# 采用独立可逆认证的图像篡改检测方法

王雨珂[1]，黄方军[1,2]
（1. 中山大学网络空间安全学院，518107，广东深圳；
2. 广东省信息安全技术重点实验室（中山大学），510006，广州）

**摘要**：针对现有的图像篡改检测方法存在篡改定位不精确以及未受篡改区域不能有效进行无损恢复的问题，提出采用独立可逆认证的图像篡改检测方法。首先，采用图像分块的独立可逆认证方法，结合可逆信息隐藏算法的嵌入原理，将现有认证方法中的认证单元由常规的 4×4 图像子块缩小为 4×3 图像子块，解决了认证单元偏大的问题，提高了篡改检测的定位精度；然后，根据子块特点，自适应地选取改进的像素值排序、局部直方图平移、差值扩展这 3 种可逆信息隐藏算法，保证每个子块至少可嵌入 1 bit 认证码的同时，将平均可嵌入认证码的数量提高到 3.6 bit，实现了图像子块独立认证与提升了篡改检测正确率；最后，考虑了 3 种可逆信息隐藏算法之间的连贯性与可逆性，以确保嵌入端的无效移位尽可能少，以及认证端认证信息的可靠提取。实验结果表明：对于未受篡改的图像区域，所提方法可以实现原始图像的无损恢复；对于剪贴攻击、常数攻击、拼贴攻击和随机篡改等常见的篡改攻击，所提方法的初始正确检测率较现有最优方法平均可提升 5.82%～22.3%，细化处理后的正确检测率可接近 100%。

**关键词**：图像篡改检测；可逆信息隐藏；像素值排序；独立可逆认证
**中图分类号**：TP391 **文献标志码**：A
**DOI**：10.7652/xjtuxb202412014 **文章编号**：0253-987X(2024)12-0141-12

# Image Tampering Detection Method Utilizing Independent Reversible Authentication

WANG Yuke[1], HUANG Fangjun [1,2]
(1. School of Cyber Science and Technology, Shenzhen Campus of Sun Yat-sen University, Shenzhen, Guangdong 518107, China; 2. Guangdong Provincial Key Laboratory of Information Security Technology (Sun Yat-sen University), Guangzhou 510006, China)

**Abstract**: To address the imprecise localization of tampering and ineffective recovery of untampered areas in current image tampering detection methods, a new independent reversible authentication-based approach for detecting image tampering is introduced. Initially, leveraging an image block-based independent reversible authentication technique and incorporating the embedding principles of reversible information hiding algorithms, the authentication units in existing methods are reduced from the conventional 4×4 image sub-blocks to 4×3 image sub-blocks. This modification tackles the issue of oversized authentication units, thereby enhancing the localization accuracy of tampering detection. Subsequently, based on the characteristics of

收稿日期：2024-05-29。 作者简介：王雨珂(2000—)，女，硕士生；黄方军(通信作者)，男，教授，博士生导师。 基金项目：国家自然科学基金资助项目(62072481，U2336208，62472454)。
网络出版时间：2024-09-03 网络出版地址：https://link.cnki.net/urlid/61.1069.t.20240902.1633.002

sub-blocks, three enhanced reversible information hiding algorithms, specifically improved pixel value sorting, local histogram shifting, and difference expansion, are adaptively selected to ensure that each sub-block is embedded with at least 1 bit authentication code. This process increases the average embeddable authentication code quantity to 3.6 bits, facilitating independent authentication of image sub-blocks and improving tampering detection accuracy. Finally, the coherence and reversibility of the three reversible information hiding algorithms are considered to minimize irrelevant shifts during the embedding stage and ensure the reliable extraction of authentication information during the authentication stage. The experimental results demonstrate that the proposed method enables lossless recovery of the original image in untampered areas. When confronted with common tampering attacks such as cut-and-paste, constant manipulation, collage alterations, and random tampering, the initial accuracy rate of the proposed method can be improved by an average of 5.82% to 22.3% compared to existing optimal techniques, with the refined accuracy rate potentially approaching 100%.

**Keywords**: image tampering detection; reversible data hiding; pixel-value-ordering; independent reversible authentication

随着图像处理手段的不断丰富和图像编辑软件的广泛流行,篡改图像内容的难度和成本都急剧降低。为了保护数字图像的完整性和真实性,人们提出了图像认证的概念[1]。一般来说,图像认证方案可分为 2 大类:基于数字签名的方案[2-4]和基于脆弱水印的方案[5-7]。基于数字签名的方法主要通过数字摘要技术和非对称加密机制实现,生成的数字签名需要存储在受信任的第三方机构中,当对图像进行验证时,需要将第三方中提取出的数字签名与图像生成的数字签名进行比对,以检测出被篡改的区域。但是,当第三方机构不再可信时,数字签名的真实性和安全性也将无法得到保障。与基于数字签名的方法不同,基于脆弱水印的方法直接将水印信息嵌入到要保护的图像中去,对图像内容的篡改十分敏感,但嵌入会对载体图像造成不可恢复的破坏,这在医学图像、军事通信等领域中是无法接受的。

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,因此也被称为无损信息隐藏。该技术在近年来得到了学者们的广泛研究,Tian[8]最早提出基于差值扩展的方法,通过扩展 2 个相邻像素之间的差值来嵌入 1 bit 信息,这类方法一般嵌入容量较大,但是引入的失真也较大。相对而言,基于直方图平移的方法[9-13]引入的失真更小,但是其嵌入容量受到峰值点的限制。结合差值扩展和直方图平移的优点,基于预测误差扩展的方法[14-18]考虑更大邻域内像素的相关性,提高了预测的准确率,可以得到更加尖锐的预测误差直方图,从而可以在完成信息嵌入的同时尽可能地减小所引入的失真。基于像素值排序的方法[19-21]本质上也属于预测误差扩展的方法,具体嵌入过程中分别用块内次大、次小像素预测最大、最小像素,从而得到峰值更高的预测误差直方图。

图像可逆认证方法可以看作是可逆信息隐藏和脆弱水印技术的结合,嵌入端利用可逆信息隐藏算法在图像中嵌入认证信息,认证端则通过提取认证信息来验证图像的完整性。如果图像被篡改,则可以定位具体的篡改区域;如果未被篡改,则可以完整地恢复图像。现有的可逆认证方案大致可以分为独立可逆认证方案和非独立可逆认证方案,两者的主要区别在于当前待认证子块是否嵌入与自身相关的认证信息,以及其是否需要借助其他子块来完成可逆认证。2014 年,Lo 等[22]首次提出非独立的可逆认证方法,将图像划分为 4×4 的子块,在每个子块中利用基于预测的直方图平移(PBHS)算法[13]复嵌入 1 bit 认证码,但由于 PBHS 算法的容量有限,生成的水印图像视觉质量一般。Nguyen 等[23]和 Yin 等[24]在 2016 年分别提出了改进算法。Nguyen 等[23]通过自适应的预测误差扩展(PEE)算法在不同纹理复杂度的子块中嵌入不同数量认证码。Yin 等[24]则利用希尔伯特曲线扫描整幅图像,然后对像素进行分组并在每个分组中通过改进的像素值排序(IPVO)算法嵌入认证码。2017 年,Hong 等[25]提出了一种将 IPVO 和最低有效位(LSB)替换相结合的可逆认证方案,在不可嵌入块中利用 LSB 替换来嵌入 1 bit 认证码,在可嵌入块中利用 IPVO 算法嵌

入与该块对应的认证码以及不可嵌入块中替换下来的 LSB 位，从而实现了对不可嵌入块和可嵌入块的认证。Yao 等[26]和王泓等[27]分别对 Hong 等[25]的方法进行了改进：Yao 等[26]根据不同的图像内容将图像以 2×2 的倍数进行子块划分，实现了最优的检测块尺寸；王泓等[27]通过置乱操作来避免攻击方获知嵌入块的划分方案，有效提升了算法的安全性。上述非独立可逆认证方案中，文献[22-24]存在大量嵌入容量为零的不可嵌入块，因此不能对所有子块进行有效保护；而文献[25-27]需要将替换下来的 LSB 位全部嵌入到可嵌入块保存，这意味着当可嵌入块遭到篡改攻击时，其所对应的不可嵌入块即使没有被攻击，也会由于 LSB 位提取失败而无法实现准确恢复。

为了解决非独立认证方案带来的问题，Nguyen 等[28]在 2016 年提出了最早的独立认证方法：对每个 8×8 大小的子块进行 2 次离散小波变换(DWT)，然后利用预测误差扩展方法在低频子带中嵌入 3 bit 认证码。由于认证码嵌入在 DWT 系数中，因此通常不存在不可嵌入块，但该方法存在认证单元过大、篡改定位能力较弱等问题。随后在 2022 年，钟等[29]提出一种基于差值扩展和预测误差直方图平移的独立认证方法，该方法的嵌入容量大且边信息少，每个 4×4 子块可以嵌入 2 到 16 bit 的认证码，并且通过在篡改检测的细化处理过程中增加膨胀和腐蚀操作，进一步提高了算法的篡改检测正确率。但是，该方法产生的水印图像视觉质量远低于其他方法，这主要是因为使用差值扩展来嵌入信息会使得预测误差翻倍，从而引入大量失真。

对比目前主流的 8 种可逆认证方法在不同维度的特点，文献[25-27, 29]都采用了多种算法的组合来实现所有子块的认证，但是由于没有考虑算法之间的连贯性，需要通过无效移位来避免认证端出现混淆的情况，从而导致不必要的失真。此外，文献[28-29]虽然实现了独立认证，但是文献[28]存在认证单元较大、嵌入容量较低的问题，而文献[29]生成的水印图像视觉质量较差。

针对以上问题，本文提出了一种新的针对图像子块的独立可逆认证方法，以 4×3 大小的子块为认证单元，根据子块自身的特点，分别选取 IPVO、局部直方图平移(LHS)和差值扩展(DE)这 3 种可逆信息隐藏算法来完成每个子块的独立认证。本文算法具体实施过程中，通过 2 次不同嵌入单元划分模式下的 IPVO 嵌入操作，可以在增加嵌入容量的同时产生失真补偿效果，有效提高算法的认证性能以及嵌入认证信息后水印图像的视觉质量。同时，本文还考虑了 3 种可逆信息隐藏算法之间的连贯性和可逆性，保证其相互之间不会产生混淆，从而尽可能地减小可逆嵌入过程中引入的无效移位，有效避免嵌入认证信息所导致的图像视觉质量降低等问题。

## 1　基于图像分块的独立可逆认证方法

本文方法的目标是检测图像是否被篡改。如果检测到图像被篡改，则可以进一步对篡改区域进行定位；如果检测到图像未被篡改，则可以提取认证信息并恢复原始图像。本文先对认证码的嵌入、提取与图像恢复进行介绍，然后对失真补偿现象进行说明，最后对子块间的互不混淆性质进行数学证明。

### 1.1　认证码的嵌入

首先，对原始图像进行预处理，然后将图像分块并为每个子块生成认证码；接下来，对每个子块进行分类并选择不同的可逆信息隐藏算法来嵌入认证码，最后合并子块，得到嵌入认证信息的水印图像。值得注意的是，在预处理和认证码嵌入阶段可能会产生辅助信息，这些信息都将通过第 2 次 IPVO($2^{nd}$ IPVO)嵌入到图像中。图 1 所示为认证码的嵌入流程，下面将按步骤对认证码的嵌入过程进行详细介绍。

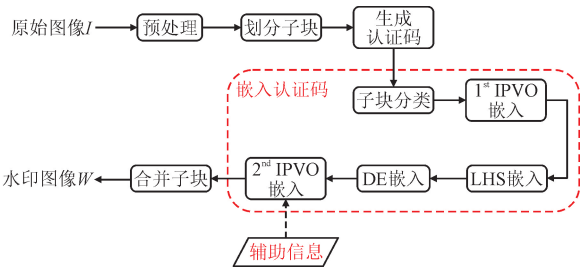


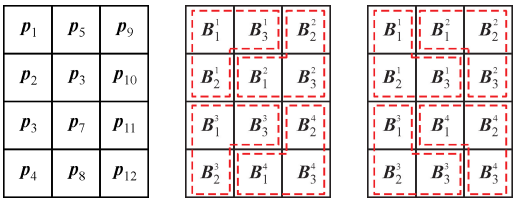

图 1　认证码的嵌入流程

Fig. 1　The embedding process of authentication codes

**步骤 1**　预处理。为了避免待认证图像在嵌入认证信息时出现部分像素值上溢或下溢的问题，在预处理过程中通过创建一个与原始图像尺寸相同的位置图(location map，LM)来记录可能产生溢出的像素位置。首先顺次扫描原始图像，当扫描到值为 0、1、2、253、254、255 的像素时，分别修改为 3、4、5、250、251、252，并在位置图对应的位置记录 1；当扫描到原始值为 3、4、5、250、251、252 的像素时，保持不变，并在位置图对应的位置记录 0；最后对位置图进行无损压缩，并将压缩后的位置图作为辅助信息的一部分嵌入到图像中。

**步骤 2**　图像分块与认证码生成。将 $M\times N$ 的原始图

像 $I$ 划分为不重叠的 4×3 子块，具体子块如图 2(a)所示。利用哈希函数($M_{\text{block}}$，$M_{\text{row}}$，$M_{\text{col}}$，$M_{\text{key}}$)为每个子块生成一个 16 bit 的认证码 $\beta=\{b_1,b_2,\cdots,b_{16}\}$，其中 $M_{\text{block}}$ 为当前块中所包含的全部像素信息，$M_{\text{row}}$ 和 $M_{\text{col}}$ 分别为当前块所在行和列，$M_{\text{key}}$ 为发送方和接收方事先约定好的密钥。




(a)未划分子块 (b)第 1 种模式 (c)第 2 种模式

图 2 2 种子块划分模式

Fig. 2 Two sub-block partitioning modes

**步骤 3** 1st IPVO 嵌入。为了使 4×3 子块在 IPVO 算法中可以嵌入更多的认证信息，按照图 2(b)所示的方式将其进一步划分为 4 个嵌入单元，每个嵌入单元包含 3 个像素。对于任意一个包含 3 个像素的嵌入单元 $B^n=\{B_1^n,B_2^n,B_3^n\}(1\leqslant n\leqslant 4)$，其中 $n$ 为嵌入单元编号，为了后续说明方便，这里忽略嵌入单元编号，将其简化记为 $B=\{B_1,B_2,B_3\}$。然后将像素按照升序排列得到 $B=\{B_{\sigma_1},B_{\sigma_2},B_{\sigma_3}\}$，其中下标 $\sigma_i$ 表示 $B_{\sigma_i}$ 在初始序列$\{B_1,B_2,B_3\}$ 中的位置，排序方式为稳定排序，即当 $B_i=B_j$ 且 $i<j$ 时，有 $\sigma_i<\sigma_j$。

预测误差 $d_{\max}$ 和 $d_{\min}$ 计算如下

$$\begin{cases} d_{\max}=B_u-B_v \\ u=\min(\sigma_3,\sigma_2) \\ v=\max(\sigma_3,\sigma_2) \end{cases} \tag{1}$$

$$\begin{cases} d_{\min}=B_s-B_t \\ s=\min(\sigma_1,\sigma_2) \\ t=\max(\sigma_1,\sigma_2) \end{cases} \tag{2}$$

分别构造关于 $d_{\max}$ 和 $d_{\min}$ 的预测误差直方图，根据下式嵌入认证信息

$$B'_{\sigma_3}=\begin{cases} B_{\sigma_3}+b, & d_{\max}\in\{0,1\} \\ B_{\sigma_3}+1, & d_{\max}\in(-\infty,0)\cup(1,+\infty) \end{cases} \tag{3}$$

$$B'_{\sigma_1}=\begin{cases} B_{\sigma_1}-b, & d_{\min}\in\{0,1\} \\ B_{\sigma_1}-1, & d_{\min}\in(-\infty,0)\cup(1,+\infty) \end{cases} \tag{4}$$

式中：$b\in\{0,1\}$ 表示要嵌入的 1 bit 认证码；$B'_{\sigma_3}$、$B'_{\sigma_1}$ 分别为 $B_{\sigma_3}$、$B_{\sigma_1}$ 嵌入后的值。

将 4 个嵌入单元的嵌入容量相加，得到整个子块在 1st IPVO 嵌入中的嵌入容量 $V_1(0\leqslant V_1\leqslant 8)$。

**步骤 4** 子块分类。任意 4×3 子块在完成 1st IPVO 嵌入后，记录下 1st IPVO 嵌入容量 $V_1$，然后将该子块按照图 2(c)所示的第 2 种方式重新进行嵌入单元划分，并再次应用 IPVO 算法计算在此划分模式下的嵌入容量，将其记为 $V_2$。值得注意的是，此时只是应用 IPVO 算法计算嵌入容量而不进行信息嵌入，容量计算结果将用于子块分类，具体分类流程如图 3 所示。如果该 4×3 子块在 2 种嵌入单元划分模式下运用 IPVO 算法计算得到的嵌入容量之和不等于 0，则将其分类为 IPVO 块；如果该子块在 2 种嵌入单元划分模式下运用 IPVO 算法计算得到的嵌入容量之和等于 0，则计算该子块在 LHS 算法下的嵌入容量 $V_{\text{LHS}}$；如果 $V_{\text{LHS}}$ 不等于 0，则将其分类为 LHS 块，反之为 DE 块。

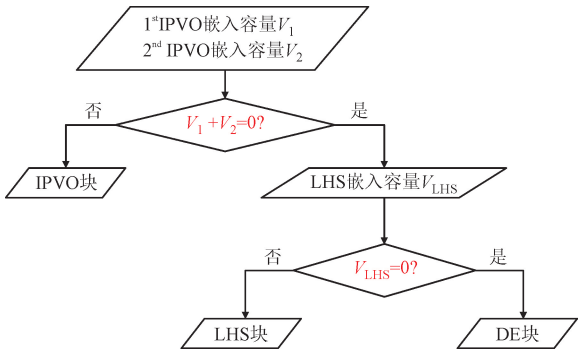



图 3 子块分类流程

Fig. 3 The classification process of sub-blocks

**步骤 5** LHS 嵌入。对于非 IPVO 块，优先考虑用 LHS 方法来嵌入认证码，没有直接选择差值扩展的原因是其通常引入的失真较大且容易产生较多的边信息。对于任意一个如图 2(a)所示的非 IPVO 块，首先得到其局部直方图，令其峰值点对应的像素值为 $P$，峰值点左邻点对应的像素值为 $P_{\text{l}}=P-1$，峰值点右邻点对应的像素值为 $P_{\text{r}}=P+1$，然后选择峰值点作为参考点，利用峰值点的左、右邻点来嵌入信息。特别地，如果子块中的像素值均匀分布(即不同像素值出现的次数相同)，则选取该子块内像素值中位数所对应的点为峰值点；如果直方图中出现多个峰值点，则选取对应像素值最大的点为峰值点。

LHS 嵌入的具体操作如下：如果峰值点 $P$ 的左邻点 $P_{\text{l}}$ 或右邻点 $P_{\text{r}}$ 存在，则当前块为 LHS 块，在嵌入过程中保持峰值点 $P$ 不变，将[1，$P-2$] 和[$P+2$，254] 范围内的像素按下式进行平移

$$p_i^{\mathrm{LHS}}=\begin{cases}p_i^{\mathrm{IPVO}}+1, & P+2\leqslant p_i^{\mathrm{IPVO}}\leqslant 254\\ p_i^{\mathrm{IPVO}}-1, & 1\leqslant p_i^{\mathrm{IPVO}}\leqslant P-2\end{cases}\tag{5}$$

式中：$p_i^{\mathrm{IPVO}}(1\leqslant i\leqslant 12)$ 表示经过 1st IPVO 嵌入后的像素。然后，在左邻点 $P_{\mathrm{l}}$ 或右邻点 $P_{\mathrm{r}}$ 上通过下式进行认证信息嵌入

$$p_i^{\mathrm{LHS}}=\begin{cases}p_i^{\mathrm{IPVO}}+b, & p_i^{\mathrm{IPVO}}=P+1\\ p_i^{\mathrm{IPVO}}-b, & p_i^{\mathrm{IPVO}}=P-1\\ p_i^{\mathrm{IPVO}}, & p_i^{\mathrm{IPVO}}=P\end{cases}\tag{6}$$

式中：$p_i^{\mathrm{LHS}}(1\leqslant i\leqslant 12)$ 表示经过 LHS 嵌入后的像素；$P$ 为峰值点；$b\in\{0,1\}$ 表示要嵌入的 1 bit 认证码。

如果峰值点 $P$ 的左邻点 $P_{\mathrm{l}}$ 和右邻点 $P_{\mathrm{r}}$ 都不存在，则当前块为 DE 块，不进行任何数据嵌入，但需要按式(5)进行平移。

**步骤 6**　DE 嵌入。对于 LHS 嵌入后仍然不能实现独立认证的非 IPVO 块，即无法嵌入自身认证信息的子块，采取 DE 方式进行认证，具体操作如下。对于任意一个 DE 块，按照光栅扫描的顺序得到初始序列 $\{p_1^{\mathrm{LHS}},p_2^{\mathrm{LHS}},\cdots,p_n^{\mathrm{LHS}}\}$，再将初始序列按升序排列得到 $\{p_{\sigma_1}^{\mathrm{LHS}},p_{\sigma_2}^{\mathrm{LHS}},\cdots,p_{\sigma_n}^{\mathrm{LHS}}\}$，然后选择在最大值 $p_{\sigma_n}^{\mathrm{LHS}}$ 或最小值 $p_{\sigma_1}^{\mathrm{LHS}}$ 中嵌入 1 bit 认证码。其中，$n$ 为子块中像素的个数($n$ 取 12)，下标 $\sigma_i(0\leqslant i\leqslant n)$ 为像素值在初始序列中对应的位置，排序方法为稳定排序。

为了尽可能地避免差值扩展可能导致的像素值溢出，本文采用一种根据子块内容自适应选取差值扩展方向的新方案。在上一步的 LHS 嵌入中可以得到每个子块局部直方图的峰值点，而峰值点在一定程度上可以反映该子块内像素的分布范围。因此，为了尽可能避免差值扩展后得到的像素值大于 255 或小于 0，设定一个全局阈值 $\theta$，在以下 2 种情况时向上扩展：① 峰值点 $P$ 小于阈值 $\theta$ 且峰值点 $P$ 不等于最大值 $p_{\sigma_n}^{\mathrm{LHS}}$，即 $P<\theta$ 且 $P\neq p_{\sigma_n}^{\mathrm{LHS}}$；② 峰值点 $P$ 大于等于阈值 $\theta$ 且峰值点 $P$ 等于最小值 $p_{\sigma_1}^{\mathrm{LHS}}$，即 $P\geqslant\theta$ 且 $P=p_{\sigma_1}^{\mathrm{LHS}}$。具体表示如下

$$\begin{cases}h=p_{\sigma_n}^{\mathrm{LHS}}-p_{\sigma_{n-1}}^{\mathrm{LHS}}\\ H=2h+b\\ p_{\sigma_n}^{\mathrm{DE}}=p_{\sigma_{n-1}}^{\mathrm{LHS}}+H\end{cases}\tag{7}$$

式中：$h$ 表示 $p_{\sigma_n}^{\mathrm{LHS}}$ 和 $p_{\sigma_{n-1}}^{\mathrm{LHS}}$ 或 $p_{\sigma_2}^{\mathrm{LHS}}$ 和 $p_{\sigma_1}^{\mathrm{LHS}}$ 的差值；$H$ 为 $h$ 扩展后的值；$b\in\{0,1\}$ 表示要嵌入的 1 bit 认证码；$p_{\sigma_i}^{\mathrm{DE}}$ 为 $p_{\sigma_i}^{\mathrm{LHS}}$ 扩展后的值。

类似地，向下扩展同样有 2 种情况：① 峰值点 $P$ 大于等于阈值 $\theta$ 且峰值点 $P$ 不等于最小值 $p_{\sigma_1}^{\mathrm{LHS}}$，即 $P\geqslant\theta$ 且 $P\neq p_{\sigma_1}^{\mathrm{LHS}}$；② 峰值点 $P$ 小于阈值 $\theta$ 且峰值点 $P$ 等于最大值 $p_{\sigma_n}^{\mathrm{LHS}}$，即 $P<\theta$ 且 $P=p_{\sigma_n}^{\mathrm{LHS}}$。具体表示如下

$$\begin{cases}h=p_{\sigma_2}^{\mathrm{LHS}}-p_{\sigma_1}^{\mathrm{LHS}}\\ H=2h+b\\ p_{\sigma_1}^{\mathrm{DE}}=p_{\sigma_2}^{\mathrm{LHS}}-H\end{cases}\tag{8}$$

本方案中全局阈值 $\theta$ 采用穷举搜索的方式得到，即针对每幅图像都在 0 到 255 范围内以 10 为步长进行搜索，当水印图像的峰值信噪比达到最大时即为最佳阈值。

值得注意的是，上述向上扩展和向下扩展中的第 2 种情况实际上并不符合避免像素值溢出的要求。但是，为了保证上一步 LHS 算法的可逆性，子块的峰值点在 DE 嵌入前后应当保持不变。因此，对于差值扩展嵌入点与峰值点恰好重合的特殊情况，需要选择另一个最值进行反向扩展。在实际嵌入过程中，这样的特殊情况出现概率较低，因此并不会产生大量的溢出像素。

如果得到的像素值 $p_{\sigma_1}^{\mathrm{LHS}}$ 或 $p_{\sigma_n}^{\mathrm{LHS}}$ 溢出，则将溢出像素的相关信息保存到序列 $S$ 中，然后将小于 0 的 $p_{\sigma_1}^{\mathrm{LHS}}$ 修改为 0，将大于 255 的 $p_{\sigma_n}^{\mathrm{LHS}}$ 修改为 255。阈值 $\theta$、溢出像素数 $K$ 和溢出像素相关信息 $S$ 将作为辅助信息的一部分通过 2nd IPVO 嵌入到 IPVO 块中，这样认证端就可以对产生溢出的像素进行恢复。

**步骤 7**　2nd IPVO 嵌入。与 1st IPVO 嵌入类似，首先将每个子块按照图 2(c)所示的方式进一步划分为 4 个嵌入单元，然后对于任意一个嵌入单元 $B''=\{B''_1,B''_2,B''_3\}$，将其中的像素按照升序的稳定排列后得到 $\{B''_{\sigma_1},B''_{\sigma_2},B''_{\sigma_3}\}$，其中下标 $\sigma_i$ 表示 $B''_{\sigma_i}$ 在初始序列 $\{B''_1,B''_2,B''_3\}$ 中的位置。

预测误差 $d_{\max}$ 和 $d_{\min}$ 计算如下

$$\begin{cases}d_{\max}=B''_u-B''_v\\ u=\min(\sigma_3,\sigma_2)\\ v=\max(\sigma_3,\sigma_2)\end{cases}\tag{9}$$

$$\begin{cases}d_{\min}=B''_s-B''_t\\ s=\min(\sigma_1,\sigma_2)\\ t=\max(\sigma_1,\sigma_2)\end{cases}\tag{10}$$

分别构造关于 $d_{\max}$ 和 $d_{\min}$ 的预测误差直方图，根据下式嵌入认证信息

$$B'''_{\sigma_3}=\begin{cases}B''_{\sigma_3}+b, & d_{\max}\in\{0,1\}\\ B''_{\sigma_3}+1, & d_{\max}\in(-\infty,0)\cup(1,+\infty)\end{cases}\tag{11}$$

$$B'''_{\sigma_1}=\begin{cases}B''_{\sigma_1}-b, & d_{\min}\in\{0,1\}\\ B''_{\sigma_1}-1, & d_{\min}\in(-\infty,0)\cup(1,+\infty)\end{cases}\tag{12}$$

式中：$b\in\{0,1\}$ 表示顺次从嵌入序列 $L$ 中取出的 1 bit 信息；$B'''_{\sigma_3}$、$B'''_{\sigma_1}$ 分别为 $B'_{\sigma_3}$、$B''_{\sigma_1}$ 嵌入后的值。注意，此时的嵌入序列 $L$ 不仅包括当前 IPVO 块自身相关的认证信息，还包括所有辅助信息。

**步骤 8** 合并子块与生成水印图像。当所有子块都已完成认证信息的嵌入，每个子块嵌入的认证码数量最少为 1，最多为 16。在实际嵌入过程中，并非所有子块都能嵌入所生成的全部认证码，但每个子块应根据自身嵌入容量嵌入尽可能多的认证码以提高认证精度。最后，将所有子块合并，得到嵌入认证码后的水印图像 $W$。

## 1.2 认证码的提取与图像恢复

图 4 所示为认证码的提取与图像恢复流程。首先对水印图像进行分块，然后通过不同的可逆信息隐藏算法对每个子块提取认证码并恢复。接下来进行篡改检测，根据第 1.1 节的哈希函数重新生成各个子块的认证码，将新生成的认证码与提取的认证码进行比对。如果不一致，认为该块已被篡改，需要进行检测和定位以得到篡改检测标记；否则，认为该块未被篡改，合并所有子块，然后经过逆预处理即可得到原始图像。特别地，2nd IPVO 提取出的信息除了认证码还包括辅助信息，这些辅助信息将用于后续的认证码提取和逆预处理操作。

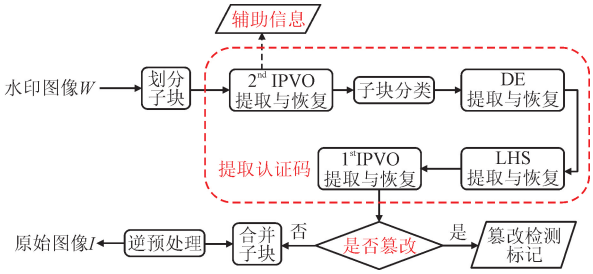



图 4 认证码的提取与图像恢复流程

Fig. 4 The process of authentication code extraction and image recovery

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。

**步骤 1** 图像分块和 2nd IPVO 提取与恢复。将大小为 $M\times N$ 的水印图像 $W$ 划分为互不重叠的 $4\times3$ 子块，将每个子块按照图 2(c) 所示的第二种方式划分成 4 个嵌入单元，将任意一个嵌入单元 $B'''=\{B'''_1,B'''_2,B'''_3\}$ 按照升序排列得到 $\{B'''_{\sigma_1},B'''_{\sigma_2},B'''_{\sigma_3}\}$，然后用嵌入时同样的方法计算预测误差 $d_{\max}$ 和 $d_{\min}$。

根据下式提取认证码 $b\in\{0,1\}$

$$b=\begin{cases}0, & d_{\max}\in\{0,1\}\\ 1, & d_{\max}\in\{-1,2\}\end{cases}\tag{13}$$

$$b=\begin{cases}0, & d_{\min}\in\{0,1\}\\ 1, & d_{\min}\in\{-1,2\}\end{cases}\tag{14}$$

由此得到 2nd IPVO 的提取序列 $L$，包括子块自身的认证信息与所有辅助信息。

根据下式恢复图像

$$B''_{\sigma_3}=\begin{cases}B'''_{\sigma_3}, & d_{\max}\in\{0,1\}\\ B'''_{\sigma_3}-1, & d_{\max}\in(-\infty,0)\cup(1,\infty)\end{cases}\tag{15}$$

$$B''_{\sigma_1}=\begin{cases}B'''_{\sigma_1}, & d_{\min}\in\{0,1\}\\ B'''_{\sigma_1}+1, & d_{\min}\in(-\infty,0)\cup(1,\infty)\end{cases}\tag{16}$$

式中：$B''_{\sigma_3}$、$B''_{\sigma_1}$ 分别为 $B'''_{\sigma_3}$、$B'''_{\sigma_1}$ 恢复后的值。

**步骤 2** 子块分类。当所有子块均已完成 2nd IPVO 提取与恢复，记录下 2nd IPVO 嵌入容量 $V_2$，然后将这些子块按照图 2(b)所示的第一种方式进行嵌入单元划分，计算 1st IPVO 嵌入容量 $V_1$。与嵌入过程中的分类原理一致，按照图 3 所示的流程对所有子块进行分类，得到与嵌入端一致的分类情况。

**步骤 3** DE 提取与恢复。在 DE 块进行提取与恢复前，需要先从辅助信息中分离出溢出像素相关信息 S，然后对 S 中记录的像素进行恢复。接下来，顺次扫描 $DE$ 块中的像素得到初始序列 $\{p_1^{\mathrm{DE}},p_2^{\mathrm{DE}},\cdots,p_n^{\mathrm{DE}}\}$，再将初始序列按升序的稳定排列得到 $\{p_{\sigma_1}^{\mathrm{DE}},p_{\sigma_2}^{\mathrm{DE}},\cdots,p_{\sigma_n}^{\mathrm{DE}}\}$，其中 $n$ 的取值为 12，下标 $\sigma_i(0\leqslant i\leqslant n)$ 表示像素值在初始序列中对应的位置。按照第 1.1 节中 LHS 嵌入时相同的规则寻找 DE 块的峰值点 $P$，如果峰值点 $P$ 小于阈值 $\theta$ 且 $P$ 不等于最大值 $p_{\sigma_n}^{\mathrm{DE}}$，或峰值点 $P$ 大于等于阈值 $\theta$ 且 $P$ 等于最小值 $p_{\sigma_1}^{\mathrm{DE}}$，则根据下式向上扩展进行还原

$$\begin{cases}H=p_{\sigma_n}^{\mathrm{DE}}-p_{\sigma_{n-1}}^{\mathrm{DE}}\\ h=\left\lfloor\dfrac{H}{2}\right\rfloor\\ p_{\sigma_n}^{\mathrm{HS}}=p_{\sigma_{n-1}}^{\mathrm{DE}}+h\end{cases}\tag{17}$$

如果峰值点 $P$ 大于等于阈值 $\theta$ 且 $P$ 不等于最小值 $p_{\sigma_1}^{\mathrm{DE}}$，或峰值点 $P$ 小于阈值 $\theta$ 且 $P$ 等于最大值 $p_{\sigma_n}^{\mathrm{DE}}$，则根据下式向下扩展进行还原

$$\begin{cases}H=p_{\sigma_2}^{\mathrm{DE}}-p_{\sigma_1}^{\mathrm{DE}}\\ h=\left\lfloor\dfrac{H}{2}\right\rfloor\\ p_{\sigma_1}^{\mathrm{HS}}=p_{\sigma_2}^{\mathrm{DE}}-h\end{cases}\tag{18}$$

式中：$H$ 表示 $p_{\sigma_n}^{\mathrm{DE}}$ 和 $p_{\sigma_{n-1}}^{\mathrm{DE}}$ 或 $p_{\sigma_2}^{\mathrm{DE}}$ 和 $p_{\sigma_1}^{\mathrm{DE}}$ 的差值；$\lfloor.\rfloor$ 表示向下取整；$h$ 为 $H$ 还原后的值；$p_{\sigma_i}^{\mathrm{HS}}$ 为 $p_{\sigma_i}^{\mathrm{DE}}$ 还原后

的值。

最后，根据下式可得到提取出的认证信息 $b$

$$b = H - 2h \tag{19}$$

与嵌入过程一致的是，如果遇到差值扩展嵌入点与局部直方图峰值点重合的情况，则需选择另一个最值进行反向还原，这可以保证 DE 算法的可逆性。

**步骤 4**　LHS 提取与恢复。对于一个非 IPVO 块，得到其局部直方图并按照 LHS 嵌入时相同的规则寻找峰值点 $P$。由于峰值点在整个认证码嵌入的过程中都保持不变，因此认证端可以找到与嵌入端相同的峰值点。根据下式从 LHS 块中提取认证信息

$$b = \begin{cases} 1, & p_i^{\mathrm{LHS}} \in \{P-2, P+2\} \\ 0, & p_i^{\mathrm{LHS}} \in \{P-1, P+1\} \end{cases} \tag{20}$$

式中，$b$ 为提取出的认证信息比特。

根据下式恢复非 IPVO 块（包括 LHS 块和 DE 块）

$$p_i^{\mathrm{IPVO}} = \begin{cases} p_i^{\mathrm{LHS}} - 1, & P+2 \leqslant p_i^{\mathrm{LHS}} \leqslant 254 \\ p_i^{\mathrm{LHS}} + 1, & 1 \leqslant p_i^{\mathrm{LHS}} \leqslant P-2 \\ p_i^{\mathrm{LHS}}, & P-1 \leqslant p_i^{\mathrm{LHS}} \leqslant P+1 \end{cases} \tag{21}$$

式中：$p_i^{\mathrm{LHS}}(1 \leqslant i \leqslant 12)$ 表示经过 DE 恢复后的像素值；$p_i^{\mathrm{IPVO}}(1 \leqslant i \leqslant 12)$ 表示经过 LHS 恢复后的像素值。

**步骤 5**　1st IPVO 提取与恢复。将所有子块按照图 2(b)所示的第一种方式划分成 4 个嵌入单元，将任意一个嵌入单元 $B' = \{B'_1, B'_2, B'_2\}$ 按照升序的稳定排列后得到 $\{B'_{\sigma_1}, B'_{\sigma_2}, B'_{\sigma_3}\}$，然后用嵌入时同样的方法分别计算预测误差 $d_{\max}$ 和 $d_{\min}$。

根据下式提取当前 IPVO 块中嵌入的认证码 $b \in \{0,1\}$，得到 1st IPVO 嵌入的认证信息序列，将其与 2nd IPVO 提取得到的认证信息进行拼接，可得到当前 IPVO 块的所有认证信息。

$$b = \begin{cases} 0, & d_{\max} \in \{0,1\} \\ 1, & d_{\max} \in \{-1,2\} \end{cases} \tag{22}$$

$$b = \begin{cases} 0, & d_{\min} \in \{0,1\} \\ 1, & d_{\min} \in \{-1,2\} \end{cases} \tag{23}$$

最后，所有子块根据下式恢复图像

$$B_{\sigma_3} = \begin{cases} B'_{\sigma_3}, & d_{\max} \in \{0,1\} \\ B'_{\sigma_3} - 1, & d_{\max} \in (-\infty, 0) \cup (1, \infty) \end{cases} \tag{24}$$

$$B_{\sigma_1} = \begin{cases} B'_{\sigma_1}, & d_{\min} \in \{0,1\} \\ B'_{\sigma_1} - 1, & d_{\min} \in (-\infty, 0) \cup (1, \infty) \end{cases} \tag{25}$$

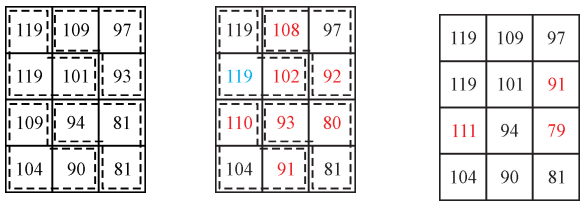
式中，$B_{\sigma_3}$、$B_{\sigma_1}$ 分别为 $B'_{\sigma_3}$、$B'_{\sigma_1}$ 恢复后的值。

**步骤 6**　合并子块与恢复原始图像。此时，所有子块均已完成认证码的提取与图像恢复，接下来根据第 1.1 节的哈希函数重新生成各个子块的认证码，然后将新生成的认证码与提取的认证码比对，如果一致则认为该块未被篡改，否则认为该块已被篡改。如果图像没有被篡改，则合并所有子块并进行逆预处理操作，最终即可得到原始图像 $I$；如果图像被篡改，则利用 Nguyen 等[23]提出的篡改检测方法和钟等[29]提出的细化处理方法对篡改子块进行定位，从而得到精确的篡改检测标记。

### 1.3　失真补偿

在两种不同的嵌入单元划分模式下进行 IPVO 嵌入有以下两点优势：一方面，两次嵌入可以扩大嵌入区域和增加嵌入容量；另一方面，不同的嵌入单元划分模式可以在局部子块内产生失真补偿像素从而提升水印图像的视觉质量。同时，失真补偿不会影响整体认证算法的可逆性，包括载体图像的无损恢复和认证信息的准确提取，

下面通过一个具体的例子来说明失真补偿。图 5 所示为 Lena 图中某一 IPVO 块依次经过 1st IPVO 嵌入和 2nd IPVO 嵌入时像素值发生的变化，其中蓝色像素为用于扩展嵌入的像素，红色像素为产生失真（即与原始值相比发生变化）的像素。图 5(a)所示为子块的原始值，根据图 2(b)所示的第 1 种模式计算出该子块的 1st IPVO 嵌入容量为 2 bit，分别对蓝色像素 119 和 81 进行扩展以嵌入“0”和“1”，其余像素平移。经过 1st IPVO 嵌入后，图 5(b)中的 7 个红色像素共产生了 7 位失真。然后，根据图 2(c)所示的第 2 种模式计算出该子块的 2nd IPVO 嵌入容量为 1 bit，对图 5(b)中的蓝色像素 119 进行扩展以嵌入“0”，其余像素平移。经过 2nd IPVO 嵌入后，图 5(c)中的 3 个红色像素共产生了 6 位失真。对比图 5(b)和(c)可以发现，中间一列的 4 个像素全部恢复回原始值，即产生了失真补偿。




(a)原始子块　(b)1st IPVO 嵌入后的子块　(c)2nd IPVO 嵌入后的子块

图 5　子块失真补偿实例

Fig. 5　An example of distortion compensation

### 1.4 子块间互不混淆证明

本文通过 3 种不同的可逆信息隐藏算法实现了对所有子块的独立认证，且考虑到多种算法组合的连贯性，本文方法中不同类别的子块在认证端不会产生相互混淆，也不需要进行无效移位。由第 1.1 节认证码的嵌入过程可知，所有子块将根据 2 次不同嵌入单元划分模式下的 IPVO 嵌入容量和 LHS 嵌入容量进行分类，分类完成后，非 IPVO 块将依次进行 LHS 嵌入和 DE 嵌入。接下来本文将证明 LHS 嵌入和 DE 嵌入并不会将非 IPVO 块与 IPVO 块混淆，因此也不会影响认证端对 IPVO 块的判定以及认证信息的提取。

(1)LHS 嵌入。尽管本文方法在子块分类中的 2 次 IPVO 容量计算分别在不同的嵌入单元划分模式下进行，但本质上均为 IPVO 嵌入，因此以非 IPVO 块在 2 种划分模式下的任意一个嵌入单元 $B=\{B_1,B_2,B_3\}$ 为例，将其按照升序排列得到 $B=\{B_{\sigma_1},B_{\sigma_2},B_{\sigma_3}\}$，该序列必定满足

$$d_{\max}\in\begin{cases}(-\infty,0), & d_{\max}=B_{\sigma_2}-B_{\sigma_3}\\(1,\infty), & d_{\max}=B_{\sigma_3}-B_{\sigma_2}\end{cases}\tag{26}$$

$$d_{\min}\in\begin{cases}(-\infty,0), & d_{\min}=B_{\sigma_1}-B_{\sigma_2}\\(1,\infty), & d_{\min}=B_{\sigma_2}-B_{\sigma_1}\end{cases}\tag{27}$$

式(26)、(27)表明，该嵌入单元在 IPVO 嵌入中没有嵌入容量。

接下来对非 IPVO 块进行 LHS 嵌入，表 1 给出了预测误差在 LHS 嵌入中可能出现的所有取值情况。根据子块的峰值点 $P$ 与嵌入单元内 3 个像素 $B_{\sigma_1}$、$B_{\sigma_2}$、$B_{\sigma_3}$ 之间的不同大小关系，$d_{\max}$ 和 $d_{\min}$ 表示该嵌入单元在 IPVO 嵌入中的预测误差，$d_{\max}^{\mathrm{LHS}}$ 和 $d_{\min}^{\mathrm{LHS}}$ 则分别表示 $d_{\max}$ 和 $d_{\min}$ 经过 LHS 嵌入后的预测误差。表 1 为预测误差在 LHS 嵌入中可能的取值情况，由表 1 可以得出以下结论：如果预测误差 $d_{\max}$ 本身大于 1，则经过 LHS 嵌入后的预测误差 $d_{\max}^{\mathrm{LHS}}$ 将保持不变或增大；如果预测误差 $d_{\min}$ 本身小于 0，则经过 LHS 嵌入后的预测误差 $d_{\min}^{\mathrm{LHS}}$ 将保持不变或减小。因此，LHS 嵌入不会使得非 IPVO 块与 IPVO 块发生混淆。

表 1 预测误差在 LHS 嵌入中可能的取值情况

Table 1 The possible values of prediction errors in LHS embedding

| 大小关系 | $d_{\max}^{\mathrm{LHS}}$ 可能的取值情况 | | $d_{\min}^{\mathrm{LHS}}$ 可能的取值情况 | |
|---|---|---|---|---|
| | $d_{\max}<0$ | $d_{\max}>1$ | $d_{\min}<0$ | $d_{\min}>1$ |
| $B_{\sigma_1}<B_{\sigma_2}<B_{\sigma_3}<P$ | $d_{\max},d_{\max}-1$ | $d_{\max},d_{\max}+1$ | $d_{\min}$ | $d_{\min}$ |
| $B_{\sigma_1}<B_{\sigma_2}<B_{\sigma_3}=P$ | $d_{\max},d_{\max}-1$ | $d_{\max}+1$ | $d_{\min},d_{\min}-1$ | $d_{\min},d_{\min}+1$ |
| $B_{\sigma_1}<B_{\sigma_2}<P<B_{\sigma_3}$ | $d_{\max},d_{\max}-1,d_{\max}-2$ | $d_{\max},d_{\max}+1,d_{\max}+2$ | $d_{\min},d_{\min}-1$ | $d_{\min},d_{\min}+1$ |
| $B_{\sigma_1}<B_{\sigma_2}=P<B_{\sigma_3}$ | $d_{\max},d_{\max}-1$ | $d_{\max}+1$ | $d_{\min},d_{\min}-1$ | $d_{\min}+1$ |
| $B_{\sigma_1}<P<B_{\sigma_2}<B_{\sigma_3}$ | $d_{\max},d_{\max}-1$ | $d_{\max},d_{\max}+1$ | $d_{\min},d_{\min}-1,d_{\min}-2$ | $d_{\min},d_{\min}+1,d_{\min}+2$ |
| $B_{\sigma_1}=P<B_{\sigma_2}<B_{\sigma_3}$ | $d_{\max},d_{\max}-1$ | $d_{\max},d_{\max}+1$ | $d_{\min},d_{\min}-1$ | $d_{\min}+1$ |
| $P<B_{\sigma_1}<B_{\sigma_2}<B_{\sigma_3}$ | $d_{\max}$ | $d_{\max}$ | $d_{\min},d_{\min}-1$ | $d_{\min},d_{\min}+1$ |

(2)DE 嵌入。由式(7)和式(8)可知，DE 嵌入会使得子块内的最小值保持不变或变小、最大值保持不变或变大。因此对于任意一个嵌入单元，其预测误差 $d_{\max}^{\mathrm{LHS}}$ 和 $d_{\min}^{\mathrm{LHS}}$ 将在 DE 嵌入后继续保持大于 1 或小于 0 的状态，故非 IPVO 块与 IPVO 块也不会产生混淆。另外，由于在选择差值扩展方向时已经避免了对局部直方图峰值点的修改，因此 DE 块和 LHS 块之间也不会产生混淆。

## 2 实验分析与讨论

本文使用 6 幅 512×512 像素标准灰度图像 Splash、Baboon、Lena、Plane、Peppers、House 进行实验，采用峰值信噪比 $R_{\mathrm{PSNR}}$ 来衡量嵌入认证信息后图像的视觉质量，使用钟等[29]提出的正确检测率 $Y$ 来评价本文方法的认证性能。

$$Y=\frac{C_{\mathrm{block}}}{T_{\mathrm{block}}}\times 100\%\tag{28}$$

式中：$C_{\mathrm{block}}$ 表示正确检测出的篡改块数；$T_{\mathrm{block}}$ 表示实际的篡改块数。

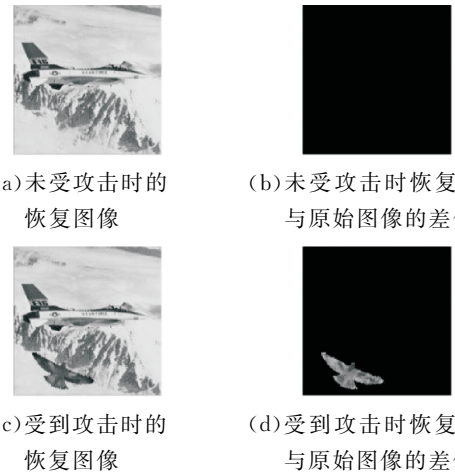
为了验证本文方法的可逆性，本文给出了未受攻击图像与受攻击图像的恢复情况。为了证明本文方法的有效性，本文从嵌入容量、视觉质量和认证精度 3 个方面将本文方法与 Hong 等[25]、Yao 等[26]、王泓等[27]和钟亦友等[29]的最新方法进行了对比。

### 2.1 可逆性验证

本文提出一种分块独立的可逆认证方法，对于图像中未受篡改攻击的子块，可以实现子块的无损恢复与认证信息的准确提取；对于受到篡改攻击的子块，可以实现精确的篡改检测与定位。

为了验证上述可逆性，接下来以 Plane 图像受

到篡改攻击为例进行实验。图 6 分别展示了未受篡改攻击和受篡改攻击后的水印图像恢复情况，以及未受篡改攻击和受篡改攻击后的恢复图像与原始图像的差值。对于未受攻击的图像，恢复图像与原始图像的差值为 0，证明原始载体图像可以被无损恢复。对于受到攻击的图像，恢复图像与原始图像的峰值信噪比为 20.25 dB，且从恢复图像与原始图像的差值可以看出，左下区域的子块由于受到攻击而无法实现无损恢复，其余区域的子块则可以实现无损恢复。



(a)未受攻击时的恢复图像　(b)未受攻击时恢复图像与原始图像的差值

(c)受到攻击时的恢复图像　(d)受到攻击时恢复图像与原始图像的差值

图 6　未受攻击图像与受攻击图像的恢复情况

Fig. 6　Recovery of untampered and tampered images

### 2.2　3 种嵌入算法的性能对比

为了进一步探究 2 次不同嵌入单元划分模式下的 IPVO、LHS 和 DE 算法在本文方法中起到的认证效果，表 2 给出了这 3 种嵌入算法在不同测试图像中的嵌入容量、认证子块数以及 2 次 IPVO 嵌入产生的失真补偿像素数。

表 2　3 种嵌入算法的性能对比

Table 2　Comparison of three embedding algorithms

| 图像 | 容量/$10^3$ bit | | | 子块数/$10^3$ bit | | | 两次 IPVO 失真补偿像素数 |
|---|---|---|---|---|---|---|---|
| | 两次 IPVO | LHS | DE | 两次 IPVO | LHS | DE | |
| Splash | 102.18 | 0.27 | 0.39 | 21.26 | 0.24 | 0.39 | 9 783 |
| Baboon | 26.95 | 1.88 | 6.87 | 13.27 | 1.76 | 6.87 | 6 401 |
| Lena | 73.54 | 0.81 | 1.76 | 19.39 | 0.73 | 1.76 | 12 550 |
| Plane | 103.47 | 0.54 | 1.52 | 19.87 | 0.50 | 1.52 | 8 358 |
| Peppers | 60.44 | 0.90 | 1.63 | 19.47 | 0.78 | 1.63 | 7 112 |
| House | 90.71 | 0.86 | 2.64 | 18.45 | 0.80 | 2.64 | 8 765 |
| 平均值 | 76.21 | 0.88 | 2.47 | 18.62 | 0.80 | 2.47 | 8 828 |

从表 2 可以看出，两次 IPVO 算法在认证中起主要作用，6 幅测试图像里平均 84.5% 的子块为 IPVO 块，且平均一个 IPVO 块可嵌入的认证码多达 4.1 bit。此外，两次 IPVO 嵌入还能产生相当数量的失真补偿像素。相比之下，通过 LHS 和 DE 算法实现认证码嵌入的子块较少，平均一个 LHS 块可嵌入 1.1 bit 认证码，而一个 DE 块固定嵌入 1 bit 认证码，但这 2 种算法在整个认证过程中都起到了重要的作用，LHS 算法可以嵌入至少 1 bit 认证码，相较 DE 算法引入的失真更少且能实现更高的篡改检测准确率，同时还不会产生额外的溢出信息；DE 算法虽然只能嵌入 1 bit 认证信息，但能够保证所有子块都嵌入认证码，是实现子块独立认证的关键。

### 2.3　嵌入容量对比

一般情况下，认证效果随着嵌入容量的增加而增加，表 3 为本文方法与文献[25-27, 29]方法的嵌入容量对比。从表中可以看出本文方法的嵌入容量远大于其他 4 种方法，这主要得益于本文采用了 2 种不同的 IPVO 嵌入单元划分模式。虽然文献[25-27]也使用了 IPVO 算法来嵌入认证信息，但平均嵌入容量却不及本文方法的一半，主要原因是这些方法都将 4×4 子块固定划分成 4 个 2×2 的嵌入单元，因此无法进行两次不同划分模式下的 IPVO 嵌入。钟等[29]采用了双层嵌入模式，但是嵌入容量也不及本文方法，这是因为本文的分块尺寸更小，在相同大小的图像中产生的子块数量更多，因此可以得到更大的嵌入容量。

表 3　不同方法的嵌入容量对比

Table 3　Comparison of embedding capacity of different methods

| 图像 | 容量/$10^4$ bit | | | | |
|---|---|---|---|---|---|
| | 文献[25] | 文献[26] | 文献[27] | 文献[29] | 本文 |
| Splash | 5.3 | 5.3 | 4.6 | 8.7 | 10.3 |
| Baboon | 1.3 | 1.3 | 1.2 | 2.0 | 3.6 |
| Lena | 3.8 | 3.8 | 3.2 | 6.0 | 7.6 |
| Plane | 5.2 | 5.2 | 4.7 | 8.7 | 10.6 |
| Peppers | 3.1 | 3.1 | 2.8 | 3.8 | 6.3 |
| House | 4.6 | 4.6 | 4.2 | 7.6 | 9.4 |
| 平均值 | 3.9 | 3.9 | 3.5 | 6.1 | 8.0 |

### 2.4 水印图像质量对比

表 4 给出了 6 幅测试图像在嵌入认证信息后的峰值信噪比，可以看到本文方法的峰值信噪比平均值要低于文献[25-27]，这主要是由于本文采用了失真较大的 DE 算法来实现子块的独立认证，而文献[25-27]采用失真极小的 LSB 替换算法来嵌入认证码，但无法实现对不可嵌入块的独立认证。与同样实现独立认证的钟等[29]方法相比，除了 Splash 图像以外，本方法得到的峰值信噪比都更高，主要原因有二：首先，IPVO 算法本身由 PEE 算法改进而来，其通过使用新的预测器可以减少移位像素，从而产生更小的失真；其次，两次不同嵌入单元划分模式下的 IPVO 算法可以产生失真补偿的效果，而钟等[29]采用的棋盘格式双层嵌入不存在失真补偿。值得注意的是，对于纹理较为复杂的 Baboon 图像，文献[25-27]由于嵌入容量过低而无法实现认证，钟等[29]的方法可以实现独立认证，但得到的峰值信噪比低于人眼视觉感知的标准(38 dB)。相比之下，本文方法不仅可以实现独立认证，且得到的水印图像视觉质量较钟等[29]得到的水印图像更好，峰值信噪比提高了 3.6 dB。

表 4 不同方法的峰值信噪比对比

Table 4 Comparison of peak signal-to-noise rate of different methods

| 图像 | 峰值信噪比/dB | | | | |
|---|---|---|---|---|---|
| | 文献[25] | 文献[26] | 文献[27] | 文献[29] | 本文 |
| Splash | 51.65 | 49.55 | 51.53 | 46.08 | 45.18 |
| Baboon | | | | 34.59 | 38.19 |
| Lena | 50.60 | 47.95 | 50.01 | 42.64 | 43.03 |
| Plane | 51.04 | 48.84 | 50.65 | 42.40 | 43.16 |
| Peppers | 50.34 | 50.62 | 50.02 | 41.39 | 43.15 |
| House | 50.37 | 47.85 | 49.96 | 41.24 | 41.98 |
| 平均值 | 50.80 | 48.96 | 50.43 | 41.39 | 42.45 |

### 2.5 认证性能对比

对于可逆认证而言，水印图像通常可以恢复出原始图像，因此相较于水印图像的视觉质量，其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质量可接受的同时，能获得更好的认证精度。实验中本文对 5 种可逆认证方法得到的 Lena 水印图像进行多种篡改攻击，并分别从初始正确检测率和细化处理[29]后的正确检测率这两个方面进行比较。

如图 7 所示，实验分别对 Lena 图像进行剪贴攻击、常数攻击、拼贴攻击和随机篡改攻击。其中，随机篡改攻击的比例为 1%，即随机选择 1%的子块，并将子块内的一个像素随机修改为不大于 255 且不小于 0 的任意值。

(a)剪贴攻击 (b)常数攻击 (c)拼贴攻击 (d)随机篡改

图 7 4 种篡改攻击

Fig. 7 Four types of tampering attack

从表 5 中可以看到，本方法在 4 种攻击下的初始正确检测率都要明显高于文献[25-27]，这是因为本方法的嵌入容量更大，子块嵌入的认证信息也更多。与钟等[29]的方法相比，本文在剪贴攻击、常数攻击以及拼贴攻击上都有更高的初始正确检测率，这依然得益于本文较高的嵌入容量，但对于随机篡改攻击，本文的初始正确检测率要低于钟等[29]方法，这是由于本方法通过差值扩展仅能嵌入 1 bit 认证码，而钟等[29]方法通过双层嵌入的差值扩展可嵌入 2 bit 认证码。经过细化处理[29]后，本文的细化正确检测率同样显著高于前 3 种认证方案，这也符合初始正确检测率的结果。与钟等[29]的方法相比，本方法在前 3 种攻击下的细化正确检测率都与其接近，但随机篡改的正确检测率较低，这同样是由差值扩展时嵌入的认证码数量差异导致的。

表 5 不同方法的初始和细化正确检测率对比

Table 5 Comparison of initial and refined detection accuracy rates corresponding to different methods

| 篡改攻击类型 | 初始正确检测率/% | | | | | 细化正确检测率/% | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 文献[25] | 文献[26] | 文献[27] | 文献[29] | 本文 | 文献[25] | 文献[26] | 文献[27] | 文献[29] | 本文 |
| 剪贴攻击 | 66.47 | 73.60 | 60.67 | 84.13 | 85.12 | 95.33 | 97.07 | 95.73 | 98.27 | 98.97 |
| 常数攻击 | 63.28 | 74.12 | 62.50 | 75.00 | 82.34 | 98.93 | 98.63 | 99.02 | 99.90 | 99.93 |
| 拼贴攻击 | 62.50 | 70.70 | 69.92 | 76.17 | 94.89 | 96.09 | 97.66 | 100.0 | 100.0 | 100.0 |
| 随机篡改 | 58.64 | 48.47 | 58.02 | 81.48 | 77.72 | 58.64 | 48.47 | 58.02 | 81.48 | 77.72 |

注：细化正确检测率由钟等[29]所提出的细化处理方法计算得出。

需要强调的是，其他 4 种方法的认证单元均为 4×4 大小的子块，而本文方法的认证单元为 4×3 大小的子块。一般来说，子块尺寸越大意味着嵌入容量越大，因此可以嵌入更多的认证信息以实现更高的正确检测率。但是子块的尺寸会影响篡改区域的定位精度，也就是说在大尺寸子块的情况下，即使确定子块被篡改也很难定位具体发生在子块的哪个区域。因此，相比于现有方法，本文方法能够在大部分的篡改攻击下实现更高的正确检测率，同时使得可逆认证的认证单元从常规的 4×4 缩小到了 4×3，实现了更加精准的篡改定位。

## 3　结　论

本文提出了一种基于图像分块的独立可逆认证方法。为了实现独立认证，本文采用了 3 种可逆信息隐藏算法来实现每个子块认证信息的嵌入，其中两次不同嵌入单元划分模式下的 IPVO 嵌入不仅可以使得大部分子块嵌入尽可能多的认证码，而且还能产生失真补偿的效果以提升视觉质量，而 LHS 嵌入在帮助一部分子块完成认证码嵌入时具有引入失真较小的优点，另外 DE 嵌入则保证了所有子块都能嵌入认证码，对子块的独立认证起到了关键作用。此外，本文还考虑了多种算法组合使用时的连贯性和可逆性，在保证不同类别的子块互不混淆的情况下，不需要进行额外的无效移位。实验结果表明，本文方法在保证视觉质量可接受的同时，比现有的可逆认证方案具有更高的篡改检测准确率和更精细的篡改定位能力。

**参考文献：**


[1] HU Yuchen, LO C C, CHEN Wulin. Probability-based reversible image authentication scheme for image demosaicking [J]. Future Generation Computer Systems, 2016, 62: 92-103.

[2] LOU D C, LIU J L. Fault resilient and compression tolerant digital signature for image authentication [J]. IEEE Transactions on Consumer Electronics, 2000, 46(1): 31-39.

[3] XIE Liehua, ARCE G R, GRAVEMAN R F. Approximate image message authentication codes [J]. IEEE Transactions on Multimedia, 2001, 3(2): 242-252.

[4] AHMED F, SIYAL M Y, UDDIN ABBAS V. A secure and robust hash-based scheme for image authentication [J]. Signal Processing, 2010, 90(5): 1456-1470.

[5] 金喜子，姜文哲. 块级篡改定位的 JPEG 图像脆弱水印 [J]. 电子学报，2010，38(7)：1585-1589.
JIN Xizi, JIANG Wenzhe. Fragile watermarking capable of locating tampered blocks in JPEG images [J]. Acta Electronica Sinica, 2010, 38(7): 1585-1589.

[6] HE Hongjie, CHEN Fan, TAI Hengming, et al. Performance analysis of a block-neighborhood-based self-recovery fragile watermarking scheme [J]. IEEE Transactions on Information Forensics and Security, 2012, 7(1): 185-196.

[7] SU Guodong, CHANG C C, CHEN C C. A hybrid-Sudoku based fragile watermarking scheme for image tampering detection [J]. Multimedia Tools and Applications, 2021, 80(8): 12881-12903.

[8] TIAN Jun. Reversible data embedding using a difference expansion [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(8): 890-896.

[9] CHEN Xianyi, SUN Xingming, SUN Huiyu, et al. Histogram shifting based reversible data hiding method using directed-prediction scheme [J]. Multimedia Tools Applications, 2015, 74(15): 5747-5765.

[10] KIM S, QU Xiaochao, SACHNEV V, et al. Skewed histogram shifting for reversible data hiding using a pair of extreme predictions [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 29(11): 3236-3246.

[11] PAN Zhibin, HU Sen, MA Xiaoxiao, et al. Reversible data hiding based on local histogram shifting with multilayer embedding [J]. Journal of Visual Communication and Image Representation, 2015, 31: 64-74.

[12] HUANG Dan, HUANG Fangjun. Reversible data hiding based on adaptive block selection strategy [J]. International Journal of Digital Crime and Forensics, 2020, 12(1): 157-168.

[13] TSAI P, HU Yuchen, YEH H L. Reversible image hiding scheme using predictive coding and histogram shifting [J]. Signal Processing, 2009, 89(6): 1129-1143.

[14] THODI D M, RODRIGUEZ J J. Expansion embedding techniques for reversible watermarking [J]. IEEE Transactions on Image Processing, 2007, 16(3): 721-730.

[15] LI Xiaolong, YANG Bin, ZENG Tieyong. Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection [J]. IEEE Transactions on Image Processing, 2011, 20(12):

3524-3533.

[16] XIAO Mengyao, LI Xiaolong, WANG Yangyang, et al. Reversible data hiding based on pairwise embedding and optimal expansion path [J]. Signal Processing, 2019, 158: 210-218.

[17] OU Bo, LI Xiaolong, ZHANG Weiming, et al. Improving pairwise PEE via hybrid-dimensional histogram generation and adaptive mapping selection [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 29(7): 2176-2190.

[18] QIN Jianqiang, HUANG Fangjun. Reversible data hiding based on multiple two-dimensional histograms modification [J]. IEEE Signal Processing Letters, 2019, 26(6): 843-847.

[19] PENG Fei, LI Xiaolong, YANG Bin. Improved PVO-based reversible data hiding [J]. Digital Signal Processing, 2014, 25: 255-265.

[20] KAUR G, SINGH S, RANI R. A high capacity reversible data hiding technique based on pixel value ordering using interlock partitioning [C]//2020 7th International Conference on Signal Processing and Integrated Networks (SPIN). Piscataway, NJ, USA: IEEE, 2020: 727-732.

[21] QU Xiaochao, KIM H J. Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding [J]. Signal Processing, 2015, 111: 249-260.

[22] LO C C, HU Yuchen. A novel reversible image authentication scheme for digital images [J]. Signal Processing, 2014, 98: 174-185.

[23] NGUYEN T S, CHANG C C, SHIH T H. A high-quality reversible image authentication scheme based on adaptive PEE for digital images [J]. KSII Transactions on Internet and Information Systems, 2016, 10(1): 395-413.

[24] YIN Zhaoxia, NIU Xuejing, ZHOU Zhili, et al. Improved reversible image authentication scheme [J]. Cognitive Computation, 2016, 8(5): 890-899.

[25] HONG W, CHEN Meijin, CHEN T S. An efficient reversible image authentication method using improved PVO and LSB substitution techniques [J]. Signal Processing: Image Communication, 2017, 58: 111-122.

[26] YAO Heng, WEI Hongbin, QIN Chuan, et al. A real-time reversible image authentication method using uniform embedding strategy [J]. Journal of Real-Time Image Processing, 2020, 17(1): 41-54.

[27] 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 [J]. 信息安全学报, 2022, 7(1): 56-65.
WANG Hong, HUANG FANG Jun. Attack and improvement of an authentication scheme based on reversible data hiding [J]. Journal of Cyber Security, 2022, 7(1): 56-65.

[28] NGUYEN T S, CHANG C C, YANG Xiaoqian. A reversible image authentication scheme based on fragile watermarking in discrete wavelet transform domain [J]. AEU-International Journal of Electronics and Communications, 2016, 70(8): 1055-1061.

[29] 钟亦友, 黄方军. 基于分块的高效图像可逆认证方法 [J]. 软件学报, 2023, 34(12): 5848-5861.
ZHONG Yiyou, HUANG FANG Jun. Efficient image reversible authentication method based on blocks [J]. Journal of Software, 2023, 34(12): 5848-5861.

（编辑 刘杨 陶晴）