文章编号:1001 - 9081(2011)04 - 0966 - 04

doi:10.3724/SP. J. 1087. 2011. 00966

精确的图像篡改定位与恢复的三水印算法

周　燕,曾凡智,卓仰俱

(佛山科学技术学院 机电与信息工程学院,广东 佛山 528000)

(zhouyan791266@163.com)

摘　要:针对目前图像篡改定位与恢复的水印算法在篡改定位精度和篡改恢复性能方面存在的不足,提出一种精确的图像篡改定位与恢复的三水印算法。该算法在最低有效位(LSB)方法的基础上,采用二进制编码的方式生成检测水印、定位水印和恢复水印等三种水印,并嵌入到图像的低位。篡改检测和篡改恢复采用基于分块的检测水印和恢复水印,篡改精确定位采用基于单像素的定位水印。仿真实验表明,该算法对任意大小的亮度图像、RGB 图像具有高精度的篡改定位能力,并且具有很好的篡改恢复性能。

关键词:三水印算法;篡改检测;篡改定位;篡改恢复;二进制编码

中图分类号: TP309.2　　**文献标志码**:A

Precise three-watermarking algorithm for image tamper localization and recovery

ZHOU Yan, ZENG Fan-zhi, ZHUO Yang-ju

(*College of Electrical and Information Engineering, Foshan University, Foshan Guangdong* 528000, *China*)

Abstract: Concerning the shortage of the tamper localization accuracy and tamper recovery performance in the existing image tampers localization and recovery algorithms, the authors proposed a precise three-watermarking algorithm. It generated three types of watermarks such as detection watermark, localization watermark and recovery watermark by binary coding based on Least Significant Bit (LSB). The watermarks were imbedded into the low bits of image. Tamper detection and recovery were implemented by detection watermark and recovery watermark based on blocks, and precise localization was implemented by localization watermark based on single pixel. The simulation results show that the proposed algorithm has precise tamper localization to any size of brightness images and RGB images, and has good tamper recovery performance.

Key words: three-watermark algorithm; tamper detection; tamper localization; tamper recovery; binary coding

0　引言

近年来,基于数字水印的图像篡改定位和恢复得到了广泛的研究[1-5]。将代表图像特征的信息作为水印,嵌入到图像本身,通过提取水印信息,检测和定位图像中被篡改的位置,并利用隐藏在图像其他位置的信息对篡改部分进行恢复。

目前基于数字水印的篡改定位可以分为两种:1)分块级的篡改定位,也称分块认证,如和红杰等人[6]提出了一种高定位精度的可恢复水印算法,可以把篡改定位到 2×2 的图像分块上。为了尽可能定位准确,分块应尽可能小。2)像素级的篡改定位,也称单像素认证,如谢建全等人[7]提出了一种用于图像内容像素级篡改认证的脆弱水印算法,能准确识别图像中被篡改的像素点。基于数字水印的篡改恢复也可以分为两种:1)精确恢复,恢复后的图像和原来的图像具有完全相同的效果;2)模糊恢复,恢复后的图像和原来的图像不完全相同。对于图像认证来说[8],只需要实现模糊恢复,因为图像认证容忍恢复后的图像和原来的图像之间存在一定程度的差别。

目前已经提出了多种基于数字水印的图像篡改定位与恢复算法,Lin 等人[1]和刘泉等人[2]提出了一种分层水印算法,通过分层检测提高自恢复水印算法的篡改检测性能。该算法的篡改恢复质量不高,尤其是图像的篡改比例较大时。Lee 等人[3]提出了一种双水印算法,该算法将一个图像块的水印信息分别嵌入在两个不同的图像块中,从而提高篡改恢复质量。但该算法存在一定的安全隐患,用于检测图像块的认证数据仅能检测图像块低位水印信息的篡改,而不能检测对图像块内容的篡改,因此攻击者可以恶意篡改图像内容,只要保持低位嵌入的水印信息不变,算法就不能检测该篡改。

针对上述问题,本文结合单像素认证算法和分块认证算法的优点,在可实现篡改检测定位并能恢复被篡改区域的图像认证算法方面进行了研究,提出了一种精确的图像篡改定位与恢复的水印算法。主要研究工作包括:1)利用像素关联性质,生成单像素篡改定位水印,嵌入到图像自身的低位;2)对含定位水印的图像进行分块,生成各个分块的篡改恢复水印;3)利用对角块映射性质,将恢复水印嵌入到映射块的低位;4)通过提取检测水印和定位水印,实现对篡改图像的精确定位;5)通过提取恢复水印,实现对篡改图像的精确恢复。

1　水印生成与嵌入

设原始图像的大小为 $N_0 \times M_0$,在生成水印前,先将原始

收稿日期:2010 - 09 - 14;**修回日期**:2010 - 11 - 12。

基金项目:广东省自然科学基金资助项目(10152800001000016;10452800001004185;9151040701000002)。

作者简介:周燕(1979 -),女,江西抚州人,讲师,硕士,主要研究方向:图像处理、数字水印;　曾凡智(1964 -),男,湖北武汉人,副教授,博士,主要研究方向:数据挖掘、图像处理;　卓仰俱(1987 -),男,广东陆丰人,主要研究方向:图像处理。

图像进行图像边缘扩充处理,使图像的大小为 $N \times M$,其中 N 和 M 都是 4 的倍数,嵌入水印后再按原始图像大小输出水印图像。

1.1　定位水印的生成与嵌入

设图像 F 是大小为 $N \times M$ 的 RGB 图像,篡改定位水印的生成与嵌入算法如下。

1) 提取 F 的三层图像 FR、FG、FB,得到大小为 $N \times M$ 的单色图像,并定义 FR、FG、FB 的每个像素值为 $fr(i,j)$、$fg(i,j)$、$fb(i,j)$,其中 $i \in [1,N]$, $j \in [1,M]$。

2) 将 FR、FG、FB 中所有像素的低三位置 0,得到图像 F_pR、F_pG、F_pB。

3) 定义 F_pR、F_pG、F_pB 的每个像素值为 $f_pr(i,j)$、$f_pg(i,j)$、$f_pb(i,j)$,其中 $i \in [1,N]$, $j \in [1,M]$。一个小块的三层图像的像素值总和为:

$$Sf = f_pr(i,j) + f_pg(i,j) + f_pb(i,j) \tag{1}$$

4) 计算定位水印的关键值 WK:

$$WK = \mathrm{rem}(Sf,K) + 1;\ K \in [2^a - 1,\ 2^{a+1} - 1] \tag{2}$$

5) 将 WK 进行二进制编码:

$$WK = d_a \times 2^a + d_{a-1} \times 2^{a-1} + \cdots + d_0 \times 2^0;$$
$$d_a, d_{a-1}, \cdots, d_0 \in (0,1) \tag{3}$$

得到如下的编码序列:

$$d_a d_{a-1} \cdots d_0 \tag{4}$$

6) 提取二进制编码的后 a 位,即 $d_{a-1} \cdots d_0$,按 R、G、B 的顺序依次嵌入到 F_pR、F_pG、F_pB 的低位上。定位水印的嵌入分布情况如表 1 所示。

7) 按步骤 4) ~ 6) 处理完所有的像素点,得到嵌入定位水印后的图像 $F_p'R$、$F_p'G$、$F_p'B$。

表 1　篡改定位水印的嵌入分布情况

像素	第 3 位	第 2 位	第 1 位
$f_pr(i,j)$	d_2	0	0
$f_pg(i,j)$	d_1	0	0
$f_pb(i,j)$	d_0	0	0

1.2　恢复水印的生成与嵌入

嵌入篡改定位水印后,利用原始图像像素的平均值生成恢复水印,恢复水印的生成与嵌入算法如下。

1) 将含有定位水印的图像 F_p' 分成 2×2 的四大块,分别为 A、Q、C、D,每一块的大小为 $N/2 \times M/2$。

2) 对每一块进行无重复分块,设每小块的像素个数为 $n \times m$(本文取 $n,m = 2$),即每块的一行有 $M/4$ 小块,每一列有 $N/4$ 小块。

3) 对各小块从左到右、从上到下进行编号,并规定四大块的序号分别为:H_h^A, H_h^Q, H_h^C, H_h^D,其中 $h \in [1, NM/16]$。

4) 采用对角块同序号对应映射关系,对 H_h^A, H_h^Q, H_h^C, H_h^D 建立映射关系,即 $H_h^A \leftrightarrow H_h^D$, $H_h^Q \leftrightarrow H_h^C$。

5) 计算每个小块 FR、FG、FB 各像素值的平均值 $P(fr)$, $P(fg)$, $P(fb)$:

$$P(fr) = \frac{1}{4}\sum_{i=k1}^{k1+1}\sum_{j=k2}^{k2+1} fr(i,j) + 1;$$
$$k1 = 1,3,5,\cdots,N/2+1,\ k2 = 1,3,5,\cdots,M/2+1 \tag{5}$$

$$P(fg) = \frac{1}{4}\sum_{i=k1}^{k1+1}\sum_{j=k2}^{k2+1} fg(i,j) + 1;$$
$$k1 = 1,3,5,\cdots,N/2+1,\ k2 = 1,3,5,\cdots,M/2+1 \tag{6}$$

$$P(fb) = \frac{1}{4}\sum_{i=k1}^{k1+1}\sum_{j=k2}^{k2+1} fb(i,j) + 1;$$
$$k1 = 1,3,5,\cdots,N/2+1,\ k2 = 1,3,5,\cdots,M/2+1 \tag{7}$$

6) 对 $P(fr)$, $P(fg)$, $P(fb)$ 进行二进制编码,得到相应的序列码元:

$$r_7 r_6 \cdots r_0,\ g_7 g_6 \cdots g_0,\ b_7 b_6 \cdots b_0 \tag{8}$$

7) 分别取序列码元的前 7 位,即 $r_7 r_6 \cdots r_1$、$g_7 g_6 \cdots g_1$、$b_7 b_6 \cdots b_1$,按某种顺序分别嵌入到对应映射小块的各层的 7 个低位,这样就形成了 A 块的低位隐藏着 D 块的恢复水印,D 块的低位隐藏着 A 块的恢复水印。恢复水印在对应映射像素点低位的嵌入分布情况如表 2 所示。

表 2　恢复水印在对应映射像素点低位的嵌入分布情况

像素	第 3 位	第 2 位	第 1 位
$f_pr(i,j)$	d_2	0	g_7
$f_pg(i,j)$	d_1	0	r_7
$f_pb(i,j)$	d_0	0	b_7
$f_pr(i+1,j)$	d_2	g_4	g_3
$f_pg(i+1,j)$	d_1	r_4	r_3
$f_pb(i+1,j)$	d_0	b_4	b_3
$f_pr(i,j+1)$	d_2	g_6	g_5
$f_pg(i,j+1)$	d_1	r_6	r_5
$f_pb(i,j+1)$	d_0	b_6	b_5
$f_pr(i+1,j+1)$	d_2	g_2	g_1
$f_pg(i+1,j+1)$	d_1	r_2	r_1
$f_pb(i+1,j+1)$	d_0	b_2	b_1

8) 按步骤 4) ~ 7) 分别处理所有小块,得到嵌入了定位水印和恢复水印的图像 F_p''。

1.3　检测水印的生成与嵌入

检测水印的生成与嵌入类似于恢复水印的生成与嵌入,最大的区别是恢复水印是基于 2×2 的小块,而检测水印是基于 4×4 的小块,每个小块的 R、G、B 层的总像素值按式(9) 计算:

$$E_{\mathrm{key}} = \mathrm{rem}\left(\sum_{i=k_1}^{k_1+3}\sum_{j=k_2}^{k_2+3}(f_pr(i,j) + f_pg(i,j) + f_pb(i,j)),\ 4095\right) + 1;\ k_1 = 1,5,9,\cdots,N+1,\ k_2 = 1,5,9,\cdots,M+1 \tag{9}$$

进行二进制编码后,取码元 $e_{11}e_{10}\cdots e_0$,嵌入到每一块剩下的 12 个低位。检测水印在 4×4 小块上的嵌入分布情况,如表 3 所示。

表 3　检测水印在 4 ×4 小块上的嵌入分布情况

像素	第 2 位	像素	第 2 位
$f_pr(i,j)$	e_{11}	$f_pr(i,j+2)$	e_5
$f_pg(i,j)$	e_{10}	$f_pg(i,j+2)$	e_4
$f_pb(i,j)$	e_9	$f_pb(i,j+2)$	e_3
$f_pr(i+2,j)$	e_8	$f_pr(i+2,j+2)$	e_2
$f_pg(i+2,j)$	e_7	$f_pg(i+2,j+2)$	e_1
$f_pb(i+2,j)$	e_6	$f_pb(i+2,j+2)$	e_0

2　篡改定位与恢复

在进行图像篡改定位与恢复前,同样先对图像大小进行扩充,图像篡改定位与恢复后,再按原始大小输出图像。

2.1　篡改检测定位

设 F'' 为待检测的图像,图像篡改检测算法如下。

1）生成一个和待检测图像大小相同的篡改检测图 T'，初始化所有值为 1（白色）。

2）提取图像 F'' 每一个 4 × 4 小块的检测水印码元序列共 12 位。

3）按照检测水印的生成算法，生成对应的检测水印二进制编码序列也是 12 位。

4）比较提取的检测水印和生成的检测水印，若每个码元都相等，表明该像素没有被篡改；否则，表明该像素被篡改，将对应图 T' 中生成该检测水印块的所有像素值改为 0（黑色）。

5）按步骤 2）～4）处理所有的小块，得到检测定位图 T'。

2.2 篡改精确定位

设 F'' 为待定位的图像，定位时无需原始图像，其算法如下。

1）生成一个和待定位图像大小相同的篡改定位图 T，初始化所有值为 1（白色）。

2）提取图像 F'' 的篡改定位水印码元序列 $d_2d_1d_0$。

3）按照篡改定位水印的生成算法，生成对应的篡改定位水印二进制编码序列 $d_2'd_1'd_0'$。

4）比较 $d_2d_1d_0$ 和 $d_2'd_1'd_0'$ 码元，若每个码元都相等，表示该像素没有被篡改；否则，表示该像素被篡改，将图 T 中生成该定位水印的单像素的值改为 0（黑色）。

5）按步骤 2）～4）分别处理所有的像素，得到篡改定位图 T。

2.3 基于分块的篡改恢复

设 F'' 为待恢复的图像，T' 为篡改检测图像。篡改恢复时无需原始图像，其算法如下。

1）初始化 12 个大小为 $N/4 \times M/4$ 的辅助数组。

2）按照恢复水印的生成算法，分别提取恢复水印码元。以 A 块为例，在 D 块中按式（10）～（12）提取所有像素点的恢复像素值：

$$Ar(i,j) = \sum_{n=2}^{8} r_n \times 2^{n-1} \tag{10}$$

$$Ag(i,j) = \sum_{n=2}^{8} g_n \times 2^{n-1} \tag{11}$$

$$Ab(i,j) = \sum_{n=2}^{8} b_n \times 2^{n-1} \tag{12}$$

3）根据篡改检测生成图像，逐个像素点检查是否被篡改。如果像素值为 0，则根据被篡改的位置找到对应辅助数组的位置，用该位置的值代替 F'' 的值。

4）按步骤 2）～3）分别处理所有小块，得到恢复图像 F''。

3 算法性能分析

3.1 安全性分析

由于本文采用了基于单像素的水印算法和基于分块的水印算法，因此水印的安全性主要考虑两种类型的攻击：量化攻击和 Oracle 攻击。量化攻击针对的是基于分块的水印算法，本文的篡改检测水印采用 4 × 4 分块，由于 RGB 图像共有三层，相当于灰色图像的 4 × 4 × 3 个像素，加上检测水印嵌入位置的不确定性以及篡改检测水印与各像素值密切相关的特点，用其他小块代替将会引起该块认证失败，因此，本文算法可以抵抗量化攻击。

Oracle 攻击针对的是基于单像素的水印算法，对于本文的篡改精确定位算法，当进行 Oracle 攻击时，即使 Oracle 可以提供篡改定位结果，但攻击者最多也只能找到式（2）中 K 的取值，从而伪造出一幅图像使篡改定位算法找不到被修改的位置。这仅仅是其中的一种定位水印嵌入位置，假设每一块的定位水印按照统一的嵌入方式，根据排列组合可知，定位水印的嵌入位置共有：

$$X = 4C_9^3 = 4 \times 9 \times 8 \times 7 = 2016 \tag{13}$$

攻击者只有知道 K 的值和定位水印的嵌入位置，才能伪造成功。如果 Oracle 不提供篡改定位结果，那么对图像任何一个像素的修改都会以较高的概率被检测到。因此本文算法可以抵抗 Oracle 攻击。

3.2 篡改检测和定位性能分析

本文采用分块算法进行篡改检测。对于本文的篡改检测算法，其检测概率取决于图像的像素值本身。对于大小为 4 × 4 的 RGB 图像分块，每块都有 3 层，每一分块的像素值总和的最大值为：

$$E_{max} = 2^8 \times 16 \times 3 \tag{14}$$

利用 12 位二进制编码就能够把一个分块的像素值总和完整记录下来，一旦分块的像素值改变，二进制的编码就发生改变，一个码元发生改变就可以认定图像被篡改，因此其检测概率接近 100%。

本文的篡改精确定位算法，其检测概率取决于嵌入图像的定位水印。对于单像素精确定位，根据式（2）可知，当 k = 7 时，在区间 [0,255] 的余数为 1、2、3、4、5、6、7，同一个余数的概率，即发生漏检的概率为：

$$p = 1/k \tag{15}$$

由于 7 个余数之间是相互独立的，符合二项式独立分布：

$$P\{x = i\} = C_n^m p^m (1-p)^{n-m} \tag{16}$$

从 7 个余数中取 1 个数，经过二进制编码后作为精确定位水印，出现漏检的概率为：

$$P = C_7^1 \times \frac{1}{7} \times \left(\frac{6}{7}\right)^6 = 0.39 \tag{17}$$

3.3 篡改恢复性能分析

本文的篡改恢复算法属于模糊恢复，恢复的像素值与原图像的像素值相差较小，以 4 × 4 像素点为单位，每单位像素值的差为 1。算法的篡改恢复能力依赖于图像篡改检测的精度，如果图像与对应的映射分块同时被篡改，则篡改无法恢复，否则，一定能够恢复被篡改的图像，但恢复的图像有较小的模糊性。

从算法时间复杂度方面考虑，算法要借助 12 个辅助数组，因此时间复杂度为 $O(N \times M)$；而文献[7] 中的混沌散列算法的时间复杂度为 $T(n) = O(K \times M \times N)$，其中 K 的取值不小于 20 才能得到混沌现象，因此本算法的时间复杂度较低。

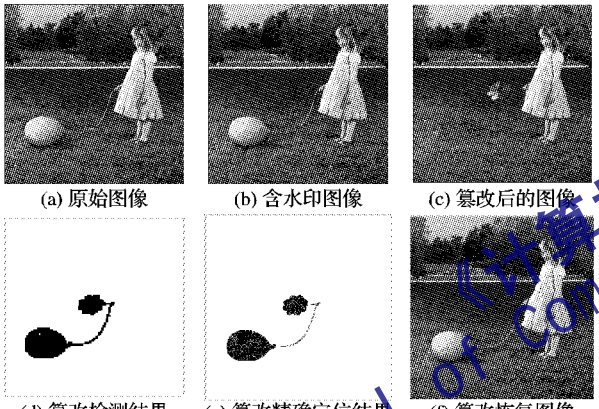
4 仿真实验

为了测试算法的性能，在 Matlab 平台上对本文算法进行了实验仿真。算法主要针对数据类型为 uint8 的图像，即用 1 b来表示一个像素值，其输入的图像类型为亮度图像或者 RGB 图像。算法中关联像素的个数取 $n = 2, m = 2, K = 7$。

图 1（a）是原始 RGB 图像，图（b）是用本文算法嵌入水印后的图像，从图中可以看出，水印透明性很好，含水印图像和原始图像的峰值信噪比（Peak Signal to Noise Ratio，PSNR）达

到了 40 dB 以上,满足水印算法对不可见性的要求。图(c)是篡改后的图像(图中的球和绳子被替换为风车),图(d)是用篡改检测算法得到的检测结果,图(e)是用篡改精确定位算法得到的定位结果,图(f)是用篡改恢复算法得到的恢复图像。从图中可以看出,篡改定位精度很高,被篡改的图像能够完整地恢复出来,恢复的图像与原始图像的 PSNR 达到 42.31 dB,恢复效果很好。

为了验证篡改检测算法的准确性,对一幅 256×256 的嵌入水印的 Lena 图像进行不同程度的篡改,然后分别采用文献[3]、文献[9]的算法以及本文算法进行篡改检测。使用 $P = N_e/N_d$ 表示检测概率(N_e 为随机篡改的像素点,N_d 为实际检测到的像素点)。三种算法的篡改检测概率曲线如图 2 所示。在文献[3]的算法中,当篡改的区域是连续的并且是连通时,才具有良好的篡改检测能力,当篡改区域较为分散时,篡改检测概率只达到 75%。在文献[9]的算法中,采用了单像素置乱及像素关联技术,该算法存在虚警和漏警,单个像素的检测概率为 $1-0.5^2=75\%$。在本文的算法中,通过嵌入检测水印和定位水印,增强了水印的抗攻击能力,提高了篡改检测能力,因此具有更高的检测率。



(a) 原始图像　(b) 含水印图像　(c) 篡改后的图像

(d) 篡改检测结果　(e) 篡改精确定位结果　(f) 篡改恢复图像

图 1　图像篡改定位与篡改恢复仿真结果

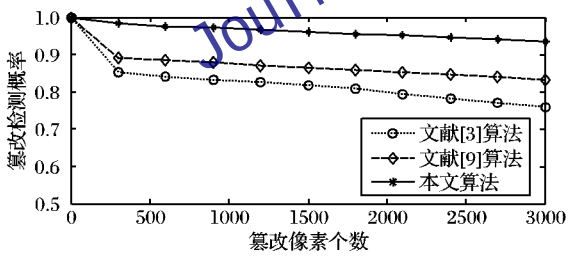


图 2　图像篡改检测概率对比

为了验证篡改恢复算法的精确性,通过对一组 256×256 的标准图像嵌入水印,对含水印图像篡改 2 000 个像素点,分别采用文献[3-4]的算法以及本文算法进行恢复,并通过恢复图像与原始图像的 PSNR 来表示图像的篡改恢复效果。表 4 列出了三种算法的篡改恢复效果。由于在本文算法中使用了恢复水印,因此恢复效果比另外两种算法都要好。

图像的篡改比例与篡改恢复效果存在一定的关系,一般地,篡改比例越低(篡改的范围越小),篡改恢复效果越好;篡改比例越高(篡改的范围越大),篡改恢复效果越差。对于某些算法,当达到一定的篡改比例后,篡改图像就无法恢复。以 256×256 的 Lena 图像为例,按不同的比例进行篡改,然后利用文献[3-4]的算法以及本文算法进行恢复,三类算法的篡改比例与篡改恢复效果的关系如图 3 所示。从图 3 中可以看出,随着篡改比例的提高,三类算法的篡改恢复效果逐渐下降。对于相同的篡改比例,本文算法的篡改恢复效果比另外两种算法要好很多。

表 4　算法的篡改恢复效果对比　　dB

图像	文献[3]算法	文献[4]算法	本文算法
Lena	28.27	32.29	46.62
Bamboo	27.11	31.89	45.42
Bird	28.42	31.64	43.98
Boat	27.91	30.32	41.57
Girl	28.73	30.81	39.67
House	27.23	30.72	40.12

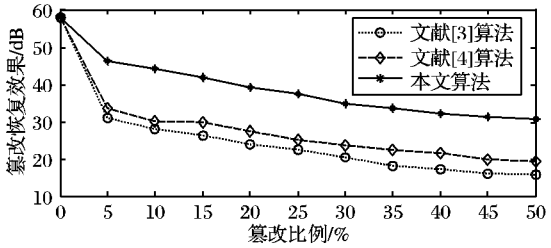


图 3　图像篡改恢复效果对比

5　结语

本文提出的三水印算法,通过生成并嵌入三种不同的水印,实现对篡改图像的篡改检测、篡改精确定位以及篡改恢复。该算法的主要特点是:1)在基于 LSB 方法的基础上,采用二进制编码的方式生成水印,水印嵌入灵活多变,能够提高水印的安全性和抗攻击能力;2)通过嵌入三种不同的水印,分别实现篡改检测、篡改精确定位和篡改恢复,其中篡改检测和篡改恢复采用基于分块的水印,篡改精确定位采用基于单像素的水印;3)通过边界像素扩展,可以处理任意大小的图像;4)算法的时间复杂度跟图像的大小有关,为 $O(N\times M)$,比目前的算法都要低;5)恢复算法利用了两层分块按同编号映射性质,能够大面积地恢复被篡改的内容,而且恢复效果好。

安全性分析表明,本文算法能够抵抗针对单像素的 Oracle 攻击以及针对分块算法的量化攻击,仿真实验也表明算法具有较好的性能。

参考文献:

[1] LIN P L, HSIEH C K, HUANG P W. A hierarchical digital watermarking method for image tampers detection and recovery [J]. Pattern Recognition, 2005, 38(12): 2519-2529.

[2] 刘泉,江雪梅.用于图像篡改定位和恢复的分层半脆弱数字水印算法[J].通信学报,2007,28(7):104-110.

[3] LEE T Y, LIN S D. Dual watermark for image tamper detection and recovery [J]. Pattern Recognition, 2008, 41(11): 3497-3506.

[4] 赵彦涛,李志全.一种改进的图像篡改定位及恢复的双水印算法[J].光电子·激光,2009,20(7):938-943.

[5] 张磊,陈帆,高辉.基于混沌的图像自恢复安全双水印算法[J].计算机应用,2010,30(1):203-206.

[6] 和红杰,张家树,陈帆.一种高定位精度的可恢复水印算法[J].中国科学:E 辑,2008,38(4):533-552.

[7] 谢建全,阳春华.一种像素级的图像篡改认证算法[J].计算机应用,2007,27(6):1337-1342.

[8] 张宪海,杨永田.基于脆弱水印的图像认证算法研究[J].电子学报,2007,35(1):34-39.

[9] 王国栋,刘粉林,汪萍,等.一种篡改检测与篡改定位分离的图像认证方案[J].计算机学报,2007,30(10):1880-1888.