空域强鲁棒零水印方案

熊祥光 [1]

摘 要 为了解决传统鲁棒水印技术不可感知性和鲁棒性间的矛盾, 对空域零水印技术进行研究, 分析了常规图像处理攻击对载体图像所有选择分块整体均值与分块均值间大小关系的影响, 结果表明此关系具有较强的稳健性. 基于此, 提出了一种新的空域强鲁棒零水印方案. 1) 利用混沌系统对初值敏感的特性映射图像分块的位置和采用混沌加密与 Arnold 空间置乱技术对原始水印信号进行预处理; 2) 采用载体图像所有选择分块整体均值与分块均值间大小关系的稳健性能来构造特征信息; 3) 采用混沌加密和 Arnold 空间置乱技术对生成的零水印信号进行后处理. 仿真实验结果表明, 本文算法对常规的图像处理、尺寸缩放、旋转和多种组合攻击等都表现出较强的抗攻击能力. 与相似的鲁棒零水印方案相比, 本文算法的平均运行时间不仅减少了约 90 %, 而且抗攻击平均性能提高了约 15 %, 表明它具有较低的计算复杂度和更优越的鲁棒性能, 适用于对载体图像质量要求较高的作品版权保护应用场合.

关键词 混沌系统, 空域, 零水印, 常规攻击, 组合攻击, 鲁棒性

引用格式 熊祥光. 空域强鲁棒零水印方案. 自动化学报, 2018, **44**(1): 160−175

DOI 10.16383/j.aas.2018.c160478

A Zero Watermarking Scheme with Strong Robustness in Spatial Domain

XIONG Xiang-Guang[1]

Abstract In order to solve the contradiction between imperceptibility and robustness of traditional robust watermarking technology, the spatial domain zero watermarking technology is researched and the effect of the numerical relationship between the overall mean of all selected blocks and block mean against common image processing attacks is analyzed. The results show that the numerical relationship has strong stability. Based on this, a new zero watermarking scheme with strong robustness in the spatial domain is proposed. Firstly, using the characteristic of sensitivity to initial value, logistic mapping is used to find the position of image block and the original copyright information is pre-processed by chaotic encryption and Arnold scrambling technologies. Secondly, the stability of the numerical relationship between the overall mean of all selected blocks and block mean is utilized to generate feature information. Finally, the generated zero watermarking signal is post-processed again by chaotic encryption and Arnold scrambling technologies. Experimental results on a large number of standard test images show that the proposed algorithm has strong robustness against common image processing, scaling, rotation, and various combination attacks. Compared with similar robust zero watermarking schemes, the proposed scheme not only saves 90 % running time but also improves robustness performance by 15 % on average. These results show that it has lower computational complexity and better performance and can be applied in copyright protection applications for high quality requirements of cover images.

Key words Chaos system, spatial domain, zero watermarking, common attack, combination attack, robustness

Citation Xiong Xiang-Guang. A zero watermarking scheme with strong robustness in spatial domain. *Acta Automatica Sinica*, 2018, **44**(1): 160−175

数字水印技术按水印信号的嵌入域可分为空域数字水印技术和变换域数字水印技术两类. 一般地, 变换域数字水印技术表现出更强的鲁棒性能, 因此现在的数字水印技术绝大多数都属于变换域算法. 变换域水印技术往往将水印信号嵌入到载体信号的变换域系数中, 具有较强的鲁棒性, 对载体信号的版权保护及认证等应用具有非常重要的意义[1−4]. 然而, 嵌入水印信号后, 往往造成载体信号的不可逆失真, 影响了载体信号的不可感知性. 为解决在载体信号中嵌入水印信号后载体信号不可感知性和鲁棒性间不可调和的矛盾, 通常采用鲁棒的图像哈希技术[5−6] 和零水印技术[7−18] 来解决. 这两种技术都是从载体信号中提取鲁棒的特征来构造能唯一标识载体信号的相关信息且不需嵌入到载体信号中, 载体信

收稿日期 2016-06-18 录用日期 2016-11-03
Manuscript received June 18, 2016; accepted November 3, 2016
国家自然科学基金 (61309006), 贵州省教育厅自然科学基金 (黔教合 KY 字 [2015]434), 贵州省教育厅创新群体重大研究项目 (黔教合 KY 字 [2016]027), 中央引导地方科技发展专项资金 (黔科中引地 [2016] 4006) 资助
Supported by National Natural Science Foundation of China (61309006), Natural Science Foundation of Educational Commission of Guizhou Province (Qian-Jiao-He KY Zi [2015]434), Major Research Program of Creative Groups of Educational Commission of Guizhou Province (Qian-Jiao-He KY Zi [2016] 027), and Central Leading Local Science and Technology Development Special Foundation (Qian-Ke-Zhong-Yin-Di [2016]4006)
本文责任编委 赖剑煌
Recommended by Associate Editor LAI Jian-Huang
1. 贵州师范大学大数据与计算机科学学院 贵阳 550001
1. School of Big Data and Computer Science, Guizhou Normal University, Guiyang 550001

号的不可感知性可得到充分的保证, 其抗攻击能力主要都是取决于提取的鲁棒特征对相应的攻击是否具有较好的稳健性能, 从这点来看, 这两种技术是相似的. 但是从目前公开的文献来看, 这两种技术也是存在差别的. 从两种技术的构造过程来看, 图像哈希技术一般需经历特征提取、量化和压缩编码等三个阶段[5], 零水印技术一般需经历特征提取和量化两个阶段[7−18]; 从评价指标来看, 图像哈希技术一般需考虑抗碰撞性/区分性、鲁棒性、单向性、随机性、传递性和摘要性等六个指标[6], 零水印技术需考虑抗碰撞性/区分性 (相似性)、鲁棒性和安全性 (随机性) 等三个指标[7−18], 单向性、传递性和摘要性等三个指标一般是不考虑的, 其生成的零水印信号长度一般由用户根据需求来决定.

目前已提出了一些新颖的零水印技术[8−18]. 文献 [8] 利用离散小波变换后低频子带分块最大奇异值最高位数字的奇偶性来构造零水印信号, 文献 [9] 结合轮廓小波变换和可视密码技术, 将原始水印信号嵌入两个分享中, 从而得到零水印信号. 文献 [10] 提出基于分块压缩感知的图像半脆弱零水印算法, 利用压缩感知理论对每一个图像块进行观测, 并将得到的观测值作为零水印信号, 具有篡改定位和恢复图像功能. 文献 [11] 利用第一主成分向量方向的稳健性, 提出一种新的零水印方案, 具有较好的抗攻击性能. 文献 [12] 利用立体载体图像左右视点小波变换域低频子带视差和离散余弦变换直流系数的稳健性能来构造零水印信号, 具有较强的安全性和鲁棒性. 文献 [13] 在已有视觉密码鲁棒水印算法的基础上, 结合视觉密码和零水印技术的思想, 提出小波域视觉密码零水印算法, 具有较强的抗攻击能力. 文献 [14] 利用各个分块奇异值分解后 U 矩阵和 V 矩阵第一列元素平方的方差间的大小关系来构造零水印信号; 文献 [15] 利用相邻两个子块奇异值矩阵小波低频子带对角线元素的均值大小关系来构造零水印信号, 实验结果表明这两种算法都具有较好的鲁棒性能, 但生成的零水印信号长度都较短. 文献 [16] 提出基于混沌系统和奇异值分解的零水印方案, 对强度不大的常规处理攻击具有较好的鲁棒性能. 文献 [17] 提出基于离散小波变换和奇异值分解的零水印方法, 利用最大奇异值构造零水印信号, 文献 [18] 利用 Arnold 和扩展频谱技术来构造零水印信号, 实验结果表明这两种算法都具有较强的抗攻击能力.

综合上述分析可知, 这些零水印方案都具有较好的抗攻击能力. 但是, 还存在如下的一些问题:

1) 在构造和检测零水印信号时, 需对载体信号采用某种变换方法从空域变换到变换域, 算法的计算复杂度往往较高[7−18].

2) 未考虑算法的安全性问题[8, 14−15], 生成零水印过程或生成零水印信号后都未对其进行任何处理. 若生成零水印的算法公开, 则算法的安全性较低.

3) 直接以生成的无意义二值信号为最终的零水印信号[7−11, 14−16], 人眼缺乏可视性, 仅靠计算提取的二值信号与保存在注册中心的二值信号的相似度来进行版权归属认证往往难以服众.

4) 未分析最终生成的零水印二值信号的分布问题[7−10, 12−18], 对于某些载体图像, 生成的二值信号中 “0” 的比例过高 (“1” 的比例过低) 或 “1” 的比例过高 (“0” 的比例过低), 未满足或近似满足随机信号的特性, 攻击者要想伪造该类零水印信号应该说是很容易的.

针对上述问题, 本文利用混沌系统对初值敏感的特性和在分析常规的图像处理攻击对载体图像所有选择分块整体均值与分块均值间大小关系稳健性的基础上, 选择有意义的二值图像作为原始的水印信号, 直接在空域提出一种基于混沌的强鲁棒零水印方案. 大量的仿真实验结果表明, 本文算法具有较强的抵抗性能. 与相似的鲁棒零水印技术相比, 本文算法的计算复杂度更低且具有更优越的鲁棒性能.

1 基础理论

1.1 Logistic 混沌系统和位置选取

为提高算法的安全性, 一些学者对混沌系统的随机性进行了大量的研究. 本文为了简单, 选择最常用的 Logistic 混沌系统产生混沌信号, 其定义如下:

$$x_{k+1} = \mu \times x_k \times (1 - x_k) \tag{1}$$

其中, $k = 1, 2, 3, \cdots$, $0 < x_k < 1$ 且 $3.5699456 < \mu \leq 4$.

利用混沌系统产生的混沌信号映射图像分块位置信息的方法如下:

步骤 1. 利用密钥 Key_1 作为 Logistic 混沌系统的初值 x_1, 产生长度至少为 $t \times l$ (l 为图像分块的个数, t 为正整数且 $t \geq 2$) 的混沌信号, $\boldsymbol{X} = \{x_1, x_2, x_3, \cdots, x_{t\times l}\}$.

步骤 2. 因当混沌系统的初值很接近时, 产生的混沌信号的前几十个值也较接近, 故从长度至少为 $2l$ 的混沌信号中选择长度为 l 的信号时, 利用密钥 Key_2 作为第一个元素的位置下标 index, 选取长度为 l 的混沌信号, $\boldsymbol{Y} = \{y_1, y_2, y_3, \cdots, y_l\} = \{x_{\text{index}}, x_{\text{index}+1}, \cdots, x_{\text{index}+l-1}\}$.

步骤 3. 得到长度为 l 的混沌信号后, 选择稳定的排序方法对信号 $\boldsymbol{Y}$ 进行升序或降序排序, 并记录排序后信号的索引 $\boldsymbol{S}$ 如下:

$$[\boldsymbol{S} \quad \boldsymbol{Z}\ ] = sort(\boldsymbol{Y}) \tag{2}$$

其中, $sort(\cdot)$ 表示排序函数, $\boldsymbol{S}$ 表示排序后得到的索引, $\boldsymbol{Z}$ 表示排序后得到的升序或降序信号.

步骤 4. 以排序后得到的索引 $\boldsymbol{S}$ 作为选取图像分块次序的依据.

1.2 常规信号处理对整体均值与分块均值大小关系的影响

零水印算法抗攻击的性能如何, 主要取决于构造零水印信号时所选取的重要特征对攻击是否表现出较强的稳健性能. 假设载体图像 I 的大小为 $M \times N$, 分块 F_d 的大小为 $m \times n$, 则可将载体图像划分为 s ($s = \lfloor M/m \rfloor \times \lfloor N/n \rfloor$, 符号 $\lfloor \cdot \rfloor$ 表示向下取整运算) 个分块. 因在对载体图像进行互不重叠的分块时, 图像的边界区域可能不包含在任何一个分块中, 故在计算整体均值时, 只计算所有选择分块 (本文选择图像中的所有分块) 的均值, 而不是整个载体图像的均值. 各个分块的均值 A_d 和载体图像所有选择分块的均值 A 可分别由式 (3) 和式 (4) 计算.

$$A_d = \frac{\sum_{p=1}^{m}\sum_{q=1}^{n} F_d(p,q)}{m \times n} \tag{3}$$

$$A = \frac{\sum_{d=1}^{s} A_d}{s} \tag{4}$$

当载体图像受到某种攻击 (例如噪声和滤波等) 时, 相当于在原始载体图像像素灰度值的基础上加上或减去某个值. 假设攻击信号为 G, 均值为 P, 各分块的均值为 P_d, 则将信号 G 叠加到整个载体图像后各个分块的均值 A'_d 和所有选择分块的整体均值 A' 可分别由式 (5) 和式 (6) 计算.

$$A'_d = \frac{\sum_{p=1}^{m}\sum_{q=1}^{n} \left(F_d(p,q) + G_d(p,q)\right)}{m \times n} = A_d + P_d \tag{5}$$

$$A' = \frac{\sum_{d=1}^{s} (A_d + P_d)}{s} = A + P \tag{6}$$

当载体图像遭受攻击后, $A' - A'_d = (A - A_d) + (P - P_d)$. 也就是说, A' 与 A'_d 的大小关系取决于攻击信号 G 整体均值与分块均值的关系. 当攻击信号是均匀信号时, 可以认为 $P = P_d$, 此时无论 A 是大于、小于或等于 A_d, 因 $P = P_d$, 故 A' 与 A'_d 的关系仍然保持不变. 但是攻击信号往往不是均匀信号, 对载体图像每个像素的修改量不一致, 导致载体图像受到攻击后, A' 与 A'_d 的关系不一定保持不变, 可能会出现大于、相等或小于的情况. 本文认为: 1) 绝大多数分块的均值与载体图像所有选择分块整体均值间的大小关系仍然保持不变; 2) 分块均值与载体图像所有选择分块整体均值差值的绝对值越大, 当载体图像受到攻击后, 分块均值与载体图像所有选择分块整体均值间的大小关系发生改变的比例越小.

为证实此结论, 假设对载体图像进行分块后, 当载体图像未受到攻击时, 分块均值与载体图像所有选择分块整体均值差值的绝对值 Dif 小于给定的阈值 T 的分块数为 l_1, 大于给定的阈值 T 的分块数为 l_2. 当载体图像受到攻击后, Dif 小于给定的阈值 T 的分块数为 l_3, 大于给定的阈值 T 的分块数为 l_4, 则当 Dif 小于给定的阈值 T 时, 分块均值与所有选择分块整体均值间的大小关系发生变化的比例为

$$P_1 = \frac{|l_1 - l_3|}{l_1} \times 100\,\% \tag{7}$$

其中, $|\cdot|$ 表示求绝对值运算. 当 Dif 大于给定的阈值 T 时, 分块均值与所有选择分块整体均值间的大小关系发生变化的比例为

$$P_2 = \frac{|l_2 - l_4|}{l_2} \times 100\,\% \tag{8}$$

当载体图像未受到攻击时, Dif 小于给定的阈值 T 的分块数占所有选择分块数的比例为

$$P_3 = \frac{l_1}{l_1 + l_2} \times 100\,\% \tag{9}$$

当载体图像受到攻击后, Dif 小于给定的阈值 T 的分块数占所有选择分块数的比例为

$$P_4 = \frac{l_3}{l_3 + l_4} \times 100\,\% \tag{10}$$

仿真实验中, 分别在 SIPI (共 146 幅图像)[19] 和 UCID (共 1 334 幅图像)[20] 图像数据库中随机选择 100 幅图像和 Kodak 图像数据库 (共 24 幅图像)[21] 中进行实验测试 (阈值 T 设置为 10, 若原始图像为彩色图像, 则先将其转换为灰度图像). 当载体图像受到常规的图像处理攻击后, 相应的实验结果如表 1 所示. 从表 1 可以看出: 1) 当载体图像未受到攻击时, $Dif > 10$ 的比例 ($100 - P_3$) 远大于 $Dif < 10$ 的比例 (表 1 中的 P_3), 且当载体图像受到攻击后, 与原始比例相比, 这些比例的波动不大 (表 1 中的 P_3 和 P_4), 表明绝大多数分块的均值与载体图像所有选择分块整体均值间的大小关系仍然保持不变; 2) 当载体图像受到攻击后, $Dif > 10$ 时出现分块均值与载体图像所有选择分块整体均值间的大小关系发生变化的比例小于 $Dif < 10$ 时的比例 (表 1 中的 P_1 和 P_2), 表明 Dif 越大, 分块均值与载体图像所有选择分块整体均值间的大小关系发生变化的比例越小.

为了进一步分析载体图像所有选择分块整体均值与分块均值差的绝对值在不同阈值条件下的大小关系变化情况, 将常规的信号处理攻击进行组合, 详见表 2. 在不同阈值 T 和常规信号处理组合攻击下, 相应的实验结果如表 3 所示. 从表 3 可以看出: 1)

在相同的阈值条件下, 随着攻击强度的不断增大, 分块均值与载体图像所有选择分块整体均值间的大小关系发生变化的比例也越大 (表 3 中的 P_1 和 P_2); 2) 阈值 T 越大, 则 Dif 小于阈值 T 的分块数占所有选择分块数的比例越大 (表 3 中的 P_3 和 P_4).

表 1　所有选择分块整体均值与分块均值间差值关系变化情况 (%)

Table 1　The changes of difference relationship between the overall mean of all selected blocks and block mean (%)

图像集	攻击方式	P_1	P_2	P_3	P_4
Kodak	JPEG 压缩 (20)	2.5417	0.8506		21.1772
	中值滤波 (3 × 3)	2.7067	0.7202		21.0870
	维纳滤波 (3 × 3)	2.6860	0.7406	21.0334	21.0992
	椒盐噪声 (0.1)	10.6074	2.7533		22.5952
	高斯噪声 (0.1)	13.5641	3.1607		22.6128
SIPI	JPEG 压缩 (20)	3.3064	2.6867		33.8072
	中值滤波 (3 × 3)	3.3476	2.4786		33.8938
	维纳滤波 (3 × 3)	3.1466	2.4220	33.6135	34.2121
	椒盐噪声 (0.1)	12.0245	9.9359		35.5716
	高斯噪声 (0.1)	13.6321	11.8315		35.8746
UCID	JPEG 压缩 (20)	3.0059	0.4684		12.9183
	中值滤波 (3 × 3)	3.2848	0.4411		12.8613
	维纳滤波 (3 × 3)	2.9985	0.3996	12.8779	12.9180
	椒盐噪声 (0.1)	13.4309	1.5459		14.1185
	高斯噪声 (0.1)	19.5050	1.7995		14.3138

表 2　组合攻击

Table 2　Combination attacks

	JPEG 压缩	中值滤波	维纳滤波	椒盐噪声	高斯噪声
攻击方式 1	20	3 × 3	3 × 3	0.1	0.1
攻击方式 2	15	5 × 5	5 × 5	0.2	0.2
攻击方式 3	10	7 × 7	7 × 7	0.3	0.3

此外, 以大小为 512 像素 × 512 像素的标准测试灰度图像 Barbara 为例, 对其进行互不重叠的 8 × 8 分块, 分析常规的信号处理攻击对载体图像所有选择分块整体均值和各个分块均值的影响, 相应的实验结果如图 1 所示 (考虑各个分块均值与载体图像所有选择分块整体均值的差值分布情况, 由于共有 4 096 个分块, 若全选, 则生成的差值分布图画面混乱, 很难看清其分布情况, 故图 1 仅随机选择 400 个分块). 从图 1 可以看出, 当载体图像受到这些常规的图像处理攻击后, 分块均值与载体图像所有选择分块均值间的差值分布与未受到攻击时的差值分布很相似, 表明载体图像受到攻击后, 绝大多数分块的均值与载体图像所有选择分块整体均值间的大小关系具有非常好的稳健性能, 仍然保持不变.

表 3　在给定阈值条件下, 所有选择分块整体均值与分块均值间差值关系变化情况 (%)

Table 3　The changes of difference relationship between the overall mean of all selected blocks and block mean with a given threshold (%)

图像集	阈值 T	攻击方式	P_1	P_2	P_3	P_4
Kodak	10	攻击方式 1	6.3325	1.6419		21.7064
		攻击方式 2	12.2779	3.1058	21.0334	22.4345
		攻击方式 3	22.4367	4.8887		23.4951
	20	攻击方式 1	5.4244	2.8775		39.5189
		攻击方式 2	11.9098	6.1418	38.2100	41.3628
		攻击方式 3	21.6383	10.0186		43.5223
SIPI	10	攻击方式 1	7.0620	5.8494		34.6629
		攻击方式 2	14.6056	10.5559	33.6135	36.0441
		攻击方式 3	25.8649	16.0384		38.2140
	20	攻击方式 1	4.7413	6.2570		59.5355
		攻击方式 2	10.2979	12.7624	58.0793	61.7652
		攻击方式 3	17.4728	20.5366		64.3733
UCID	10	攻击方式 1	8.5528	0.9181		13.4140
		攻击方式 2	19.8770	1.9943	12.8779	14.1996
		攻击方式 3	39.3653	3.7169		15.6343
	20	攻击方式 1	8.1239	2.1013		26.2574
		攻击方式 2	18.2561	4.6279	25.0641	27.9303
		攻击方式 3	33.9565	8.3950		30.4609

综合表 1、表 3 和图 1 可以发现, 当载体图像遭受到常规的信号处理攻击后, 载体图像所有选择分块整体均值与绝大多数分块均值间的大小关系仍未发生变化, 表现出较强的稳健性. 也就是说, 该特征 (分块均值与载体图像所有选择分块整体均值间的大小关系) 对常规的图像处理攻击具有较强的免疫力, 可利用该重要特征来构造鲁棒的零水印信号.

2　零水印构造算法

设原始载体图像 I 的大小为 $M \times N$, 要生成的零水印信号长度为 $H \times K$, 从此载体图像中构造零水印信号的详细步骤如下:

步骤 1. 对载体图像 I 进行互不重叠的大小为 $m \times n$ ($m = \lfloor M/H \rfloor$, $n = \lfloor N/K \rfloor$) 的分块, 设相应的分块为 F_d, $d = 1, 2, 3, \cdots, \lfloor M/m \rfloor \times \lfloor N/n \rfloor$.

步骤 2. 选择 Logistic 混沌系统的初值 x_1 (密钥

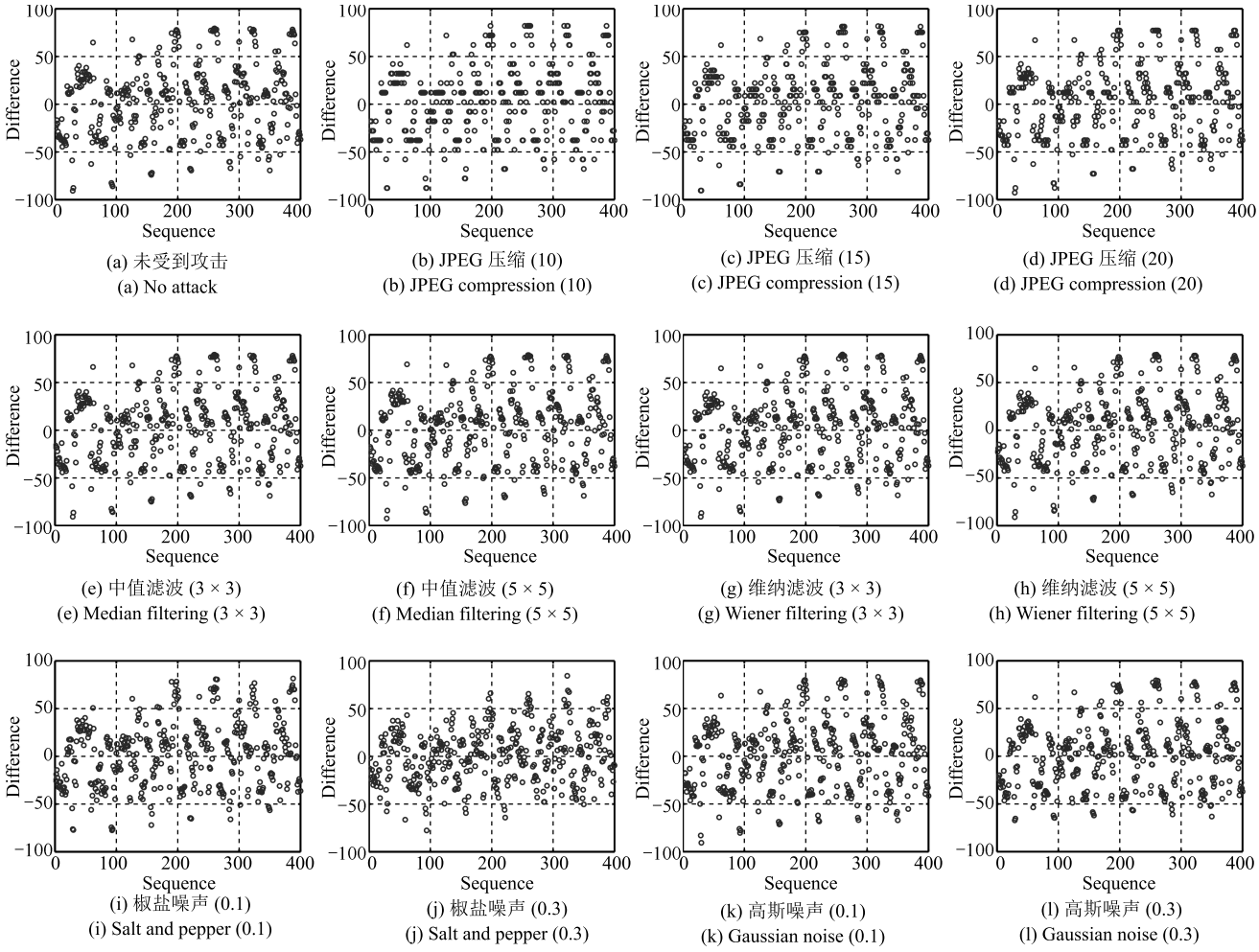


图 1 所有选择分块整体均值与各个分块均值间差值关系图

Fig. 1 The flowchart of differences relationship between the overall mean of all selected blocks and each block mean

Key_1), 利用式 (1) 产生足够长度的混沌信号, 随后从生成的混沌信号中以某一个位置 $index$ (密钥 Key_2) 为基准选取长度为 $H \times K$ 的信号.

步骤 3. 采用稳定的排序方法对选取的混沌信号进行升序或降序排序, 得到排序后信号的索引.

步骤 4. 基于得到的索引利用式 (3) 计算各个分块的均值 A_d 和利用式 (4) 计算载体图像所有选择分块的整体均值 A.

步骤 5. 利用载体图像所有选择分块整体均值 A 和各个分块均值 A_d 间的大小关系, 构造原始载体图像的稳健特征信息 B(每个分块仅构造一比特信息), 其构造方法为: 若分块均值 A_d 大于所有选择分块的整体均值 A, 则生成的信号 B_d 为 1; 否则, 生成的信号 B_d 为 0. 即

$$B_d = \begin{cases} 1, & A_d > A \\ 0, & A_d \leq A \end{cases} \tag{11}$$

步骤 6. 读取有意义的原始水印信号 W, 采用文献 [4] 中的混沌加密和 Arnold 空间置乱方法, 先利用密钥 Key_3 (产生 Logistic 混沌信号的初值) 对水印信号 W 进行混沌加密, 之后再利用密钥 Key_4 (Arnold 空间置乱的迭代次数) 进行 Arnold 空间置乱, 得到混沌加密和 Arnold 置乱后的水印信号 D.

步骤 7. 利用步骤 5 构造生成的稳健特征信息 B 与步骤 6 预处理后生成的水印信号 D 进行异或运算, 生成零水印信号 E.

步骤 8. 为进一步增强零水印信号的安全性, 与步骤 6 相同, 对水印信号 E 再次利用密钥 Key_5 (产生 Logistic 混沌信号的初值) 进行混沌加密和利用密钥 Key_6 (Arnold 空间置乱的迭代次数) 进行 Arnold 空间置乱处理, 生成最终的零水印信号 F.

步骤 9. 从时间戳权威机构申请得到时间戳, 并将最终生成的零水印信号 F 与其进行绑定, 之后将绑定后得到的信号在知识产权数据库 (Intellectual property right database, IPRD) 中注册, 零水印信号的构造和注册过程全部结束.

3　零水印提取算法

从可能已遭受攻击的载体图像中提取零水印信号的详细步骤如下:

步骤 1. 与零水印构造算法的步骤 1～4 相同, 读取待检测图像, 根据零水印信号构造过程分块的大小, 对其进行互不重叠的分块, 计算载体图像所有选择分块的整体均值 A 和基于得到的索引计算各个分块的均值 A_d.

步骤 2. 与零水印信号构造过程相同, 利用载体图像所有选择分块整体均值 A 和各个分块均值 A_d 间的大小关系, 构造原始载体图像的稳健特征信息 B, 其构造方法为: 若分块均值 A_d 大于所有选择分块的整体均值 A, 则生成的信号 B_d 为 1; 否则, 生成的信号 B_d 为 0. 即

$$B_d = \begin{cases} 1, & A_d > A \\ 0, & A_d \le A \end{cases} \tag{12}$$

步骤 3. 从 IPRD 中分离出零水印信号 F 和时间戳信息. 与零水印信号的构造过程相同, 利用文献 [4] 中的混沌解密和逆 Arnold 空间置乱方法, 对水印信号 F 先利用密钥 Key_6 进行逆 Arnold 空间置乱, 之后再利用密钥 Key_5 进行混沌解密, 得到逆 Arnold 空间置乱和混沌解密后的水印信号 E.

步骤 4. 利用步骤 2 生成的图像稳健特征信息 B 和步骤 3 得到的水印信号 E 进行异或运算, 得到还未逆 Arnold 空间置乱和混沌解密的水印信号 D.

步骤 5. 与步骤 3 相同, 利用文献 [4] 中的混沌解密和逆 Arnold 空间置乱方法, 对水印信号 D 利用密钥 Key_4 进行逆 Arnold 空间置乱和利用密钥 Key_3 进行混沌解密, 得到最终提取的水印信号 EW.

步骤 6. 若提取的水印信号 EW 人眼可直接识别且提取的时间戳通过认证, 则认为版权申诉者具有该载体作品的合法版权, 否则认为版权申诉者不具有该载体作品的合法版权.

4　实验结果及分析

本实验选择在笔记本电脑上使用 Windows 7 操作系统 (32 位) 和 Matlab R2010a 平台来模拟一些常见的图像处理操作和几何攻击. 选取大小为 64 像素 × 64 像素, 标识 "贵州师大" 的二值图像作为原始水印信号, 如图 2 (a) 所示. 实验选取的原始载体图像为 512 像素 × 512 像素的标准灰度图像, 分别为 Aerial, Barbara, Boat, Couple, Elain, Frog, Goldhill 和 Zelda, 如图 3 所示. 实验过程中, 不重叠分块的大小为 8 像素 × 8 像素. 以 Barbara 载体图像为载体信号, 利用本文算法, 生成的零水印信号如图 2 (b) 所示. 从图 2 (b) 可以看出, 未对生成的零水印信号进行处理之前, 该零水印信号杂乱无章, 人眼是不可识别的, 具有较好的保密性能.

(a) 原始水印　(b) 生成的零水印

(a) Original watermarking　(b) Generated zero watermarking

图 2　原始的水印和生成的零水印

Fig. 2　Original watermarking and generated zero watermarking

4.1　分块大小对算法性能的影响

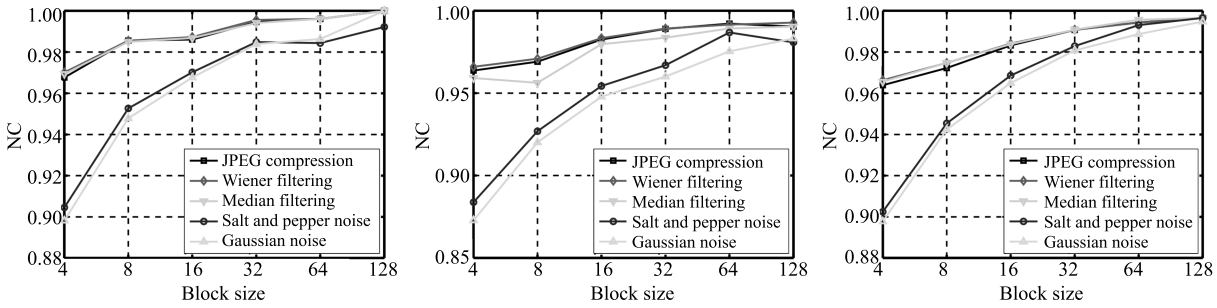
为了测试本文算法的抗攻击能力, 采用有关水印算法文献中普遍使用的归一化互相关 (Normalize correlation, NC)[4] 系数来客观评判原始的水印信号与提取的水印信号的相似程度. 一般地, NC 值越大, 表明提取的水印信号与原始水印信号越相似, 水印算法的抗攻击能力就越强; 反之, 水印算法的抗攻击能力就越弱.

对于本文算法, 因为仅在每一分块中构造一位零水印信号, 所以分块大小会对最终生成的零水印信号的长度和鲁棒性能造成影响. 显然, 分块的大小越大, 生成零水印信号的长度就越短; 反之, 生成零水印信号的长度就越长. 为测试分块大小对本文算法性能的影响, 实验时分别以 4×4, 8×8, 16×16, 32×32, 64×64 和 128×128 为分块的大小, 在文中选择的 8 幅载体图像、SIPI 图像集中大小为 512 × 512 的 26 幅图像 (misc 文件夹) 和 Kodak 图像集中的 24 幅图像进行测试, 相应的实验结果如图 4 所示 (若原始图像为彩色图像, 则先将其转换为灰度图像).

(a) Aerial　(b) Barbara　(c) Boat　(d) Couple　(e) Elain　(f) Frog　(g) Goldhill　(h) Zelda

图 3　原始的测试图像

Fig. 3　Original test image



(a) 文中选择的 8 幅图像 (a) Selected 8 images (b) SIPI 图像集 (b) SIPI image set (c) Kodak 图像集 (c) Kodak image set

图 4 不同分块大小对本算法性能的影响

Fig. 4 The effect of the algorithm performance for different block size

从图 4 可以看出, 随着分块大小不断增大, 对于这五种攻击 (其他攻击类型的变化趋势与这五种攻击相似) 来说, 算法的抗攻击性能总体上都是处于上升的趋势. 也就是说, 在已知载体图像大小和最终要生成的零水印信号长度的情况下, 应尽可能选择更大的分块, 使零水印算法的抗攻击性能得到进一步的提高. 一般地, 若载体图像大小为 $M \times N$, 要生成的零水印信号大小为 $H \times K$, 则将分块的大小设置为 $\lfloor M/H \rfloor \times \lfloor N/K \rfloor$ 即可. 对于本文算法来说, 由于选择的载体图像大小为 512×512, 原始的水印信号大小为 64×64, 故分块的大小设置为 8×8.

对于零水印技术来说, 算法的抗攻击能力强弱主要取决于构造的特征信息. 因此, 这里对本文算法和文献 [14−16] 构造生成的特征信息的抗攻击能力进行比较. 对文中选择的 8 幅载体图像, 以 4×4, 8×8, 16×16 和 32×32 为分块大小, 测试四种算法抵抗 JPEG 压缩 (10)、窗口大小为 3×3 的维纳滤波和中值滤波与噪声强度为 0.1 的高斯噪声和椒盐噪声攻击的能力, 相应的实验结果如图 5 所示 (图中的每一个点都是五种攻击的平均值). 从图 5 可以看出, 随着分块大小不断增大, 本文算法和文献 [14−16] 的鲁棒性能都有所提高, 且本文算法比其他三种算法具有更好的性能.

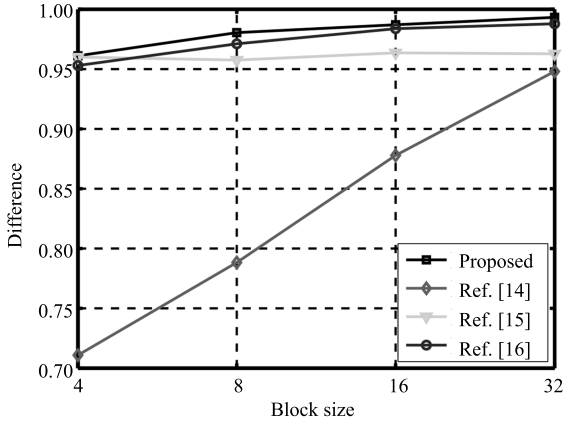


图 5 不同分块大小实验结果

Fig. 5 Experimental results with different block size

4.2 零水印均衡性测试

本文算法生成的零水印信号为二值信号, 若生成的信号中 “0” 和 “1” 的个数基本相等, 即近似满足均匀分布, 则算法的安全性能就更好. 设 L, N_0, N_1 和 E 分别表示生成的二值信号中总的个数、“0” 的个数、“1” 的个数和信号的均衡性, 则 E 的定义如下:

$$E = \frac{|N_0 - N_1|}{L} \tag{13}$$

其中, $|\cdot|$ 表示求绝对值运算. 通过计算, 不同载体图像构造的特征信息和最终生成的零水印信号的均衡性测试结果如表 4 所示.

表 4 不同载体图像生成的特征信息和零水印均衡性测试

Table 4 Balance test of generated feature information and zero watermarking from different cover images

	特征信息 B			最终生成的零水印		
	N_0	N_1	E	N_0	N_1	E
Aerial	1 652	2 444	0.1934	2 055	2 041	0.0034
Barbara	2 083	2 013	0.0171	2 086	2 010	0.0186
Boat	1 366	2 730	0.3330	2 055	2 041	0.0034
Couple	1 825	2 271	0.1089	1 998	2 098	0.0244
Elain	2 088	2 008	0.0195	2 085	2 011	0.0181
Frog	2 027	2 069	0.0103	2 042	2 054	0.0029
Goldhill	2 226	1 870	0.0869	2 005	2 091	0.0210
Zelda	1 943	2 153	0.0513	2 078	2 018	0.0146
平均值	1 901	2 195	0.1026	2 051	2 046	0.0133

从表 4 可以看出, 从这些载体图像构造的特征信息 B 和最终生成的零水印信号的均衡性都非常低, 满足或近似满足均匀分布的基本要求, 其均衡性能都较好.

此外, 将本文算法与文献 [13−16] 在选择的 8 幅载体图像中进行均衡性性能比较, 8 幅图像的平均实验结果如表 5 所示.

表 5　不同算法生成的特征信息和零水印均衡性测试

Table 5　Balance test of generated feature information and zero watermarking from different algorithms

	8 幅载体图像均衡性结果的平均值 (E)	
	特征信息 B	最终生成的零水印
本文	0.1026	0.0133
文献 [13]	0.0046	0.0000
文献 [14]	0.1707	0.0130
文献 [15]	0.0087	0.0060
文献 [16]	0.0111	0.0107

从表 5 可以看出, 文献 [13] 最终生成的零水印的均衡性值为 0, 其原因主要是因为该算法在生成最终的零水印信号时, 不是直接采用生成的特征信息 B 与水印信号进行异或操作生成, 而是基于特征信息 B 中的值来构造, 且在构造零水印信号的过程中, “0” 和 “1” 的个数始终相等, 故最终生成的零水印信号中 “0” 和 “1” 的个数是相等的, 从而均衡性值为 0. 从表 5 也可以看出, 其他四种算法的均衡性值也都很低, 均衡性能都很好.

4.3　安全性测试

本文算法的安全性主要依赖于 Logistic 混沌系统的初值和其他的几个密钥 (详见零水印构造算法). 也就是说, 即使构造零水印的算法完全公开, 攻击者没有正确的密钥 (即使仅有一个密钥是错误的), 生成的零水印信号也是错误的. 限于篇幅, 仅分析密钥 Key_1 的初值对本文算法安全性能的影响 (假设其他的密钥都正确). 在实验中, 假设选取的初值 x 为 0.123456. 为了验证本文算法的安全性, 以 0.123407 为第一个密钥, 以 0.000001 为步长, 以 0.123506 为最后一个密钥, 根据提出的算法计算这 100 个密钥提取的零水印信号与原始水印的相似性, 相应的实验结果如图 6 所示. 从图 6 可以看出: 1) 正确密钥提取出的零水印信号与原始水印信号的相似性为 1.0000; 2) 虽然这些错误密钥与正确的密钥相差较小 (特别是错误密钥 0.123455 和 0.123457 与正确的密钥 0.123456 都只相差 0.000001), 但是, 提取的零水印信号与原始水印信号的相似性也仅为 0.5 左右. 需要注意的是, 上面的分析是基于密钥 Key_1 的初值位数已知的情况下进行穷举搜索得到相关结果的. 若初值位数未知, 则整个搜索空间是非常大的, 采用穷举搜索到真值的概率微乎其微.

另外, 即使侵权者根据本文算法生成了代表他的零水印信号也注册在 IPR 数据库中, 也是不能证明作品的版权归宿是属于侵权者的. 因为本文算法在对生成的零水印信号进行注册时, 附加了通过权威机构认证的时间戳, 在出现版权纠纷需进行版权归属认证时, 当然只认可注册时间在前的零水印信号, 后面注册的零水印信号都可以认为是伪造的, 不合法的.

4.4　零水印相似度测试

零水印技术由于未在载体信号中嵌入水印信号, 因此构造的零水印信号应与载体信号的内容高度相关, 这样才能唯一识别载体信号的版权. 从图 6 可以

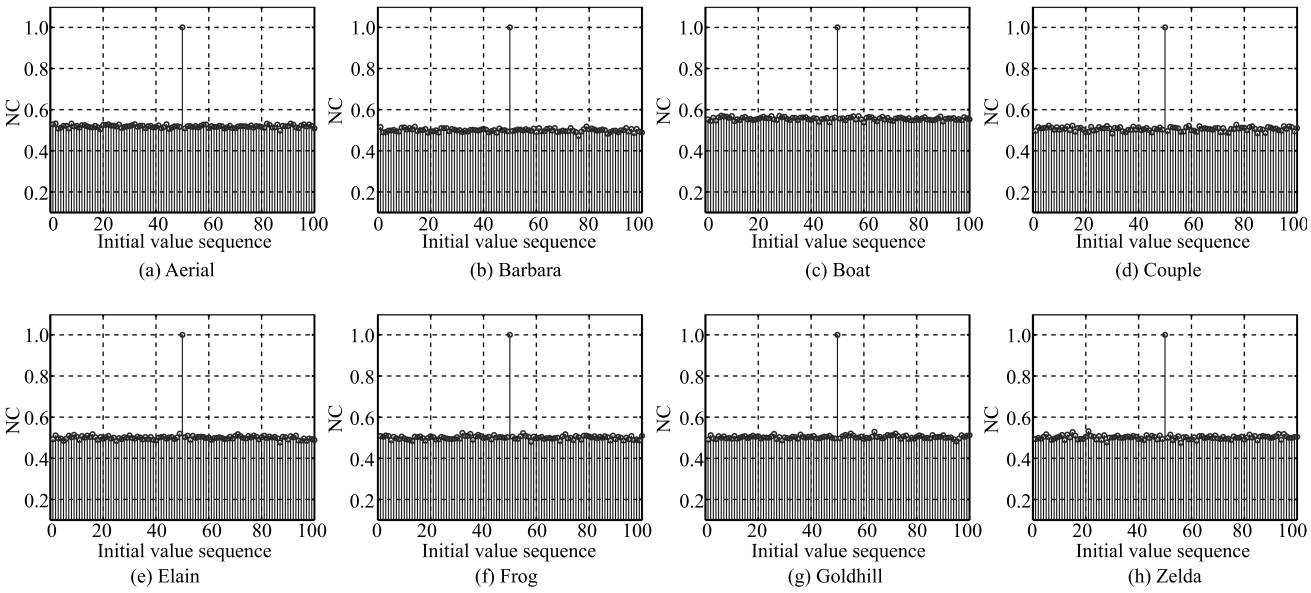


图 6　本文算法安全性测试

Fig. 6　Security testing of the proposed algorithm

表 6 不同载体图像零水印间的相似度

Table 6 Similarities between the generated zero watermarking from different cover images

	Aerial	Barbara	Boat	Couple	Elain	Frog	Goldhill	Zelda
Aerial	1.0000	0.5149	0.5498	0.4900	0.4810	0.4651	0.5513	0.5344
Barbara	0.5149	1.0000	0.5330	0.4878	0.5300	0.4897	0.4944	0.4850
Boat	0.5498	0.5330	1.0000	0.5881	0.4756	0.4734	0.5894	0.4465
Couple	0.4900	0.4878	0.5881	1.0000	0.4880	0.4951	0.5461	0.4292
Elain	0.4810	0.5300	0.4756	0.4880	1.0000	0.4832	0.5659	0.5051
Frog	0.4651	0.4897	0.4734	0.4951	0.4832	1.0000	0.4890	0.4912
Goldhill	0.5513	0.4944	0.5894	0.5461	0.5659	0.4890	1.0000	0.5002
Zelda	0.5344	0.4850	0.4465	0.4292	0.5051	0.4912	0.5002	1.0000

看出, 采用不同的密钥从同一幅载体图像中生成的这些零水印信号的相似度都维持在 0.5 上下波动. 因此, 在计算不同载体图像间生成的零水印的相似度时, 算法中所有的参数都相同, 仅是操作不同的载体图像. 对选择的 8 幅载体图像进行实验测试, 相应的实验结果如表 6 所示. 从表 6 可以看出, 每一幅载体图像生成的零水印信号与载体图像的内容息息相关, 8 幅载体图像生成的零水印信号间的相似度的最大值为 0.5894, 最小值为 0.4292, 平均值为 0.5052, 方差为 0.0016, 表明不同载体图像间生成的零水印信号是不相同的, 具有可辨别性.

此外, 统计了本文算法和文献 [13−16] 四种算法在 8 幅不同载体图像零水印间的相似度, 相应的实验结果如表 7 所示. 从表 7 可以看出, 对于这五种算法, 相似度最大值为 0.6292 (文献 [16]), 最小值为 0.4292 (本文算法); 平均相似度最大值为 0.5044 (文献 [16]), 最小值为 0.4971 (文献 [13]); 方差最大值为 0.0022 (文献 [15−16]), 最小值为 0.0008 (文献 [14]), 在不同载体图像间生成的零水印信号的平均相似度都在 0.5 左右, 表明这五种算法生成的零水印在相似度性能方面都具有较好的性能.

表 7 不同算法零水印间的相似度

Table 7 Similarities between the generated zero watermarking from different algorithms

	本文	[13]	[14]	[15]	[16]
最大值	0.5894	0.5952	0.5798	0.6418	0.6292
最小值	0.4292	0.4900	0.4954	0.5017	0.5266
平均值	0.5052	0.4971	0.5093	0.5040	0.5044
方差	0.0016	0.0022	0.0008	0.0022	0.0020

从表 6、表 7 和均衡性测试结果可以看出, 本文算法生成的零水印信号中的 “0” 和 “1” 的个数基本相等, 近似满足均匀分布. 为了进一步测试不同的载体图像生成的零水印信号与随机生成的二值信号间的相似性, 实验中产生 99 个近似服从均匀分布的伪随机二值信号, 之后计算这些伪随机二值信号与这些载体图像生成的零水印信号的相似度, 相应的实验结果如图 7 所示. 对于图 7 中的各个子图, 第 50 个信号为各个载体图像生成的零水印信号. 从图 7 可以看出, 伪随机产生的二值信号与每一幅载体图像生成的零水印信号间的相似度明显小于 1.0000, 基本上都在 0.5 上下小幅度波动. 综合表 6、表 7 和图 7 的实验结果可以看出, 采用本文算法从一幅载体图像中构造出的零水印信号是可以作为标志此载体图像的版权信息的.

4.5 抗常规信号处理攻击测试

仿真实验中发现, 若 8 幅测试图像都没有受到任何的攻击, 则从每一幅测试图像中提取出来的零水印信号与其原始的零水印信号的相似度 (NC 值) 都为 1.0000. 当受到相应的攻击时, 本文算法也表现出较强的鲁棒性. 为了进一步测试本文算法的性能, 将本文算法与文献 [13−16] 中的零水印算法进行鲁棒性性能比较. 当载体图像受到某种类型的攻击处理时, 本文算法与其他算法相比, 抗攻击性能的平均提高率 AE_v 定义如下:

$$AE_v = \frac{NC_u - NC_v}{NC_v} \times 100\,\% \tag{14}$$

其中, NC_u 和 NC_v ($v \in \{1,2,3,4\}$) 分别表示本文算法和其他四种算法的平均 NC 值.

4.5.1 噪声攻击测试

对选择的 8 幅载体图像分别利用 Matlab 平台中的 $imnoise(\cdot)$ 函数添加均值为 0, 噪声强度不等的椒盐噪声和高斯噪声, 相应的实验结果如表 8 所示 (采用峰值信噪比 (Peak signal to noise ratio, PSNR)[4] 来客观衡量受攻击图像的质量, 表中的数据是 8 幅图像的平均值且对最大值进行了加粗处理).

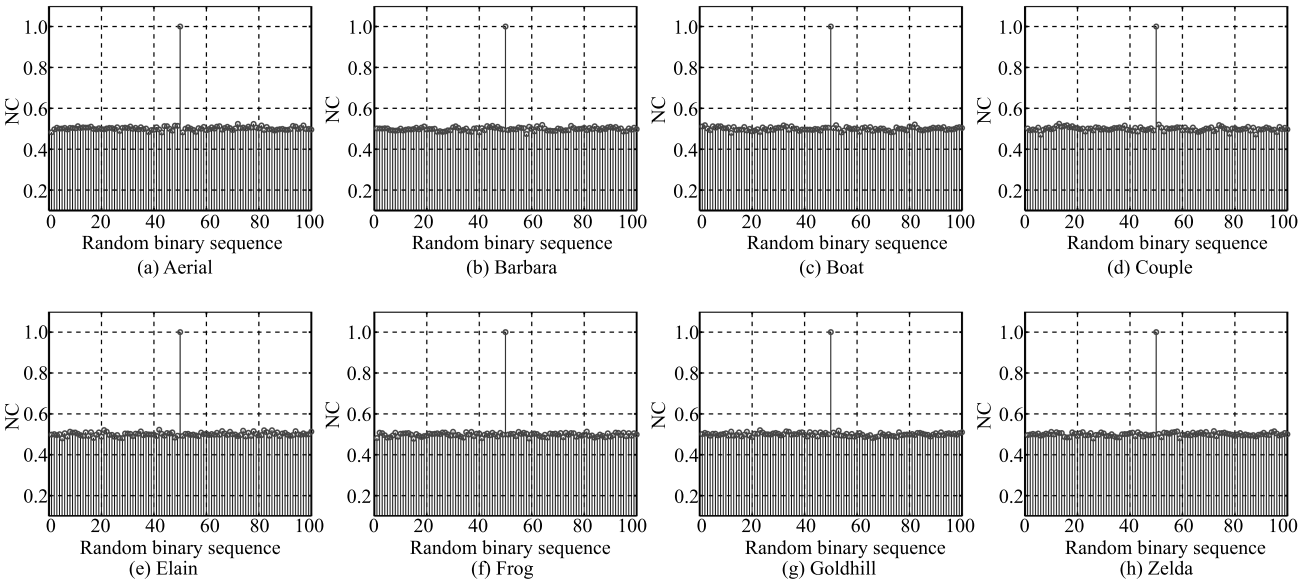


图 7　生成的零水印与伪随机二值信号间的相似度

Fig. 7　Similarities between the generated zero watermarking and random binary signal

从表 8 可以看出, 随着噪声强度的不断增大, 五种算法的 NC 值都有所下降, 但是本文算法的 NC 值也大于 0.82, 具有较强的抗攻击能力. 与文献 [13−16] 中的算法相比, 本文算法的抵抗能力明显优于文献 [13−16]. 就平均性能而言, 对于椒盐噪声攻击, 本文算法分别提高了大约 47 %, 60 %, 23 % 和 14 %; 对于高斯噪声攻击, 本文算法分别提高了大约 21 %, 36 %, 19 % 和 3 %.

表 8　抗噪声攻击实验结果

Table 8　Experimental results against noise attacks

攻击方式	噪声强度	*PSNR*	*NC* 本文	[13]	[14]	[15]	[16]
椒盐噪声	0.1	15.3992	**0.9510**	0.6901	0.6318	0.7859	0.8789
	0.2	12.3662	**0.9239**	0.6235	0.5754	0.7461	0.8213
	0.3	10.6169	**0.8932**	0.5927	0.5458	0.7316	0.7741
	0.4	9.3830	**0.8640**	0.5718	0.5292	0.7040	0.7347
	0.5	8.4076	**0.8289**	0.5594	0.5139	0.6700	0.6960
平均值		11.2345	**0.8922**	0.6075	0.5592	0.7275	0.7810
高斯噪声	0.1	17.1752	**0.9756**	0.7915	0.7084	0.8910	0.9298
	0.2	13.3711	**0.9712**	0.7885	0.7094	0.8769	0.9263
	0.3	10.6398	**0.9550**	0.7893	0.7016	0.8271	0.9207
	0.4	8.7613	**0.9241**	0.7731	0.6821	0.7332	0.9067
	0.5	7.4853	**0.8804**	0.7500	0.6617	0.6197	0.8783
平均值		11.4865	**0.9413**	0.7785	0.6927	0.7896	0.9124

4.5.2　滤波攻击测试

对选择的 8 幅载体图像分别利用 Matlab 平台中的 $medfilt2(\cdot)$ 函数和 $wiener2(\cdot)$ 函数进行中值滤波和维纳滤波攻击实验测试, 8 幅图像的平均实验结果如表 9 所示.

从表 9 可以看出, 对于所有的中值滤波和维纳滤波攻击, 与文献 [13−16] 中的算法相比, 本文算法的抗攻击能力都是最好的. 就平均性能而言, 对于中值滤波攻击, 本文算法分别提高了大约 12 %, 20 %, 4 % 和 3 %; 对于维纳滤波攻击, 本文算法分别提高了大约 8 %, 17 %, 4 % 和 3 %.

表 9　抗滤波攻击实验结果

Table 9　Experimental results against filtering attacks

攻击方式	窗口大小	*PSNR*	*NC* 本文	[13]	[14]	[15]	[16]
中值滤波	3 × 3	30.0745	**0.9922**	0.9132	0.9006	0.9678	0.9816
	5 × 5	27.4016	**0.9846**	0.8826	0.8505	0.9503	0.9650
	7 × 7	26.1452	**0.9754**	0.8652	0.8089	0.9353	0.9482
	9 × 9	25.1367	**0.9669**	0.8495	0.7746	0.9226	0.9335
	11 × 11	24.3752	**0.9590**	0.8348	0.7429	0.9106	0.9207
平均值		26.6266	**0.9756**	0.8691	0.8155	0.9373	0.9498
维纳滤波	3 × 3	33.3994	**0.9959**	0.9417	0.9211	0.9733	0.9878
	5 × 5	30.7947	**0.9915**	0.9237	0.8762	0.9600	0.9745
	7 × 7	29.3403	**0.9855**	0.9094	0.8373	0.9469	0.9598
	9 × 9	28.2680	**0.9790**	0.8965	0.8062	0.9335	0.9455
	11 × 11	27.4357	**0.9710**	0.8876	0.7792	0.9228	0.9346
平均值		29.8476	**0.9846**	0.9118	0.8440	0.9473	0.9605

4.5.3　JPEG 压缩攻击测试

对选择的 8 幅载体图像分别进行 JPEG 压缩攻击实验测试, 8 幅图像的平均实验结果如表 10 所示.

表 10 抗 JPEG 压缩攻击实验结果

Table 10 Experimental results against JPEG compression attacks

品质百分数 (%)	*PSNR*	*NC*				
		本文	[13]	[14]	[15]	[16]
5	25.5419	**0.9559**	0.8640	0.6803	0.8941	0.9054
10	27.9958	**0.9856**	0.8998	0.7787	0.9364	0.9513
15	29.3245	**0.9837**	0.9138	0.8188	0.9498	0.9661
20	30.2552	**0.9866**	0.9227	0.8468	0.9571	0.9727
25	30.9669	**0.9861**	0.9274	0.8721	0.9607	0.9780
30	31.5407	**0.9955**	0.9317	0.8845	0.9651	0.9813
35	32.0416	**0.9932**	0.9358	0.8966	0.9656	0.9835
40	32.4319	**0.9922**	0.9381	0.9019	0.9680	0.9864
45	32.8186	**0.9941**	0.9424	0.9135	0.9688	0.9870
50	33.1562	**0.9952**	0.9431	0.9221	0.9706	0.9879
平均值	30.6073	**0.9868**	0.9219	0.8515	0.9536	0.9700

从表 10 可以看出, 本文算法对 JPEG 压缩具有较好的抗攻击能力. 与文献 [13−16] 中的算法相比, 文献 [14] 的鲁棒性能较差, 本文算法和文献 [16] 中的算法性能较好. 就平均性能而言, 对于 JPEG 压缩攻击, 本文算法分别提高了大约 7 %, 16 %, 3 % 和 2 %.

4.5.4 常规图像处理组合攻击测试

对选择的 8 幅载体图像分别进行不同的常规图像处理组合攻击实验测试, 8 幅图像的平均实验结果如表 11 所示. 从表 11 可以看出, 本文算法对这些组合攻击都具有较好的抵抗性能和具有比文献 [13−16] 更强的抗攻击能力. 就平均性能而言, 对于这些组合攻击, 本文算法分别提高了大约 22 %, 36 %, 11 % 和 7 %.

4.6 抗几何攻击测试

4.6.1 偏移行列攻击测试

对选择的 8 幅载体图像分别进行偏移行列攻击实验测试, 8 幅图像的平均实验结果如表 12 所示. 在表 12 中, 右偏移 2 列是指整个载体图像向右平移 2 列, 最右的 2 列像素丢失, 最左的 2 列像素全设置为 0. 左偏移 2 列、上偏移 2 行及下偏移 2 行的处理过程与右偏移 2 列的基本过程大致相同, 只是方向不同而已. 从表 12 可以看出, 其他四种算法的抗攻击性能都优于文献 [13] 中的算法. 就平均性能而言, 对于偏移行列攻击, 本文算法分别提高了大约 23 %, 16 %, 5 % 和 4 %.

4.6.2 偏移行列组合攻击测试

对选择的 8 幅载体图像分别进行偏移行列组合攻击性能测试, 8 幅图像的平均实验结果如表 13 所示. 从表 13 可以看出, 对于这些组合攻击, 五种算法的抵抗能力都有所下降, 但是提取的零水印信号与原始的零水印信号的相似度仍然较高, 其他四种算法的性能都强于文献 [13] 中的算法. 就平均性能而言, 对于这些组合攻击, 本文算法分别提高了大约 24 %, 23 %, 7 % 和 6 %.

4.6.3 缩放攻击测试

利用 Matlab 平台中的 $imresize(\cdot)$ 函数对选择

表 11 抗常规图像处理组合攻击实验结果

Table 11 Experimental results against common image processing combination attacks

攻击方式	*PSNR*	*NC*				
		本文	[13]	[14]	[15]	[16]
中值滤波 (5 × 5) + 椒盐噪声 (0.3)	10.5309	**0.8901**	0.5784	0.5299	0.7235	0.7685
中值滤波 (5 × 5) + 高斯噪声 (0.3)	10.4693	**0.9501**	0.7434	0.6449	0.8141	0.9107
维纳滤波 (5 × 5) + 椒盐噪声 (0.3)	10.5801	**0.8926**	0.5836	0.5341	0.7244	0.7699
维纳滤波 (5 × 5) + 高斯噪声 (0.3)	10.5371	**0.9566**	0.7641	0.6583	0.8242	0.9139
中值滤波 (5 × 5) + JPEG 压缩 (10)	26.1020	**0.9762**	0.8676	0.7212	0.9219	0.9341
维纳滤波 (5 × 5) + JPEG 压缩 (10)	27.1870	**0.9819**	0.8835	0.7425	0.9264	0.9429
JPEG 压缩 (10) + 椒盐噪声 (0.3)	10.5540	**0.8929**	0.5866	0.5481	0.7268	0.7714
JPEG 压缩 (10) + 高斯噪声 (0.3)	10.5560	**0.9551**	0.7610	0.6920	0.8228	0.9099
JPEG 压缩 (10) + 放大 2 倍 + 缩小 0.5 倍	28.3582	**0.9856**	0.9039	0.8112	0.9417	0.9555
逆时针旋转 2 度 + JPEG 压缩 (10)	17.6329	**0.8628**	0.6854	0.6457	0.8010	0.8060
平均值	18.8576	**0.9356**	0.7641	0.6879	0.8438	0.8772

表 12 抗偏移行列攻击实验结果

Table 12 Experimental results against row and column shifting attacks

攻击方式	*PSNR*	*NC*				
		本文	[13]	[14]	[15]	[16]
右偏移 2 列	21.2879	**0.9489**	0.7796	0.8425	0.9056	0.9129
左偏移 2 列	21.4324	**0.9497**	0.7831	0.8425	0.9068	0.9118
上偏移 2 行	21.9757	**0.9522**	0.7966	0.8473	0.9172	0.9237
下偏移 2 行	21.6893	**0.9504**	0.7897	0.8445	0.9104	0.9198
右偏移 2 列 + 上偏移 2 行	19.8069	**0.9271**	0.7444	0.7919	0.8827	0.8882
左偏移 2 列 + 上偏移 2 行	19.8413	**0.9220**	0.7468	0.7906	0.8792	0.8862
右偏移 2 列 + 下偏移 2 行	19.5717	**0.9239**	0.7410	0.7882	0.8728	0.8807
左偏移 2 列 + 下偏移 2 行	19.7364	**0.9248**	0.7423	0.7910	0.8747	0.8837
平均值	20.1291	**0.9296**	0.7528	0.8012	0.8840	0.8917

表 13 抗偏移行列组合攻击实验结果

Table 13 Experimental results against row and column shifting combination attacks

攻击方式	*PSNR*	*NC*				
		本文	[13]	[14]	[15]	[16]
右偏移 2 列 + 上偏移 2 行 + 逆时针旋转 2 度	17.1498	**0.8561**	0.6787	0.6987	0.8225	0.8287
左偏移 2 列 + 上偏移 2 行 + 逆时针旋转 2 度	17.1585	**0.8554**	0.6760	0.7019	0.7764	0.7808
右偏移 2 列 + 下偏移 2 行 + 放大 2 倍 + 缩放 0.5 倍	19.9653	**0.9242**	0.7479	0.7871	0.8733	0.8811
左偏移 2 列 + 下偏移 2 行 + 缩放 0.5 倍 + 放大 2 倍	20.9519	**0.9254**	0.7674	0.7762	0.8727	0.8839
平均值	18.8166	**0.8993**	0.7268	0.7304	0.8378	0.8503

的 8 幅载体图像分别进行抗缩放攻击实验测试, 8 幅图像的平均实验结果如表 14 所示 (在计算最后的平均 $PSNR$ 值时, 因最后的两个值为 $+\infty$, 故未参与计算). 从表 14 可以看出, 对载体图像先进行缩放 x 倍, 再缩放 $1/x$ 倍使图像恢复到原始的大小后, 从中构造的零水印信号与原始的零水印信号的相似度都较高, 这五种零水印算法对这类尺寸缩放攻击都具有较好的鲁棒性能. 无论采用哪种插值方法, 本文算法都具有更强的抗攻击能力. 就平均性能而言, 对于先缩放 x 倍再缩放 $1/x$ 倍的缩放攻击, 本文算法分别提高了大约 7 %, 10 %, 3 % 和 2 %.

若在进行缩放攻击时, 仅采取缩小或放大 x 倍的缩放方式, 由于采用这种方式处理后的载体图像大小与原始图像的大小不一致. 因此, 为能构造 4 096 比特的零水印信号, 需对载体图像分块的大小进行调整. 一般地, 若受到攻击后的图像大小为 $M_1 \times N_1$, 要生成的零水印信号大小为 $H_1 \times K_1$, 则可将分块的大小设置为 $\lfloor M_1/H_1 \rfloor \times \lfloor N_1/K_1 \rfloor$, 这样可确保生成的零水印信号长度正好为 $H_1 \times K_1$ (每个分块仅生成一比特的零水印信号). 当然, 也可以选择比 $\lfloor M_1/H_1 \rfloor \times \lfloor N_1/K_1 \rfloor$ 小的分块, 随后再随机选择足够多的分块来生成零水印信号也是可以的. 对 8 幅载体图像仅进行放大或缩小攻击的平均实验结果如表 15 所示. 从表 15 可以看出, 无论采用哪种插值方法, 本文算法的性能都较好. 就平均性能而言, 对于仅缩小或放大 x 倍的缩放攻击, 本文算法分别提高了大约 53 %, 5 %, 2 % 和 1 %.

4.6.4 旋转攻击测试

利用 Matlab 平台中的 $imrotate(\cdot)$ 函数 (bbox = crop) 对选择的 8 幅载体图像分别进行抗旋转攻击实验测试, 8 幅图像的平均实验结果如表 16 所示 (对于表 16 中的攻击方式 x 度和 $-x$ 度, 分别表示对载体图像进行逆时针和顺时针旋转 x 度). 从表 16 可以看出, 在旋转相同角度的条件下, 逆时针旋转和顺时针旋转得到的结果基本相同, 且随着旋转角度的增大, 五种算法的性能都有所下降, 但是对于所有的攻击, 无论采用哪种插值方法, 本文算法的性能都是最好的. 但需注意的是, 因本文算法未对载体图像进行任何的变换操作,使得本文算法不能抵抗大角度

表 14 抗缩放攻击实验结果

Table 14 Experimental results against scaling attacks

插值方法	攻击方式	*PSNR*	*NC*				
			本文	[13]	[14]	[15]	[16]
bilinear	缩小 0.25 倍 + 放大 4 倍	25.4763	**0.9786**	0.8584	0.8108	0.9246	0.9475
	缩小 0.5 倍 + 放大 2 倍	28.3491	**0.9917**	0.8994	0.8825	0.9492	0.9735
	放大 4 倍 + 缩小 0.25 倍	34.2862	**0.9976**	0.9483	0.9454	0.9742	0.9898
	放大 2 倍 + 缩小 0.5 倍	33.5935	**0.9972**	0.9433	0.9397	0.9725	0.9891
bicubic	缩小 0.25 倍 + 放大 4 倍	26.4544	**0.9910**	0.8710	0.8348	0.9399	0.9663
	缩小 0.5 倍 + 放大 2 倍	29.9015	**0.9968**	0.9129	0.9123	0.9626	0.9875
	放大 4 倍 + 缩小 0.25 倍	39.3801	**0.9989**	0.9710	0.9699	0.9859	0.9954
	放大 2 倍 + 缩小 0.5 倍	39.0522	**0.9987**	0.9702	0.9671	0.9855	0.9950
nearest	缩小 0.25 倍 + 放大 4 倍	23.1304	**0.9564**	0.8201	0.6959	0.9019	0.9089
	缩小 0.5 倍 + 放大 2 倍	25.6881	**0.9790**	0.8653	0.8456	0.9430	0.9522
	放大 4 倍 + 缩小 0.25 倍	$+\infty$	**1.0000**	1.0000	1.0000	1.0000	1.0000
	放大 2 倍 + 缩小 0.5 倍	$+\infty$	**1.0000**	1.0000	1.0000	1.0000	1.0000
	平均值	30.5312	**0.9905**	0.9217	0.9003	0.9616	0.9754

表 15 抗仅缩小或放大缩放攻击实验结果

Table 15 Experimental results against only reduce/enlarge scaling attacks

插值方法	攻击方式	*NC*				
		本文	[13]	[14]	[15]	[16]
bilinear	缩小 0.5 倍	**0.9963**	0.6365	0.9079	0.9587	0.9856
	缩小 2 倍	**0.9982**	0.7503	0.9607	0.9818	0.9925
	放大 4 倍	**0.9982**	0.5657	0.9651	0.9837	0.9928
bicubic	缩小 0.5 倍	**0.9976**	0.6371	0.9202	0.9643	0.9896
	放大 2 倍	**0.9989**	0.7507	0.9795	0.9914	0.9965
	放大 4 倍	**0.9988**	0.5658	0.9791	0.9906	0.9960
nearest	缩小 0.5 倍	**0.9790**	0.6202	0.8456	0.9432	0.9522
	放大 2 倍	**1.0000**	0.7513	1.0000	1.0000	1.0000
	放大 4 倍	**1.0000**	0.5693	0.9999	1.0000	1.0000
平均值		**0.9963**	0.6497	0.9509	0.9793	0.9895

的旋转攻击. 例如, 当采用 bilinear 插值方法分别进行逆时针旋转 10° 和 30° 时, 提取的 NC 值为 0.6633 和 0.5582, 此时提取的零水印信号人眼已不可识别. 就平均性能而言, 对于旋转攻击, 本文算法分别提高了大约 25 %, 17 %, 5 % 和 5 %.

4.7 性能提高情况

与文献 [13−16] 中的算法相比, 在抗攻击性能方面, 对于文中选择的 8 幅载体图像, 本文算法的平均性能提高情况如表 17 所示 (表中数据基于表 8 ∼16 的分析结果). 从表 17 可以看出, 对于所有的攻击, 本文算法分别提高了大约 23 %, 23 %, 8 % 和 5 %, 平均性能提高了大约 15 %.

4.8 普适性测试

从文中选择的 8 幅载体图像的实验结果来看, 本文算法与文献 [13−16] 相比, 具有更强的抗攻击能力. 为进一步测试本文算法与文献 [13−16] 的鲁棒性能, 从 SIPI 图像数据集[19] (选择大小为 512 像素 × 512 像素和 1 024 像素 × 1 024 像素的图像, 共 131 幅) 中随机选择 100 幅图像进行实验, 100 幅图像对每一种攻击类型 (每一种攻击类型的攻击方式见表 8 ∼ 14 和表 16 的说明) 的平均实验结果如表 18 所示. 从表 18 可以看出, 对于所有的攻击类型, 本文算法具有比文献 [13−16] 更优越的鲁棒性能. 就平均性能而言, 对于所有的攻击类型, 本文算法分别提高了约 18 %, 35 %, 8 % 和 4 %, 平均性能提高了约 16 %.

表 19 统计了随机选择的 100 幅载体图像对于每一种攻击的 NC 值的方差的平均值 (表中值最小的那一项对其进行了加粗处理), 从表 19 可以看出, 对于随机选择的 100 幅图像, 五种算法对于每一种攻击在不同图像的 NC 值的方差都较小, 表明对于同一种攻击, 不同图像中 NC 值的离散程度较小. 就平均值而言, 本文算法的方差是最小的 (对绝大多数的攻击类型, 本文算法的方差也是最小的).

综合上述实验结果表明, 本文算法对噪声、滤波、偏移行列、尺寸缩放、旋转和多种组合攻击等都具有较强的抗攻击能力. 与文献 [13−16] 中的算法相

表 16　抗旋转攻击实验结果

Table 16　Experimental results against rotation attacks

插值方法	攻击方式	*PSNR*	*NC* 本文	[13]	[14]	[15]	[16]
bilinear	1°	20.0928	**0.9209**	0.7514	0.7812	0.8750	0.8809
	−1°	20.2172	**0.9216**	0.7476	0.7753	0.8781	0.8812
	3°	16.1739	**0.8209**	0.6407	0.6640	0.7670	0.7680
	−3°	16.3061	**0.8223**	0.6172	0.6645	0.7720	0.7713
	5°	14.5964	**0.7571**	0.5945	0.6242	0.7097	0.7081
	−5°	14.7210	**0.7607**	0.5645	0.6264	0.7191	0.7177
	10°	12.8664	**0.6633**	0.5369	0.5844	0.6292	0.6315
	−10°	12.9698	**0.6709**	0.5140	0.5825	0.6436	0.6416
	30°	10.8797	**0.5582**	0.4880	0.5235	0.5487	0.5541
	−30°	10.9389	**0.5634**	0.5093	0.5236	0.5525	0.5489
bicubic	1°	19.8760	**0.9208**	0.7466	0.7815	0.8739	0.8797
	−1°	19.9959	**0.9211**	0.7437	0.7773	0.8776	0.8802
	3°	16.0703	**0.8205**	0.6376	0.6664	0.7655	0.7672
	−3°	16.1995	**0.8221**	0.6147	0.6667	0.7710	0.7702
	5°	14.5213	**0.7573**	0.5934	0.6247	0.7083	0.7072
	−5°	14.6440	**0.7610**	0.5623	0.6280	0.7190	0.7171
	10°	12.8152	**0.6631**	0.5359	0.5847	0.6290	0.6313
	−10°	12.9175	**0.6708**	0.5147	0.5847	0.6434	0.6410
	30°	10.8483	**0.5582**	0.4877	0.5245	0.5488	0.5538
	−30°	10.9075	**0.5630**	0.5083	0.5239	0.5520	0.5491
nearest	1°	19.6571	**0.9211**	0.7445	0.7818	0.8752	0.8796
	−1°	19.7749	**0.9214**	0.7403	0.7780	0.8779	0.8821
	3°	15.9727	**0.8203**	0.6365	0.6663	0.7640	0.7680
	−3°	16.0995	**0.8221**	0.6150	0.6668	0.7713	0.7717
	5°	14.4531	**0.7569**	0.5928	0.6267	0.7071	0.7080
	−5°	14.5743	**0.7610**	0.5625	0.6247	0.7180	0.7171
	10°	12.7704	**0.6636**	0.5345	0.5843	0.6281	0.6309
	−10°	12.8715	**0.6705**	0.5141	0.5836	0.6425	0.6421
	30°	10.8207	**0.5579**	0.4895	0.5212	0.5493	0.5539
	−30°	10.8800	**0.5634**	0.5089	0.5224	0.5530	0.5486
平均值		14.8811	**0.7458**	0.5949	0.6356	0.7090	0.7101

比, 具有更优越的鲁棒性能.

4.9　构造零水印信息时间测试

文献 [13−16] 和本文算法的测试环境都为 Windows 7 操作系统 (旗舰版, Service Pack 1) 和 Matlab R2010a 平台, 测试电脑的 CPU 为 Intel (R) Core (TM) i5−4200U, 主频为 1.60 GHz 和 2.30 GHz, 内存容量 4.00 GB (2.45 GB 可用), 硬盘容量 500 GB. 本文算法直接在空域中进行, 文献 [13] 在小波域中进行, 文献 [15] 在离散小波变换和奇异值分解变换域中进行, 文献 [14] 和文献 [16] 在奇异值分解变换域中进行. 相同条件下, 在每一幅图像中构造零水印信号所需的平均时间如表 20 所示 (对于每一幅载体图像, 算法都是运行 10 次, 之后取其平均值作为该幅载体图像的平均运行时间).

为便于比较本文算法与其他四种算法的运行时间性能, 定义平均节省率 JS_v 如下:

$$JS_v = \frac{RT_v - RT_u}{RT_v} \times 100\,\% \tag{15}$$

表 17　本文算法与其他算法抗攻击性能的提高率 (%)

Table 17　Improvement performance against attacks compared this algorithm with other algorithms (%)

攻击方式	[13]	[14]	[15]	[16]
椒盐噪声	47	60	23	14
高斯噪声	21	36	19	3
中值滤波	12	20	4	3
维纳滤波	8	17	4	3
JPEG 压缩	7	16	3	2
常规信号组合	22	36	11	7
偏移行列	23	16	5	4
偏移行列组合	24	23	7	6
先缩放 x 倍, 再缩放 $1/x$ 倍	7	10	3	2
仅缩小或放大 x 倍	53	5	2	1
旋转	25	17	5	5
平均值	23	23	8	5

表 18　不同算法在 SIPI 图像数据集的实验结果

Table 18　Experimental results on the SIPI image database from different algorithms

攻击方式	*PSNR*	本文	[13]	[14]	[15]	[16]
椒盐噪声	11.0086	**0.8587**	0.6164	0.5690	0.7542	0.7940
高斯噪声	12.2500	**0.9170**	0.7599	0.6611	0.7533	0.9042
中值滤波	24.8580	**0.9609**	0.8411	0.6904	0.9160	0.9365
维纳滤波	27.5159	**0.9697**	0.8878	0.7106	0.9328	0.9484
JPEG 压缩	29.5265	**0.9770**	0.9091	0.7514	0.9456	0.9663
常规信号组合	15.6869	**0.8959**	0.7145	0.6211	0.7929	0.8560
偏移行列	19.8269	**0.9085**	0.7460	0.6755	0.8703	0.8756
偏移行列组合	17.5026	**0.8222**	0.6688	0.6184	0.7712	0.7743
缩放	29.0496	**0.9843**	0.9105	0.7622	0.9536	0.9721
旋转	13.7218	**0.6798**	0.5567	0.5665	0.6422	0.6416
平均值	20.0947	**0.8974**	0.7611	0.6626	0.8332	0.8669

表 19 不同算法在 SIPI 图像数据集实验结果的方差
Table 19 The variance of experimental results on the SIPI image database from different algorithms

攻击方式	本文	[13]	[14]	[15]	[16]
椒盐噪声	**0.0026**	0.0030	0.0077	0.0051	0.0034
高斯噪声	**0.0025**	0.0055	0.0205	0.0338	0.0030
中值滤波	**0.0009**	0.0043	0.0242	0.0022	0.0020
维纳滤波	**0.0008**	0.0026	0.0260	0.0016	0.0015
JPEG 压缩	0.0006	0.0011	0.0405	0.0008	**0.0005**
常规信号组合	**0.0033**	0.0034	0.0143	0.0182	0.0034
偏移行列	**0.0040**	0.0066	0.0180	0.0072	0.0065
偏移行列组合	0.0085	**0.0053**	0.0100	0.0106	0.0107
缩放	**0.0002**	0.0012	0.0401	0.0007	0.0004
旋转	0.0085	**0.0041**	0.0050	0.0080	0.0081
平均值	**0.0032**	0.0037	0.0206	0.0088	0.0040

表 20 不同算法构造零水印运行时间 (s)
Table 20 The running time for constructing zero watermarking from different algorithms (s)

	本文	[13]	[14]	[15]	[16]
Aerial	0.0343	0.5554	0.3541	6.7782	0.1607
Barbara	0.0328	0.5429	0.3494	6.7424	0.1591
Boat	0.0374	0.5429	0.3510	6.8297	0.1560
Couple	0.0328	0.5476	0.3479	6.8282	0.1560
Elain	0.0328	0.5491	0.3510	6.8079	0.1607
Frog	0.0312	0.5491	0.3416	6.7814	0.1513
Goldhill	0.0343	0.5444	0.3494	6.7814	0.1576
Zelda	0.0328	0.5600	0.3557	6.8172	0.1622
平均时间	**0.0335**	0.5489	0.3500	6.7958	0.1580

其中, RT_u 和 RT_v $(v \in \{1,2,3,4\})$ 分别表示本文算法和其他四种算法的平均运行时间. 从表 20 可以看出, 对于这些测试图像, 本文算法构造零水印信号的平均运行时间仅为 0.0335 s. 与文献 [13−16] 中的算法相比, 本文算法的平均运行时间明显小于文献 [13−16] 中的算法, 分别节省了约 93 %, 90 %, 99 % 和 78 % 的时间, 平均节省了约 90 %.

5 结论

与常见的鲁棒零水印技术通常在变换域中构造零水印信号不同, 本文在分析分块均值与载体图像所有选择分块整体均值间大小关系具有较强稳健性能的基础上, 提出了一种基于混沌的空域强鲁棒零水印方案, 具有如下的特点:

1) 选择有意义的二值图像作为原始水印信号, 解决了提取的水印信号缺乏可视性的问题;

2) 在零水印信号构造和检测过程中, 未对载体图像进行任何的变换, 而是直接在空域利用载体图像所有选择分块整体均值与分块均值间的大小关系来构造稳健的图像特征信息, 没有对载体图像进行变换的计算复杂性, 降低了算法的计算复杂度;

3) 算法直接在空域进行, 比较简单且易于实现, 可操作性强;

4) 算法以对初值极度敏感的混沌系统为基础, 利用对混沌信号排序生成的索引作为选取图像分块的次序, 算法安全性高;

5) 算法将混沌加密和 Arnold 空间置乱技术引入到原始水印信号的预处理和生成的零水印信号的后期处理中, 进一步提高了算法的安全性能;

6) 生成的零水印信号满足或近似满足均匀分布, 相当于伪随机二值信号, 其均衡性较好;

7) 对生成的零水印信号进行注册时, 附加了经过权威机构认证的时间戳, 这样即使侵权者后来据本文算法生成了代表他的零水印信号也注册在 IPR 数据库中, 也是不能证明其版权是属于侵权者的;

8) 算法采用了零水印技术, 确保原始的载体图像没有被任何信息干扰, 适用于对载体图像质量要求较高的作品版权保护应用场合.

大量的仿真实验结果表明, 本文算法对噪声、滤波、行列偏移、尺寸缩放、旋转和多种组合攻击等都具有较强的鲁棒性能. 与文献 [13−16] 中的鲁棒零水印算法相比, 本文算法的平均运行时间不仅减少了约 90 %, 而且抗攻击性能平均提高了约 15 %, 具有计算复杂度更低、实现更简单和抗攻击性能更强的特点.

References

1 Qi Ke, Xie Dong-Qing. Watermarking scheme against geometrical attacks based on second generation Bandelet. *Acta Automatica Sinica*, 2012, **38**(10): 1646−1653
(綦科, 谢冬青. 基于第二代 Bandelet 变换的抗几何攻击图像水印. 自动化学报, 2012, **38**(10): 1646−1653)

2 Lin Xiao-Dan. DCT-domain watermark detection using Gaussian mixture model. *Acta Automatica Sinica*, 2012, **38**(9): 1445−1448
(林晓丹. 基于高斯混合模型的 DCT 域水印检测方法. 自动化学报, 2012, **38**(9): 1445−1448)

3 Cui Han-Guo, Liu Jian-Xin, Li Zheng-Min. STL model watermarking algorithm based on pyramid technique. *Acta Automatica Sinica*, 2013, **39**(6): 852−860
(崔汉国, 刘健鑫, 李正民. 基于金字塔技术的 STL 模型数字水印算法. 自动化学报, 2013, **39**(6): 852−860)

4 Xiong Xiang-Guang, Wei Li, Xie Gang. A robust color image watermarking algorithm based on 3D-DCT and SVD. *Computer Engineering and Science*, 2015, **37**(6): 1039−1100
(熊祥光, 韦立, 谢刚. 基于 3D-DCT 和 SVD 的鲁棒彩色图像水印算法. 计算机工程与科学, 2015, **37**(6): 1039−1100)

5 Swaminathan A, Mao Y N, Wu M. Robust and secure image hashing. *IEEE Transaction on Information Forensics and Security*, 2006, **1**(2): 215−230

6 Niu Xia-Mu, Jiao Yu-Hua. An overview of perceptual hashing. *Acta Electronica Sinica*, 2008, **36**(7): 1405−1411
(牛夏牧, 焦玉华. 感知哈希综述. 电子学报, 2008, **36**(7): 1405−1411)

7 Wen Quan, Sun Tan-Feng, Wang Shu-Xun. Concept and application of zero-watermark. *Acta Electronica Sinica*, 2003, **31**(2): 214−216
(温泉, 孙锬锋, 王树勋. 零水印的概念与应用. 电子学报, 2003, **31**(2): 214−216)

8 Ye Tian-Yu, Ma Zhao-Feng, Niu Xin-Xin, Yang Yi-Xian. A zero-watermark technology with strong robustness. *Journal of Beijing University of Posts and Telecommunications*, 2010, **33**(3): 126−129
(叶天语, 马兆丰, 钮心欣, 杨义先. 强鲁棒零水印技术. 北京邮电大学学报, 2010, **33**(3): 126−129)

9 Jin Wei, Li Jin-Xiang, Yin Cao-Qian. An image zero-watermarking scheme based on visual cryptography utilizing contour-wavelet. *Journal of Optoelectronics · Laser*, 2009, **20**(5): 653−656
(金炜, 励金祥, 尹曹谦. 一种基于可视密码的轮廓小波图像零水印方案. 光电子 · 激光, 2009, **20**(5): 653−656)

10 Zhao Chun-Hui, Liu Wei. Block compressive sensing based image semi-fragile zero-watermarking algorithm. *Acta Automatica Sinica*, 2012, **38**(4): 609−617
(赵春晖, 刘巍. 基于分块压缩感知的图像半脆弱零水印算法. 自动化学报, 2012, **38**(4): 609−617)

11 Fu Jian-Jing, Wang Ke. Image zero-watermark based on direction stability of first principal component vector. *Journal of Image and Graphics*, 2012, **17**(7): 756−769
(付剑晶, 王珂. 基于第一主成分方向稳定性的图像零水印. 中国图象图形学报, 2012, **17**(7): 756−769)

12 Zhou Wu-Jie, Yu Mei, Yu Si-Min, Jiang Gang-Yi, Ge Ding-Fei. A zero-watermarking algorithm of stereoscopic image based on hyperchaotic system. *Acta Physica Sinica*, 2012, **61**(8): 080701
(周武杰, 郁梅, 禹思敏, 蒋刚毅, 葛丁飞. 一种基于超混沌系统的立体图像零水印算法. 物理学报, 2012, **61**(8): 080701)

13 Qu Chang-Bo, Yang Xiao-Tao, Yuan Duo-Ning. Zero-watermarking visual cryptography algorithm in the wavelet domain. *Journal of Image and Graphics*, 2014, **19**(3): 365−372
(曲长波, 杨晓陶, 袁铎宁. 小波域视觉密码零水印算法. 中国图象图形学报, 2014, **19**(3): 365−372)

14 Ye Tian-Yu. A robust zero-watermarking algorithm using variance in singular value decomposition domain. *Acta Photonica Sinica*, 2011, **40**(6): 961−966
(叶天语. 基于方差的奇异值分解域鲁棒零水印算法. 光子学报, 2011, **40**(6): 961−966)

15 Ye Tian-Yu. A robust zero-watermarking algorithm resisting JPEG compression and geometric attacks. *Acta Photonica Sinica*, 2012, **41**(2): 210−217
(叶天语. 抗 JPEG 压缩和几何攻击的鲁棒零水印算法. 光子学报, 2012, **41**(2): 210−217)

16 Song Wei, Hou Jian-Jun, Li Zhao-Hong, Huang Liang. A novel zero-watermarking algorithm based on Logistic chaotic system and singular value decomposition. *Acta Physica Sinica*, 2009, **58**(7): 4449−4456
(宋伟, 侯建军, 李赵红, 黄亮. 一种基于 Logistic 混沌系统和奇异值分解的零水印算法. 物理学报, 2009, **58**(7): 4449−4456)

17 Rani A, Bhullar A K, Dangwal D, Kumar S. A zero-watermarking scheme using discrete wavelet transform. *Procedia Computer Science*, 2015, **70**: 603−609

18 Sun L, Xu J C, Zhang X X, Dong W, Tian Y. A novel generalized Arnold transform-based zero-watermarking scheme. *Applied Mathematics and Information Sciences*, 2015, **9**(4): 2023−2035

19 The USC-SIPI image database [Online], available: http://sipi.usc.edu/database/, May 31, 2016

20 UCID-uncompressed colour image database [Online], available: http://homepages.lboro.ac.uk/~cogs/datasets/UCID/ucid.html, May 31, 2016

21 Kodak lossless true color image suite [Online], available: http://r0k.us/graphics/kodak/, May 31, 2016

熊祥光 贵州师范大学大数据与计算机科学学院副教授. 主要研究方向为多媒体信息安全和数字水印技术.
E-mail: xxg0851@163.com
(**XIONG Xiang-Guang** Associate professor at the School of Big Data and Computer Science, Guizhou Normal University. His research interest covers multimedia information security and digital watermarking technology.)