入容量下多个测试图像的峰值信噪比 (PSNR, peak signal-to-noise ratio) 和结构相似度 (SSIM, structural similarity index), 其均值结果如图 3 所示. 由图 3 可知,随着嵌入容量的增大,测试图像的 PSNR 和 SSIM 均下降,因此图像失真相应增大,这是由于更多的像素值需要修改以完成水印信息嵌入.

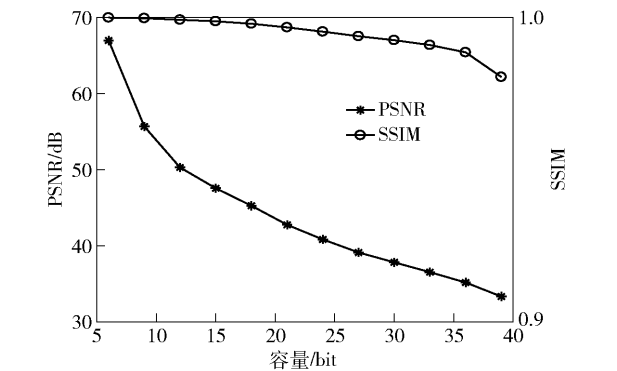


图 3 不同嵌入容量下多个水印图像的 PSNR 和 SSIM 的均值

2.3 水印鲁棒性分析

对于水印的鲁棒性,通常情况下,用比特错误率 (BER, bit error rate) 来衡量. 所提算法做了不同强度的攻击实验,并与文献[3,8-9]提出的当前主流算法在不同嵌入容量下进行了对比,结果如图 4 所示. 由图 4 可知,所提算法可以抵抗裁剪和旋转攻击,整体优于其他主流算法;在加性高斯白噪声攻击下,所提算法稍微弱于其他算法,是因为所提算法引入了比例相等的关系,在遭受严重攻击后,像素修改量的变化降低了算法的提取准确度. 从上述结果可知,所提算法可以抵抗一定的几何攻击和信号处理攻击. 此外,由于水印信息的嵌入主要依赖于直方图的统计特性,与图像的大小无关,因此,所提算法可以用于任何大小的图像.

为了进一步分析水印鲁棒性,当水印嵌入容量  $C = 25, 30$  时,表 3 和表 4 定量比较了所提算法

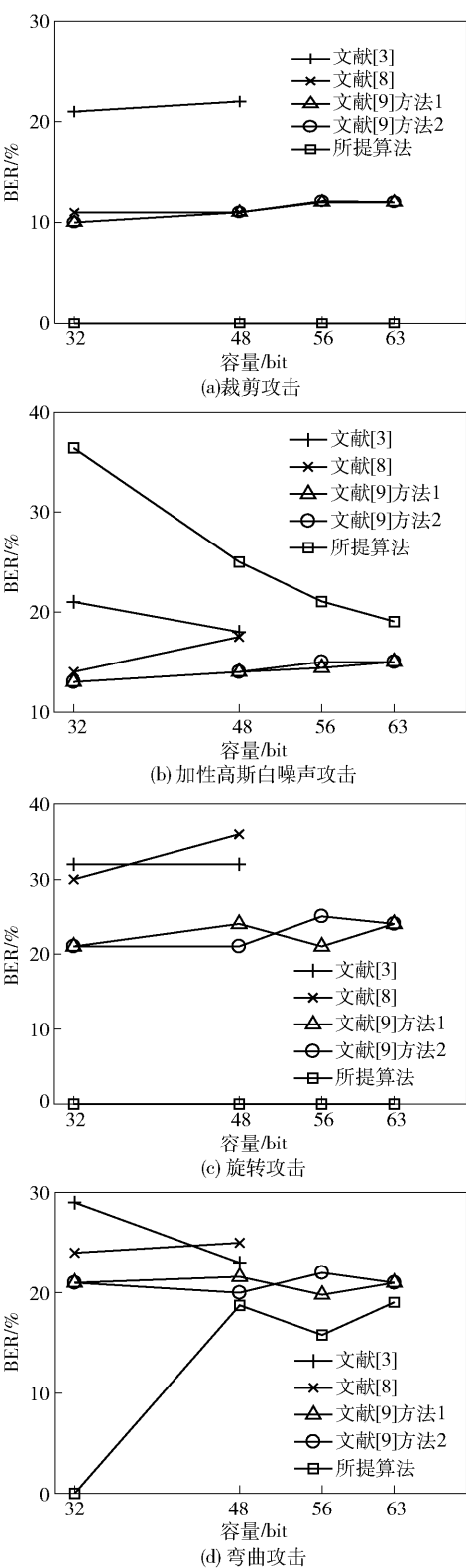


图 4 所提算法与主流算法在攻击下 BER 的比较

与其他算法<sup>[3,5,8,12]</sup> 在遭受攻击下的水印提取比特错误率. 从表 3 和表 4 可知,相较于其他算法,所提算法在全球扭曲攻击、椒盐噪声攻击、裁剪攻击



和旋转攻击上均表现出明显的强鲁棒性. 而针对其他类型攻击,所提算法的水印鲁棒性也均在可接受范围内.

表3 不同攻击下 BER 比较 ( $C=25$ )					%
攻击类型	文献 [3]	文献 [5]	文献 [8]	文献 [12]	所提 算法
抖动 1%	5.3	7.6	0.7	0	12.5
全局扭曲	7.1	19.7	2.5	5.6	0
椒盐噪声	5.0	7.4	0.6	0	0
中值滤波	9.1	11.8	5.6	0	25.0
加性高斯噪声	7.1	14.6	2.7	47.3	12.5
旋转 10°	7.5	22.8	2.5	26.7	0
旋转 25°	8.3	23.0	3.3	30.7	0
裁剪 10%	6.0	5.3	0.6	0	0
裁剪 20%	13.8	7.3	0.9	0	0
裁剪 30%	14.2	7.6	1.9	0	25.0

表4 不同攻击下比特错误率比较 ( $C=30$ )					%
攻击类型	文献 [3]	文献 [5]	文献 [8]	文献 [12]	所提 算法
抖动 1%	6.8	8.6	3.1	0	7.7
全局扭曲	9.1	22.7	6.1	7.1	0
椒盐噪声	6.6	8.6	2.3	0	0
中值滤波	11.2	12.1	9.9	0	23.1
加性高斯噪声	9.6	18.9	5.8	47.6	7.7
旋转 10°	9.1	23.7	7.1	26.8	0
旋转 25°	10.6	25.1	7.3	31.0	0
裁剪 10%	8.0	7.3	2.9	0	0
裁剪 20%	16.0	8.3	3.6	0	0
裁剪 30%	16.2	8.3	4.8	0	0

综上所述,所提算法不仅使嵌入容量明显提高,而且还能够抵抗一定程度的攻击,因此所提算法在解决现有问题上具有明显优势.

### 3 结束语

通过研究已有直方图修改数字水印算法,提出了多 Bin 多进制概念,设计了一种广义的多 Bin 多进制数字水印算法,并通过实验验证了 3 个 Bin 二十七进制水印算法性能. 该算法引入了全局处理策略,解决了水印嵌入过程中连续 Bin 间比例相等问题,以及可能的像素溢出问题. 实验结果表明,所提算法在满足图像视觉质量和鲁棒性的基础上,极大地提高了同类算法的水印嵌入容量. 如何进一步在密文图像中通过直方图修改方法完成水印信息的嵌入和提取以满足用户隐私保护的需求是未来的研究方向.

#### 参考文献:

[1] Li Xiaolong, Li Jian, Li Bin, et al. High-fidelity reversi-

ble data hiding scheme based on pixel-value-ordering and prediction-error expansion[J]. Signal Processing, 2013, 93(1): 198-205.

[2] Kumar R, Chand S, Singh S. An improved histogram-shifting-imitated reversible data hiding based on HVS characteristics[J]. Multimedia Tools and Applications, 2018, 77(12): 13445-13457.

[3] Xiang Shijun, Kim H J, Huang Jiwu. Invariant image watermarking based on statistical features in the low-frequency domain[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2008, 18(16): 777-790.

[4] Deng Cheng, Gao Xinbo, Peng Hao, et al. Histogram modification based robust image watermarking approach[J]. International Journal of Multimedia Intelligence and Security, 2010, 1(12): 153-168.

[5] He Xuansen, Zhu Tau, Yang Gaobo. A geometrical attack resistant image watermarking algorithm based on histogram modification[J]. Multidimensional Systems and Signal Processing, 2015, 26(1): 291-306.

[6] Deng Cheng, Gao Xinbo, Li Xuelong, et al. Local histogram based geometric invariant image watermarking[J]. Signal Processing, 2010, 90(12): 3256-3264.

[7] Hu Xuefeng, Wang Daoshun. A histogram based watermarking algorithm robust to geometric distortions[C] // ICECEE 2015. Jinan: Atlantis Press, 2015: 773-779.

[8] Zong Tianrui, Xiang Yong, Natgunanathan I, et al. Robust histogram shape-based method for image watermarking[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2015, 25(5): 717-729.

[9] Feng Bingwen, Weng Jian, Lu Wei. Improved algorithms for robust histogram shape-based image watermarking[C] // IWDW 2017. Magdeburg: Springer, 2017: 275-289.

[10] Yue Zhen, Li Zichen, Song Peifei, et al. A high-capacity watermarking algorithm using two bins histogram modification[C] // COMPSAC 2018. Tokyo: IEEE Press, 2018: 382-387.

[11] Hua Guang, Xiang Yong, Zhang Leoyu. Informed histogram-based watermarking[J]. IEEE Signal Processing Letters, 2020, 27: 236-240.

[12] Li Xinwei, Guo Baolong, Li Leida, et al. A new histogram based image watermarking scheme resisting geometric attacks[C] // ICIAS 2009. Xi'an: IEEE Press, 2009: 239-242.

[13] Yue Zhen, Li Zichen, Ren Hua, et al. A large capacity histogram-based watermarking algorithm for three consecutive bins[J]. Applied Sciences, 2018, 8(12): 1-16.

[14] Miscellaneous gray level images [EB/OL]. (2014-03-13) [2020-06-11]. <http://decsai.ugr.es/cvg/dbimages/g512.php>.