软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(12):5848−5861 [doi: 10.13328/j.cnki.jos.006803]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于分块的高效图像可逆认证方法*

钟亦友[1,2], 黄方军[1,2]

[1](中山大学 网络空间安全学院, 广东 深圳 518107)

[2](广东省信息安全技术重点实验室, 广东 广州 510006)

通信作者: 黄方军, E-mail: huangfj@mail.sysu.edu.cn

摘　要: 图像可逆认证是一项将可逆信息隐藏和脆弱水印相结合的新技术, 其既能实现对图像的脆弱认证, 还能在提取认证信息的同时无失真地恢复出原始载体, 对图像的原始性和完整性认证具有非常重要的意义. 针对现有可逆认证方法认证精度低、对具有复杂纹理的图像或图像中部分纹理复杂区域无法实现有效保护的问题, 提出一种新的图像可逆认证方法. 首先对待认证图像进行分块, 根据每个子块可嵌入容量将其分为差分块和平移块, 并采用不同的可逆嵌入方法对不同类型的块进行认证码嵌入操作. 为了增大嵌入容量以提高对每个子块的认证效果, 还采取了分层嵌入的方式. 在认证方, 可以通过从每个子块中提取认证码实现子块的篡改检测和定位. 此外, 所提方法还可与形态学中的膨胀和腐蚀操作结合以细化篡改检测标记, 进一步提高检测效果. 实验结果表明, 所提方法能够在同样的认证精度下对纹理平滑和纹理复杂的图像进行保护, 同时还能够实现对几乎所有子块的独立认证和恢复, 具有广泛的适用性.

关键词: 图像认证; 篡改检测; 脆弱水印; 可逆信息隐藏

中图法分类号: TP391

中文引用格式: 钟亦友, 黄方军. 基于分块的高效图像可逆认证方法. 软件学报, 2023, 34(12): 5848−5861. http://www.jos.org.cn/1000-9825/6803.htm

英文引用格式: Zhong YY, Huang FJ. Efficient Image Reversible Authentication Method Based on Blocks. Ruan Jian Xue Bao/Journal of Software, 2023, 34(12): 5848−5861 (in Chinese). http://www.jos.org.cn/1000-9825/6803.htm

Efficient Image Reversible Authentication Method Based on Blocks

ZHONG Yi-You[1,2], HUANG Fang-Jun[1,2]

[1](School of Cyber Science and Technology, Sun Yat-sen University, Shenzhen 518107, China)

[2](Guangdong Key Laboratory of Information Security Technology, Guangzhou 510006, China)

Abstract: As a new technology that combines reversible data hiding and fragile watermarking, image reversible authentication (RA) can not only realize the fragile authentication of images but also recover the original carrier image without distortion while extracting the authentication code. Thus, it is of great significance to authenticate the originality and integrity of images. Existing reversible authentication methods have low authentication accuracy and cannot effectively protect images with complex textures or some areas with complex textures in the images. To this end, this study proposes a new reversible authentication method. Firstly, images to be authenticated are divided into blocks, and the obtained sub-blocks are classified as differential blocks (DB) and shifting blocks (SB) according to their embedding capacity. Different reversible embedding methods are employed to embed the authentication codes into different types of blocks. It also adopts a hierarchical embedding strategy to increase embedding capacity and improve the authentication effects of each sub-block. On the authentication side, tamper detection and localization can be realized by the authentication code extracted from each sub-block. In addition, this method can be combined with dilation and corrosion in morphology to refine tamper detection marks and further improve the detection accuracy rate. Experimental results show that the proposed method can protect images with

* 基金项目: 国家自然科学基金 (62072481, 61772572); 广州市科技计划 (202201011587); 河南省网络空间态势感知重点实验室开放课题 (HNTS2022014)
收稿时间: 2022-02-01; 修改时间: 2022-06-22; 采用时间: 2022-09-14; jos 在线出版时间: 2023-04-13
CNKI 网络首发时间: 2023-04-13

smooth texture and complex texture under the same authentication accuracy, and can also realize independent authentication and restoration of almost all sub-blocks, which has widespread applicability.

Key words: image authentication; tamper detection; fragile watermarking; reversible data hiding

随着互联网和多媒体技术的快速发展, 数字图像、音视频作品已遍布人们日常生活的每个角落. 同时随着计算机软件技术的迅速发展, 图像、音视频处理软件也越来越多, 不法分子利用这些软件工具能够轻易地对多媒体文件进行篡改, 如何确保海量多媒体文件在存储和传输过程中的原始性和完整性是一个亟待解决的问题[1,2].

为了实现对特定图像的保护, 脆弱水印[3−7]是一种较为常见的认证方法. 通过在需要保护的图像中提前嵌入脆弱水印, 一旦图像在存储和传输过程中受到有意或无意的篡改, 那么可以从图像中提取出所嵌入的脆弱水印来对图像的原始性和完整性进行认证, 甚至对篡改区域进行定位. 当前, 利用脆弱水印技术开发出高效可靠的篡改检测方法受到了较多关注. 但由于信息嵌入所具有的入侵特性, 传统信息隐藏技术在嵌入信息的同时必然会破坏原始载体的某些视听觉特性. 常见的脆弱水印方法[3−7]大多采用这种有损嵌入方式. 通常而言, 有损嵌入方案可以在图像视觉质量保持、嵌入容量、篡改区域定位等方面取得较为满意的性能, 但是有损嵌入不可避免地会导致原始载体图像的某些信息丢失[3].

可逆信息隐藏是一种新的无损信息嵌入方式, 它可以在提取所嵌入信息的同时, 无失真地恢复原始载体, 因此也被称之为无损信息隐藏[8,9]. 该技术在近年来受到了学者们的广泛关注, 其中差值扩展[10]和直方图平移[11]是两种基本的方法. 差值扩展 (difference expansion, DE) 最早由 Tian[10]提出, 通过扩展像素对的差值来嵌入信息. 直方图平移 (histogram shifting, HS) 方法由 Ni 等人[11]提出, 通过移动图像直方图峰值点和零点间的像素得到可嵌入空间, 并将信息嵌入到峰值点中. 基于差值扩展的方法通常具有较大的容量, 但容易引入较大的失真; 而基于直方图平移的方法, 虽然容量受到峰值点的限制, 但是带来的失真较小, 能够保持较高的图像质量. 在这两种方法的基础上, 利用图像相邻像素之间的相关性, 构造更加尖锐的预测误差直方图[12−15], 通过对预测误差直方图进行平移的方式[16−18]以提高嵌入容量并降低嵌入失真, 成为目前可逆信息隐藏领域的主流方法[8]. 后续基于预测误差直方图平移的思想, 研究者们发展了一系列新的方法, 如基于像素排序 (pixel value ordering, PVO)[19]及其改进算法 (improved PVO, IPVO)[20−22], 以及基于高维预测误差直方图平移和扩展的方法等[23−26].

可逆认证 (reversible authentication, RA)[27−32]可以看作是可逆信息隐藏和脆弱水印技术的结合[32], 它既能实现对图像的脆弱认证, 还能在提取认证信息同时无失真地恢复原始载体图像. 在一些对于图像质量要求比较严格的领域, 如医学、司法、军事、监控等领域具有重要意义. 近年来, 可逆认证方案受到越来越广泛的关注. 为了确保对图像中被篡改区域的定位, 通常可逆认证方案以块为单位来进行信息嵌入和提取. 2014 年, Lo 等人[28]首次提出了具有篡改定位能力的可逆认证方法, 通过将图像分成 N 个子块, 选取每个子块中间的像素作为基准值, 利用基准值和块内其他像素的差值构造预测误差直方图, 然后运用预测误差直方图平移方法将 N 比特认证码嵌入到 N 个子块中. 在 Lo 等人的基础上, Yin 等人[29]进一步提出首先利用希尔伯特曲线对图像进行扫描, 然后将得到的像素分成 N 个子块, 最后运用 IPVO 方法将 N 比特认证码嵌入到 N 个子块中. 但上述方案在实际中可能存在部分子块的嵌入容量为 0, 因此对这部分非可嵌入子块无法实现有效认证的情况. 另外, 由于上述方法中嵌入的认证码与当前子块自身信息无关, 因此可以选择对部分子块进行常数攻击 (子块内所有的像素同时加上或减去一个常数). 由于常数攻击并不会改变该子块所对应的预测误差直方图[30], 因此不影响认证信息的提取, 从而可能导致篡改无法被有效检测. 在文献 [30] 中, Hong 等人对上述方法进行了改进, 将最低有效位 (least significant bit, LSB) 替换方法和 IPVO 方法相结合, 实现了对非可嵌入子块的认证. Yao 等人[31]在 Hong 等人[30]的基础上提出了一种自适应分块的可逆认证方法, 能够将图像划分为更小的子块, 提高了认证精度. 此外王泓等人[32]指出, Hong 等人[30]的方法对块中的部分像素不能有效认证, 存在一定的安全隐患. 针对这一点, 王泓等人[32]成功对 Hong 等人[30]的方法进行了攻击. 同时为了进一步提高方法的安全性, 王泓等人提出可以将更多像素引入认证码的生成过程中并在嵌入前对图像块进行置乱.

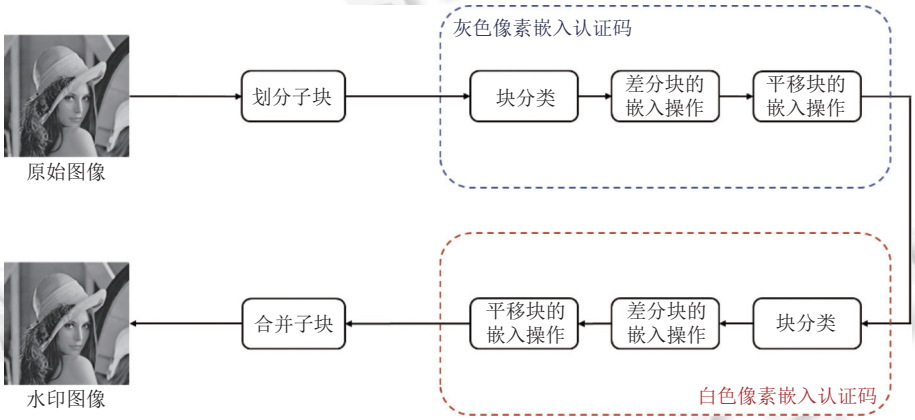
文献 [30−32] 虽然能够对所有子块进行认证, 但是需要保存大量的边信息, 对于具有复杂纹理的图像, 其容量可能不足以将边信息完整保存, 在不改变分块大小的情况下难以实现对原始载体的可逆恢复. 此外由于在每个子块中嵌入的认证码位数较少, 通常会导致对于篡改区域的正确检测率较低. 针对这两点, 本文提出了一种新的可逆认证方法, 根据每个子块的可嵌入容量将图像块分为差分块和平移块, 分别采用差值扩展和预测误差直方图平移

的方式嵌入认证码以保证每个子块的独立恢复, 对于具有复杂纹理的图像或图像的纹理复杂区域能实现更有效保护. 同时为了充分利用图像中的信息冗余, 本文采取两层嵌入操作, 确保在每个块内至少可以嵌入 2 比特认证码, 有效提高了对潜在恶意篡改的检测效率. 此外, 本文方法还可与形态学中的膨胀和腐蚀操作结合以细化篡改检测标记, 进一步提高检测效果.

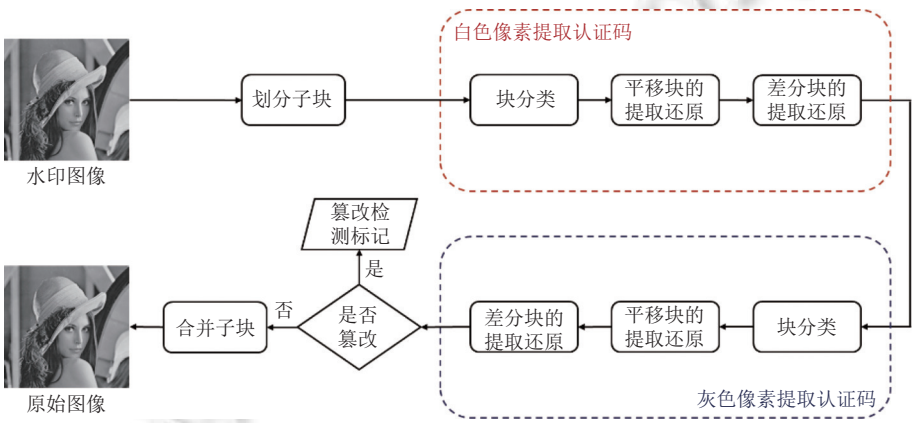
本文第 1 节详细介绍本文所提出的可逆认证方法. 第 2 节通过对比实验证明了本文方法的有效性. 第 3 节是本文总结.

1 基于分块的图像可逆认证方法

本文算法的整体框架如图 1 所示, 其中图 1(a) 为认证码的嵌入流程, 图 1(b) 为认证码的提取与图像恢复流程. 在认证码的嵌入流程中, 首先将大小为 $M\times N$ 原始图像 I 划分为不重叠的 4×4 子块, 然后将子块中的像素按照如图 2(a) 所示棋盘格方式划分为两个集合, 即灰色像素集合和白色像素集合, 认证码的嵌入采取如下两层嵌入方式: 第 1 层用白色像素对灰色像素进行预测, 根据所得到的预测误差直方图统计当前块的可嵌入容量, 根据嵌入容量将不同的块定义为差分块或平移块, 先对差分块采用差值扩展方式嵌入认证码, 再对平移块采用预测误差直方图平移方式嵌入认证码; 第 2 层用灰色像素对白色像素进行预测, 并进行类似的分块和嵌入操作. 当两层嵌入完成后, 合并所有的 4×4 子块, 输出含水印图像. 认证码的提取认证与图像恢复为认证码嵌入的逆操作, 第 1.1–1.3 节将以第 1 层嵌入为例对本文方法进行具体说明 (在第 1 层嵌入过程中, 只对灰色像素进行改变, 白色像素在嵌入前后保持不变, 而在第 2 层嵌入过程中, 只对白色像素进行改变, 灰色像素在嵌入前后保持不变).



(a) 认证码的嵌入流程



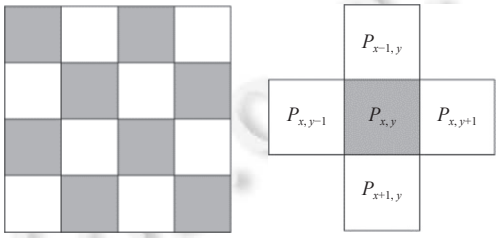
(b) 认证码的提取与图像恢复流程

图 1 本文算法框架

1.1 块分类和认证码生成

为了避免值为 0, 1, 254 或 255 的像素在嵌入过程中产生上溢或者下溢, 在每一层操作前均统一对输入图像进行如下预处理操作. 具体如下: 首先创建一个空的动态数组, 并将其作为预处理定位表 *LM* (location map), 然后顺次扫描图像, 当扫描到值为 0、1、254 或 255 的像素, 则分别将其修改为 2、3、252 或 253, 并在 *LM* 中添加字符 1; 当扫描到原始值为 2、3、252 或 253 的像素, 则直接在 *LM* 中添加字符 0. 预处理完成后, *LM* 中的 0、1 字符的个数即为输入图像中值为 0、1、2、3、252、253、254 和 255 的像素个数. 为了减少 *LM* 占用的空间, 文中我们使用 JBIG2 码[33]对其进行了压缩处理.

预处理完成后, 将每个 4×4 子块按照如图 2(a) 棋盘格的方式划分为灰色像素和白色像素集合, 以灰色像素集合为例, 根据如图 2(b) 的菱形预测器及公式 (1) 和公式 (2) 得到灰色像素 $P_{x,y}$ 的预测值 $\bar{P}_{x,y}$ 和预测误差 $e_{x,y}$. 其中 x 和 y 为像素对应的横纵坐标 ($1 \leqslant x \leqslant M, 1 \leqslant y \leqslant N$), $P_{x-1,y}, P_{x+1,y}, P_{x,y-1}, P_{x,y+1}$ 分别表示 $P_{x,y}$ 上下左右 4 个相邻的像素, [·] 表示四舍五入取整 (每块中位于角落或边上的像素只利用其相邻 2 或 3 个像素的平均值作为该点的预测值).



(a) 集合划分 (b) 菱形预测器

图 2 集合划分与菱形预测器

根据得到的预测误差值 $e_{x,y}$. 统计该子块的可嵌入容量 $V = \#\{e_{x,y} | e_{x,y} = PK_1 \| e_{x,y} = PK_2\}$, 其中#表示集合中元素的个数, PK_1 和 PK_2 分别为在该子块预测误差直方图中选定的两个峰值点. 若 V 为 0, 则将该块定义为差分块 (differential block, DB), 后期将采用差分扩展的方式在该子块中嵌入认证信息; 否则将其定义为平移块 (shifting block, SB), 后期将采用预测误差直方图平移的方式在该子块中嵌入认证信息. 由于预测误差直方图的峰值点一般位于零点附近, 为了获得更多的平移块, 本方法中一般设置 PK_1 和 PK_2 的值为−1 和 0.

$$\bar{P}_{x,y} = \left\lfloor \frac{P_{x-1,y} + P_{x+1,y} + P_{x,y-1} + P_{x,y+1}}{4} \right\rfloor \tag{1}$$

$$e_{x,y} = P_{x,y} - \bar{P}_{x,y} \tag{2}$$

对块进行分类后, 利用哈希函数 $Hash(block, row, col, key)$ 为每个 4×4 子块生成一个 8 比特的认证码 (authentication code, *AC*), 即 $AC = \{b_1, b_2, \ldots, b_8\}$, 其中 *block* 为当前块中所包含像素的相关信息, *row* 和 *col* 分别为当前块所在的行和列, *key* 为发送方和接收方事先约定好的密钥. 为了有效抵抗拼贴攻击、常数攻击以及对任意像素的篡改等, 本文在生成认证码的过程中将块中的所有像素信息以及块的位置信息等作为参数. 上述 8 比特认证码并不一定都能嵌入到当前子块中, 但考虑到嵌入更多比特的认证码能够有效提高对子块的认证精度, 在嵌入过程中我们将根据当前块的可嵌入容量嵌入尽可能多的认证信息.

1.2 差分块的认证方法

差分块的认证方法主要分为两部分, 即认证码的嵌入和认证码的提取. 在嵌入端, 采用差值扩展方法将 1 比特认证码 $AC = \{b_1\}$ 嵌入差分块中; 在认证端, 采用相应的逆操作将嵌入的 *AC* 提取并进行认证. 下面我们将以第 1 层嵌入为例详细介绍差分块的认证方法.

1.2.1 认证码的嵌入

为了尽可能减少差值扩展过程中引入的失真以及可能产生的溢出, 我们采取先对预测误差排序, 然后根据灰度值自适应地选择扩展方向, 最后利用预测误差之间的差值进行扩展的方式. 具体做法如下所示.

Step 1. 对于任意给定 4×4 差分块, 按从上到下从左到右的次序扫描得到灰色像素序列$\{P_1,P_2,\ldots,P_{n-1},P_n\}$, 然后根据公式 (1) 和公式 (2) 得到灰色像素序列对应的预测值序列$\{\bar{P}_1,\bar{P}_2,\ldots,\bar{P}_{n-1},\bar{P}_n\}$及预测误差序列$\{e_1,e_2,\ldots,e_{n-1},e_n\}$, 并按从小到大的顺序对预测误差序列进行排序得到$\{e_{\sigma_1}\leqslant e_{\sigma_2}\leqslant\ldots\leqslant e_{\sigma_{n-1}}\leqslant e_{\sigma_n}\}$. 其中 n 为子块中灰色像素的个数 (本文一般采取 4×4 分块, 故 n 的取值为 8), 下标$\sigma_i(0\leqslant i\leqslant n)$为预测误差$e_{\sigma_i}$在初始序列$\{e_1,e_2,\ldots,e_{n-1},e_n\}$中对应的位置. 排序方法为稳定排序, 即当$e_i\leqslant e_j(0\leqslant j\leqslant n)$且$i<j$时, 有$\sigma_i\leqslant\sigma_j$.

Step 2. 考虑到灰度值接近 0 或 255 的像素在差值扩展后可能溢出, 在具体嵌入前首先统计该子块内灰度值在闭区间 [0, 7] 和闭区间 [248, 255] 中的白色像素个数, 分别记为记为A_1和A_2. 若$A_1\leqslant A_2$, 说明该子块中可能有更多的像素灰度值接近 255, 则选择将预测误差值e_{σ_1}向下扩展, 即对应的被修改像素值减小, 从而避免向上溢出; 反之, 若$A_1<A_2$, 说明该子块中可能有更多的像素灰度值接近 0, 则选择将预测误差值e_{σ_n}向上扩展, 即对应的被修改像素值增大, 从而避免向下溢出. 向下扩展如公式 (3) 所示, 其中 h 表示$e_{\sigma_1},e_{\sigma_2}$的差值, H 为 h 扩展后的值, $b\in\{0,1\}$表示要嵌入的 1 比特认证码 (即第 1.1 节所生成的该子块认证码AC的第 1 个比特), e'_{σ_1}为e_{σ_1}扩展后的值.

$$\begin{cases} h=e_{\sigma_2}-e_{\sigma_1} \\ H=2h+b \\ e'_{\sigma_1}=e_{\sigma_2}-H \end{cases} \tag{3}$$

向上扩展如公式 (4) 所示. 其中, h 表示$e_{\sigma_{n-1}},e_{\sigma_n}$的差值, H 为 h 扩展后的值, $b\in\{0,1\}$表示要嵌入的 1 比特认证码 (即第 1.1 节所生成的该子块认证码AC的第 1 个比特), e'_{σ_n}为e_{σ_n}扩展后的值..

$$\begin{cases} h=e_{\sigma_n}-e_{\sigma_{n-1}} \\ H=2h+b \\ e'_{\sigma_n}=e_{\sigma_{n-1}}+H \end{cases} \tag{4}$$

Step 3. 经过 Step 1 和 Step 2 得到新的预测误差序列为$\{e'_1,e'_2,\ldots,e'_{n-1},e'_n\}$, 为避免在提取过程中将差分块与平移块混淆, 按公式 (5) 对预测误差值$e'_i(0\leqslant i\leqslant n)$进行平移操作得到平移后的预测误差值$e''_i(0\leqslant i\leqslant n)$, 根据公式 (6) 计算可得嵌入认证信息后的灰色像素值$P'_i(0\leqslant i\leqslant n)$.

$$e''_i=\begin{cases} e'_i-2, \text{if } e'_i\leqslant PK_1 \\ e'_i+2, \text{if } e'_i\geqslant PK_2 \\ e'_i, \text{otherwise} \end{cases} \tag{5}$$

$$P'_i=\bar{P}_i+e''_i \tag{6}$$

Step 4. 如果得到的像素值P'_i溢出, 则将溢出像素P'_i在原始图像中对应的横纵坐标 x、y 以及溢出量 (*value*) 的绝对值|*value*|作为二进制序列$(x,y,|value|)_2$顺次保存在预先设定的动态数组 S 中. 然后将小于 0 的P'_i修改为 0, 将大于 255 的P'_i修改为 255. 认证方在提取认证码前需要对产生溢出的像素进行恢复.

Step 5. 重复以上操作, 处理所有的差分块. 最后将预处理定位表 LM 与动态数组 S 的进行合并得到完整的边信息序列 M, 并在后续将其嵌入到部分指定的平移块中.

1.2.2 认证码的提取

认证方根据嵌入方法的逆操作提取认证码并进行认证, 具体做法如下所示.

Step 1. 从平移块中提取数组 S 的相关信息并恢复上述差分嵌入中产生溢出的像素.

Step 2. 在嵌入信息后的子块中找到灰色像素序列$\{P'_1,P'_2,\ldots,P'_{n-1},P'_n\}$, 根据公式 (1) 和公式 (2) 计算其对应的预测值序列$\{\bar{P}_1,\bar{P}_2,\ldots,\bar{P}_{n-1},\bar{P}_n\}$和预测误差值序列$\{e''_1,e''_2,\ldots,e''_{n-1},e''_n\}$, 并按照公式 (7) 将所有预测误差$e''_i(0\leqslant i\leqslant n)$平移复原得到$e'_i(0\leqslant i\leqslant n)$.

$$e'_i=\begin{cases} e''_i+2, \text{if } e''_i\leqslant PK_1-2 \\ e''_i-2, \text{if } e''_i\geqslant PK_2+2 \\ e''_i, \text{otherwise} \end{cases} \tag{7}$$

Step 3. 按照与嵌入时同样的规则将序列$\{e'_1,e'_2,\ldots,e'_{n-1},e'_n\}$进行排序得到$\{e'_{\sigma_1},e'_{\sigma_2},\ldots,e'_{\sigma_{n-1}},e'_{\sigma_n}\}$, 利用差值扩展的逆操作进行还原. 对于向下扩展按照公式 (8) 进行还原, 其中 H 表示预测误差e'_{σ_1}和e'_{σ_2}的差值, $\lfloor\cdot\rfloor$表示向下取整, h 为 H 还原后的值, e_{σ_1}为e'_{σ_1}还原后的值.

$$\begin{cases} H = e'_{\sigma_2} - e'_{\sigma_1} \\ h = \left\lfloor \dfrac{H}{2} \right\rfloor \\ e_{\sigma_1} = e'_{\sigma_2} - h \end{cases} \tag{8}$$

对于向上扩展按照公式 (9) 进行还原, 其中 H 表示预测误差 $e'_{\sigma_{n-1}}$ 和 e'_{σ_n} 的差值, $\lfloor\cdot\rfloor$ 表示向下取整, h 为 H 还原后的值, e_{σ_n} 为 e'_{σ_n} 还原后的值.

$$\begin{cases} H = e'_{\sigma_n} - e'_{\sigma_{n-1}} \\ h = \left\lfloor \dfrac{H}{2} \right\rfloor \\ e_{\sigma_n} = e'_{\sigma_{n-1}} + h \end{cases} \tag{9}$$

Step 4. 经过以上操作得到的还原后的预测误差序列为 $\{e_1, e_2, \ldots, e_{n-1}, e_n\}$, 根据公式 (10) 和公式 (11) 即可得到还原后的像素值 P_i $(0 \leqslant i \leqslant n)$ 及提取出的认证信息 b.

$$P_i = \bar{P}_i + e_i \tag{10}$$

$$b = H - 2h \tag{11}$$

根据第 1.1 节的哈希函数重新生成该子块的认证码, 如果其与提取的认证信息 b 相同则认为该块未被篡改; 否则说明该块已被篡改. 重复以上操作, 处理所有的差分块.

1.3 平移块的认证方法

平移块的认证方法同样分为认证码的嵌入和认证码的提取两部分. 在嵌入端, 采用预测误差直方图平移方法嵌入认证码, 并按照子块的容量调整嵌入的认证码位数; 在认证端, 采用相应的逆操作将嵌入的认证码提取并进行认证. 下面我们将以第 1 层嵌入为例详细介绍平移块的认证方法.

1.3.1 认证码的嵌入

平移块的容量 V 为 1–8 比特, 为了提高认证能力, 降低从篡改块中提取的认证码与重新生成的认证码相等的概率, 当 $V<3$ 时只嵌入第 1.1 节所生成的该子块认证码 AC 的前 V 比特, 即嵌入序列 $L = \{b_1, b_2, \ldots, b_V\}$; 当 $V \geqslant 3$ 时, 从第 1.2 节生成的边信息数组 M 中顺次提取 1 比特 m, 并将其拼接到第 1.1 节所生成的该子块认证码 AC 的前 V–1 比特认证码之后, 得到 V 比特嵌入序列 $L = \{b_1, b_2, \ldots, b_{V-1}, m\}$ (若 M 已经嵌入完成, 则只嵌入 V 比特认证码即可). 具体嵌入做法如下.

Step 1. 对于任意给定 4×4 平移块, 按从上到下从左到右的次序扫描得到灰色像素序列 $\{P_1, P_2, \ldots, P_{n-1}, P_n\}$, 然后根据公式 (1) 和公式 (2) 得到灰色像素序列对应的预测值序列 $\{P_1, P_2, \ldots, P_{n-1}, P_n\}$ 及预测误差序列 $\{e_1, e_2, \ldots, e_{n-1}, e_n\}$, 按顺序访问序列中的预测误差 e_i $(0 \leqslant i \leqslant n)$ 并根据公式 (12) 进行嵌入操作得到修改后的预测误差 e'_i, 其中 PK_1 和 PK_2 为上文所述用于平移嵌入的两个峰值位置, $b \in \{0, 1\}$ 表示顺次从嵌入序列 L 中取出的 1 比特信息.

$$e'_i = \begin{cases} e_i - 1, \text{ if } e_i < PK_1 \\ e_i + 1, \text{ if } e_i > PK_2 \\ e_i - b, \text{ if } e_i = PK_1 \\ e_i + b, \text{ if } e_i = PK_2 \\ e_i, \text{ otherwise} \end{cases} \tag{12}$$

Step 2. 根据公式 (13) 得到信息嵌入后像素值 P'_i $(0 \leqslant i \leqslant n)$.

$$P'_i = \bar{P}_i + e'_i \tag{13}$$

Step 3. 重复以上操作, 处理所有的平移块.

1.3.2 认证码的提取

认证方根据嵌入方法的逆操作提取认证码并进行认证, 具体做法如下所示.

Step 1. 在嵌入信息后的子块中找到灰色像素序列 $\{P'_1, P'_2, , P'_{n-1}, P'_n\}$, 根据公式 (1) 和公式 (2) 计算对应的预测值序列 $\{\bar{P}_1, \bar{P}_2, \ldots, \bar{P}_{n-1}, \bar{P}_n\}$ 及预测误差值序列 $\{e'_1, e'_2, \ldots, e'_{n-1}, e'_n\}$, 根据预测误差 e'_i $(0 \leqslant i \leqslant n)$ 统计该块的容量, 按顺序访问预测误差序列并利用公式 (14) 从 e'_i 中提取嵌入的信息 $b \in \{0, 1\}$, 得到该子块的嵌入序列 L.

$$b=\begin{cases}0, \text{ if } e'_i=PK_1 \text{或} e'_i=PK_2\\ 0, \text{ if } e'_i=PK_1-1 \text{或} e'_i=PK_2+1\end{cases} \tag{14}$$

Step 2. 按照公式 (15) 对所有 $e'_i(0\leqslant i\leqslant n)$ 进行平移恢复得到还原后的预测误差序列 $\{e_1,e_2,\ldots,e_{n-1},e_n\}$, 并根据公式 (16) 计算还原后的像素值 P_i.

$$e_i=\begin{cases}e'_i+1, \text{ if } e'_i<PK_1\\ e'_i-1, \text{ if } e'_i>PK_2\\ e'_i, \text{ otherwise}\end{cases} \tag{15}$$

$$P'_i=\bar{P}_i+e'_i \tag{16}$$

Step 3. 从提取出的 V 比特序列 L 中得到 V–1 比特认证码和 1 比特附加信息 m (当 V<3 时, 只得到 V 比特认证码), 将 m 顺次保存在动态数组 M 中. 根据第 1.1 节的哈希函数重新生成该子块的认证码, 如果其与提取的认证信息 b 相同则认为该块未被篡改; 否则说明该块已被篡改.

Step 4. 重复以上操作, 处理所有的平移块. 当所有平移块都处理完成后, 从动态数组 M 中分离出存放差分块溢出信息的数组 S 和预处理定位表 LM.

在上述第 1.1–1.3 节中, 我们详细介绍了在第 1 层中, 认证码的嵌入提取及图像恢复具体流程. 在第 1 层嵌入完成后, 紧跟着以同样的方式进行第 2 层嵌入操作. 认证码的提取和图像恢复是上述嵌入过程的逆过程.

1.4 篡改检测标记细化处理

根据前文描述, 本文认证码的嵌入采取了分层嵌入的方式. 以 4×4 子块为例, 每一层嵌入过程中可以嵌入 1–8 比特认证码, 其中差分块固定为 1 比特, 而平移块至少 1 比特, 最多为 8 比特. 因此, 经过两层嵌入后每个子块中可嵌入 2–16 比特认证码. 在认证端, 同样采取分层提取的方式, 从每个子块中可以提取的认证码最少为 2 比特, 最多为 16 比特. 如果该子块中所嵌入的认证码偏少, 对于部分被篡改的区域, 可能出现提取的认证码和重新生成的认证码恰好相等的情况. 以子块中嵌入 2 比特认证码为例, 这种巧合的概率理论上为 1/4.

考虑到实际应用中篡改区域一般不会小于 4×4 范围, 因此为了进一步提高对篡改区域的检测和定位效果, 本文结合形态学中的膨胀和腐蚀操作对常见的细化处理方式[28]进行了改进. 具体如下: 首先对检测到的原始篡改子块使用如图 3 的结构元进行膨胀操作, 结构元素中的每一个小正方形对应于一个 4×4 子块, 膨胀操作时将结构元素的中心依次与初始篡改标记中的每个篡改块重叠, 将与该篡改块相邻的 8 个子块也标记为篡改块; 然后利用文献 [28] 中 Lo 等人提出的细化方式, 即依次检查如图 4 所示的任意 4×4 子块 B 的水平方向 (图 4(a))、垂直方向 (图 4(b))、主对角线方向 (图 4(c)) 或副对角线方向 (图 4(d)) 上相邻的两块, 若任意一个方向上的相邻两块都被标记为篡改块, 则将块 B 也标记为篡改块, 迭代处理直到没有新的块被标记为篡改块; 最后, 用如图 3 所示的结构元素对篡改区域进行腐蚀操作, 将结构元素的中心依次与每个篡改块重叠, 若与该篡改块相邻的 8 个子块均被标记为篡改则将其标记为最终篡改块, 处理完所有块后, 得到所有的最终篡改块. 本文后续实验表明, 在迭代处理前执行膨胀操作可以进一步连接篡改标记边缘的断点并填充内部的孔洞, 在迭代处理结束后用相同的结构元素进行腐蚀操作减少细化过程中带来误检块数.

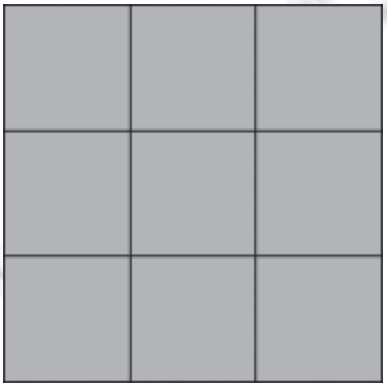
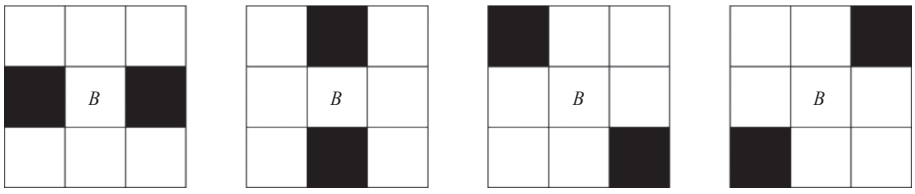


图 3　3×3 的结构元素



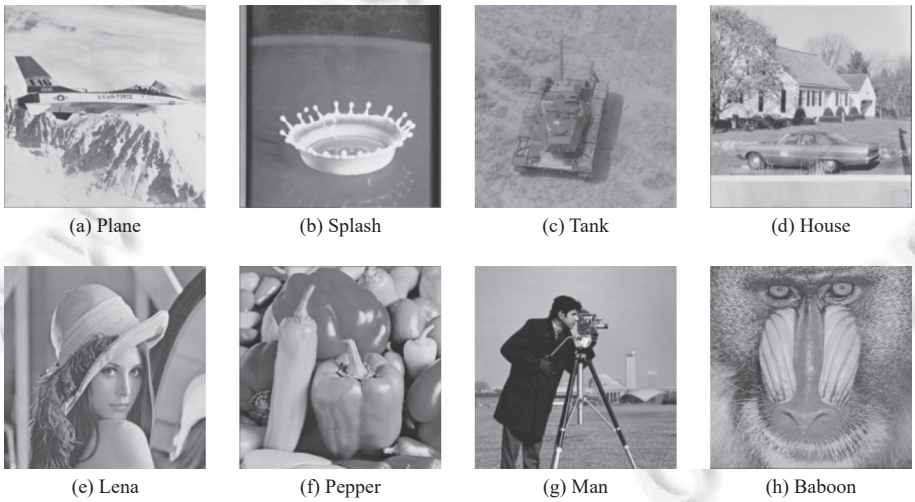
(a) 水平方向 (b) 垂直方向 (c) 主对角线方向 (d) 副对角线方向

图 4 块 B 的 4 种相邻关系

2 实验分析

实验部分我们主要针对图 5 中的 8 幅 512×512 大小的标准灰度图像进行了测试. 使用峰值信噪比 (peak signal-to-noise rate, PSNR) 和结构相似度 (structural similarity, SSIM) 两个性能指标来衡量嵌入认证信息后图像的视觉质量, 使用正确检测率 (correct detection rate, CR) 来评价本文方法的认证性能. 本文使用的正确检测率计算方式如公式 (17), 其中 C_{block} 表示正确检测出的篡改块数目, T_{block} 表示实际的篡改块数目. 为了证明本文方法的有效性, 我们从嵌入容量、视觉质量以及认证精度 3 个方面将我们算法与 Hong 等人[30]、Yao 等人[31]和王泓等人[32]的最新方法进行了对比.

$$CR=\frac{C_{\text{block}}}{T_{\text{block}}}\times 100\% \tag{17}$$



(a) Plane (b) Splash (c) Tank (d) House

(e) Lena (f) Pepper (g) Man (h) Baboon

图 5 实验图像

2.1 嵌入容量和边信息长度对比

表 1 为本文方法与 Hong 等人[30]、Yao 等人[31]和王泓等人[32]的方法提供的嵌入容量及边信息长度对比. 一般情况下, 嵌入容量越大且所需保存的边信息越少, 则能够嵌入更多认证码从而达到更好的认证效果. 从表 1 中可以看到, 在 4×4 分块下本文方法的嵌入容量远大于其他 3 种方法. 其中一个重要的原因在于本文采取了双层嵌入的方式, 子块中的所有像素均被用于提供嵌入空间, 而文献 [30−32] 中只有不到一半的像素用于提供嵌入空间. 同时, 由于文献 [30−32] 中为了实现容量为 0 的块 (非可嵌入块) 的可逆认证, 采用了 LSB 替换方式来保存原始图像非可嵌入块中的部分信息, 从而产生了更多的边信息, 而本文方法对于容量为 0 的块 (差分块) 采用差值扩展方式能够实现该块的独立认证和恢复, 仅需保存个别溢出像素的信息, 产生的边信息远少于其他 3 种方法. 从表 1 中可以看到, 特别是对于类似 Baboon 等具有复杂纹理的图像, 文献 [30,32] 将产生大量的边信息. 由于嵌入容量的限制, 文献 [30,32] 中的方法可能出现部分认证码或边信息无法嵌入的情况, 因而无法实现有效的可逆认证. 而文献 [31]

虽然通过自适应划分为更大的子块, 能够对具有复杂纹理的图像进行可逆认证, 但是会导致认证精度下降. 本文方法由于产生的边信息较少, 因此对于具有复杂纹理的图像可以提供更有效保护.

表 1 嵌入容量和边信息长度对比

图像	Hong等人[30]		Yao等人[31]		王泓等人[32]		本文方法	
	容量 (10^4 bit)	边信息 (bit)	容量 (10^4 bit)	边信息 (bit)	容量 (10^4 bit)	边信息 (bit)	容量 (10^4 bit)	边信息 (bit)
Plane	5.2	1 794	5.2	7 742	4.7	2 384	**8.7**	**26**
Splash	5.3	739	5.3	5 744	4.6	1 109	**8.7**	**104**
Tank	2.8	2 867	2.8	2 867	2.5	3 527	**3.9**	**52**
House	4.6	2 981	4.6	10 856	4.2	3 784	**7.6**	**130**
Lena	3.8	2 237	3.8	10 000	3.2	3 360	**6.0**	**26**
Pepper	3.1	2 566	3.1	2 568	2.8	3 197	**3.8**	**26**
Man	7.2	1 506	7.1	6 106	5.9	2 112	**14.1**	**156**
Baboon	1.3	7 705	1.3	3 818	1.2	8 346	**2.0**	**416**
平均值	4.2	2 799	4.2	6 213	3.6	3 477	**6.8**	**117**

2.2 水印图像质量比较

表 2 给出了测试图像在嵌入认证码后的 PSNR 和 SSIM 值. 从表 2 中可以看到本文方法得到的水印图像 PSNR 值和 SSIM 值要低于其他 3 种方法, 这是因为本文方法为了确保对每一个子块的有效认证, 在差分块中采取了差值扩展的嵌入方式, 从而导致对原始图像的修改量较大. 而文献 [30−32] 采用 IPVO 的嵌入方式产生的修改量较小. 虽然本文方法的视觉质量有所下降, 但是一般情况下人眼对 PSNR 大于 38 dB 的失真感知通常是可接受的, 除了 Baboon 之外, 本文方法所得到的其他水印图像的 PSNR 值均大于 38 dB 的, 同时本文方法的 SSIM 值均保持在 0.98 以上, 说明水印图像与原始图像的结构相似性仍然较高. 从表 2 中也可以看到, 在 4×4 分块下, 对于具有复杂纹理的图像如 Baboon, 本文方法依然能够进行有效的保护, 而文献 [31] 中为了实现对 Baboon 的保护需要增大分块大小, 文献 [30,32] 在 4×4 分块下由于容量的限制无法对 Baboon 实现有效认证. 通常对于可逆认证而言, 由于原始载体是可恢复的, 因此相对于水印图像的视觉质量, 认证性能更值得关注. 实验结果表明, 本文方法在确保水印图像视觉质量在可以接受范围的同时, 能够获得更好的认证性能.

表 2 PSNR 和 SSIM 对比

图像	Hong等人[30]		Yao等人[31]		王泓等人[32]		本文方法	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
Plane	51.04	0.9964	48.84	0.9953	50.65	0.9963	42.40	0.9908
Splash	51.65	0.9957	49.55	0.9934	51.31	0.9955	46.08	0.9887
Tank	50.17	0.9966	50.43	0.9968	49.86	0.9964	42.32	0.9846
House	50.37	0.9972	47.85	0.9960	49.96	0.9971	41.24	0.9912
Lena	50.60	0.9963	47.95	0.9937	50.01	0.9960	42.64	0.9878
Pepper	50.34	0.9960	50.62	0.9964	50.02	0.9959	41.39	0.9813
Man	51.51	0.9965	49.89	0.9965	50.97	0.9962	44.35	0.9932
Baboon	—	—	49.69	0.9986	—	—	34.59	0.9834
平均值	50.81	0.9964	49.35	0.9958	50.40	0.9962	41.88	0.9876

2.3 认证性能比较

实验中对 4 种方法得到的 Lena 水印图像进行多种篡改攻击, 并且为了更直观地比较本文方法与其他方法的认证能力以及分块大小对认证性能的影响, 我们首先对不同分块大小下的初始正确检测率进行了比较, 然后结合细化处理进一步展示了本文算法的性能.

2.3.1 正确检测率比较

图 6 为以 4×4 分块作为篡改检测单元, 水印图像遭受剪贴攻击、常数攻击、拼贴攻击和随机篡改攻击 (篡改

比例为 1%, 即随机选择 1% 的子块, 并将子块内的像素随机修改为不大于 255 且不小于 0 的任意值) 后得到的初始检测标记, 其中图 6(a) 为被篡改的 Lena 水印图像, 图 6(b)–图 6(e) 分别是 Hong 等人[30], Yao 等人[31]和王泓等人[32]的方法以及本文方法得到的初始检测标记, 图中每个点表示一个 4×4 的子块. 可以明显看到相比于其他 3 种方法, 本文方法在不同种类的攻击下都能够检测出更多篡改块.

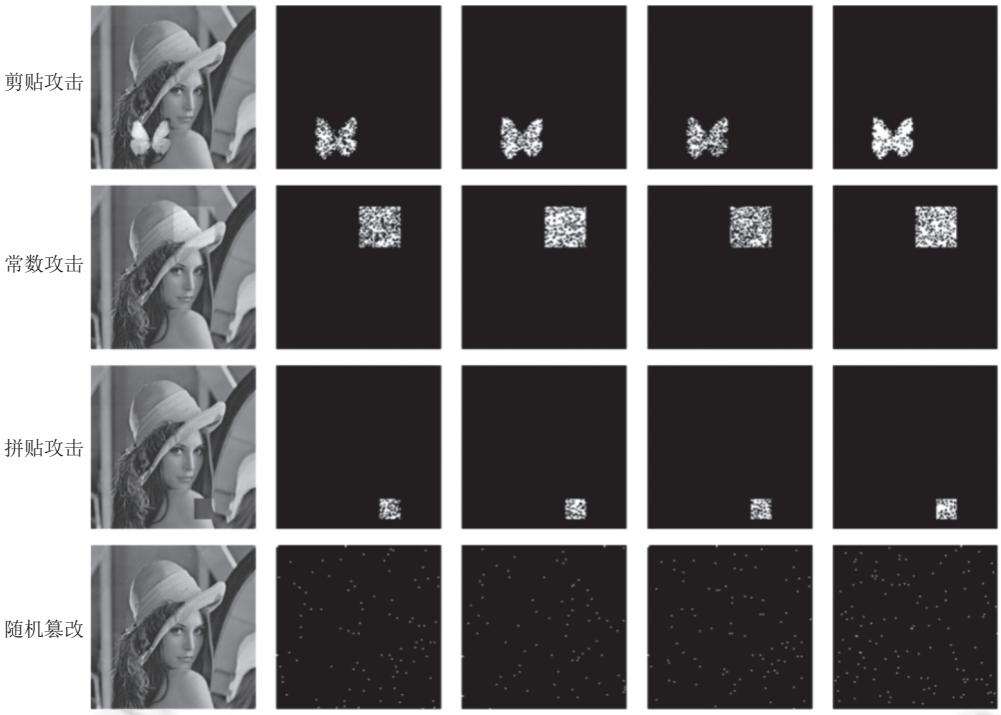


图 6 初始篡改检测标记

表 3 为分别以 4×4、8×8 和 16×16 分块为篡改检测单元得到的初始正确检测率对比, 可以看到在 4×4 分块大小下本文方法的正确检测率要明显高于其他 3 种方法. 这是因为在文献 [30,32] 中, 存在较多块只能嵌入 1 比特认证码, 文献 [32] 中为了统一篡改检测能力, 在所有子块中嵌入相同比特长度的认证码. 而本文方法中, 每块能嵌入至少 2 比特认证码, 并且部分块还能够嵌入 9–16 比特认证码, 减小了在篡改块中提取的认证码与重新生成的认证码相同的概率. 同时, 从表 3 中可以看到, 增大篡改检测单元的分块大小能够提高初始正确检测率, 这主要是因为在更大的分块条件下, 每个子块拥有更大的容量以嵌入更多的认证信息, 同时低容量的子块 (例如容量为 0 或 1) 的数量将会减少, 在一定程度上提高了检测率, 但是随着子块的增大, 相对而言会导致篡改定位的效果变差. 如以 16×16 大小的子块作为篡改检测单元为例, 如表 3 所示, 上述方法对所有 16×16 子块是否被篡改的正确检测率基本都可以达到 90% 以上, 但具体到 16×16 子块内部, 哪些 4×4 或 8×8 的子块曾经被篡改, 并不能被准确定位.

表 3 初始正确检测率对比 (%)

篡改攻击	Hong等人[30]			Yao等人[31]			王泓等人[32]			本文方法		
	4×4	8×8	16×16	4×4	8×8	16×16	4×4	8×8	16×16	4×4	8×8	16×16
剪贴攻击	66.47	95.73	98.53	73.60	95.26	97.06	60.67	88.63	98.53	84.13	96.68	98.53
常数攻击	63.28	96.48	100.0	74.12	99.61	100.0	62.50	87.89	96.88	75.00	99.61	100.0
拼贴攻击	62.50	95.31	100.0	70.70	100.0	100.0	69.92	95.31	100.0	76.17	100.0	100.0
随机篡改	58.64	87.50	90.00	48.47	57.50	60.00	58.02	82.50	90.00	81.48	85.00	90.00
平均值	61.97	93.76	97.13	66.72	88.09	89.26	62.78	88.58	96.35	78.81	95.32	97.13

2.3.2 细化效果比较

使用 Lo 等人[28]和本文的细化方法分别对图 6 中的篡改检测标记进行处理, 结果如图 7 和后文图 8 所示, 每行中对应的篡改攻击分别是剪贴攻击、常数攻击、拼贴攻击和随机篡改攻击 (篡改比例为 1%). 可以看到, 采用本文的细化方式能够得到更好的细化的效果. 后文表 4 为以 4×4 分块为篡改检测单元得到的细化后正确检测率对比, 从表中可以看到, 在大多数情况下使用本文方法能够将更多漏检的块标记出来, 尤其对于形状比较规则的篡改区域如图 7 和图 8 第 2 和 3 行能够明显提高正确检测率. 但是对于随机篡改攻击, 由于产生的标记是离散的, 篡改区域不集中, 漏检的块周围可能没有被标记的块, 无法在细化的过程中将漏检的块进行标记. 从攻击方的角度考虑, 要在空域图像上产生有意义的篡改, 篡改区域一般是连续、集中的, 因此本文方法通常能更高效地标记漏检块.

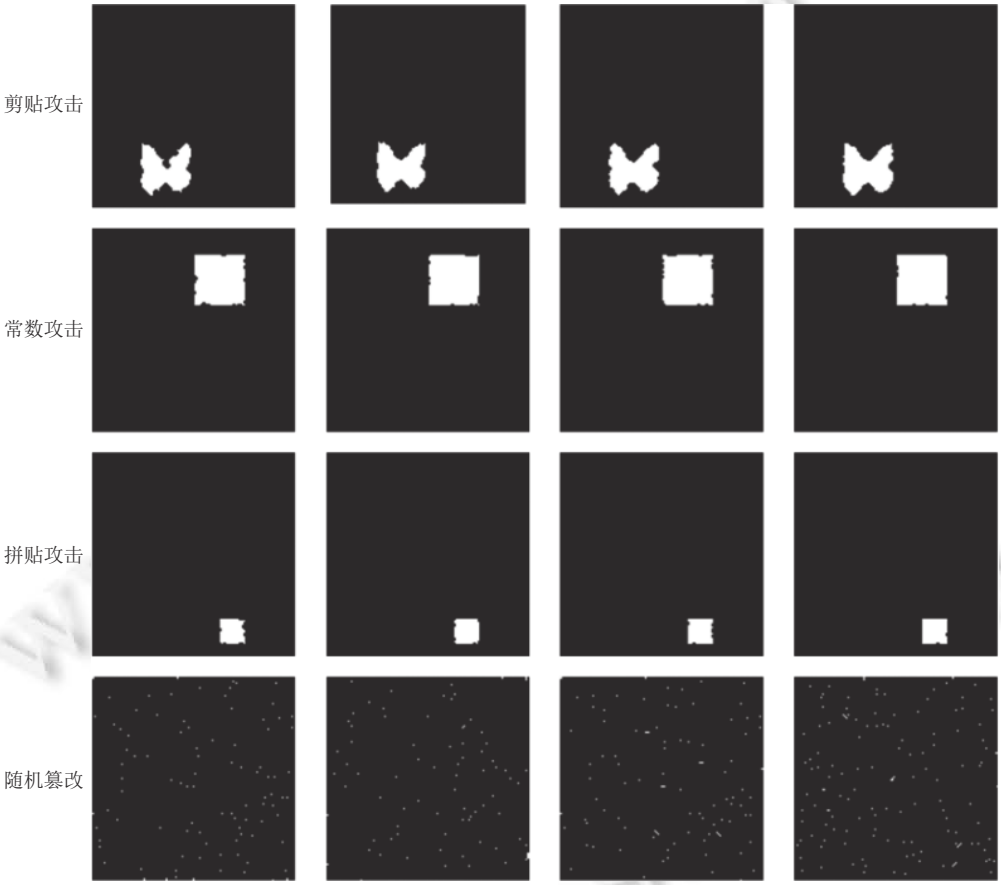


图 7 使用 Lo 等人[28]的方式对篡改检测标记细化处理

3 总 结

本文提出了一种基于分块的高效图像可逆认证方法. 与现有的可逆认证方法相比, 本文方法可以充分利用图像子块中的信息冗余, 减少了边信息的长度, 几乎能对所有子块进行独立认证. 同时通过采用两层嵌入方式, 进一步提高了嵌入容量. 实验结果表明, 本文方法的整体认证精度高于现有最优的方法, 而且能对具有复杂纹理的图像或图像中部分纹理复杂区域实现有效保护. 此外, 本文还结合形态学中的膨胀和腐蚀操作对已有的细化方案进行了改进, 进一步提高了算法的检测效率. 总体而言, 相比于比现有的可逆认证方法, 本文方法具有更高的篡改检测和定位能力.

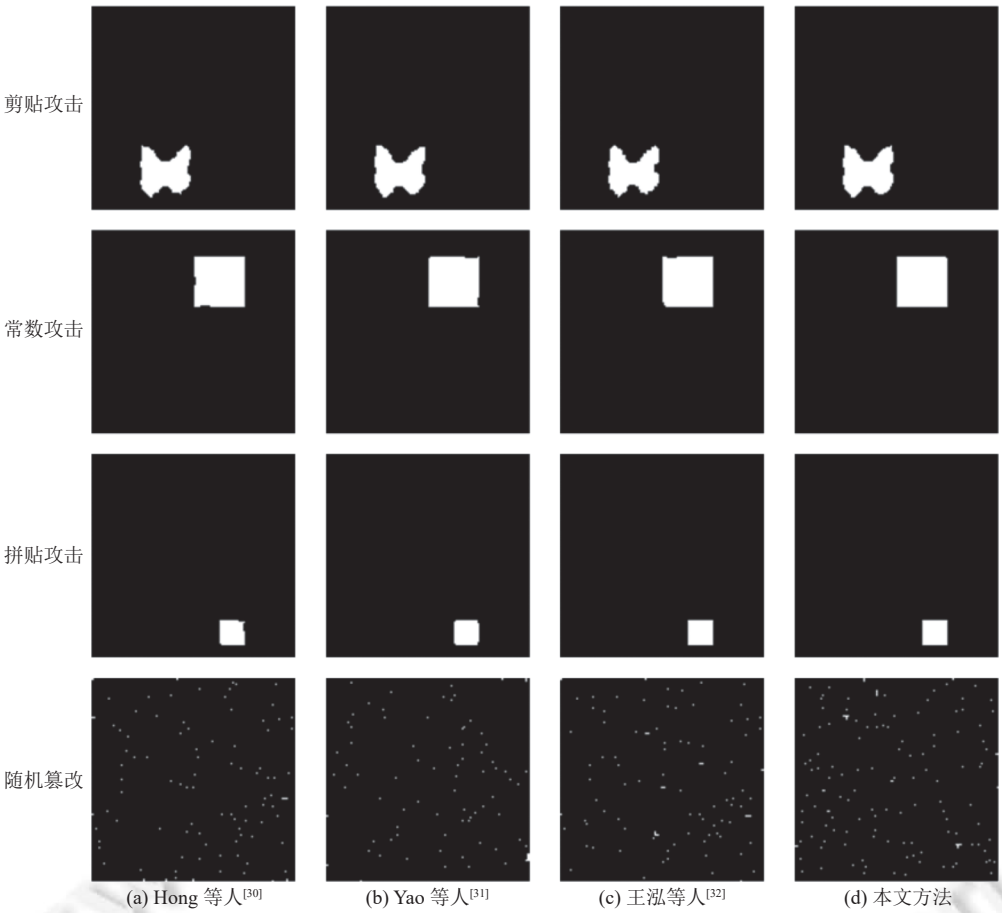


图 8　使用本文方法对篡改检测标记细化处理

表 4　不同细化方式正确检测率对比 (%)

篡改攻击	Hong等人[30]		Yao等人[31]		王泓等人[32]		本文方法	
	Lo等人[28]	本文	Lo等人[28]	本文	Lo等人[28]	本文	Lo等人[28]	本文
剪贴攻击	92.80	**95.33**	96.27	**97.07**	94.00	**95.73**	97.33	**98.27**
常数攻击	96.78	**98.93**	97.46	**98.63**	96.87	**99.02**	98.73	**99.90**
拼贴攻击	91.80	**96.09**	95.31	**97.66**	95.70	**100.0**	98.44	**100.0**
随机篡改	58.64	58.64	48.47	48.47	58.02	58.02	81.48	81.48
平均值	85.00	**87.25**	84.38	**85.46**	86.15	**88.20**	94.00	**94.91**

References:

[1] Ghosal SK, Mandal JK. Binomial transform based fragile watermarking for image authentication. Journal of Information Security and Applications, 2014, 19(4–5): 272–281. [doi: 10.1016/j.jisa.2014.07.004]

[2] Vyas C, Lunagaria M. A review on methods for image authentication and visual cryptography in digital image watermarking. In: Proc. of the 2014 IEEE Int'l Conf. on Computational Intelligence and Computing Research. Coimbatore: IEEE, 2014. 1–6. [doi: 10.1109/ICCIC.2014.7238504]

[3] Chang CC, Chen KN, Lee CF, Liu LJ. A secure fragile watermarking scheme based on chaos-and-hamming code. Journal of Systems and Software, 2011, 84(9): 1462–1470. [doi: 10.1016/j.jss.2011.02.029]

[4] Huo YR, He HJ, Chen F. Alterable-capacity fragile watermarking scheme with restoration capability. Optics Communications, 2012, 285(7): 1759–1766. [doi: 10.1016/j.optcom.2011.12.044]

[5] Qin C, Ji P, Zhang XP, Dong J, Wang JW. Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. Signal Processing, 2017, 138: 280–293. [doi: 10.1016/j.sigpro.2017.03.033]

[6] Su GD, Chang CC, Chen CC. A hybrid-Sudoku based fragile watermarking scheme for image tampering detection. Multimedia Tools and Applications, 2021, 80(8): 12881–12903. [doi: 10.1007/s11042-020-10451-1]

[7] Kim C, Yang CN. Self-embedding fragile watermarking scheme to detect image tampering using AMBTC and OPAP approaches. Applied Sciences, 2021, 11(3): 1146. [doi: 10.3390/app11031146]

[8] Zhang XP, Yin ZX. Data hiding in multimedia. Chinese Journal of Nature, 2017, 39(2): 87–95 (in Chinese with English abstract). [doi: 10.3969/j.issn.0253-9608.2017.02.002]

[9] Li XL. A review on image reversible data hiding. Journal of Information Security Research, 2016, 2(8): 729 –734 (in Chinese with English abstract).

[10] Tian J. Reversible data embedding using a difference expansion. IEEE Trans. on Circuits and Systems for Video Technology, 2003, 13(8): 890–896. [doi: 10.1109/TCSVT.2003.815962]

[11] Ni ZC, Shi YQ, Ansari N, Su W. Reversible data hiding. IEEE Trans. on Circuits and Systems for Video Technology, 2006, 16(3): 354–362. [doi: 10.1109/TCSVT.2006.869964]

[12] Thodi DM, Rodríguez JJ. Expansion embedding techniques for reversible watermarking. IEEE Trans. on Image Processing, 2007, 16(3): 721–730. [doi: 10.1109/TIP.2006.891046]

[13] Sachnev V, Kim HJ, Nam J, Suresh S, Shi YQ. Reversible watermarking algorithm using sorting and prediction. IEEE Trans. on Circuits and Systems for Video Technology, 2009, 19(7): 989–999. [doi: 10.1109/TCSVT.2009.2020257]

[14] Coltuc D. Improved embedding for prediction-based reversible watermarking. IEEE Trans. on Information Forensics and Security, 2011, 6(3): 873–882. [doi: 10.1109/TIFS.2011.2145372]

[15] Hu RW, Xiang SJ. CNN prediction based reversible data hiding. IEEE Signal Processing Letters, 2021, 28: 464–468. [doi: 10.1109/LSP.2021.3059202]

[16] Luo JG, Han GQ, Yan W. Novel reversible data hiding based on difference expansion. Journal on Communications, 2016, 37(2): 53–62 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000-436x.2016030]

[17] Jia YJ, Yin ZX, Zhang XP, Luo YL. Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting. Signal Processing, 2019, 163: 238–246. [doi: 10.1016/j.sigpro.2019.05.020]

[18] Kim S, Qu XC, Sachnev V, Kim HJ. Skewed histogram shifting for reversible data hiding using a pair of extreme predictions. IEEE Trans. on Circuits and Systems for Video Technology, 2019, 29(11): 3236–3246. [doi: 10.1109/TCSVT.2018.2878932]

[19] Li XL, Li J, Li B, Yang B. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. Signal Processing, 2013, 93(1): 198–205. [doi: 10.1016/j.sigpro.2012.07.025]

[20] Peng F, Li XL, Yang B. Improved PVO-based reversible data hiding. Digital Signal Processing, 2014, 25: 255–265. [doi: 10.1016/j.dsp.2013.11.002]

[21] Wu HR, Li XL, Zhao Y, Ni RR. Improved PPVO-based high-fidelity reversible data hiding. Signal Processing, 2020, 167: 107264. [doi: 10.1016/j.sigpro.2019.107264]

[22] Fan GJ, Pan ZB, Gao ED, Gao XY, Zhang XR. Reversible data hiding method based on combining IPVO with bias-added rhombus predictor by multi-predictor mechanism. Signal Processing, 2021, 180: 107888. [doi: 10.1016/j.sigpro.2020.107888]

[23] Ou B, Li XL, Zhao Y, Ni RR, Shi YQ. Pairwise prediction-error expansion for efficient reversible data hiding. IEEE Trans. on Image Processing, 2013, 22(12): 5010–5021. [doi: 10.1109/TIP.2013.2281422]

[24] Ou B, Li XL, Zhang WM, Zhao Y. Improving pairwise PEE via hybrid-dimensional histogram generation and adaptive mapping selection. IEEE Trans. on Circuits and Systems for Video Technology, 2019, 29(7): 2176–2190. [doi: 10.1109/TCSVT.2018.2859792]

[25] Qin JQ, Huang FJ. Reversible data hiding based on multiple two-dimensional histograms modification. IEEE Signal Processing Letters, 2019, 26(6): 843–847. [doi: 10.1109/LSP.2019.2909080]

[26] Zhang T, Li XL, Qi WF, Guo ZM. Location-based PVO and adaptive pairwise modification for efficient reversible data hiding. IEEE Trans. on Information Forensics and Security, 2020, 15: 2306–2319. [doi: 10.1109/TIFS.2019.2963766]

[27] Lee SK, Suh YH, Ho YS. Reversible image authentication based on watermarking. In: Proc. of the 2006 IEEE Int'l Conf. on Multimedia and Expo. Toronto: IEEE, 2006. 1321–1324. [doi: 10.1109/ICME.2006.262782]

[28] Lo CC, Hu YC. A novel reversible image authentication scheme for digital images. Signal Processing, 2014, 98: 174–185. [doi: 10.1016/j.sigpro.2013.11.028]

[29] Yin ZX, Niu XJ, Zhou ZL, Tang J, Luo B. Improved reversible image authentication scheme. Cognitive Computation, 2016, 8(5): 890–899. [doi: 10.1007/s12559-016-9408-6]

[30] Hong W, Chen MJ, Chen TS. An efficient reversible image authentication method using improved PVO and LSB substitution techniques. Signal Processing: Image Communication, 2017, 58: 111–122. [doi: 10.1016/j.image.2017.07.001]

[31] Yao H, Wei HB, Qin C, Tang ZJ. A real-time reversible image authentication method using uniform embedding strategy. Journal of Real-time Image Processing, 2020, 17(1): 41–54. [10.1007/s11554-019-00904-8]

[32] Wang H, Huang FJ. Attack and improvement of an authentication scheme based on reversible data hiding. Journal of Cyber Security, 2022, 7(1): 56–65 (in Chinese with English abstract). [doi: 10.19363/J.cnki.cn10-1380/tn.2022.01.04]

[33] Howard PG, Kossentini F, Martins B, Forchhammer S, Rucklidge WJ. The emerging JBIG2 standard. IEEE Trans. on Circuits and Systems for Video Technology, 1998, 8(7): 838–848. [doi: 10.1109/76.735380]

附中文参考文献:

[8] 张新鹏, 殷赵霞. 多媒体信息隐藏技术. 自然杂志, 2017, 39(2): 87–95. [doi: 10.3969/j.issn.0253-9608.2017.02.002]

[9] 李晓龙. 图像可逆隐藏综述. 信息安全研究, 2016, 2(8): 729–734.

[16] 罗剑高, 韩国强, 沃焱. 新颖的差值扩展可逆数据隐藏算法. 通信学报, 2016, 37(2): 53–62. [doi: 10.11959/j.issn.1000-436x.2016030]

[32] 王泓, 黄方军. 基于可逆信息隐藏技术的认证方案的攻击与改进. 信息安全学报, 2022, 7(1): 56–65. [doi: 10.19363/J.cnki.cn10-1380/tn.2022.01.04]

钟亦友(1996－), 男, 硕士生, 主要研究领域为可逆信息隐藏, 数字水印.

黄方军(1973－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为 AI 安全, 多媒体内容安全.