中国科学 E 辑: 信息科学
2007 年 第 37 卷 第 2 期: 315~328
http://www.scichina.com

《中国科学》杂志社
SCIENCE IN CHINA PRESS

一种高定位精度的安全 JPEG 图像认证水印算法

余 淼* 和红杰 张家树

(西南交通大学信号与信息处理四川省重点实验室, 成都 610031)

摘要 分析讨论了现有的 JPEG 图像认证水印算法存在的定位精度低和安全性差等问题, 并实现了对现有认证水印算法的两种伪造攻击. 在此基础上, 提出了一种高定位精度的安全 JPEG 图像认证水印算法, 推导给出了该算法在一般区域篡改和拼贴攻击下的篡改检测概率和虚警概率. 该算法对每个图像块固定选取 4 个中频系数嵌入水印并利用剩余的 DCT 系数生成 4 比特水印信息, 然后基于不同密钥分别选取每比特水印信息嵌入的图像块, 使得每个图像块的 4 位水印随机嵌入在不同图像块对应的中频系数中. 认证时根据图像块提取出的 4 比特水印信息并结合其九邻域判定该图像块内容是否被篡改. 理论分析和实验结果表明: 该算法不仅具有精确的篡改定位精度, 而且具有很高的安全性和抵抗拼贴攻击的能力.

关键词 脆弱水印 JPEG 压缩 定位精度 拼贴攻击

数字图像水印 [1—7]作为信息隐藏 [8]技术研究领域的重要组成部分, 特别是结合广泛使用的JPEG图像压缩标准对其内容完整性认证的水印算法 [4—7], 已成为国内外广泛关注的研究热点课题. 现有的JPEG图像水印算法可以分为两类: 一类是能够容忍一定程度的JPEG压缩, 并能够检测空域和变换域恶意篡改的半脆弱水印算法 [4,5]; 另一类是将水印算法融入到图像的压缩算法中, 在压缩的过程中嵌入水印, 在解压的过程中提取水印, 用来对JPEG图像进行精确认证的脆弱数字水印算法 [6,7]. 其中, 用于JPEG图像认证的脆弱数字水印算法主要用于网络传输中图像的认证及内容完整性证明等领域, 已引起了众多学者和产业界的广泛关注.

早期的JPEG图像认证水印算法为保证水印的不可见性, 只在几个选定的DCT系数上嵌入了水印 [7], 而大部分的DCT系数未得到保护, 致使这类JPEG图像认证水印算法存在严重的安全性隐患. 为了增强其安全性, Li提出了一种新的脆弱水印算法 [6]. 该算法将DCT系数分为水

收稿日期: 2006-07-15; 接受日期: 2006-09-30
国家自然科学基金(批准号: 60572027)、教育部新世纪优秀人才支持计划(批准号: NCET-05-0794)、四川省青年科技基金(批准号: 03ZQ026-033)、国防预研基金(批准号: 51430804QT2201)和四川省应用基础研究(批准号: 2006J13-10)资助项目
* 联系人, E-mail: zealyu@163.com

印嵌入系数和非水印嵌入系数两类, 首先利用所有的非水印嵌入系数生成水印, 然后将其嵌入到水印嵌入系数上, 从而使图像块的所有DCT系数都得到保护; 而基于九邻域的水印生成方案提高了该算法抵抗拼贴攻击 [9]的能力. 虽然Li提出的JPEG图像认证水印算法在安全性方面有明显提高, 但不幸的是该算法仍然存在安全隐患, 主要表现为: (ⅰ)篡改检测的定位精度不高; (ⅱ)不能有效保护图像的平滑区域; (iii)不能完全抵抗拼贴攻击 [9].

针对上述问题, 本文提出了一种高定位精度的安全JPEG图像认证水印方案. 该方案对每个图像块固定选取 4 个中频系数作为水印的嵌入位, 并利用剩余的DCT系数生成 4bits水印, 然后基于不同的密钥分别选取每比特水印信息嵌入的图像块, 使每个图像块的 4bits水印随机嵌入到 4 个图像块中. 认证时根据图像块提取的水印信息并结合其九邻域判定该图像块内容是否被篡改. 通过水印嵌入引入图像块之间的相关性, 不仅使算法拥有好的篡改定位精度, 而且大大提高了算法抵抗拼贴攻击 [9]的能力; 结合其九邻域判定图像块是否被篡改, 可以在保持算法篡改检测概率的同时降低算法的虚警概率, 并从理论上推导了在一般区域篡改和拼贴攻击 [9]下, 算法的篡改检测概率和虚警概率. 理论分析和实验仿真表明: 无论是对平滑区域的篡改, 还是采用拼贴攻击 [9]对图像的篡改, 该算法都能精确定位图像被篡改的位置.

1 对Li[6]的算法的安全性分析

与文献 [7]的算法相比, Li提出的JPEG图像认证水印算法在安全性方面有明显提高, 不过该算法仍然存在安全隐患. 下面对该算法存在的安全隐患进行详细分析:

(ⅰ) 篡改定位精度低.

篡改定位精度是指在含水印图像遭到篡改后, 算法能够既准确又精确的定位图像中那些被篡改的区域. 对定位型认证水印算法而言, 篡改定位精度是衡量算法性能的一个很重要的方面. 影响算法篡改定位精度的因素主要有两个: ①算法的篡改检测概率, 即在篡改区域上检测到篡改的概率. ②算法的虚警概率, 即在非篡改区域上检测到篡改的概率. 一个算法只有同时拥有高的篡改检测概率和低的虚警概率时才具有高的定位精度.

在 Li 的算法中, 水印是基于九邻域生成的. 含水印图像的任何一个图像块发生篡改时, 都极有可能造成其周围九邻域中的未被篡改的图像块被检测出篡改, 也就是该算法的虚警概率高, 造成了其定位精度低.

为了降低算法的虚警概率, Li引入了一个阈值k. 在图像块被检测到篡改后, 再考察其九邻域中被检测到篡改的图像块的个数(此个数为k), 然后决定是否将此图像块判定为已篡改. 阈值k的引入可以有效降低算法的虚警概率, 但同时也降低了算法的篡改检测概率. 换句话 说, 这不仅没有有效的提高算法的定位精度, 而且会给算法带来安全隐患. 在本节第三部分对Li的算法进行拼贴攻击 [9]的一个实例中, 将对此作更为详细的分析.

(ⅱ)不能保护图像的平滑区域.

Li 的算法要求在每个图像块中选取 4 个非零的中频系数作为水印的嵌入位, 但是由于在 JPEG 压缩下, 部分较平滑的图像块的大部分系数为零, 其非零的中频的个数小于 4, 导致在这些图像块上无法完成水印嵌入. 下面以 Clock, F-16, Lena, Peppers, Baboon 为例, 分别统计它们在经过各种质量因子的 JPEG 压缩后, 频率不大于中频 h(这里 h 取 19)的非零系数的个数小于 4 的图像块占整个图像的比例. 实验结果如图 1 所示.

由图 1 可知: 在相同的 JPEG 压缩质量因子下, 不同的图像无法完成水印嵌入的图像块的

比例相差较大. 但对于大多数图像而言, 在小于85的JPEG压缩质量因子下, 都存在不同程度无法完成水印嵌入的图像块, 特别是那些含平滑区域较多的图像, 这样的图像块所占的比例更大. 这些无法嵌入水印的图像块将给 Li 算法带来安全性隐患, 不能有效地保护图像中的平滑区域.

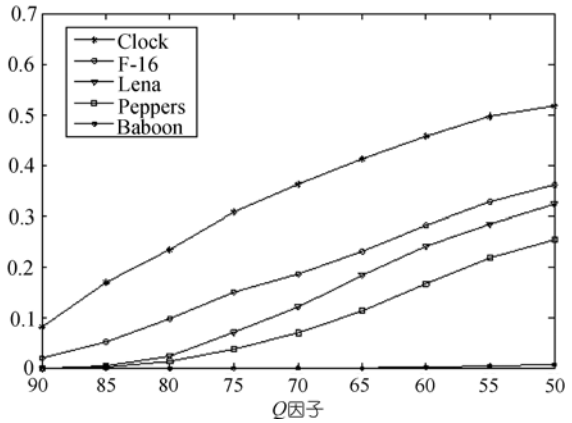

图 1　在各种质量因子下无法嵌入水印的图像块的比例

图 2 和图 3 以 Clock 图像为例, 直观地展示 Li 算法不能保护图像中的平滑区域的安全性隐患. 图 2 是经过质量因子为 80 的 JPEG 压缩后的图像块分类示意图, 其中 Clock 中频率不大于 19 的非零系数的个数等于 4 的图像块用黑色表示, 小于 4 的图像块用灰色表示, 其他图像块的灰度值保持不变. 图 2 中的大片灰色区域图像块均处于原始图像中较平滑的区域, 而这些平滑区域由于没有足够的非零中频系数, 在 Li 的算法中这些区域将无法完成水印的嵌入, 也将跳过水印的检测. 所以 Li 的算法不能保护图像的平滑区域, 这给算法带来了安全隐患. 图 3 是对 Li 算法的一个伪造攻击示意图. 在这个伪造攻击中, 首先将 Lena 图像中的眼睛区域的图像块进行质量因子为 80 的 JPEG 压缩, 然后只保留频率不大于 19 的前 3 个非零系数, 并将其他所有的系数全部置 0, 用它们替代图 2 中的那些灰色区域即得到了对 Li 的算法的一个伪造攻击. 由于这些图像块的非零系数的个数均小于 4, 则图 3 所示的伪造结果也将能够完全通过 Li 算法的水印检测.

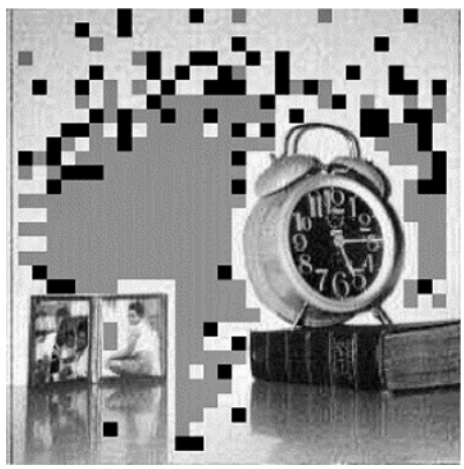

图 2　$Q = 80$ 时图像块的分类

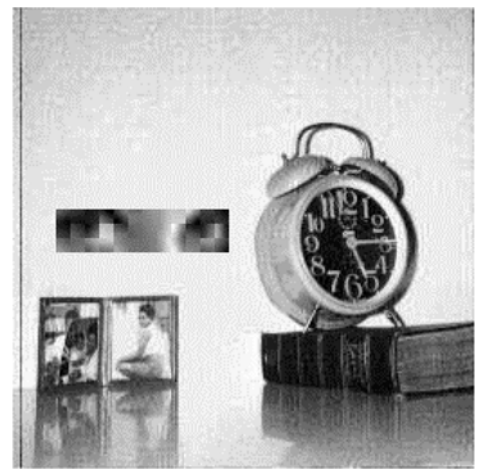

图 3　对图像中的平滑区域进行的伪造攻击

(iii) 不能完全抵抗拼贴攻击 [9].

Li采用基于九邻域的水印生成技术来引入图像块之间的相关性, 从而提高算法抵抗拼贴攻击[9]的能力. 在拼贴攻击[9]的区域较小时, Li算法的效果较好. 遗憾的是, 当拼贴攻击[9]的区域较大时, 篡改区域内部图像块的篡改不能被检测, 这是由于: ①图像块及其九邻域的所有图像块均未改变, 因而基于这些图像块生成的水印不会改变; ②图像块的水印嵌入在自身, 拼贴攻击后从该图像块提取的水印也不会改变. 所以, 拼贴攻击[9]的区域较大时, 篡改区域内部的

图像块能够通过认证, 能够检测到篡改的只是篡改区域内外边界上的图像块, 因此Li的算法不能完全抵抗拼贴攻击.

为了验证上述分析结果, 对 Li 的算法抵抗拼贴攻击的能力进行了仿真实验研究. 仿真研究中, Lena 和 Peppers 在 JPEG 压缩质量因子为 80 的条件下, 采用相同的密钥按 Li 的算法进行水印的嵌入, 然后用含水印的 Lena 图像中的部分图像块替代含水印的 Peppers 图像中相同位置的图像块, 得到经过拼贴攻击的 Peppers 图像. 对这个伪造的图像按 Li 的算法进行认证, 在阈值 k 分别取 0, 4, 6 的情形下, 得到的检测结果分别如图 4(a)—(c)所示. 图中灰色的部分, 表示被检测出篡改的块.

图 4 在各种 k 值下拼贴攻击的检测结果

(a) $k = 0$ 时的篡改检测结果; (b) $k = 4$ 时的篡改检测结果;(c) $k = 6$ 时的篡改检测结果

由图 4(a)—(c)可以看出: ①Li 的算法不能检测出拼贴区域内部图像块的篡改; ②以一定概率检测出拼贴区域内外边界图像块的篡改, 该检测概率随阈值 k 的增大而急剧下降. 如图 4(c)所示, 在 $k = 6$ 时, 该伪造图像基本通过了认证(仅检测出一个图像块被篡改), 而 Li 给出的仿真中, k 的取值就为 6. 由此可知, 阈值 k 的引入不仅没有提高算法的定位精度, 反而给算法带来严重的安全性隐患.

综上所述, 造成Li的算法存在上述安全性隐患的主要原因有: ①基于九邻域的水印生成技术降低了算法的定位精度. ②水印嵌入在 4 个非零的中频系数上, 而平滑区域的图像块由于找不到足够的非零中频系数, 所以得不到算法的保护. ③图像块水印被嵌入在该图像块自身中, 使算法不能完全抵抗拼贴攻击 [9].

2 本文提出的用于 JPEG 图像认证的高定位精度的安全脆弱水印算法

为了克服Li的算法安全性隐患, 本文提出了具有较高定位精度的安全脆弱水印算法. 新算法基于图像块本身产生图像块的 4bits水印, 从而提高算法的定位精度; 固定选取了 4 个中频系数作为水印的嵌入系数, 以便保证图像中所有的图像块均能够完成水印的嵌入, 从而实现图像平滑区域的保护; 为了完全抵抗拼贴攻击 [9], 每个图像块的 4bits水印分别根据其所对应的密钥随机嵌入到其他 4 个图像块的对应位置中. 这样, 在拼贴攻击下, 篡改区域图像块的水印很可能位于非篡改区域, 从而使得这些篡改区域的图像块检测出篡改. 新算法具体过程描述如下:

(ⅰ) 水印的嵌入算法.

Step 1 对原始图像进行分块 DCT 变换, 然后用指定的 JPEG 压缩的质量因子 Q 对应的量化表对其进行量化, 得到 DCT 块 X_i, i=1, 2, …, N (N 代表图像块的个数). 然后对每个图像块的

系数进行 zigzag 排序, $X_i(j)$代表 X_i 经 zigzag 排序后的第 j 个系数.

Step 2　利用密钥 k_1 产生与原始图像大小相同的二值随机序列 A, 其组织方式与 X 相同, 即 $A_i(j)$代表 A 中第 i 块(即 A_i)按照 zigzag 排序后的第 j 个比特.

Step 3　对于所有的 DCT 块 X_i 而言, 频率 0 至 h'的系数为水印生成系数. 由所有图像块的水印生成系数产生 4 份水印 W_k (k=1, 2, …, 4), 然后将其分别嵌入所有 DCT 块的 h'+1 至 h'+4 这 4 个水印嵌入系数中. 下面是 W_k 的生成过程:

①对每个 DCT 块 X_i 生成 4 个 $S_i(k)$ (k = 1, 2, 3, 4), 其公式为

$$S_i(k)=\sum_{n\in[0,\ h']}(A_i(n)\oplus A_i(h'+k))\cdot X_i(n), \tag{1}$$

其中⊕代表异或运算. 由公式(1)可以看出, $S_i(k)$依据其水印嵌入系数的 $A_i(h'+k)$与生成水印的系数的 $A_i(n)$的异或运算结果, 来选取 $X_i(n)$参与求和运算. 令 Parity($S_i(k)$)表示先将 $S_i(k)$转换为其补码, 然后再统计其补码中的 1 的个数, 若 1 的个数为奇数, 则 Parity($S_i(k)$)=1, 否则 Parity($S_i(k)$)=0. 最后, 按如下的公式对每个图像块 X_i, 生成 4 位水印 $w_i(k)$ (k=1, 2, 3, 4):

$$w_i(k)=\text{Parity}(S_i(k))\oplus A_i(h'+k). \tag{2}$$

由公式(2)可以看出, 每个图像块的 4 位水印是通过其所生成的 $S_i(k)$的 Parity($S_i(k)$)的值与其对应的水印嵌入位的 $A_i(h'+k)$经过异或运算产生.

②W_k 为所有 DCT 块的第 k 位水印 $w_i(k)$的集合, 即 $W_k=\{w_1(k),w_2(k),\cdots,w_N(k)\}$.

Step 4　生成分别与 4 个 W_k 对应的水印嵌入位置序列 $f_k(i)$ (k=1, 2, 3, 4, i=1, 2, …, N), 其中 f_1 为 W_1 的嵌入位置序列, 其余 3 组的对应关系依此类推. 下面是其生成方法:

①利用密钥 k_2, k_3, k_4, k_5 分别生成长度为 N 的随机序列 R^1, R^2, R^3, R^4, 下面以 f_1 为例来说明其生成方法, 剩余 3 个与此类似.

② $R^1=(r_1^1,r_2^1,\cdots,r_N^1)$,采用稳定排序法生成有序序列 $R_a^1=(r_{a_1}^1,r_{a_2}^1,\cdots,r_{a_N}^1)$, 则得到其索引序列 $I^1=(a_1,a_2,\cdots,a_N)$.

③ 令 $f_1(i)=a_i$, i=1, 2, …, N.

Step 5　将 W_k(k=1, 2, 3, 4)分别按其水印嵌入位置序列 f_k 将其嵌入到所对应的 DCT 块的对应位中, 即对每个 W_k 进行如下操作:

对于 W_k 中的每一个 $w_i(k)$, 根据其嵌入位置序列 f_k 找到其所嵌入的 DCT 块 $X_{f_k(i)}$, 然后修改此 DCT 块中对应的水印嵌入系数 $X_{f_k(i)}(h'+k)$, 使其满足公式(3), 以达到嵌入 $w_i(k)$的目的.

$$\text{Parity}(X_{f_k(i)}(h'+k))=w_i(k), \tag{3}$$

其中 Parity 的含义见 Step 3 中所述.

(ⅱ) 水印的提取算法.

Step 1　对接收到的 JPEG 图像进行解码, 得到其 DCT 块 X_i, i=1,2,…,N.

Step 2　与水印嵌入算法中的 Step 2 相同, 利用密钥 k_1 产生二值随机序列 A, 然后按与嵌入算法中 Step 3 相同的方法生成 W_k, 并利用密钥 k_2, k_3, k_4, k_5 按嵌入算法中 Step 4 中的方法生成与 W_k, 对应的水印嵌入位置序列 f_k, k=1,2,3,4.

Step 3　生成大小为 N 的全 0 序列 v, 其第 i 位 $v(i)$代表 DCT 块 X_i 中检测到篡改的水印的个数. 对每个 W_k, 进行如下的验证:

对 W_k 中的每一位水印 $w_k(i)$, 可知 $w_k(i)$代表 DCT 块 X_i 的第 k 个水印, 根据其所对应的嵌入位置序列 f_k 找到其水印嵌入块 $X_{f_k(i)}$, 然后验证其与 $w_k(i)$对应的水印嵌入的系数 $X_{f_k(i)}(h'+k)$ 是否满足公式(3), 若不满足, 则将此 DCT 块 X_i 对应的 $v(i)$加 1.

Step 4 对每个 DCT 块 X_i, 若其所对应的 $v(i)>0$, 则考察其九邻域内(包括 X_i 本身在内)的图像块中满足 $v(i)\geqslant 2$ 的个数, 若此个数不小于 2, 则将此图像块标记为篡改, 否则认为此图像块通过了验证.

在水印提取算法的最后一步中, 在检测到图像块出现篡改后, 通过考察其九邻域内检测到篡改的图像块的个数, 并结合其篡改的位数, 确定是否将此图像块确认为篡改, 从而提高了新算法的定位精度.

3 本文算法的性能分析

脆弱数字水印算法的性能主要有两个评价指标: ①算法的安全性, ②算法的篡改定位精度. 作为一种保护多媒体信息的重要手段, 脆弱数字水印算法的安全性至关重要. 而脆弱数字水印算法作为认证型水印算法的一种, 其篡改定位精度反映了算法对篡改的定位能力, 是衡量算法性能的重要指标. 因此, 本文将分别从这两个方面入手, 讨论新算法的性能.

3.1 算法的安全性分析

从水印的嵌入算法可知: 由于固定选取了所有图像块的 4 个水印嵌入位, 保证了对于所有的图像块都能够嵌入水印, 这样就保护了含水印图像的所有区域, 从而克服了 Li 的算法不能有效地保护含水印图像的平滑区域的缺陷, 提高了 JPEG 图像认证水印算法的安全性.

由于在JPEG图像认证水印算法中, 篡改检测概率越高, 表明认证水印算法抵抗攻击的能力越强, 即它的安全性也越高. 为了更好地分析本文的新算法抵抗拼贴攻击 [9]的能力, 此处分别在一般区域篡改和拼贴攻击 [9]下, 通过考察新算法的篡改检测概率来分析新算法的安全性.

(ⅰ) 在一般区域篡改下的篡改检测概率.

由公式(1)和(2)可知: 一个图像块 X_i 的 4 个 $w_i(k)$根据其所对应的 $A_i(h'+k)$不同, 只有两种可能的取值情形. 不妨将其分别记为 $w_i^0(k)$ 和 $w_i^1(k)$, 分别对应于 $A_i(h'+k)$取值为 0 和 1 这两种情形. 在一般区域篡改下, 经过篡改后的图像块所生成的 $w_i^0(k)$ 和 $w_i^1(k)$ 与原图像块不同的概率均为 0.5. 由于 A 是二值随机序列, 故在 4 个水印嵌入位上 $A_i(h'+k)$为 0 和 1 的概率为 0.5, 即在每个水印嵌入位上 $w_i^0(k)$ 和 $w_i^1(k)$ 出现的概率相同. 设 p_{w0}, p_{w1}, p_{w2}, p_{w3}, p_{w4} 分别表示在一般篡改下, 图像块 X_i 生成的 4 个水印发生改变的个数分别为 0, 1, 2, 3, 4 的概率, 其计算公式如下:

$$p_{w0}=0.5^2+C_2^1 0.5^2\times 0.5^4. \tag{4}$$

(4)式右边第一项代表 $w_i^0(k)$ 和 $w_i^1(k)$ 与原始图像块相同的概率, 第二项代表当 $w_i^0(k)$ 和 $w_i^1(k)$ 中有一个与原始图像块不同时, 这个不同的 $w_i(k)$在 4 个水印嵌入位上均不出现的概率. 在这两种情形下, 图像块在遭到一般区域篡改后生成的 4 个水印与原始图像块相同. 与此分析类似, 可以得到 p_{w1}, p_{w2}, p_{w3}, p_{w4} 的计算公式.

$$p_{w1}=C_2^1 0.5^2\times C_4^1 0.5^4, \tag{5}$$

$$p_{w2}=C_2^1 0.5^2\times C_4^2 0.5^4, \tag{6}$$

$$p_{w3} = C_2^1 0.5^2 \times C_4^3 0.5^4, \tag{7}$$

$$p_{w4} = 0.5^2 + C_2^1 0.5^2 \times 0.5^4. \tag{8}$$

而由一般区域篡改的特点可知, 它在使篡改后的图像块生成的水印与原图像块不同的同时, 也篡改了水印嵌入位的水印, 且篡改后的每个水印嵌入位的水印与原水印相同的概率均为 0.5. 故可知, 无论篡改后的 $w_i(k)$是否相同, 只要其嵌入在篡改区域的内部, 则其检测出篡改的概率均为 0.5, 若 $w_i(k)$相同, 且其嵌入在篡改区域外部(即非篡改区域), 则其不会被检测出篡改, 反之, 若 $w_i(k)$相同且其被嵌入在非篡改区域, 则其一定会被检测出篡改. 设 a 为篡改区域占整个图像的比例, 且由本文提出的算法可知, $w_i(k)$嵌入在所有图像块的概率服从均匀分布. 设 p_e 为 $w_i(k)$相同时被检测到篡改的概率, p_{ne} 为 $w_i(k)$不同时被检测到篡改的概率, 则由上面的分析, 其分别为

$$p_e = (1-a)\times 0 + a \times 0.5 = 0.5a, \tag{9}$$

$$p_{ne} = (1-a)\times 1 + a \times 0.5 = 1 - 0.5a. \tag{10}$$

设 P_{m0} 为图像块 X_i 的 4 个水印 $w_i(k)$均未检测到篡改的概率, 则可知

$$\begin{aligned} P_{m0} = {} & p_{w0} \times (1-p_e)^4 + p_{w1} \times (1-p_e)^3 \times (1-p_{ne}) + p_{w2} \times (1-p_e)^2 \times (1-p_{ne})^2 \\ & + p_{w3} \times (1-p_e) \times (1-p_{ne})^3 + p_{w4} \times (1-p_{ne})^4. \end{aligned} \tag{11}$$

(11)式右边第一项表示, 当 4 位 $w_i(k)$均相同时, 这 4 位均检测不到篡改的概率, 第二项表示, 当 4 位 $w_i(k)$有一位不相同时, 此不相同的一位和其余相同的 3 位均检测不到篡改的概率, 剩下的 3 项, 以此类推.

类似的, 还可以得到 4 个嵌入位中只能检测到一位篡改的概率 P_{m1} 如下:

$$\begin{aligned} P_{m1} = {} & p_{w0} \times C_4^1 p_e \times (1-p_e)^3 + p_{w1} \times (C_3^1 p_e \times (1-p_e)^2 \times (1-p_{ne}) + (1-p_e)^3 \times p_{ne}) \\ & + p_{w2} \times (C_2^1 p_e \times (1-p_e) \times (1-p_{ne})^2 + C_2^1 p_{ne} \times (1-p_{ne}) \times (1-p_e)^2) \\ & + p_{w3} \times (p_e \times (1-p_{ne})^3 + C_3^1 p_{ne} \times (1-p_{ne})^2 \times (1-p_e)) + p_{w4} \times C_4^1 p_{ne} \times (1-p_{ne})^3. \end{aligned} \tag{12}$$

(12)式右边的第一项, 表示在 4 位 $w_i(k)$均相同时, 仅有一位被检测到的概率, 第二项表示在 4 位 $w_i(k)$中有一个不同, 另外 3 个相同, 在这种情形下, 或者这个不同的 $w_i(k)$被检测到篡改, 同时另外 3 个相同的 $w_i(k)$未检测到篡改, 此概率为 $p_{ne} \times (1-p_e)^3$ 第二项中括号内的第二项, 或者 3 个相同的 $w_i(k)$中有一位检测到篡改, 同时另外一个相同的 $w_i(k)$未检测到篡改, 此概率为 $C_3^1 p_e \times (1-p_e)^2 \times (1-p_{ne})$, 即第二项中括号内的第一项. 后 3 项的概率以此类推, 就得到了 P_{m1} 的表达式.

用 P_{m2a} 表示在 4 个嵌入位中能检测到大于等于两位篡改的概率, 则易知

$$P_{m2a} = 1 - P_{m0} - P_{m1}. \tag{13}$$

由水印检测算法可知, 在一般的区域篡改下的检测概率 P_d 为

$$P_d = (1-P_{m0}) \times (1-(1-P_{m2a})^9 - C_9^1 P_{m2a} (1-P_{m2a})^8). \tag{14}$$

(14)式右边第一个括号内的 $1-P_{m0}$ 表示篡改块被检测到有不小于 1 位篡改的概率, 第二个括号表示其周围九邻域内, 有两个及以上满足其被检测到篡改的位数不小于 2 的图像块, 两者相乘, 即表示在一般区域篡改下, 篡改块的检测概率 P_d.

下面通过实验统计 P_{m0}, P_{m1}, P_{m2a} 的实验值 Pm0simu, Pm1simu, Pm2asimu, 在 0.02 至 0.2

之间, 取 10 个 a 值, 对于每个 a 值, 分别进行 50 次一般区域篡改, 统计 P_{m0}, P_{m1}, P_{m2a} 的实验值 Pm0simu, Pm1simu, Pm2asimu, 然后对这 50 次统计的结果求均值, 将其作为在此 a 下的实验值. 将它们画在图中, 如图 5 所示.

图 5 中的 Pm0 为在一般区域篡改下篡改区域图像块的 4 个水印嵌入位中检测不出篡改的概率的理论值, Pm0simu 为其实验值, 它们是图 5 中最上方的两条曲线, 可见这两条曲线较接近, 说明其理论值和实验值符合得较好. Pm1 为仅能检测出一位水印篡改的概率的理论值, Pm1simu 为其实验值, 它们是图 5 中中间的两条曲线, 其理论值和实验值符合得较好. Pm2a 是能检测出两位及以上篡改的概率的理论值, Pm2asimu 为其实验值, 它们位于图 5 的最下方, 其理论值和实验值也符合得较好.

再统计 P_d 的实验值 Pdsimu0 和 Pdsimu1, 其中, Pdsimu0 在统计篡改块时, 只统计了篡改区域内的篡改块的个数, 而 Pdsimu1 将篡改区域及与其边界相邻的图像块中所有被检测到篡改的块, 全部统计在内. 其统计结果如图 6 所示.

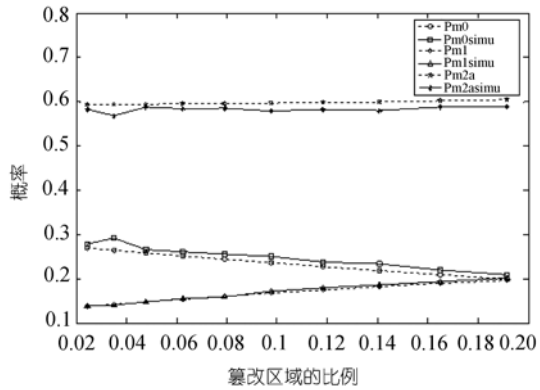

图 5 一般区域篡改下篡改区域图像块检测不出篡改、检测出 1 位、检测出 2 位及以上篡改的概率的理论值和实验值

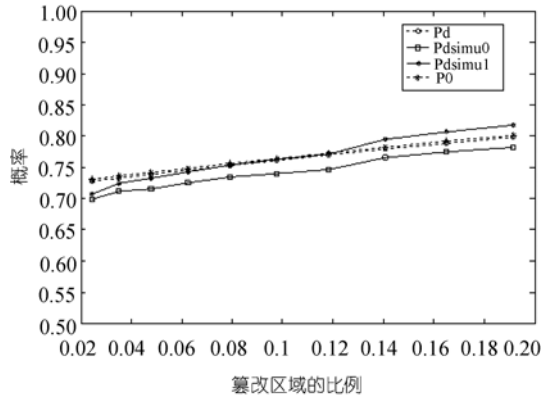

图 6 一般区域篡改下篡改检测概率的理论值及其实验值

图 6 中间两条虚线分别对应为 Pd 和 P0, 其中 Pd 为在一般区域篡改下检测概率的理论值, P0 为$(1-P_{m0})$. 由上面的分析可知, P0 表示未进行检测算法最后一步时的检测概率, 为理论上检测概率的极限值. 这两条曲线十分接近表明: 在一般区域篡改下, 经过本文算法最后一步的处理后, 本文提出的新的用于 JPEG 图像认证的水印算法的篡改检测概率基本保持不变.

图 6 中的 Pdsimu0 为只统计了篡改区域内的篡改块而得到的篡改检测概率的实验值, 略小于篡改检测概率的理论值 Pd. 这是由于在篡改区域的外边界上检测出篡改的概率较低, 影响了篡改区域内边界上的篡改块的检测概率. 而篡改区域外部的这些检测出篡改的块, 对篡改区域的定位是有很大帮助的, 故将其计入被检测到的篡改块的数目, 从而得到 Pdsimu1. 从图 6 所示结果可以看出, Pdsimu1 略大于其理论值 Pd.

由图 6 可以看出, 两条实线分别代表在两种情形下统计得到的篡改检测概率的实验值, 它们都与图 6 中间那条虚线(篡改检测概率的理论值)很接近, 说明通过实验统计出的篡改检测概率的值与我们推导得出的理论值符合得较好, 进一步证实了本文推导的在一般区域篡改下的检测概率的正确性.

(ⅱ) 在拼贴攻击[9]下的篡改检测概率.

在拼贴攻击[9]下, 图像块X_i生成的 4 位水印与原始图像块不相同的个数, 与在一般区域篡改下相同, 见公式(4)—(8). 由拼贴攻击的特点可知, 其与一般区域篡改的区别在于, 当生成的$w_i(k)$不同时, 若$w_i(k)$位于篡改区域内, 则一定检测不出篡改, 若其位于篡改区域外, 则定能检测出篡改; 而当生成的$w_i(k)$相同时, 无论$w_i(k)$位于篡改区域内还是篡改区域外, 均检测不出篡改. 与在一般区域篡改下的分析类似, 4 个嵌入位均检测不到篡改的概率P_{Tm0}, 只能检测到一个篡改的概率P_{Tm1}, 及能检测到两个及两个以上篡改的概率P_{Tm2a}分别为

$$P_{Tm0} = p_{w0} + p_{w1} \times a + p_{w2} \times a^2 + p_{w3} \times a^3 + p_{w4} \times a^4 , \tag{15}$$

$$P_{Tm1} = p_{w1} \times (1-a) + p_{w2} \times C_2^1 (1-a)a + p_{w3} \times C_3^1 (1-a)a^2 + p_{w4} \times C_4^1 (1-a)a^3 , \tag{16}$$

$$P_{Tm2a} = 1 - P_{Tm0} - P_{Tm1} . \tag{17}$$

同理, 在拼贴攻击[6]下本文所提出算法的篡改检测概率P_{Td}为

$$P_{Td} = (1-P_{Tm0}) \times (1-(1-P_{Tm2a})^9 - C_9^1 P_{Tm2a} (1-P_{Tm2a})^8) . \tag{18}$$

采用与在一般区域篡改中类似的方法, 统计 P_{Td} 的实验值 PTdsimu0 和 PTdsimu1, 其中, PTdsimu0 在统计篡改块时, 只统计了篡改区域内的篡改块的个数, 而 PTdsimu1 将篡改区域及与其边界相邻的图像块中所有检测到篡改的块全部统计在内, 如图 7 所示.

图 7 中的PTd为在拼贴攻击[9]下检测概率的理论值, P0 为$(1-P_{Tm0})$表示未进行检测算法最后一步时的检测概率, 它们在图 7 中为中间的两条虚线. 这两条虚线十分接近, 这表明: 即使在拼贴攻击[9]下, 经过本文算法最后一步的处理后, 本文算法的篡改检测概率基本保持不变.

与在一般区域篡改下的分析类似, PTdsimu0 略小于检测概率的理论值, PTdsimu1 略大于检测概率的理论值, 它们分别对应为图 7 中的两条实线, 且与篡改检测概率的理论值(虚线)很接近, 这表明: 在拼贴攻击[9]下, 实验统计出的篡改检测概率与本文推导得出的理论值符合得较好, 进一步证实了本文的理论值的正确性.

由上面的分析可知, 在实际检测中, 一般使用 PTdsimu1 和 Pdsimu1 作为在拼贴攻击和一般区域篡改下的检测概率的实验值. 图 8 给出了在一般区域篡改下检测概率的理论值 P_d、实验值 Pdsimu1 和在拼贴攻击下的检测概率的理论值 P_{Td}、实验值 PTdsimu1.

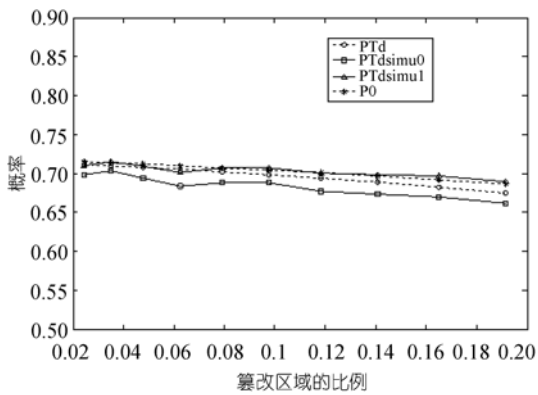

图 7　拼贴攻击下篡改检测概率的理论值及其实验值

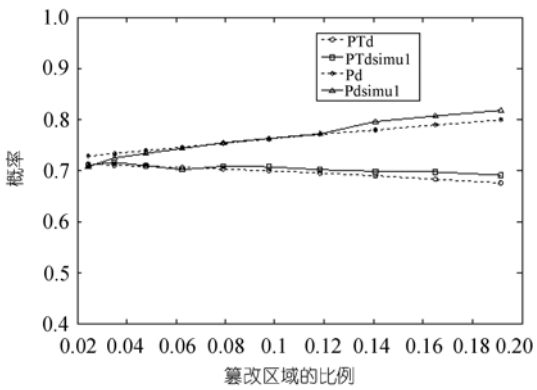

图 8　在一般区域篡改和拼贴攻击下检测概率的比较

由图 8 可以看出, 在一般区域篡改和拼贴攻击下, 本文所提出的算法均具有较高的篡改检测概率; 在拼贴攻击下, 随着篡改区域比例的增大, 算法的检测概率有所下降, 但即使在篡改

区域占整个图像的比例达到 0.2 时, 其检测概率仍然大于 0.65.

这些结果表明: ①采用水印提取算法最后一步操作后, 在一般区域篡改和拼贴攻击下, 算法的篡改检测概率基本不变, 这说明水印提取算法最后一步对算法的安全性没有影响. ②在一般区域篡改和拼贴攻击下, 算法的篡改检测概率都较高. 这说明算法的安全性高, 能以较大的概率检测到篡改. ③在拼贴攻击下, 算法的篡改检测概率下降不明显, 说明算法能够很好的抵抗拼贴攻击.

3.2 算法的篡改定位精度分析

从前面对 Li 算法的安全性分析可知: 一个算法只有在同时拥有高的篡改检测概率和低的虚警概率时才具有高的定位精度. 除了已经分析了的篡改检测概率, 有必要讨论本文所提出的算法在一般区域篡改和拼贴攻击下的虚警概率.

(ⅰ) 一般区域篡改下算法的虚警概率.

在一般区域篡改下非篡改区域的图像块的水印若落入篡改区域外, 则检测不出篡改; 若落入篡改区域内, 则有 0.5 的概率检测出篡改. 由此可知: 非篡改区域图像块中每一位 $w_i(k)$检测出篡改的概率为

$$p_{nd} = (1-a)\times 0 + a\times 0.5 = 0.5a\,. \tag{19}$$

图像块中 4 位均检测不出篡改的概率 P_{n0} 及只能检测出一位篡改的概率 P_{n1} 分别为

$$P_{n0} = (1-p_{nd})^4\,, \tag{20}$$

$$P_{n1} = C_4^1 p_{nd}(1-p_{nd})^3\,. \tag{21}$$

且易知能检测出两位及以上篡改的概率 P_{n2a} 为

$$P_{n2a} = 1-P_{n0}-P_{n1}\,. \tag{22}$$

与以上的讨论类似, 在一般的区域篡改下的虚警概率 P_{nx} 为

$$P_{nx} = (1-P_{n0})\times(1-(1-P_{n2a})^9 - C_9^1 P_{n2a}(1-P_{n2a})^8)\,. \tag{23}$$

P_{n0}, P_{n1}, P_{n2a} 的实验统计值 Pn0simu, Pn1simu, Pn2asimu 如图 9 所示.

图 9 中处于最上方的两条曲线为在一般区域篡改下, 非篡改区域的图像块的 4 个水印检测不出篡改的概率的理论值 Pn0(虚线)及实验值 Pn0simu(实线), 这两条曲线几乎重合, 表明其理论值和实验值符合得很好. 图 9 中的中间两条代表仅能检测出一个水印篡改概率的理论值和实验值的曲线, 最下方的两条曲线代表能检测出两位及以上篡改的理论值和实验值的曲线. 可以看出: 它们的理论值和实验值几乎重合. 这说明: 在一般区域篡改下非篡改区域的图像块的 4 个水印检测不出篡改, 仅能检测出一个水印篡改和能检测出两个及以上篡改的概率的理论值和实验值符合得非常好, 从而验证了我们理论推导的正确性.

P_{nx} 的实验统计值 Pxsimu0 和 Pxsimu1 如图 10 所示. 图 10 中最下方的虚线 Pnx 代表 P_{nx} 为虚警概率的理论值, 最上方的虚线 P0 代表$(1-P_{n0})$为未经过水印提取算法最后一步时的虚警概率的值; 两条实线分别代表在两种计算方法下本文算法虚警概率的实验值, 其中上方的实线为 Pxsimu0, 它在计算虚警概率时统计了篡改区域外的所有篡改块的个数, 由于与篡改区域边界相邻的图像块的虚警概率较大, 所以造成了 Pxsimu0 偏大, 故其位于上方.

由图 10 可见， 经过本文算法最后一步处理后, 检测算法的虚警概率大大下降, 且达到一个很低的水平, 这进一步说明: 在一般区域篡改下, 本文嵌入水印算法的最后一步处理能有效

地降低水印检测算法的虚警概率. 当将这些与篡改区域边界相邻的那些篡改块去除以后, 求出的概率 Pxsimu1 才较真实的反映了 P_{nx}, 故代表 Pxsimu1 的实线与理论值 Pnx 更接近.

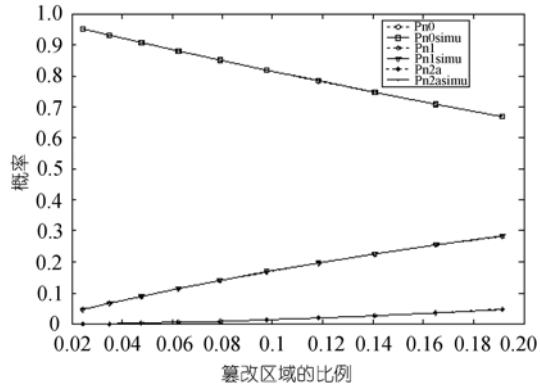

图 9　一般区域篡改下非篡改区域图像块检测不出篡改、检测出 1 位、检测出 2 位及以上篡改的概率的理论值和实验值

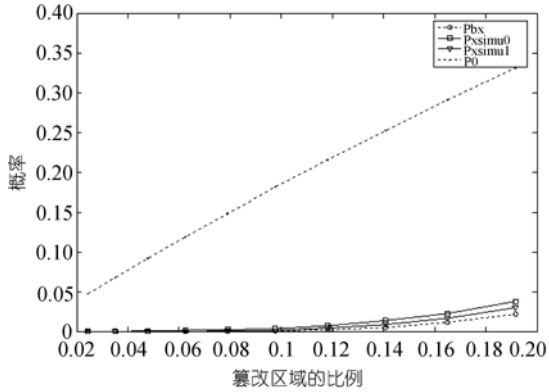

图 10　一般篡改下虚警概率的理论值及其实验值

(ⅱ) 拼贴攻击下的虚警概率.

在拼贴攻击下, 当篡改区域外的图像块所对应的 $w_i(k)$位于篡改区域外时, 此位检测不出篡改, 当篡改区域外的图像块所对应的 $w_i(k)$位于篡改区域内时, 若此 $w_i(k)$与进行拼贴攻击的原始图像中相同位置的图像块生成的水印相同时, 也检测不出篡改, 若不同, 则此位将检测出篡改. 根据这两个图像块生成的 4 个水印中不同的概率与在拼贴攻击下检测概率的讨论相同, 参见公式(4)—(8). 当生成的 $w_i(k)$不同时, 若 $w_i(k)$位于篡改区域内, 则一定能检测出篡改, 若其位于篡改区域外, 则一定检测不出篡改; 而当生成的 $w_i(k)$相同时, 无论 $w_i(k)$位于篡改区域内还是篡改区域外, 均检测不出篡改. 与在拼贴攻击下检测概率中的讨论相比较, 可知只需将公式(15)和(16)中的 a 换成$(1-a)$, 即可得到在拼贴攻击下篡改区域外的图像块检测不到篡改的概率 P_{Tx0} 和只能检测到一位篡改的概率 P_{Tx1} 如下:

$$P_{Tx0} = p_{w0} + p_{w1}\times(1-a) + p_{w2}\times(1-a)^2 + p_{w3}\times(1-a)^3 + p_{w4}\times(1-a)^4, \tag{24}$$

$$P_{Tx1} = p_{w1}\times a + p_{w2}\times C_2^1 a(1-a) + p_{w3}\times C_3^1 a(1-a)^2 + p_{w4}\times C_4^1 a(1-a)^3. \tag{25}$$

用 P_{Tx2a} 表示能检测到不小于 2 位的篡改, 显见

$$P_{Tx2a} = 1 - P_{Tx0} - P_{Tx1}. \tag{26}$$

由水印检测算法可知, 在拼贴攻击下的虚警概率 P_{Tx} 为

$$P_{Tx} = (1-P_{Tx0})\times(1-(1-P_{Tx2a})^9 - C_9^1 P_{Tx2a}(1-P_{Tx2a})^8). \tag{27}$$

然后通过实验统计 P_{Tx} 的实验值 PTxsimu0 和 PTxsimu1, 其中 PTxsimu0 在统计篡改块时, 统计了篡改区域外的所有篡改块的个数, 而 PTxsimu1 在统计篡改块时, 忽略了与篡改区域边界相邻的那些篡改块. 然后将 P_{Tx}, PTxsimu0, PTxsimu1, $(1-P_{Tx0})$画在图 11 中.

图 11 中最下方的虚线 PTx 代表 P_{Tx} 为虚警概率的理论值, 最上方的虚线 P0 表示$(1-P_{Tx0})$为未经过算法最后一步时的虚警概率的值. 由图可知: 经过算法最后一步后, 算法的虚警概率下降到了一个很低的水平, 说明水印提取算法中最后一步在拼贴攻击下同样能够有效的降低

算法的虚警概率.

图 11 中的两条实线为用两种方法统计出的算法的虚警概率的实验值. 与上面在一般区域篡改下的虚警概率的讨论可知: 实验值 PTxsimu0 大于理论值 P_{nx}, PTxsimu1 才较真实的反映了 P_{Tx}, 它与 P_{Tx} 更接近.

比较在一般区域篡改和拼贴攻击下算法的虚警概率. 由上面的分析可知, Pxsimu1 和 PTxsimu1 更能代表算法在一般区域篡改下虚警概率的理论值 P_{nx} 和在拼贴攻击下的虚警概率的理论值 P_{Tx}, 故将这 4 条曲线画在图 12 中.

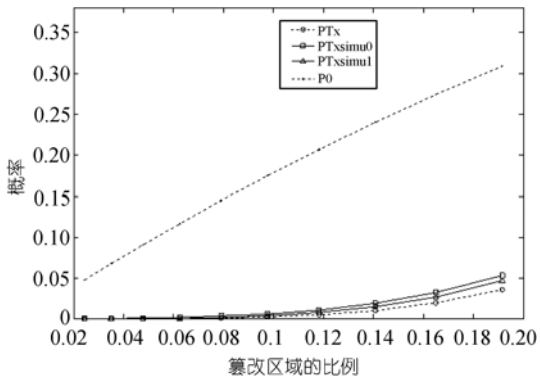

图 11 拼贴攻击下虚警概率的理论值及其实验值

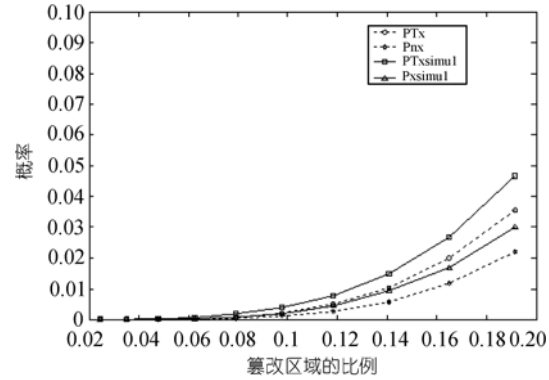

图 12 在一般区域篡改和拼贴攻击下虚警概率的比较

图 12 最上方的虚线 PTx 代表 P_{Tx}, 为算法在拼贴攻击下的虚警概率的理论值, 下方的虚线 Pnx 代表 P_{nx}, 为算法在一般区域篡改下虚警概率的理论值. 由图可知: 在一般区域篡改和拼贴攻击下算法的虚警概率极低; 在拼贴攻击下, 随着篡改区域比例的增大, 算法的虚警概率有所升高, 但即使在篡改区域占整个图像的比例达到 0.2 时, 算法的虚警概率仍低于 0.05.

由以上对一般区域篡改和拼贴攻击下的虚警概率的理论分析及实验验证可以得到以下 3 点: ①采用水印提取算法最后一步操作后, 在一般区域篡改和拼贴攻击下, 算法的虚警概率大大下降, 达到一个很低的水平. ②在一般区域篡改和拼贴攻击下, 算法的虚警概率都很低. ③在拼贴攻击下, 算法的虚警概率上升不明显.

结合本节对一般区域篡改和拼贴攻击下篡改检测概率和虚警概率的讨论可知: ①在采用水印提取算法最后一步的操作后, 无论是在一般区域篡改还是在拼贴攻击下, 算法的篡改检测概率基本保持不变, 具有高的篡改检测概率, 算法的虚警概率大大降低, 达到一个很低的水平. 这说明本算法的最后一步十分有效的提高了算法的定位精度, 使算法在一般区域篡改和拼贴攻击下都具有很高的定位精度. ②在拼贴攻击下, 算法的检测概率下降很少, 虚警概率的提高也很少, 其篡改定位精度基本不变.

由上面对本文所提出的算法的安全性和篡改定位精度的分析可知, 本文所提出的算法具有较高的定位精度, 它保护了图像的平滑区域, 并且具有很高的安全性, 能够完全抵抗拼贴攻击.

4 实验仿真

在实验仿真中, h'取为 18, 即在每个图像块的第 19—22 这 4 个中频分量上嵌入水印, JPEG

压缩的质量因子 Q 取 80. 对嵌入水印的 Peppers 图像进行区域篡改后, 进行两次检测, 在第一次检测中, 没有进行水印提取算法的最后一步, 即只要图像块 X_i 所对应的 $v(i)>0$, 就将此图像块标记为已篡改; 在第二次检测中, 采用算法最后一步降低图像的虚警概率. 这两次检测的结果分别如图 13(a)和(b)所示.

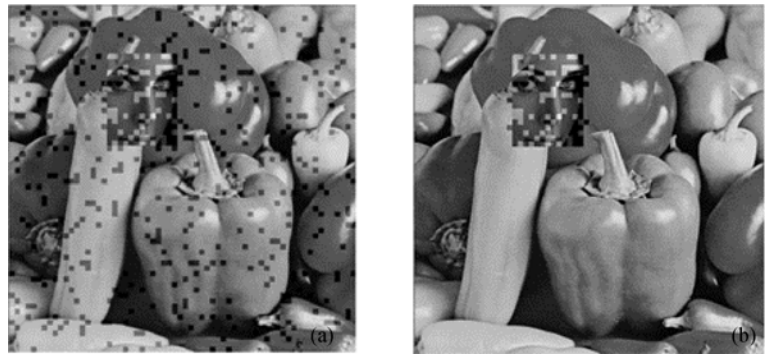

图 13　比较降低虚警概率后的效果

(a) 未进行降低虚警概率的操作; (b) 进行了降低虚警概率的操作

由图 13(a)可以看出, 在未进行水印提取算法最后一步降低虚警的操作时, 篡改区域的外部有大量的图像块被检测出篡改, 同时篡改区域内部的图像块大部分都被检测出了篡改. 在进行了水印提取算法的最后一步后, 由图 13(b)可以看出, 除了篡改区域的外边界外, 其他所有的虚警块都被标记为未篡改, 且在图 13(a)中篡改区域内部被正确检测出的篡改块, 在图 13(b)中都基本得到了保留, 这说明本算法的最后一步在保持篡改检测概率基本不变的情形下, 极大地降低了虚警概率, 有效地提高了算法的定位精度. 且由图 13(b)可以看出, 在篡改区域内, 大部分篡改的图像块都被正确地检测出来了, 算法的篡改检测概率高, 而在篡改区域外部, 只有很少的图像块检测出篡改, 算法的虚警概率低, 故算法的定位精度高. 这与本文在第 3 节对算法的定位精度的分析所得出的结论一致.

下面再分别对嵌入水印的 Clock 图像和 Peppers 图像进行与第 2 节对 Li 的算法所做的相同的伪造攻击, 然后分别对其进行认证, 实验结果如图 14 和 15 所示.

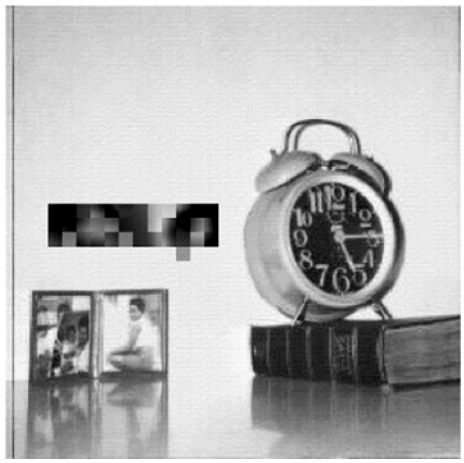

图 14　对平滑区域篡改后的检测效果

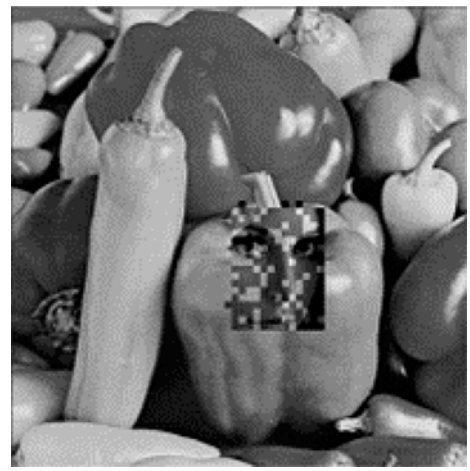

图 15　拼贴攻击下的检测效果

图 14 是按照与第 1 节的图 3 相同的方式对Clock中的平滑区域进行的伪造攻击, 由图 14 可以看出, 在本文所提出的算法中, 篡改区域被精确的定位, 这说明本文的算法能保护图像中的所有区域. 图 15 是与图 4 相同的拼贴攻击, 可以看出, 遭受拼贴攻击的大部分图像块均被检测出篡改. 通过与图 13(b)的一般区域篡改的检测结果的比较可以看出, 与一般区域篡改相比, 在拼贴攻击下, 算法的篡改检测概率没有明显的下降, 虚警概率也基本保持不变, 故在拼贴攻击 [9]下, 算法基本保持了和在一般区域篡改下的高的定位精度, 算法抵抗拼贴攻击的能力很强, 验证了在上一节中的结论.

5 结论

本文首先讨论了 Li 的算法存在的缺点及其可能带来的安全隐患, 在此基础上提出了一种新的用于 JPEG 图像认证的具有高定位精度和高安全性的水印算法. 该算法在水印嵌入过程中对每个图像块固定选取 4 个中频系数作为水印的嵌入位, 并利用剩余的 DCT 系数生成 4bits 水印, 然后基于不同的密钥选取每比特水印信息嵌入的图像块, 这样就使得每个图像块的 4bits 水印被随机地嵌入到了其他 4 个图像块中, 使得图像块之间的关系是非确定性的, 提高了算法的安全性, 使其能抵抗类似于拼贴攻击这样的伪造攻击. 在认证时根据图像块水印被篡改的个数并结合其九邻域判定该图像块是否被篡改. 理论分析及实验仿真结果表明该算法不仅拥有高的定位精度, 且具有很高的安全性.

参 考 文 献

1 Liu R Z, Tan T N. Watermarking for digital images. In: Proceedings of ICSP'98, Beijing, China, 1998. 944—947

2 Sun Z W, Feng D G. A multiplicative watermark detection algorithm for digital images in the DCT Domains. J Softw, 2005, 16(10): 1798—1804 [DOI]

3 Jin C, Peng J X. A robust detection method of blind digital watermark based on image projective sequence. J Softw, 2005, 16(2): 295—302 [DOI]

4 Ho C K, Li C T. Semi-fragile watermarking scheme for authentication of JPEG images. In: Proceedings of ITCC'2004, Apr, 2004, Vol 2: 7—11

5 Lin C Y, Chang S F. A robust image authentication method distinguishing JPEG compression from malicious manipulation. IEEE Trans on Circuits and Systems of Video Technology, 2001, 11(2): 153—168 [DOI]

6 Li C-T. Digital fragile watermarking scheme for authentication of JPEG images. IEE Proceedings-Vision, Image, and Signal Processing, Dec, 2004. 460—466

7 Fridrich J, Goljan M, Du R. Invertible authentication watermark for JPEG images. In: Proceedings of ITCC'2001, Las Vegas, NV, USA, Apr, 2001. 446—449

8 Cox I J, Miller M L. Review of watermarking and the importance of perceptual modeling. In: Bernice E R, Thrasyvoulos N P, eds. Human Vision and Electronic Imaging II, Vol 3016. Bellingham: SPIE Press, 1997. 92—99

9 Matthew H, Nasir M. Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. IEEE Trans on Image Processing, March, 2000. 432—441