中国科学 E 辑：信息科学 2008 年 第 38 卷 第 4 期：533 ~ 552
www.scichina.com info.scichina.com
《中国科学》杂志社
SCIENCE IN CHINA PRESS

一种高定位精度的可恢复水印算法

和红杰[①*]，张家树[①]，陈帆[②]

① 西南交通大学信号与信息处理四川省重点实验室，成都 610031；
② 西南交通大学信息安全与国家计算网格实验室，成都 610031
* E-mail: hehojie@mars.swju.edu.cn

收稿日期: 2007-02-12；接受日期: 2007-04-23
西南交通大学博士生创新基金(2007)、教育部新世纪优秀人才支持计划(批准号: NECT-05-0794)和四川省应用基础研究(批准号: 2006J13-10)资助项目

摘要 针对现有可恢复水印算法定位精度低和虚警概率高的问题，提出一种高定位精度的可恢复脆弱水印算法，分别推导给出了随机篡改与区域篡改条件下算法的虚警概率和漏警概率，并定义了衡量篡改恢复图像质量的 4 个指标. 基于 2×2 图像块生成水印信息不仅能有效提高算法的定位精度，而且降低了算法对随机篡改的虚警概率；基于密钥随机生成水印嵌入位置并结合图像块 8-邻域被篡改的情形，使得算法对区域篡改的虚警概率接近于 0. 与现有可恢复水印算法相比，不仅有效解决了可恢复水印算法的篡改定位问题，而且提高了可恢复水印算法对随机噪声的鲁棒性；同时，也为定量分析认证水印算法的性能提出一种客观的评价指标.

关键词
数字水印
篡改恢复
虚警概率
漏警概率

计算机技术的飞速发展与信息媒体的数字化以及各种图像编辑软件的出现，使得攻击者可以毫不费力且不留痕迹地篡改数字图像的内容. 中国有句古话: “耳听为虚，眼见为实”，亲眼所见是人们区分真伪的依据，但是多媒体数据的易操作性使人们通过眼睛难辨真伪. 如何检测数字图像的完整性、真实性等问题引起了研究者的广泛关注. 传统的数字签名技术在鉴别数字图像真实性时，存在不能适应图像处理需求和不能定位图像被篡改位置等缺陷[1]. 近年来，基于数字水印的数字图像认证技术已成为国内外的前沿性研究课题.

基于数字水印的图像认证技术分为精确认证(hard authentication)和模糊认证(soft authentication)两类[2]，它们要解决的核心问题是鉴别数字图像的真实性(篡改检测)和定位数字图像被篡改的位置(篡改定位)[3, 4]，并由此推断图像被篡改的程度和方式[5~7]. 1999 年Fridrich等人[8]首次提出了一种自嵌入水印算法，该算法不仅能定位图像被篡改的位置，而且还能近似恢复被篡改图像块的内容，充分展示了基于数字水印的图像认证技术优势. 近几年来，出现了不少有关可恢复

水印算法的研究报道 [9~16]. 总结起来, 可恢复水印算法有以下特点: 通过分块实现篡改定位; 基于图像块内容生成水印以实现篡改恢复; 加密水印信息并基于密钥选取水印嵌入位置来提高算法的安全性; 兼顾不可见性和水印嵌入容量, 水印信息一般被嵌入在数字图像的低位平面LSBs(less significant bits). 在现有可恢复水印算法中, 对"篡改恢复图像质量"的讨论多是在"水印信息不被篡改"的条件下进行的 [8~13], 而实际应用中被测图像有少量水印信息改变是完全可能的: 一方面通过网络传输的数字图像不可避免地会受信道噪声(随机篡改)的影响; 另一方面, 攻击者在篡改图像内容(区域篡改)时也不可能刻意保持水印信息不变以利于篡改恢复. 已有研究者注意到该问题 [14~16]并提出了相应的解决方案, 例如, Lin等人 [16]提出利用CRC(cyclic redundancy check)来检测水印信息是否被篡改. 只有当水印信息没有被篡改时, 隐藏于其中的压缩信息才用于恢复对应图像块的原始内容. 该方法有效避免了使用错误的水印信息恢复未被篡改的图像块, 但同时也放弃了对某些篡改块的恢复, 从而使被恢复图像的可信性大大降低. 因此, Lin等提出的"提高篡改恢复质量"方法是不可行的.

通过对现有可恢复水印算法深入研究发现, 以下两个方面的问题是导致部分水印信息被篡改时, 现有可恢复水印算法恢复质量不高的主要原因:

1) 虚警概率高. 可恢复水印算法中, 基于图像块内容生成的水印信息不是嵌入图像块自身之中, 而是嵌入在另一图像块的 LSBs. 假设图像块 X_1 的水印信息嵌入在图像块 X_2 中, 当 X_2 被篡改, 根据现有可恢复水印的篡改检查算法, 图像块 X_1 和 X_2 将同时被检测, 即真实的图像块 X_1 误判为被篡改(虚警).

2) 篡改定位精度低. 现有可恢复水印算法中, 图像块大小至少为 8×8, 算法的篡改定位精度低.

当被测图像有水印信息篡改时, 上述两个问题使现有可恢复水印算法的篡改恢复质量急剧下降. 一方面, 对误判为篡改(虚警)的真实图像块的"恢复"无疑严重降低了篡改恢复图像的质量; 另一方面, 为提高可恢复水印算法抵抗各种伪造攻击 [17~19]的能力, 基于图像块内容生成的水印信息一般采用成熟的加密体制(如DES, RSA等)加密后再嵌入其他图像块的LSBs. 由加密算法的特性 [20]可知, 只要图像块LSBs的少量(即使 1 bit)信息被改变, 就无法通过解密得到原始的水印信息. 因此, 现有可恢复水印算法中, 1 bit水印信息(如信道噪声的影响)的改变就会导致一个图像块(8×8)不能进行有效恢复. 显然, 图像块越大造成的误恢复区域也越大.

针对上述问题, 本文从提高可恢复水印算法的定位精度出发, 以降低虚警概率为主线, 提出了一种高定位精度的可恢复水印算法. 该算法首先将原始图像分为 2×2 的图像块, 然后把利用图像块内容生成的水印信息基于密钥随机嵌入到其他图像块的LSBs. 认证时, 通过比较 8-邻域块的内容与水印的匹配情形进行篡改定位, 以降低可恢复水印算法的虚警概率. 同时, 为定量地分析算法的篡改恢复性能, 本文定义了 4 个衡量篡改恢复图像质量的指标及其与虚警/漏警概率的关系. 分别推导了本文和张鸿宾等人 [11]算法在随机篡改与区域篡改条件下的虚警概率和漏警概率, 比较了它们在相同篡改条件下的篡改恢复质量. 理论分析和实验仿真结果表明, 该算法不仅提高了可恢复水印算法的篡改定位精度, 而且有效降低了被测图像有水印信息篡改时的虚警概率, 从而大大提高可恢复水印算法的篡改恢复质量.

1 虚/漏警概率和篡改恢复质量评价

“篡改恢复质量”无疑是衡量可恢复水印算法优劣的最直接标准, 而算法的篡改检测性能对篡改恢复质量又起着至关重要的作用. 因此, 首先定义可恢复水印算法的虚/漏警概率, 并推导衡量“篡改恢复质量”的 4 个指标及其与虚/漏警概率之间的关系, 然后定量地讨论现有可恢复水印算法的虚/漏警概率. 为便于描述, 首先简单介绍文中使用的概率论知识和相应的符号表示.

1.1 概率论基础

假设随机变量 T, T_1 相互独立, 且分别服从参数为(n, p)和(n, p_1)的二项分布, 即 $T \sim B(n, p)$, $T_1 \sim B(n, p_1)$, 则:

(1) T 等于 a 的概率 $P_{=a}(n, p)$:

$$P_{=a}(n, p) = \binom{n}{a} p^a (1-p)^{n-a},\ a \in [0, n]. \tag{1}$$

(2) T 小于 a 的概率 $P_{<a}(n, p)$:

$$P_{<a}(n, p) = \sum_{b=0}^{a-1} \binom{n}{b} p^b (1-p)^{n-b}. \tag{2}$$

(3) T 小于 T_1 的概率 $P_{<}(n, p, p_1)$:

$$P_{<}(n, p, p_1) = \sum_{a=1}^{n} \sum_{b=0}^{a-1} \binom{n}{a} \binom{n}{b} p_1^a (1-p_1)^{n-a} p^b (1-p)^{n-b}. \tag{3}$$

其中, $a \in [0, n]$ 为整数, 上述公式的证明过程, 可以在任一本概率论教材 [21]中找到. 下文公式推导过程中, 直接用符号$P_{=a}(n, p)$, $P_{<a}(n, p)$和$P_{<}(n, p, p_1)$分别表示相应的概率.

1.2 虚/漏警概率与篡改恢复质量

可恢复水印算法作为认证水印算法的一种, 其篡改检测性能可以通过考察算法的漏警概率(PFA: probability of false acceptance)和虚警概率(PFR: probability of false rejection)来衡量 [22]. 所谓漏警概率是指图像块被篡改而算法不能检测出该篡改的概率; 虚警概率是指图像块没有被篡改, 而算法判定其被篡改的概率.

被测图像中的图像块可分为两个集合: 篡改图像块(记为 H_0)和真实图像块(记为 H_1). 显然, $H_0 \cap H_1 = \Phi$, $H_0 \cup H_1 = H$ (H 为被测图像中所有图像块的集合), 故可恢复水印算法的 PFA 和 PFR 分别为:

$$\begin{cases} P_{fa} = 1 - P_{D|H_0}, \\ P_{fr} = P_{D|H_1}. \end{cases} \tag{4}$$

其中, $P_{D|H_0}$ 表示图像块 $Y_i \in H_0$ 条件下算法的检测概率, $P_{D|H_1}$ 表示图像块 $Y_i \in H_1$ 条件下算法的检测概率.

为了便于比较不同篡改条件下恢复图像的质量, 定义如下评价指标(设 P_{H_0} 和 P_{H_1} 分别表示被篡改和真实图像块在整个图像中所占的比例, 显然 $P_{H_0}+P_{H_1}=1$).

● 漏检率和虚检率: 漏检和误检图像块在被测图像中的比例, 即

$$\begin{cases}\Gamma_{fa}=P_{fa}P_{H_0},\\ \Gamma_{fr}=P_{fr}P_{H_1}.\end{cases} \tag{5}$$

● 检测率: 被检测图像块在被测图像中的比例, 被检测图像块包括被检测的篡改块和没有篡改而被误检的图像块, 即

$$\Gamma_D=(1-P_{fa})P_{H_0}+P_{fr}P_{H_1}. \tag{6}$$

● 误恢复率: 不真实图像块在篡改恢复图像中的比例. 恢复图像中的不真实图像块可分为 3 种: ① 图像块被篡改, 但算法没能检测出该篡改; ② 图像块被篡改且算法检测出该篡改, 但其相应水印信息被改变; ③ 图像块没有被篡改, 但其相应的水印信息被改变(虚警). 因此, 误恢复率为

$$\Gamma_F=P_{fa}P_{H_0}+(1-P_{fa})P_{H_0}P_W+P_{fr}P_{H_1}. \tag{7}$$

显然, 在相同的篡改条件下, Γ_{fa}, Γ_{fr}, Γ_D 和 Γ_F 的值越小, 可恢复水印算法的性能越好.

1.3 现有可恢复水印算法的虚/漏警概率

典型的可恢复水印算法 [11]包括两部分.

1) 水印嵌入.

① 将原始图像 X 分为大小为 $m_b\times n_b$ 的图像块 X_i(i=1,2, …, N), 将图像块 X_i 的 LSBs(记作 X_{Li})置 0, 生成该图像块内容 X_{Ci}, 即 $X_i=X_{Ci}+X_{Li}$. 将 X_{Ci} 按一定比例压缩生成的二值编码作为图像块 X_i 的水印信息 $w_i=G(X_{Ci})$, 其中 $G(\cdot)$表示对图像块的压缩编码过程.

② 给定加密密钥 k 和生成水印嵌入位置的密钥 k_1, 将 $E_k(w_i)$ 嵌入到图像块 $X_{f(i)}$ 的 LSBs 生成含水印图像块 $X^w_{f(i)}$, 其中 $f(i)$为水印嵌入位置生成函数, $E_k(\cdot)$ 为利用密钥 k 的加密过程.

2) 篡改检测及恢复

①对含水印图像的每个图像块 $Y_i=Y_{Ci}+Y_{Li}$, 如果图像没有被篡改则 $G(Y_{Ci})=G(X^{\mathrm{w}}_{Ci})$, $D_k(Y_{Li})=D_k(X^w_{Li})$, 其中 $D_k(\cdot)$ 为 $E_k(\cdot)$ 的解密过程.

② 如果 $G(Y_{Ci})=D_k(Y_{Lf(i)})$, 判定图像块 Y_i 没有被篡改, 否则判定 Y_i 被篡改并利用相应的水印信息 $Y_{Lf(i)}$ 重构图像块 Y_i 的内容.

利用上述可恢复水印算法生成的含水印图像中, 含被测含水印图像块 $Y_i=Y_{Ci}+Y_{Li}$ (i=1,2, …, N)的相应的水印信息为 $Y_{Lf(i)}$, 设 $P_{C|H_0}$, $P_{L|H_0}$ 和 $P_{W|H_0}$ 分别表示 $Y_i\in H_0$ 条件下 Y_{Ci}, Y_{Li} 和 $Y_{Lf(i)}$ 发生变化的概率, 设 $P_{C|H_1}$, $P_{L|H_1}$ 和 $P_{W|H_1}$ 分别为 $Y_i\in H_1$ 条件下 Y_{Ci}, Y_{Li} 和 $Y_{Lf(i)}$ 发生变化的概率. 显然, $P_{C|H_1}=P_{L|H_1}=0$. 在可恢复水印算法中, 图像块的水印信息基于密钥嵌

入在其他图像块的 LSBs, 因此无论图像块是否被篡改, 其相应水印信息被篡改的概率近似相同且与被测图像中低位被篡改的概率成正比, 即

$$P_W = P_{W|H_0} = P_{W|H_1} = P_{L|H_0}P_{H_0} + P_{L|H_1}P_{H_1} = P_{L|H_0}P_{H_0}. \tag{8}$$

设 $P_{U|H_0}$ 和 $P_{U|H_1}$ 分别表示图像块 Y_i 属于 H_0 和 H_1 条件下 $G(Y_{Ci}) \neq D_k(Y_{Lf(i)})$ (unequal)的概率. 根据图像块内容 Y_{Ci} 和相应水印信息 $Y_{Lf(i)}$ 是否被篡改, 分为以下 4 种情形:

1) Y_{Ci} 和 $Y_{Lf(i)}$ 都没有改变: 由水印算法可知, 如果 Y_{Ci} 和 $Y_{Lf(i)}$ 都没有被篡改, 则 $G(Y_{Ci}) = D_k(Y_{Lf(i)})$, 即

$$P_{00}\{G(Y_{Ci}) \neq D_k(Y_{Lf(i)})\} = 0. \tag{9}$$

2) Y_{Ci} 不改变而 $Y_{Lf(i)}$ 改变: $Y_{Lf(i)}$ 改变意味着 $D_k(Y_{Li}) \neq D_k(X_{Li}^w)$, 而 $G(Y_{Ci}) = G(X_{Ci}^w)$, 因此图像块内容不变而相应水印信息改变时一定有 $G(Y_{Ci}) \neq D_k(Y_{Lf(i)})$, 即

$$P_{01}\{G(Y_{Ci}) \neq D_k(Y_{Lf(i)})\} = 1. \tag{10}$$

3) Y_{Ci} 改变而 $Y_{Lf(i)}$没有改变: 此时 $G(Y_{Ci})$ 与 $G(X_{Ci}^w)$ 可能相等也可能不等, 如果 $G(Y_{Ci}) = G(X_{Ci})$, 由水印生成算法可知 Y_i 近似等于 X_i (即 $Y_i \approx X_i$), 因此从保证图像块真实性的角度, 可以认为

$$P_{10}\{G(Y_{Ci}) \neq D_k(Y_{Lf(i)})\} \approx 1. \tag{11}$$

4) Y_{Ci} 和 $Y_{Lf(i)}$ 都改变: 由加密算法的性质可知, $D_k(\cdot)$ 的取值在整个明文空间均匀地分布, 设图像块的二值水印信息长度为 L_w, 则改变后的水印信息 $P\{D_k(Y_{Lf(i)}) = D_k(X_{Lf(i)}^w)\} = 1/2^{L_w}$. 假设图像内容被随机篡改, 可以认为 $P\{G(Y_{Ci}) = G(X_{Ci}^w)\} = 1/2^{L_w}$, 则

$$P_{11}\{G(Y_{Ci}) \neq D_k(Y_{Lf(i)})\} = 1 - \binom{L_w}{1}\left(\frac{1}{2^{L_w}}\right)^2 = 1 - \frac{1}{2^{L_w}}. \tag{12}$$

根据全概率公式并结合公式(8)可知, 图像块被篡改时 $G(Y_{Ci}) \neq D_k(Y_{Lf(i)})$ 的概率 $P_{U|H_0}$ 为

$$\begin{aligned} P_{U|H_0} &= (1-P_{C|H_0})(1-P_{W|H_0})P_{00}\{G(Y_{Ci}) \neq D_k(Y_{Lf(i)})\} + (1-P_{C|H_0})P_{W|H_0}P_{01}\{G(Y_{Ci}) \neq D_k(Y_{Lf(i)})\} \\ &\quad + P_{C|H_0}(1-P_{W|H_0})P_{10}\{G(Y_{Ci}) \neq D_k(Y_{Lf(i)})\} + P_{C|H_0}P_{W|H_0}P_{11}\{G(Y_{Ci}) \neq D_k(Y_{Lf(i)})\} \\ &\approx (1-P_{C|H_0})P_{W|H_0} + P_{C|H_0}(1-P_{W|H_0}) + P_{C|H_0}P_W\left(1-\frac{1}{2^{L_w}}\right) \\ &= P_{W|H_0} + P_{C|H_0} - P_{W|H_0}P_{C|H_0}\left(1+\frac{1}{2^{L_w}}\right) \\ &= P_{L|H_0}P_{H_0} + P_{C|H_0} - P_{L|H_0}P_{H_0}P_{C|H_0}\left(1+\frac{1}{2^{L_w}}\right). \end{aligned} \tag{13}$$

相应地, 图像块不被篡改时 $G(Y_{Ci}) \neq D_k(Y_{Lf(i)})$ 的概率 $P_{U|H_1}$ 为

$$\begin{aligned}P_{U|H_1} &= (1-P_{C|H_1})(1-P_{W|H_1})P_{00}\{G(Y_{Ci}) \neq D_k(Y_{Lf(i)})\} + (1-P_{C|H_1})P_{W|H_1}P_{01}(G(Y_{Ci}) \neq D_k(Y_{Lf(i)})) \\ &\quad + P_{C|H_1}(1-P_{W|H_1})P_{10}\{G(Y_{Ci}) \neq D_k(Y_{Lf(i)})\} + P_{C|H_1}P_{W|H_1}P_{11}\{G(Y_{Ci}) \neq D_k(Y_{Lf(i)})\} \\ &= P_{W|H_1}P_{01}\{G(Y_{Ci}) \neq D_k(Y_{Lf(i)})\} \\ &= P_{W|H_1} = P_{L|H_0}P_{H_0}.\end{aligned} \tag{14}$$

根据篡改检测算法可知, 现有可恢复水印算法的篡改检测概率等于 $G(Y_{Ci}) \neq D_k(Y_{Lf(i)})$ 的概率, 即 $P_{D|H_0} = P_{U|H_0}$ 和 $P_{D|H_1} = P_{U|H_1}$. 将(8), (13)和(14)式代入(4)式, 则现有可恢复水印算法的漏警概率和虚警概率可分别表示为

$$\begin{cases}P_{fa} = 1 - P_{C|H_0} - P_{L|H_0}P_{H_0} + P_{L|H_0}P_{H_0}P_{C|H_0}\left(1+\dfrac{1}{2^{L_w}}\right), \\ P_{fr} = P_{L|H_0}P_{H_0}.\end{cases} \tag{15}$$

下面分别讨论区域篡改和随机篡改时, 现有可恢复水印算法($m_b \times n_b = 8\times 8$, 水印信息嵌入在最低位, 即 L_w=64)的虚/漏警概率与被篡改比例的关系.

- 区域篡改(area tampering)

设被篡改区域占整个图像的比例为 p, 则区域篡改(area tampering)条件下被测图像中图像块被篡改比例 $P_{H_0|A} \approx p$. 假定被篡改像素的每个比特是否变化的可能性相同, 则被篡改图像块的 Y_{Li} 和 Y_{Ci} 发生改变的概率分别为 $P_{L|H_0A} = 1-1/2^{64} \approx 1$ 和 $P_{C|H_0A} = 1-1/2^{448} \approx 1$. 将其代入(15)式, 得到现有可恢复水印算法在区域篡改条件下的虚/漏警概率分别为

$$\begin{cases}P_{fa|A} \approx \dfrac{p}{2^{64}} < \dfrac{1}{2^{64}} \to 0, \\ P_{fr|A} = p - \dfrac{p}{2^{64}} \approx p.\end{cases} \tag{16}$$

- 随机篡改(random tampering)

所谓随机篡改是指被测图像中每个像素被篡改的概率相等(如信道噪声对被测图像的污染). 假设被测图像中每个像素被篡改的概率为 p, 图像块中只要有一个像素被篡改就认为该图像块被篡改, 此时被测图像中图像块被篡改概率为 $P_{H_0|R} = 1-(1-p)^{L_w}$. 因此当图像块被随机篡改时, 被篡改图像块的 Y_{Li} 和 Y_{Ci} 发生改变的概率分别为

$$\begin{cases}P_{C|H_0R} = \left(\dfrac{1}{P_{H_0|R}}\right)\displaystyle\sum_{i=1}^{64} P_{=i}(64, p)\left(1 - P_{=0}(7i, 0.5)\right) \approx 1, \\ P_{L|H_0R} = \left(\dfrac{1}{P_{H_0|R}}\right)\displaystyle\sum_{i=1}^{64} P_{=i}(64, p)\left(1 - P_{=0}(i, 0.5)\right).\end{cases} \tag{17}$$

因此, 现有可恢复水印算法在随机篡改条件下的虚/漏警概率分别为

$$\begin{cases} P_{fa|R} = \dfrac{P_{fr|R}}{2^{64}} < \dfrac{1}{2_{64}} \to 0, \\ P_{fr|R} = \displaystyle\sum_{i=1}^{64} P_{=i}(64, p)\left(1 - \frac{1}{2^i}\right). \end{cases} \tag{18}$$

由上述讨论可以看出, 无论是随机篡改还是区域篡改, 现有可恢复水印算法的漏警概率都趋近于 0, 而虚警概率则与被测图像中水印信息被篡改的比例正相关. 因此, 如何降低虚警概率, 是可恢复水印算法走向实际应用必须解决的一个关键问题.

2 定位精确的可恢复水印算法

本文要解决的一个关键问题是如何降低可恢复水印算法的虚警概率. 算法基于密钥随机生成图像块的水印嵌入位置, 为解决这一问题提供了可能. 图 1 给出了部分图像块的水印随机嵌入位置示意图. 假设区域 A 中 16 个图像块的水印信息随机嵌入在图 1 的 16 个灰色图像块中(记为集合 B), 即对 $\forall Y_i \in A, Y_{f(i)} \in B$. 相应地, 图 1 中的 16 个黑色图像块(记为集合 C)的水印信息被嵌入在区域 A 中, 即 $\forall Y_j \in C, Y_{f(j)} \in A$. 当区域 A 被篡改时, 由于集合 C 中图像块对应的水印信息被改变, 区域 A 和集合 C 中的图像块都将被检测, 而集合 B 中的图像块不会被检测. 要想降低虚警概率, 必须将被篡改区域 A 和实际上未被篡改的集合 C 区分开. 由图 1 可以看出:

- 对 $\forall Y_i \in A$, $Y_{f(i)} \in B$, 即 Y_i 周围被篡改图像块的个数大于 $Y_{f(i)}$ 周围被篡改图像块的个数;
- 对 $\forall Y_j \in C$, $Y_{f(j)} \in A$, 即 Y_j 周围被篡改图像块的个数小于 $Y_{f(j)}$ 周围被篡改图像块的个数.

因此, 通过比较被测图像块与其水印嵌入块的 8-邻域中被篡改图像块的个数, 能够将区域 A 和集合 C 中被检测的图像块区分开, 从而降低可恢复水印算法的虚警概率.

2.1 现有可恢复水印算法的虚/漏警概率

与传统的可恢复水印算法相似, 水印嵌入及篡改检测算法都是以图像块为单位进行的. 为提高算法的定位精度, 本文将原始图像 $X_{m\times n}$ 分为大小为 2×2 的图像块 X_i(i=1,2, …, N), 为便于算法描述, 图 2 给出了块编号及结构示意图. 图像块 X_i(i=1,2,…, N)为

$$X_i = \begin{bmatrix} x_{i0} & x_{i1} \\ x_{i2} & x_{i3} \end{bmatrix}. \tag{19}$$

与图像块 X_i 相邻的 8 个图像块定义为该图像块的 8-邻域(记为 $N_8(X_i)$), 按图 2 中图像块的编号方式, 图像块 X_i 的 8-邻域为

$$N_8(X_i) = \{X_{i-n/2-1}, X_{i-n/2}, X_{i-n/2+1}, X_{i-1}, X_{i+n/2-1}, X_{i+n/2}, X_{i+n/2+1}, X_{i+1}\}. \tag{20}$$

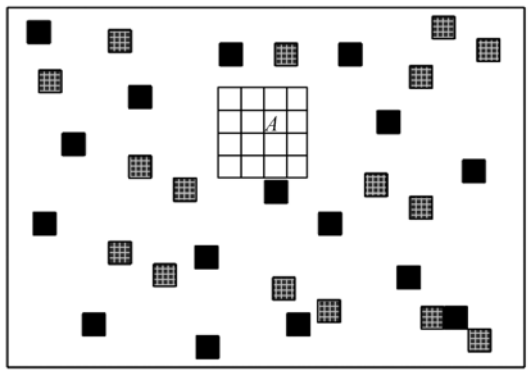

图 1　水印随机嵌入位置示意图

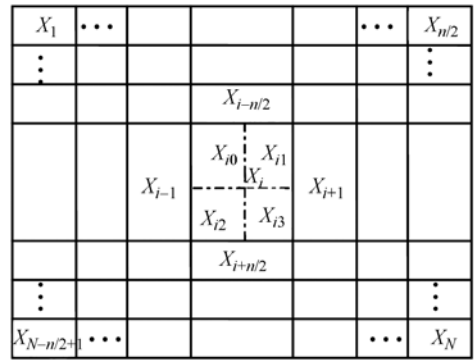

图 2　图像块编号方式及块结构示意图

图 3 为本文提出的图像块的水印生成及嵌入算法框图, 算法的详细步骤如下:

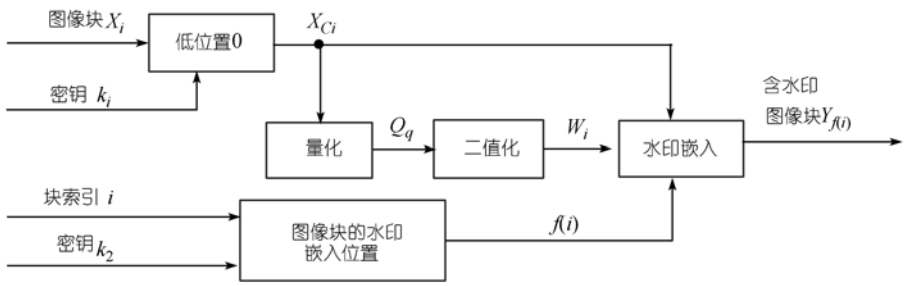

图 3　图像块的水印生成及嵌入框图

步骤 1　基于密钥 k_1 生成图像块内容 X_{Ci}: 基于密钥 k_1 生成长度为 $2N$ 的二值随机序列 $Z=\{z_1,z_2,\cdots,z_{2N}\}$, 根据该序列将像素 $x_{i(2z_{2i}+z_{2i-1})}$ 的次低位置 0, 并将所有元素的最低位置 0 生成图像块内容 $X_{Ci}=\begin{bmatrix}\tilde{x}_{i0} & \tilde{x}_{i1}\\ \tilde{x}_{i2} & \tilde{x}_{i3}\end{bmatrix}$, 即

$$\tilde{x}_{ij}=\lfloor x_{ij}/4\rfloor\times 4+2\lfloor \mathrm{mod}(x_{ij},4)/2\rfloor\times\left|\mu_{ij}-1\right|. \tag{21}$$

其中, mod(·)为取余函数, $|\cdot|$ 为绝对值符号, μ_{ij} 的值是根据二值混沌序列 Z 产生的, 即

$$\mu_{ij}=\begin{cases}1, & j=2z_{2i}+z_{2i-1},\\ 0, & \text{否则},\end{cases}\quad j=0,\ 1,\ 2,\ 3. \tag{22}$$

步骤 2　生成量化序列 $Q_C=\{Q_{Ci}\mid i=1,\cdots,N\}$: 对每个块内容 X_{Ci} 的平均值 $\mathrm{Ave}(X_{Ci})$ 做 5 bit 标量量化, 即

$$Q_{Ci}=\begin{cases}a, & \mathrm{Ave}(X_{Ci})\in[\min+aq,\min+(a+1)q),\\ N_q-1, & \mathrm{Ave}(X_{Ci})=\max.\end{cases} \tag{23}$$

其中, N_q=2^5, $q=(\max-\min)/N_q$ 称为量化步长, max 和 min 分别为所有 Ave(X_{Ci}) ($i=1,2,\cdots,N$) 的最大值和最小值, $a\in[0,N_q-1]$.

步骤 3 生成 X_i 二值水印 W_i: 对元素 Q_{Ci} 进行二值编码生成图像块 X_i 的二值水印 $W_i=(w_{i4}w_{i3}w_{i2}w_{i1}w_{i0})$, 此处 $w_{ij}\in\{0,1\}$, $i=1,2,\cdots,N$, $j=0,1,2,3,4$, 且满足

$$Q_{Ci}=16w_{i4}+8w_{i3}+4w_{i2}+2w_{i1}+w_{i0}. \tag{24}$$

步骤 4 水印嵌入: 基于密钥 k_2 生成的水印嵌入位置向量 f (i), i=1,2,$\cdots$, N (见下文 2.3 节描述水印嵌入位置向量的生成方法), 根据 f (i)将 W_i 嵌入图像块 $X_{f(i)}$ 的置 0 位, 生成含水印图像块 $X^w_{f(i)}$:

$$X^w_{f(i)}=X_{Cf(i)}+\begin{bmatrix} w_{i0}+2w_{i4}\mu_{f(i)0} & w_{i1}+2w_{i4}\mu_{f(i)1} \\ w_{i2}+2w_{i4}\mu_{f(i)2} & w_{i3}+2w_{i4}\mu_{f(i)3}\end{bmatrix}. \tag{25}$$

其中, $\mu_{f(i)j}$ 是根据公式(22)得到的.

2.2 篡改检测与恢复

图像块的篡改检测算法框图(被测图像用 Y 表示)如图 4 所示, 本文提出的篡改检测及恢复算法步骤如下:

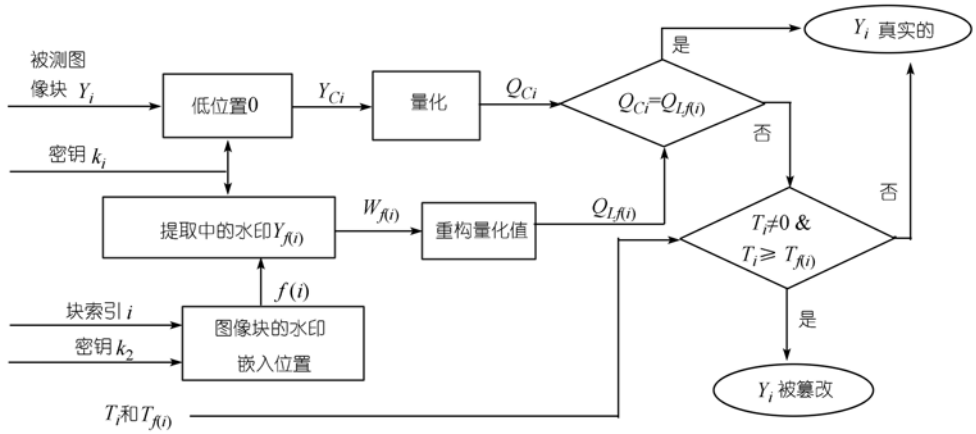

图 4 图像块的篡改检测框图

步骤 1 利用密钥 k_1, 按水印嵌入算法步骤 2 和 3 基于被测图像块内容 Y_{Ci} 生成量化值 $Q_C=\{Q_{Ci}\,|\,i=1,2,\cdots,N\}$.

步骤 2 根据提取的水印信息重构的量化序列 $Q_L=\{Q_{Li}\,|\,i=1,2,\cdots,N\}$. 根据密钥 k_0 生成长度为 2N 的二值随机序列 $Z=\{z_1,z_2,\cdots,z_{2N}\}$, 从图像块 Y_i, i=1,2,$\cdots$, N 中提取水印信息

$$\begin{cases} w'_{ij} = \mathrm{mod}(y_{ij}, 2), & j = 0,1,2,3; \\ w'_{ij} = \left\lfloor \mathrm{mod}(y_{id}, 4)/2 \right\rfloor. & j = 4. \end{cases} \tag{26}$$

其中,

$$d = 2z_{2i} + z_{2i-1}. \tag{27}$$

根据提取的水印信息, 重构图像块低位 Y_{Li} 保存的量化值

$$Q_{Li} = \mathrm{mod}\left(\sum_{j=0}^{4} 2^j w'_{ij}, \quad 32 \right). \tag{28}$$

步骤 3　对任一图像块 Y_i, 设 T_i 表示 Y_i 的 8-邻域中满足 $Q_{Cj} \neq Q_{Lf(j)}$ ($j \in N_8(Y_i)$)的图像块的个数, 则

1) 如果 $Q_{Ci} = Q_{Lf(i)}$, 判定 Y_i 没有被篡改;

2) 如果 $Q_{Ci} \neq Q_{Lf(i)}$ 且 $T_i = 0$, 判定 Y_i 没有被篡改;

3) 如果 $Q_{Ci} \neq Q_{Lf(i)}$ 且 $T_i \geqslant T_{f(i)}$ ($T_i = 0$ 除外), 则判定 Y_i 被篡改, 否则判定 Y_i 没有被篡改.

步骤 4　如果被测图像块 Y_i 被篡改, 利用相应水印信息恢复的量化值 $Q_{Lf(i)}$, 采用逆量化和插值操作恢复图像块 Y_i 的内容.

本文提出的篡改检测算法的主要贡献在于, 当 $Q_{Ci} \neq Q_{Lf(i)}$ 时, 不是简单地认为该图像块被篡改, 而是根据其8-邻域块的内容与水印的匹配情形进一步对其判定, 最大程度地降低了可恢复水印算法在区域篡改时的虚警概率. 当 $T_i = 0$ 时, 说明与图像块 Y_i 相邻的 8 个图像块都没有被篡改, 此时造成 Y_i 的内容与水印信息不匹配的原因是该图像块相应的水印信息被改变, 或者是噪声干扰, 因此我们可以认为该图像块没有被篡改. 当 $T_i \neq 0$ 时, 由前面的讨论可知, 当该图像块 Y_i 周围被篡改图像块的个数大于 $Y_{f(i)}$ 周围被篡改图像块的个数时, 该图像块被篡改的可能较大, 因此, 本文算法规定当 $T_i \geqslant T_{f(i)}$ 时判定图像块 Y_i 被篡改, 反之判定图像块 Y_i 没有被篡改.

2.3　水印嵌入位置生成函数

为尽可能减少图像块内容和相应水印信息同时被篡改的可能性, 在可恢复水印算法中基于图像块内容生成的水印信息不是嵌入在该图像块自身之中, 而是嵌入在其他图像块的LSBs中. 根据水印嵌入位置的实际意义, 水印嵌入位置 $f(i), i = 1,2,\cdots,N$ 需满足两个必要条件 [19]:

1) 对 $\forall i \in [1, N]$, 有 $f(i) \in [1, N]$, 即每一个图像块水印都必须放在该图像中某个图像块的最低位;

2) 对 $\forall i, j \in [1, N]$, $i \neq j$, 有 $f(i) \neq f(j)$, 即两个图像块的水印不能放在同一个图像块的最低位.

为提高算法的安全性, $f(i)$应基于密钥在整个图像中的近似均匀分布, 因此本文利用随机序列来生成水印嵌入位置向量 $f(i)$, 方法如下:

步骤 1 根据密钥 k_2 生成长度为 N 的实值随机序列 $B=(b_1,b_2,\cdots,b_N)$;

步骤 2 采用稳定排序法对随机序列 $B=(b_1,b_2,\cdots,b_N)$ 排序, 生成有序序列 $(b_{a_1},b_{a_2},\cdots,b_{a_N})$, 则 B 的有序索引序列为 $A=(a_1,a_2,\cdots,a_i,\cdots,a_N)$;

步骤 3 令第 i 个图像块的水印嵌入位置为 a_i, 即

$$f(i)=a_i,\ i=1,2,\cdots,N. \tag{29}$$

因为 a_i 是随机序列 $B=(b_1,b_2,\cdots,b_N)$ 的下标, 故长度为 N 的随机序列地址范围为 $1,\cdots,N$, 即 $f(i)\in[1,N]$; 其次, 序列 $B=(b_1,b_2,\cdots,b_N)$ 中任何 2 个元素 b_i 和 b_j 在有序序列中的位置都不会重合, 因此 $\forall i,j\in[1,N]$, $i\neq j$, 有 $f(i)\neq f(j)$. 显然, 采用上述方法生成的 $f(i)$ 满足水印嵌入位置向量的两个必要条件. 图 5 给出了 N=200, 密钥 k_2 分别为 10^9, 10^{9-1}, 0, $1-10^{10}$, -10^{10} 时 $f(i)$ 的分布图(随机序列是利用 Matlab 提供的 rand()函数生成的). 由图 5 可以看出, 利用上述方法生成的水印嵌入位置在区间[1, N]上近似均匀分布. 水印嵌入位置 $f(i)$ 的这个特性与篡改检测算法中的步骤 3 结合, 还能有效降低可恢复水印算法的虚警概率.

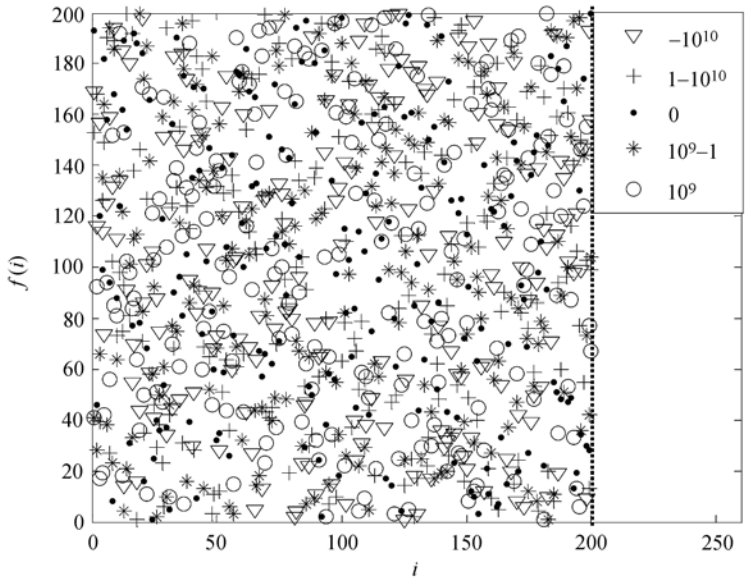

图 5 不同密钥下的图像块水印的嵌入位置分布图

3 性能分析

该部分首先推导本文可恢复水印算法的虚/漏警概率, 然后分别讨论本文算法在区域篡改和随机篡改条件下的篡改检测与恢复性能.

3.1 虚/漏警概率

根据篡改检测算法的步骤3和公式(4)可知, 本文篡改检测算法的虚警概率和漏警概率分别为

$$P_{fr}=P_{U|H_1}P\left\{\left(T_i\geqslant T_{f(i)}\right)\cap\left(T_i\neq 0\right)|\,Y_{\mathrm{i}}\in H_1\right\} \tag{30}$$

和

$$P_{fa}=\left(1-P_{U|H_0}\right)+P_{U|H_0}P\left\{\left(T_i=0\right)\cup\left(T_i<T_{f(i)}\right)|\,Y_i\in H_0\right\}. \tag{31}$$

本文算法中图像块的大小为 2×2, 图像块水印信息的长度 L_w=5, 由公式(13)可得

$$P_{U|H_0}=P_{L|H_0}P_{H_0}+P_{C|H_0}-33P_{L|H_0}P_{H_0}P_{C|H_0}/32. \tag{32}$$

3.2 区域篡改(area tampering)

设被篡改区域占整个图像的比例为 p, 篡改区域中像素的每个 bit 是否变化是独立的, 在此条件下,

$$\begin{cases} P_{H_0|A}\approx p, \\ P_{L|H_0A}=1-P_{=0}\left(5,0.5\right)=1-1/2^5=31/32, \\ P_{C|H_0A}=1-P_{=0}\left(4\times 8-5,0.5\right)=1-1/2^{27}\approx 1. \end{cases} \tag{33}$$

(33)式结合(14)和(32)式得

$$P_{U|H_1A}=P_{L|H_0A}P_{H_0|A}=31p/32m, \tag{34}$$

$$\begin{aligned} P_{U|H_0A}&=P_{L|H_0\mathrm{A}}P_{H_0|\mathrm{A}}+P_{C|H_0\mathrm{A}}-33P_{L|H_0\mathrm{A}}P_{H_0|\mathrm{A}}P_{C|H_0\mathrm{A}}/32 \\ &=31p/32+1-33*\left(31p/32\right)/32 \\ &=1-31p/32^2. \end{aligned} \tag{35}$$

3.2.1 漏警概率

将(35)式代入(31)式得, 本文算法区域篡改时的漏警概率 $P_{fa|A}$ 为

$$P_{fa|A}=31p/32^2+\left(1-31p/32^2\right)P\left\{\left(T_i=0\right)\cup\left(T_i<T_{f(i)}\right)|\,Y_i^*\in H_0\right\}. \tag{36}$$

根据图像块水印嵌入位置的随机性可知, 对 $\forall Y_i\in\Omega$, 其相应水印信息所在的图像块 $Y_{f(i)}$ 位于篡改区域内、外的概率分别为

$$\begin{cases} P\left\{Y_{f(i)}\in H_0\right\}=p, \\ P\left\{Y_{f(i)}\notin H_0\right\}=1-p. \end{cases} \tag{37}$$

相应地(为简单起见, 我们忽略了 $Y_{f(i)}$ 位于篡改区域内、外边界的情形),

$$T_{f(i)}\sim\begin{cases} B\left(8,P_{U|H_0A}\right), & \text{当}\quad Y_{f(i)}\in H_0; \\ B\left(8,P_{U|H_1A}\right), & \text{当}\quad Y_{f(i)}\notin H_0. \end{cases} \tag{38}$$

对 $\forall Y_i\in H_0$, 当 Y_i 位于篡改区域内时, 该图像块 8-邻域中的 8 个图像块均被篡改; 当 Y_i 位于篡改区域边界时, 该图像块 8-邻域中只有部分图像块被篡改. 也就是说, Y_i 在篡改区域内和篡改边界时, $T(i)$的分布不同. 设篡改区域边界在整个篡改区域的比例为 μ, 则区域篡改时算

法的漏警概率为

$$P_{fa|A}=\mu P_{fa|BA}+(1-\mu)P_{fa|IA}, \tag{39}$$

其中, $P_{fa|IA}$ 和 $P_{fa|BA}$ 分别表示 Y_i 在篡改区域内(inner)和篡改区域的边界(boundary)时的漏警概率.

- 篡改区域内(inner)H_{0I}

当 Y_i 在篡改区域内(inner), 即 $Y_i\in H_{01}$ 时, $T_i\sim B\left(8,P_{U|H_0A}\right)$, 因此,

$$P\{T_i=0\,|\,Y_i\in H_{0I}\}=P_{=0}\left(8,P_{U|H_0A}\right). \tag{40}$$

根据全概率公式, 当 $Y_i\in H_{01}$ 时 $T_i<T_{f(i)}$ 的概率为

$$\begin{aligned}P\{T_i<T_{f(i)}\,|\,H_{0I}\}&=P\{Y_{f(i)}\in H_0\}P_{<}\left(8,P_{U|H_0A},P_{U|H_0A}\right)+P\{Y_{f(i)}\notin H_0\}P_{<}\left(8,P_{U|H_0A},P_{U|H_1A}\right)\\&=pP_{<}\left(8,P_{U|H_0A},P_{U|H_0A}\right)+(1-p)P_{<}\left(8,P_{U|H_0A},P_{U|H_1A}\right)\end{aligned} \tag{41}$$

因此,

$$\begin{aligned}&P\left\{\left(T_i=0\right)\cup\left(T_i<T_{f(i)}\right)\,|\,Y_i\in H_{0I}\right\}\\&=P\left\{\left(T_i=0\right)\,|\,Y_i\in H_{0I}\right\}+P\left\{\left(T_i\neq 0\right)\,|\,Y_i\in H_{0I}\right\}P\left\{\left(T_i<T_{f(i)}\right)\,|\,Y_i\in H_{0I}\right\}\\&=P_{=0}\left(8,P_{U|H_0A}\right)+\left(1-P_{=0}\left(8,P_{U|H_0A}\right)\right)\left(pP_{<}\left(8,P_{U|H_0A},P_{U|H_0A}\right)+(1-p)P_{<}\left(8,P_{U|H_0A},P_{U|H_1A}\right)\right).\end{aligned} \tag{42}$$

将(42)式代入(36)式得, $Y_i\in H_{01}$ 时的漏警概率为

$$P_{fa|IA}=31p/32^2+\left(1-31p/32^2\right)P\left\{\left(T_i=0\right)\cup\left(T_i<T_{f(i)}\right)\,|\,Y_i^*\in H_{0I}\right\}. \tag{43}$$

- 篡改区域边界(boundary)H_{0B}

当 $Y_i\in H_{0B}$, 其8-邻域中包含内容已被篡改和没有被篡改的两类图像块, 令 $T_i=T_{i1}+T_{i2}$, 其中, T_{i1} 表示内容被篡改图像块(其个数记为 N_{Tc})中被检测图像块的个数, T_{i2} 表示内容没有被篡改图像块中被检测图像块的个数, 也就是说

$$\begin{cases}T_{i1}\sim B(N_{Tc},P_{U|H_0A}),\\T_{i2}\sim B((8-N_{Tc}),P_{U|H_1A}).\end{cases} \tag{44}$$

因此, 设 a 为区间[0, 8]内的整数,

$$\begin{aligned}P\{T_i=a\,|\,Y_i\in H_{0B}\}&=P\{T_{i1}+T_{i2}=a\}\\&=\sum_{b=0}^{a}P\{T_{i1}=b\}P\{T_{i2}=a-b\}\\&=\sum_{b=0}^{a}P_{=b}(N_{Tc},P_{U|H_0A})P_{=a-b}((8-N_{Tc}),P_{U|H_1A}).\end{aligned} \tag{45}$$

特别地, 当 a=0 时,

$$P\{T_i=0\,|\,Y_i\in H_{0B}\}=P_{=0}(N_{Tc},P_{U|H_0A})P_{=0}((8-N_{Tc}),P_{U|H_1A}), \tag{46}$$

$$\begin{aligned}P\{T_i < T_{f(i)} \mid Y_i \in H_{0B}\} &= \sum_{a=1}^{8} P\{T_{f(i)} = a\} P\{T_i < a\} \\ &= \sum_{a=1}^{8} (pP_{=a}(8, P_{U|H_0A}) + (1-p)P_{=a}(8, P_{U|H_1A})) \sum_{d=0}^{a-1} P\{T_i = d \mid Y_i \in H_{0B}\}.\end{aligned} \tag{47}$$

因此,

$$\begin{aligned}&P\{(T_i = 0) \cup (T_i < T_{f(i)}) \mid Y_i \in H_{0B}\} \\ &\quad = P\{(T_i = 0) \mid Y_i^* \in H_{0B}\} + P\{(T_i \neq 0) \mid Y_i \in H_{0B}\} P\{(T_i < T_{f(i)}) \mid Y_i \in H_{0B}\}.\end{aligned} \tag{48}$$

将(46)~(48)式代入(36)式得, $Y_i \in H_{0B}$ 时的漏警概率为

$$P_{fa|BA} = 31p/32^2 + (1-31p/32^2) P\{(T_i = 0) \cup (T_i < T_{f(i)}) \mid Y_i^* \in H_{0B}\}. \tag{49}$$

3.2.2 虚警概率

讨论虚惊概率的前提条件是图像块 Y_i 没有被篡改(即 $\forall Y_i \notin H_0$). 图像块 Y_i 没被篡改而满足 $Q_{Ci} \neq Q_{Lf(i)}$, 则其相应的水印信息 $Y_{Lf(i)}$ 一定被篡改, 即 $Y_{f(i)} \in H_0$, 所以 $T_{f(i)} \sim B(8, P_{U|H_0A})$.

与漏警概率的讨论相似, 没被篡改图像块 Y_i 分为两种情形: ① Y_i 与篡改区域相邻; ② Y_i 与篡改区域不相邻. 考虑到与篡改区域相邻的非篡改点较少, 因此, 我们仅考虑 Y_i 与篡改区域不相邻的情形, 此时 $T_i \sim B(8, P_{U|H_1A})$. 所以, $\forall Y_i \in H_1$ 有

$$\begin{cases} P\{T_i = 0 \mid Y_i \in H_1\} = P_{=0}(8, P_{U|H_1A}), \\ P\{T_i < T_{f(i)} \mid Y_i \in H_1\} = P_{<}(8, P_{U|H_1A}, P_{U|H_0A}). \end{cases} \tag{50}$$

因此,

$$\begin{aligned}&P\{(T_i \geqslant T_{f(i)}) \cap (T_i \neq 0) \mid Y_i \in H_1\} \\ &\quad = (1 - P\{(T_i \geqslant T_{f(i)}) \mid Y_i \in H_1\})(1 - P\{(T_i = 0) \mid Y_i \in H_1\}) \\ &\quad = (1 - P_{<}(8, P_{U|H_1}, P_{U|H_0}))(1 - P_{=0}(8, P_{U|H_1})).\end{aligned} \tag{51}$$

将(35)和(51)式代入(30)式得, 本文算法在区域篡改条件下的虚警概率 $P_{fr|A}$ 为

$$P_{fr|A} = 31p/32(1 - P_{=0}(8, P_{U|H_1A}))(1 - P_{<}(8, P_{U|H_1A}, P_{U|H_0A})). \tag{52}$$

为验证上述理论推导的正确性, 我们以正方形篡改区域为例, 对不同测试图像做了大量的实验仿真. 设 $m_t \times m_t$ 为被篡改区域中的图像块数, 此时, 篡改区域边界在整个篡改区域的比例为

$$\begin{cases} \mu = 4(m_t - 1)/m_t^2, \\ p = 4m_t^2/(m \cdot n). \end{cases} \tag{53}$$

其中, m_t 的变化范围为 $\left\lfloor 5, \left\lfloor (m+n)/4\sqrt{2} \right\rfloor \right\rfloor$, 相应地, p 的取值范围在 0~0.5 之间. 测试时, 将选定区域的每个像素随机替换为一个[0, 255]之间的整数得到区域篡改图像. 通过比较含水印图像与篡改图像及其检测结果, 分别统计得到 M_T: 篡改图像块个数; D_T: 被检测的篡改块个数; D_F: 被检测的真实图像块个数; M_F: 误恢复图像块个数. 则实验得到

$$\begin{cases} P_{fa} = D_T / M_T, \\ P_{fr} = D_F /((m\times n)/4 - M_T), \\ \Gamma_{fa} = 4(M_T - D_T)/(m\times n), \\ \Gamma_{fr} = 4D_F /(m\times n), \\ \Gamma_P = 4(D_T + D_F)/(m\times n), \\ \Gamma_F = 4M_F /(m\times n). \end{cases} \tag{54}$$

图 6(a)为区域篡改时本文算法的虚/漏警概率与 p 的关系曲线. 根据公式(43)和(49)计算得到的篡改区域内和篡改边界的漏警概率分别如图 6(a)中的 $P_{fa|IA}$ 和 $P_{fa|BA}$ 曲线, 根据公式(39)和(52)计算得到的区域篡改条件下的虚/漏警概率如图 6(a)中的 $P_{fr|A}$ 和 $P_{fa|A}$ 理论曲线. 根据实验统计结果得到的虚/漏警概率分别如图 6(a)中的 $P_{fa|A}$ 和 $P_{fa|A}$ 实验曲线. 由图 6(a)可以看出, $P_{fa|A}$ 和 $P_{fr|A}$ 的理论与实验曲线都拟合得很好, 且 $P_{fr|A}$ 接近于 0.

图 6(b)给出了区域篡改条件下本文算法的漏检率、虚检率、检测率和误恢复率的理论和实验曲线. 将理论计算得到的 $P_{fr|A}$ 和 $P_{fa|A}$ 代入公式(5)~(7)分别得到的区域篡改的漏检率、虚检率、检测率和误恢复率的理论值分别表示为图 6(b)的 $\Gamma_{fa|A}$, $\Gamma_{fr|A}$, $\Gamma_{D|A}$ 和 $\Gamma_{F|A}$ 的理论曲线. 根据实验统计结果计算得到区域篡改的漏检率、虚检率、检测率和误恢复率分别如图 6(b)的 $\Gamma_{fa|A}$, $\Gamma_{fr|A}$, $\Gamma_{D|A}$ 和 $\Gamma_{F|A}$ 的实验曲线. 由图 6(b)可以看出, $\Gamma_{fa|A}$, $\Gamma_{fr|A}$, $\Gamma_{D|A}$ 和 $\Gamma_{F|A}$ 的理论和实验曲线拟合得很好, 进一步验证了上述理论推导的正确性.

3.3 随机篡改(random tampering)

设被测图像中随机篡改像素的比例为 p, 2×2 图像块中只要有一个像素被篡改就认定该图像块被篡改, 因此随机篡改像素的比例为 p 时, 图像块被篡改和不被篡改的概率分别为

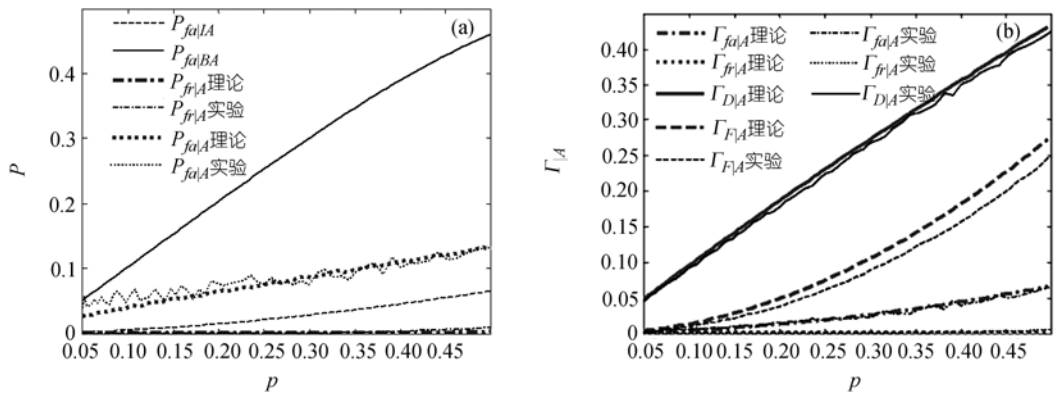

图 6 区域篡改时的篡改检测及恢复性能

(a) 虚/漏警概率与 p 的关系曲线; (b) 恢复质量评价指标与 p 的关系曲线

$$\begin{cases} P_{H_0|R} = \sum_{i=1}^{4} P_{=i}(4,p) = 1-(1-p)^4, \\ P_{H_1|R} = P_{=0}(4,p) = (1-p)^4. \end{cases} \tag{55}$$

当图像块 Y_i 中的像素被随机篡改, 该图像块的内容和低位被篡改的概率分别为

$$\begin{cases} P_{L|H_0R} = \dfrac{1}{P_{H_0|R}} \sum_{i=1}^{4} P_{=i}(4,p)\left[\dfrac{i}{4}(1-P_{=0}(i+1,0.5)) + \dfrac{4-i}{4}(1-(P_{=0}(i,0.5)))\right], \\ P_{C|H_0R} = \dfrac{1}{P_{H_0|R}} \sum_{i=1}^{64} P_{=i}(4,p)\left[\dfrac{i}{4}(1-P_{=0}(7i-1,0.5)) + \dfrac{4-i}{4}(1-(P_{=0}(7i,0.5)))\right]. \end{cases} \tag{56}$$

将(55)和(56)式分别代入(14)和(32)式得

$$\begin{aligned} P_{U|H_1R} &= P_{L|H_0R}P_{H_0|R} \\ &= \sum_{i=1}^{4} P_{=i}(4,p)\left[\frac{i}{4}(1-P_{=0}(i+1,0.5)) + \frac{4-i}{4}(1-(P_{=0}(i,0.5)))\right] \\ &= \sum_{i=1}^{4} P_{=i}(4,p)\left[\frac{i}{4}(1-0.5^{(i+1)}) + \frac{4-i}{4}(1-0.5^{i})\right], \end{aligned} \tag{57}$$

$$P_{U|H_0R} = P_{L|H_0R}P_{H_0R} + P_{C|H_0R} - 33P_{L|H_0R}P_{H_0R}P_{C|H_0R}/32. \tag{58}$$

随机篡改条件下, T_i 和 $T_{f(i)}$ 服从参数相同的二项分布 $B(8,P_{U|HR})$, 此处 $P_{U|HR}$ 为 $\forall Y_i \in H_1 \cup H_0$ 满足 $Q_{iC} \neq Q_{f(i)L}$ 的概率, 即

$$\begin{aligned} P_{U|HR} &= P_{U|H_0R}P_{H_0|R} + P_{U|H_1R}P_{H_1|R} \\ &= (P_{W|R} + P_{C|H_0R} - 33P_{W|R}P_{C|H_0R}/32)P_{H_0|R} + P_{W|R}P_{H_1|R} \\ &= P_{W|R}(P_{H_0} + P_{H_1}) + P_{C|H_0R}P_{H_0} - 33P_{W|R}P_{C|H_0R}P_{H_0}/32 \\ &= P_{W|R} + P_{C|H_0R}P_{H_0|R} - 33P_{W|R}P_{C|H_0R}P_{H_0|R}/32. \end{aligned} \tag{59}$$

根据公式(30)和(31)得, 本文可恢复水印算法在随机篡改条件下的虚/漏警概率分别为

$$\begin{cases} P_{fa|R} = (1-P_{U|H_0R}) + P_{U|H_0R}(P\{T_i=0\} + P\{T_i < T_{f(i)}\}P\{T_i \neq 0\}) \\ \qquad = (1-P_{U|H_0R}) + P_{U|H_0R}(P_{=0}(8,P_{U|HR}) + P_{<}(8,P_{U|HR},P_{U|HR})(1-P_{=0}(8,P_{U|HR}))) \\ P_{fr|R} = P_{W|R}(1-P_{=0}(8,P_{U|HR}))(1-P_{<}(8,P_{U|HR},P_{U|HR})) \end{cases} \tag{60}$$

由公式(60)可知, 随机篡改的虚/漏警概率与 $P_{U|HR}$ 直接相关. 为此下述实验验证公式(59)的正确性及本文算法在随机篡改条件下的篡改恢复性能.

利用 Rand()函数随机选取 M 个像素并将其随机替换为[0, 255]内的一个整数, 则篡改图像中像素被随机篡改的比例 $p = M/(m*n)$. 比较篡改图像与含水印图像及其检测结果, 统计得到 M_T, D_T, D_F 和 M_F 4 个统计量. 同时分别统计篡改图像中内容改变和水印改变的图像块个数 M_C 和 M_W, 并统计篡改图像中满足 $Q_{iC} \neq Q_{f(i)L}$ 的图像块的个数 M_U, 则实验得到的 $P_{U|H_1R} = 4M_W/(m\times n)$, $P_{U|H_0R}P_{H_0|R} = 4M_C/(m\times n)$ 和 $P_{U|HR} = 4M_U/(m\times n)$.

对不同的 p(通过改变随机选取像素的个数 M), 根据实验结果得到的 $P_{U|H_1R}$, $P_{U|H_0R}P_{H_0|R}$ 和 $P_{U|HR}$ 的 3 条实验曲线如图 7(a)所示, $P_{U|H_1R}$, $P_{U|H_0R}P_{H_0|R}$ 和 $P_{U|HR}$ 的理论曲线是根据公式(57)~(59)计算得到的. 图 7(b)是随机篡改条件下本文算法的 4 个评价指标与 p 的关系曲线, 其中, $\Gamma_{fa|A}$, $\Gamma_{fr|A}$, $\Gamma_{D|A}$ 和 $\Gamma_{F|A}$ 的实验曲线是根据上述统计量利用公式(54)计算得到的, $\Gamma_{fa|A}$, $\Gamma_{fr|A}$, $\Gamma_{D|A}$ 和 $\Gamma_{F|A}$ 的理论曲线是将(57)和(59)式的计算结果代入(5)~(7)式计算得到的. 由图 7 可以看出, 各项指标的理论与实验曲线都拟合得很好, 验证了上述理论推导的正确性.

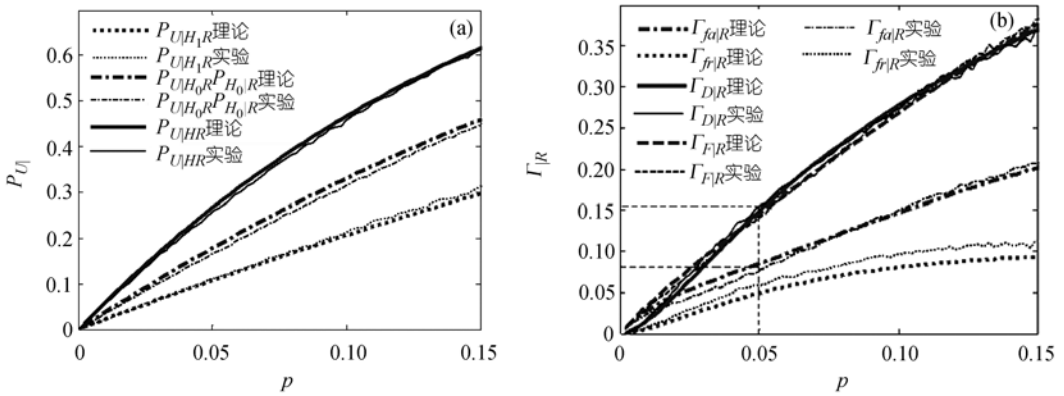

图 7 随机篡改时算法的篡改恢复性能

(a) $P_{U|}$ 与 p 的关系曲线; (b) 恢复质量评价指标与 p 的关系曲线

4 实验仿真

为验证本文算法的篡改检测和恢复性能, 下面给出相同篡改条件下, 本文算法与现有可恢复水印算法 [11]的两组(区域篡改和随机篡改)实验仿真比较结果.

4.1 区域篡改

为验证本文算法区域篡改时的篡改定位及篡改恢复能力, 图 8 给出了本文算法与文献 [11] 算法的篡改定位及恢复结果比较, 其中, 图 8(a)为利用本文算法生成的含水印图像, 与原始图像的峰值信噪比为 48.14 dB; 图 8(b)为篡改图像: 使用Photoshop编辑软件将Napoleon的头部替换为Mona Lisa的头部, 区域篡改像素比例 $p \approx 0.185$, 篡改图像与含水印图像的峰值信噪比为 17.13 dB; 图 8(c)为本文算法的篡改定位结果. 显然, 篡改区域被精确定位(白色区域), 篡改区域外仅有少数几个图像块(2×2)被误判, 有效降低了可恢复水印算法的虚警概率; 图 8(d)为本文算法得到的篡改恢复图像, 与含水印图像的峰值信噪比为 24.31 dB, 尽管篡改恢复质量不高, 但篡改前Napoleon的脸部仍能清晰可辨.

采用相同的篡改方式, 文献 [11]的篡改检测结果如图8(e)所示, 篡改区域之外存在较多被误判的篡改块, 这些误判块在图像中的分布依赖于密钥; 图8(f)为文献 [11]得到的篡改恢复图像, 与含水印图像的峰值信噪比为 15.76 dB, 根据篡改恢复图像很难辨别Napoleon的脸部. 需要说明的是, 文献 [11]中水印嵌入位置生成函数为 $f(i) = \mathrm{mod}(k_0 + k_1 \cdot i, r) + 1$, 其中, k_0 和 k_1 为

偏移值密钥. 图 8(e)和(f)所示的结果是k_0=253, k_1=71 时得到的.

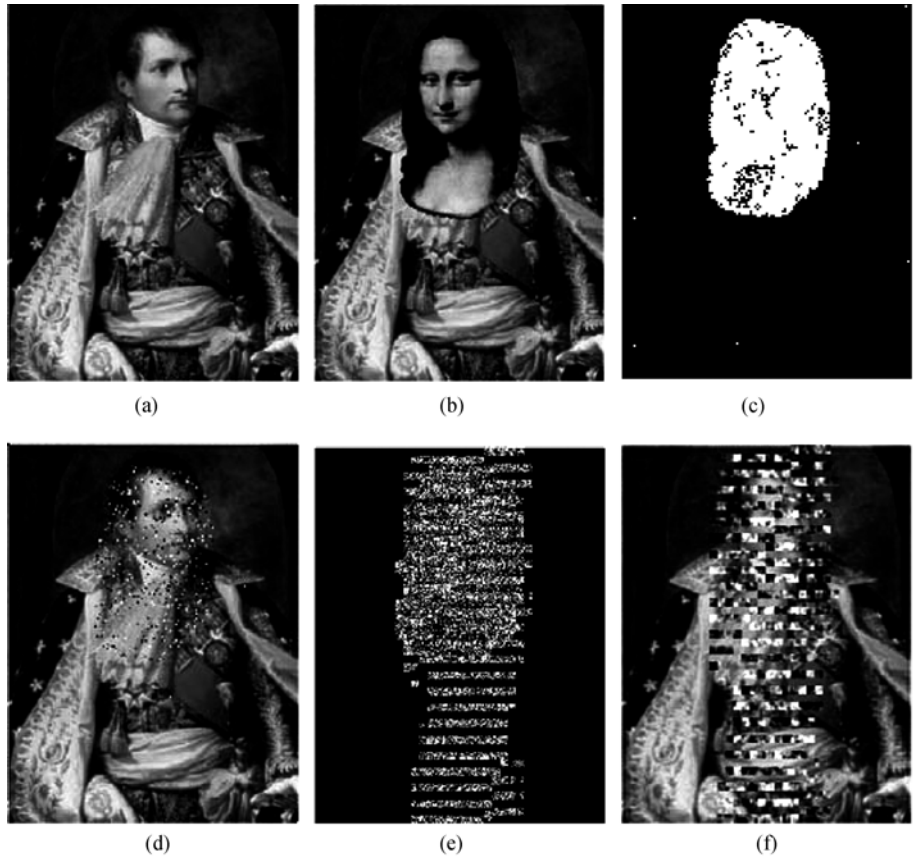

图 8　区域篡改时的篡改检测与恢复结果

(a) 含水印图像; (b) 篡改图像; (c) 本文算法的篡改定位结果; (d) 本文算法的篡改恢复图像; (e) 文献 [11]的篡改定位结果; (f) 文献 [11]的篡改恢复图像

4.2　随机篡改

图 9 为随机篡改的篡改检测与恢复结果. 图 9(a)是利用本文算法生成的含水印图像; 图 9(b)是篡改图像, 篡改方式为: 首先将车牌号“F6o6YVG”修改为“F606FVF”, 然后再在被测图像中添加随机噪声(噪声比例 $p \approx 0.008$), 篡改图像与含水印图像的峰值信噪比为 30.38 dB; 图 9(c)为本文算法的篡改定位结果. 尽管本文算法对随机篡改的漏警概率较高, 但仍能准确定位改变图像真实性的小区域(车牌号中的两个被篡改的字母)篡改; 图 9(d)为本文算法的篡改恢复结果, 与含水印图像的峰值信噪比为 31.12 dB. 虽然被测图像中被随机篡改的噪声点不能有效恢复, 但是被篡改的车牌号(即使像“车牌号”这样小区域)有效地被恢复出来. 采用相同的篡改方法, 图 9(e)和(f)分别为文献 [11]算法的篡改检测和篡改恢复结果. 由于文献 [11]的图像块为 8×8, 因此图像块中一个像素被篡改, 则导致一个 8×8 的图像块被检测, 使得恢复图像中出现较严重的方块效应, 根据图 9(f)难以推测出篡改前的车牌号.

比较图9(d)和(f)可以看出, 本文算法对噪声的鲁棒性远远高于文献 [11], 一定程度上解决了含水印图像的传输问题, 为自嵌入认证水印算法走向实际提供了可能.

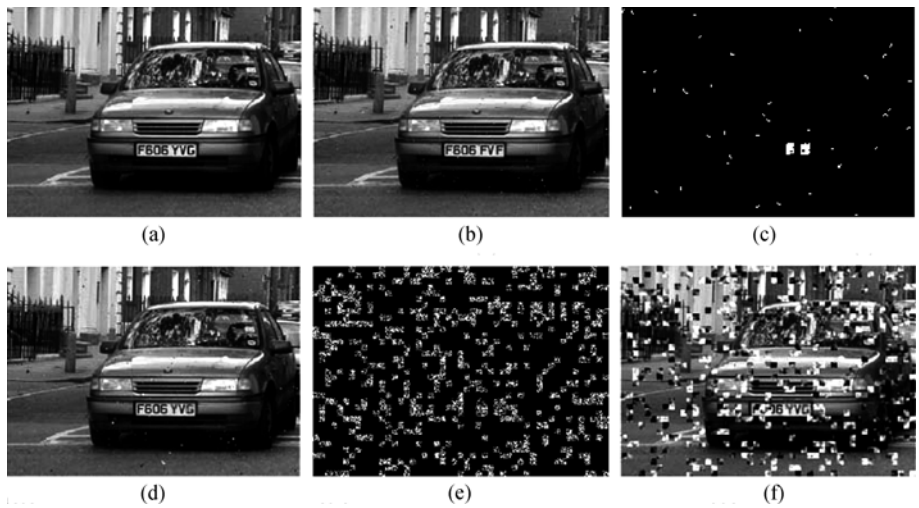

图 9 随机篡改时的篡改检测与恢复结果

(a)被测图像; (b)篡改图像; (c)本文算法的篡改定位结果; (d)本文算法的恢复图像; (e)文献 [11]的篡改定位结果; (f)文献 [11]的恢复图像

从图 8 和 9 的仿真结果可以看出: 无论是区域篡改还是随机篡改时, 本文算法的虚警概率都远远低于现有可恢复水印算法的虚警概率, 有效降低了篡改恢复图像的误恢复率; 尽管本文算法的漏警概率较高, 但由于本文算法图像块小, 漏检篡改块一般不会影响被测图像的真实性. 因此, 当被测图像中有水印信息篡改时, 本文算法大大提高了篡改恢复图像的质量.

5 结论

针对现有可恢复水印算法定位精度低和虚警概率高的问题, 本文提出一种高定位精度的可恢复水印算法. 分别推导给出了在随机篡改与区域篡改条件下, 现有可恢复水印算法和本文算法的虚警概率和漏警概率, 并通过实验验证了理论推导的正确性. 定义了衡量篡改恢复图像质量的 4 个评价指标及其与虚警概率和漏警概率的关系, 为定量地分析可恢复水印算法的性能提出了一种客观的评价指标. 与现有可恢复水印算法相比, 本文提出的算法具有以下优点:

1) 提高了可恢复水印算法的定位精度;
2) 提高了可恢复水印算法对随机噪声的鲁棒性;
3) 有效解决了可恢复水印算法的篡改定位问题;
4) 大大提高了被测图像中有水印信息篡改时的篡改恢复图像的质量.

如何提高可恢复水印算法对 JPEG 压缩的鲁棒性是我们下一步的研究内容.

参考文献

1 吴金海, 林福宗. 基于数字水印的图像认证技术. 计算机学报, 2004, 27(9): 1153—1161

2 Zhu B B, Swanson M D, Tewfik A H. When seeing isn't believing. IEEE Signal Process Mag, 2004, 3: 40—49 [DOI]

3 Wong P W, Memon N. Secret and public key image watermarking schemes for image authentication and ownership verification. IEEE Trans Image Process, 2001, 10(10): 1593—1601 [DOI]

4 Suthaharan S. Fragile image watermarking using a gradient image for improved localization and security. Pattern Recognit Lett, 2004, 25: 1893—1903 [DOI]

5 和红杰, 张家树, 田蕾. 能区分图像或水印篡改的脆弱水印方案. 电子学报, 2005, 33(9): 1557—1561

6 Kundur D, Hatzinakos D. Digital watermarking for telltale tamper proofing and authentication. Proc IEEE, 1999, 87(7): 1167—1180 [DOI]

7 沃炎, 韩国强, 张波. 一种新的基于特征的图像内容认证方法. 计算机学报, 2005, 28(1): 105—112

8 Fridrich J, Goljan M. Images with self-correcting capabilities. In: Proc ICIP'99. Kobe: IEEE Press, 1999. 25—28

9 Fridrich J, Goljan M. Protection of digital images using self embedding. In: Proceedings of NJIT Symposium on Content Security and Data Hiding in Digital Media. New Jersey Institute of Technology, 1999

10 何孝富, 黄继凤, 张培君, 等. 一种具有自我恢复功能的脆弱性水印技术. 上海师范大学学报, 2004, 33(1): 56—62

11 张鸿宾, 杨成. 图像的自嵌入及篡改的检测和恢复算法. 电子学报, 2004, 32(2): 196—199

12 王永杰, 赵耀, 潘正祥. 可以自恢复和篡改定位的可逆数字水印. 哈尔滨工业大学学报(Suppl), 2007, 38(7): 791—795

13 郑江滨, 冯大淦, 张艳宁, 等. 可恢复的脆弱数字图像水印. 计算机学报. 2004, 27(3): 371—376

14 和红杰, 张家树. 一种安全的自嵌入及篡改检测和恢复算法. 哈尔滨工业大学学报(Suppl), 2007, 38(7): 889—893

15 和红杰, 张家树. 基于混沌置乱的分块自嵌入水印算法. 通信学报, 2006, 27(7): 80—87

16 Lin P L, Huang P W, Peng A W. A fragile watermarking scheme for image authentication with localization and recovery. In: Proceedings of the 6th IEEE International Symposium on Multimedia Software Engineering. Miami, 2004. 399—407

17 Holliman M, Memon N. Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. IEEE Trans Image Process, 2000, 3(9): 432—441 [DOI]

18 Fridrich J, Goljan M, Memon N. Cryptanalysis of the Yeung-Mintzer fragile watermarking technique. J Electron Imaging, 2002, 11(4): 262—274 [DOI]

19 He H J, Zhang J S, Wang H X. Synchronous counterfeiting attacks on self-embedding watermarking schemes. IJCSNS, 2006, 6(1B). 251—257

20 毛文波. 现代密码学理论与实践. 王继林, 等译. 北京: 电子工业出版社, 2004. 137—163

21 李裕奇. 概率论与数理统计. 北京: 国防工业出版社, 2001. 1—253

22 He H J, Zhang J S, Tai H M. A wavelet-based fragile watermarking scheme for secure image authentication. Lecture Notes in Computer Science, vol 4283. Berlin: Springer-Verlag, 2006. 422—432