

基于像素值排序与块再分的可逆数据隐藏算法

任方^{1,2}, 杨益萍^{1,2}, 薛斐元¹

(1.西安邮电大学 网络空间安全学院,西安 710121; 2.西安邮电大学 无线网络安全技术国家工程实验室,西安 710121)

摘要: 基于像素值排序的可逆数据隐藏算法通过修改图像块中的最大像素和最小像素嵌入数据,但并未充分利用图像块内的每一个像素,从而影响嵌入性能。结合块再分原理,提出基于像素值排序的可逆数据隐藏算法。将原始图像划分为3×3的图像块,计算每一个图像块的局部复杂度。设计12种分块模式将局部复杂度小于阈值的图像块细分为子块A和B。根据子块A和B的不同局部特征分别采用2种不同的扫描顺序读取像素。子块A的像素序列使用次小值预测最小值和次大值预测最大值的方法,获得2个预测误差值,子块B的像素序列利用中值像素连续预测其余4个像素的方法,得到4个预测误差值。在此基础上,利用图像块中预测误差值为0和1的像素嵌入隐藏数据。实验结果表明,该算法在一个图像块中最高可嵌入6 bit的数据,在较低计算复杂度的情况下能够有效提高像素的嵌入性能。

关键词: 像素值排序;可逆数据隐藏;嵌入性能;局部复杂度;阈值

开放科学(资源服务)标志码(OSID):



中文引用格式:任方,杨益萍,薛斐元.基于像素值排序与块再分的可逆数据隐藏算法[J].计算机工程,2022,48(10):130-137.

英文引用格式:REN F, YANG Y P, XUE F Y. Reversible data hiding algorithm based on pixel value ordering and block subdivision[J]. Computer Engineering, 2022, 48(10): 130-137.

Reversible Data Hiding Algorithm Based on Pixel Value Ordering and Block Subdivision

REN Fang^{1,2}, YANG Yiping^{1,2}, XUE Feiyuan¹

(1.School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China;

2.National Engineering Laboratory for Wireless Network Security Technology,
Xi'an University of Posts and Telecommunications, Xi'an 710121, China)

[Abstract] The reversible data hiding algorithm based on Pixel Value Ordering (PVO) embeds data by modifying the largest and smallest pixels in the block without completely using each pixel in image block, thereby affecting the embedding performance. Combined with the principle of block subdivision, this study proposes an reversible data hiding algorithm based on PVO by dividing the original image into 3×3 image blocks to calculate the local complexity of each. Twelve block modes are constructed for subdividing the image block, whose local complexity is less than the threshold, into sub-blocks A and B. According to the different local features of sub-blocks A and B, two different scanning sequences are used to read pixels. In the pixel sequence of sub-block A, two prediction error values are obtained using the second smallest pixel to predict the minimum pixel and the second largest pixel to predict the maximum pixel. In the pixel sequence of sub-block B, four prediction error values are obtained by continuously predicting the remaining four pixels using the median pixel. On this basis, hidden data are embedded using pixels with prediction error values of zero and one in the image block. Experiments reveal that the algorithm can embed up to six bit of data in an image block and effectively improve the embedding performance of pixels with a low computational complexity.

[Key words] Pixel Value Ordering (PVO); reversible data hiding; embedding performance; local complexity; threshold

DOI: 10. 19678/j. issn. 1000-3428. 0063019

0 概述

可逆数据隐藏是一种只需要对载体图像的像素进行轻微修改,并将数据隐藏在载体图像中的技术。该

技术从被嵌入数据的载体图像中无损地提取数据,同时完全恢复出原载体图像^[1-2]。可逆数据隐藏技术在医学图像^[3-5]、军事图像、法律取证等领域中传输和使用数据时,能够确保数据不失真^[6-8]。

基金项目: 国家自然科学基金(61902315);陕西省自然科学基金计划(2021JM-463);西安邮电大学研究生创新基金(CXJLY2021077)。

作者简介: 任方(1981—),男,副教授、博士,主研方向为数据隐藏与数据安全;杨益萍、薛斐元,硕士研究生。

收稿日期: 2021-10-21 **修回日期:** 2022-01-19 **E-mail:** 1214801226@qq.com

早期的可逆数据隐藏技术^[9-11]通过压缩载体图像的一个位平面来创造空间, 并进行数据嵌入, 但其可逆性高度依赖于压缩算法的性能。文献[12]提出DE算法, 通过对载体图像进行整数小波变换, 同时计算并扩展差值, 以嵌入可逆数据。为充分利用自然图像的冗余空间, 预测误差扩展(Prediction Error Expansion, PEE)^[13]采用预测误差代替DE算法中的差值, 受到许多研究人员的关注^[14-16]。

文献[17]结合 PEE 提出像素值排序(Pixel Value Ordering, PVO)算法。该算法将载体图像分成非重叠同等大小的块, 并对块内的像素进行排序, 通过计算预测误差为块中最大(小)值与次大(小)值的差值, 以修改最大(小)值嵌入数据。在 PVO 算法中, 预测误差等于 1 的图像块被用于嵌入数据, 预测误差等于 0 的图像块未被利用。文献[18]提出改进的 PVO(IPVO)算法, 根据最大(小)值和次大(小)值的空间位置预测误差, 误差等于 0 和误差等于 1 的图像块都被用于嵌入数据, 以提高 PVO 的嵌入容量。文献[19]提出 PVO-K, 同时修改 k 个最大(小)值以嵌入 1 bit 数据, 当 $k=1$ 时, PVO-K 退化为 PVO。文献[20]提出 PPVO, 打破了 PVO 中分块的限制, 将每个像素作为一个嵌入单元。文献[21]将 PVO 生成的预测误差两两分为一组, 采用 2D 的方式成对修改误差以嵌入数据。研究人员利用可逆约束对像素进行动态预测, 从而提升 PVO 的嵌入容量^[22]。为获得更大的嵌入容量, 文献[23]将图像块的复杂度分为 m 个级别, 根据不同级别动态地修改块内像素的个数。

本文提出基于像素值排序的块再分算法。构建 12 种分块模式将 1 个图像块分为 2 个子块, 对每个子块采用不同的扫描顺序及嵌入策略, 以充分利用图像块中每个像素之间的相关性。此外, 设计一种局部复杂度的计算方式, 当嵌入数据时, 只处理复杂度小于阈值的图像块, 提高嵌入数据后图像的视觉质量。

1 基于像素值排序的可逆数据隐藏算法

PVO 算法^[17]通过对图像进行分块, 并将块内的 n 个像素按照升序排列, 使得 $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \dots \leq x_{\sigma(n)}$, 在此基础上, 计算预测误差 $e_{\max} = x_{\sigma(n)} - x_{\sigma(n-1)}$ 。PVO 和 IPVO 算法的预测误差直方图如图 1 所示。

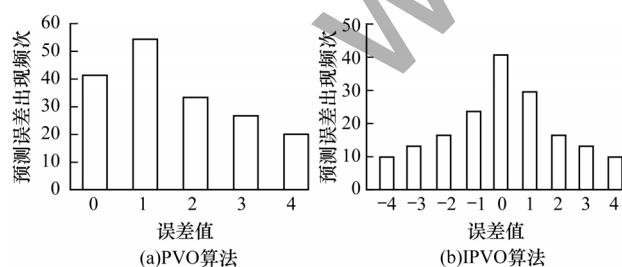


图 1 PVO 和 IPVO 算法的预测误差直方图

Fig.1 Prediction error histogram of PVO and IPVO algorithms

POV 预测误差 e_{\max} 的直方图误差范围在 $(0, +\infty)$, 峰值点为 1。PVO 算法通过 $e_{\max}=1$ 来嵌入数据。

文献[18]提出改进的像素值排序(IPVO)算法, 其目的是利用在 PVO 算法中被放弃的最大(小)值与次大(小)值相等的图像块来嵌入数据。IPVO 计算得到的预测误差 $d_{\max} = x_u - x_v$, 其中 $u = \min(\sigma(n), \sigma(n-1))$, $v = \max(\sigma(n), \sigma(n-1))$ 。IPOV 预测的误差直方图如图 1(b) 所示, 预测误差 d_{\max} 的直方图误差范围在 $(-\infty, +\infty)$, 峰值点为 0。IPVO 利用 $d_{\max}=0$ 和 $d_{\max}=1$ 来嵌入数据, 大幅增大了 PVO 的嵌入容量。

当数据被嵌入到图像块后, 图像块的最大值被修改为 $x'_{\sigma(n)} = x_{\sigma(n)} + b$, 最小值被修改为 $x'_{\sigma(1)} = x_{\sigma(1)} - b$, 其他像素保持不变, 保证了算法的可逆性。在提取过程中, 水印误差为 $d'_{\max} = x'_u - x'_v$, 如果 $d'_{\max} \in \{1, 2\}$, 则提取的数据 $b = d'_{\max} - 1$, 如果 $d'_{\max} \in \{-1, 0\}$, 则提取的数据 $b = -d'_{\max}$, 最终恢复的原始像素 $x_{\sigma(n)} = x'_{\sigma(n)} - b$ 。

IPVO 在 3×3 图像块最大值上嵌入和提取数据的过程如图 2 所示。

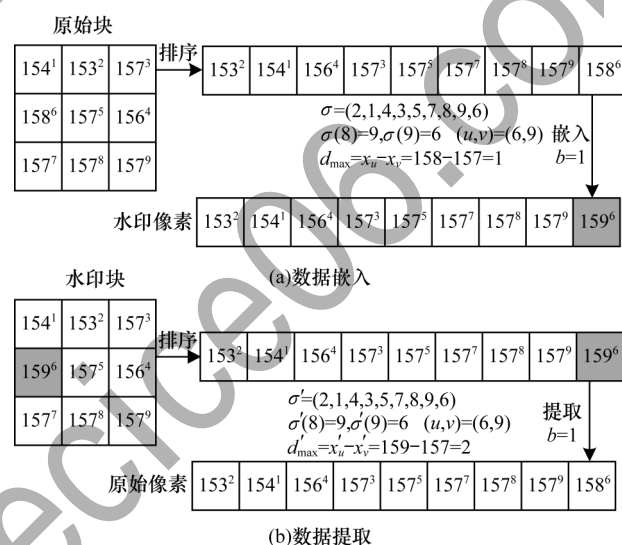


图 2 IPVO 算法在最大像素上嵌入和提取数据的过程

Fig.2 The process embedding and extraction data of IPVO algorithm on maximum pixel

当原始块的 9 个像素排序后, 最大值 $x_{\sigma(9)}$ 为 158, 次大值 $x_{\sigma(8)}$ 为 157, 其中 $\sigma(9)=6, \sigma(8)=9$, 预测误差 $d_{\max} = x_{\sigma(9)} - x_{\sigma(8)} = x_6 - x_9 = 1$ 。假设嵌入数据 $b=1$, 则 $d'_{\max} = d_{\max} + b$, 因此, 最大值 $x_{\sigma(9)}$ 被修改为 $x'_{\sigma(9)} = x_{\sigma(9)} + b = 158 + 1 = 159$ 。在提取数据的过程中, IPVO 同样对水印块的像素进行排序, 水印块的位置序列 σ' 与原始块的位置序列 σ 相同, 水印误差 $d'_{\max} = x'_{\sigma(9)} - x'_{\sigma(8)} = x'_6 - x'_9 = 2$ 。因为 $d'_{\max} \in \{1, 2\}$, 所以提取数据 $b = d'_{\max} - 1 = 1$, 同时原始像素被恢复为 $x_{\sigma(9)} = x'_{\sigma(9)} - b = 159 - 1 = 158$ 。

2 本文算法

2.1 分块模式

分块模式将原始图像分成 3×3 不重叠的块, 将每

一个块内包含的9个像素划分为2个像素集,即像素集A和像素集B。像素集A包含4个像素,采用次大像素值预测最大像素值和次小像素值的方法。像素集B包含当前块内剩下的5个像素,采用中值像素分别预测其余4个像素的方法。这样的划分方式共有 C_4^9 种,其目的是为了更充分利用像素之间的相关性。由于像素之间的相关性越高,产生的预测误差直方图分布越集中,因此可被用于嵌入数据的像素就越多。本文考虑到像素与像素之间的距离越近,其相关性越好,因此,像素集的划分应尽可能地让像素集内的像素彼此相连。在这 C_4^9 种划分方式中,12种细分块的模式如图3所示。

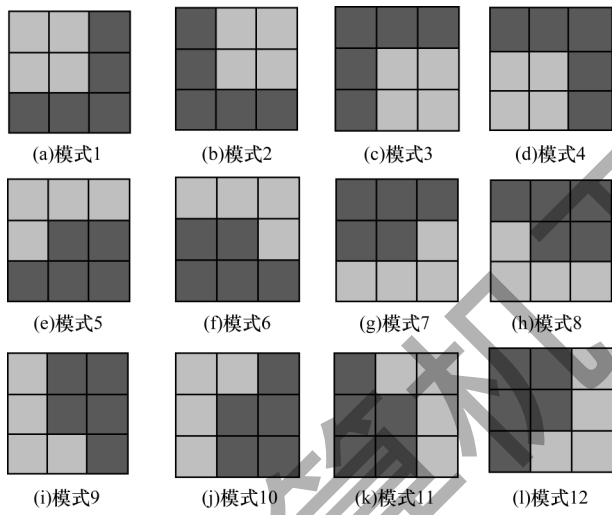


图3 12种分块模式

Fig.3 12 block modes

2.2 数据嵌入与提取

本文采用图3中的模式1进行数据嵌入与提取。

2.2.1 数据嵌入

数据分块示意图如图4所示。

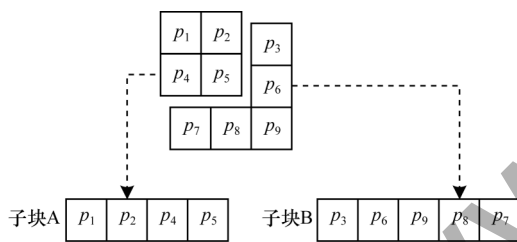


图4 数据分块示意图

Fig.4 Schematic diagram of data block

在数据嵌入过程中,对于图4中的子块A,首先逐行读取子块A的像素,获得像素序列 $\{p_1, p_2, p_4, p_5\}$,然后将其按照升序排列,得到 $\{p_{\omega(1)}, p_{\omega(2)}, p_{\omega(3)}, p_{\omega(4)}\}$,其中, ω 是一个双射,使得 $p_{\omega(1)} \leq p_{\omega(2)} \leq p_{\omega(3)} \leq p_{\omega(4)}$ 。如果像素 $p_{\omega(w)} = p_{\omega(r)}$,并且 $w < r$,则 $\omega(w) < \omega(r)$,计算预测误差 $e_{1,j}$,如式(1)所示:

$$e_{1,j} = p_u - p_v \quad (1)$$

其中: $j \in \{1, 2\}$; 当 $j=1$ 时, $u = \min\{\omega(j), \omega(2)\}$, $v = \max\{\omega(j), \omega(2)\}$; 当 $j=2$ 时, $u = \min\{\omega(j+1), \omega(4)\}$, $v = \max\{\omega(j+1), \omega(4)\}$ 。

如果 b 为嵌入的数据, $b \in \{0, 1\}$,那么计算得到的水印误差 $e'_{1,j}$ 如式(2)所示:

$$e'_{1,j} = \begin{cases} e_{1,j} - b, & e_{1,j} = 0 \\ e_{1,j} + b, & e_{1,j} = 1 \\ e_{1,j} - 1, & e_{1,j} < 0 \\ e_{1,j} + 1, & e_{1,j} > 1 \end{cases} \quad (2)$$

相应地,当数据被嵌入后,子块A的最小像素 $p_{\omega(1)}$ 被修改为:

$$p'_{\omega(1)} = \begin{cases} p_{\omega(1)} - b, & e_{1,j} = 1 \text{ or } e_{1,j} = 0 \\ p_{\omega(1)} - 1, & e_{1,j} > 1 \text{ or } e_{1,j} < 0 \end{cases} \quad (3)$$

其中: $p_{\text{former}} = p_{\omega(1)}$; $p'_{\text{former}} = p'_{\omega(1)}$ 。

最大像素 $p_{\omega(4)}$ 被修改为:

$$p'_{\omega(4)} = \begin{cases} p_{\omega(4)} + b, & e_{1,j} = 1 \text{ or } e_{1,j} = 0 \\ p_{\omega(4)} + 1, & e_{1,j} > 1 \text{ or } e_{1,j} < 0 \end{cases} \quad (4)$$

其中: $p_{\text{latter}} = p_{\omega(4)}$; $p'_{\text{latter}} = p'_{\omega(4)}$ 。

对于子块B,采用从上向下和从右至左的顺序读取像素,得到像素序列 $\{p_3, p_6, p_9, p_8, p_7\}$,将其按照升序排列得到 $\{p_{\tau(1)}, p_{\tau(2)}, p_{\tau(3)}, p_{\tau(4)}, p_{\tau(5)}\}$,其中, τ 是双射,使得 $p_{\tau(1)} \leq p_{\tau(2)} \leq p_{\tau(3)} \leq p_{\tau(4)} \leq p_{\tau(5)}$ 。如果 $p_{\tau(w)} = p_{\tau(r)}$,并且 $w < r$,则 $\tau(w) < \tau(r)$,预测误差 $e_{2,j}$ 如式(5)所示:

$$e_{2,j} = p_s - p_t \quad (5)$$

其中: $j \in \{1, 2, 3, 4\}$; 当 $j \in \{1, 2\}$ 时, $s = \min\{\tau(j), \tau(3)\}$, $t = \max\{\tau(j), \tau(3)\}$; 当 $j \in \{3, 4\}$ 时, $s = \min\{\tau(j+1), \tau(3)\}$, $t = \max\{\tau(j+1), \tau(3)\}$ 。

本文根据式(2)修改预测误差 $e_{2,j}$ 以嵌入数据 b 。

当 $j \in \{1, 2\}$ 时,像素 $p_{\tau(j)}$ 根据式(3)被修改为 $p'_{\tau(j)}$,其中, $p_{\text{former}} = p_{\tau(j)}$, $p'_{\text{former}} = p'_{\tau(j)}$; 当 $j \in \{3, 4\}$ 时,像素 $p_{\tau(j+1)}$ 根据式(4)被修改为 $p'_{\tau(j+1)}$,其中, $p_{\text{latter}} = p_{\tau(j+1)}$, $p'_{\text{latter}} = p'_{\tau(j+1)}$ 。

本文算法将 3×3 的图像块修改为可以嵌入6 bit 图像块,与PVO和IPVO等算法相比,大幅提升了嵌入容量。IPVO和本文算法的预测误差值对比如图5所示。图5选取了Lena的 3×3 的图像块,如果采用IPVO算法,则只能得到2个可以被用于嵌入数据的预测误差。如果使用本文算法,将得到6个可以用来嵌入数据的预测误差。

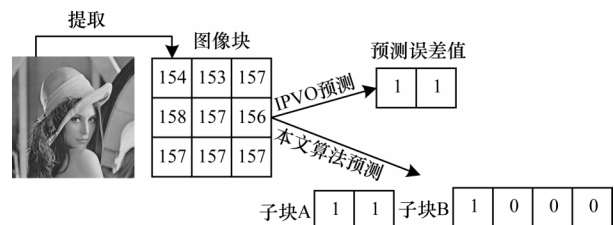


图5 IPVO算法和本文算法的预测误差值对比

Fig.5 Prediction error values comparison between IPVO algorithm and the proposed algorithm

2.2.2 数据提取

在数据的提取过程中,假设水印块A排序后的

像素序列为 $\{p'_{\omega(1)}, p'_{\omega(2)}, p'_{\omega(3)}, p'_{\omega(4)}\}$, 已知 $p'_{\omega(1)} \leq p_{\omega(1)}, p'_{\omega(2)} = p_{\omega(2)}, p'_{\omega(3)} = p_{\omega(3)}, p'_{\omega(4)} \geq p_{\omega(4)}$, 即水印块 A 的像素按升序排列后与原始块 A 的像素排序后得到的位置序列相同。因此, 原始块 A 的嵌入算法是可逆的, 同理, 原始块 B 的嵌入算法也是可逆的。水印块 A 的预测误差 $e'_{1,j} = p'_u - p'_v$, 其中, j, u, v 的定义与式(1)相同。提取得到的数据 b 如式(6)所示:

$$b = \begin{cases} e'_{1,j} - 1, & e'_{1,j} \in \{1, 2\} \\ -e'_{1,j}, & e'_{1,j} \in \{0, -1\} \end{cases} \quad (6)$$

将水印块 A 中的像素 $p'_{\omega(1)}$ 恢复为 $p_{\omega(1)}$, 如式(7)所示:

$$p_{\text{former}} = \begin{cases} p'_{\text{former}} + e'_{1,j} - 1, & e'_{1,j} \in \{1, 2\} \\ p'_{\text{former}} - e'_{1,j}, & e'_{1,j} \in \{0, -1\} \\ p'_{\text{former}} + 1, & e'_{1,j} < -1 \text{ or } e'_{1,j} > 2 \end{cases} \quad (7)$$

其中: $p_{\text{former}} = p_{\omega(1)}; p'_{\text{former}} = p'_{\omega(1)}$

根据式(8)将像素 $p'_{\omega(4)}$ 恢复为 $p_{\omega(4)}$:

$$p_{\text{latter}} = \begin{cases} p'_{\text{latter}} - e'_{1,j} + 1, & e'_{1,j} \in \{1, 2\} \\ p'_{\text{latter}} + e'_{1,j}, & e'_{1,j} \in \{0, -1\} \\ p'_{\text{latter}} - 1, & e'_{1,j} < -1 \text{ or } e'_{1,j} > 2 \end{cases} \quad (8)$$

其中: $p_{\text{latter}} = p_{\omega(4)}; p'_{\text{latter}} = p'_{\omega(4)}$

对于水印块 B, 假设像素按照升序排列后得到的像素序列为 $\{p'_{\tau(1)}, p'_{\tau(2)}, p'_{\tau(3)}, p'_{\tau(4)}, p'_{\tau(5)}\}$, 同样计算预测误差 $e'_{2,j} = p'_s - p'_t$, 其中, j, s, t 的定义同式(5)相同。本文根据式(6)提取数据 b , 恢复出块内除中值像素外的所有像素。当 $j \in \{1, 2\}$ 时, 根据式(7)将 $p'_{\tau(j)}$ 恢复为 $p_{\tau(j)}$, 其中, $p_{\text{former}} = p_{\tau(j)}, p'_{\text{former}} = p'_{\tau(j)}$; 当 $j \in \{3, 4\}$ 时, 根据式(8)将 $p'_{\tau(j+1)}$ 恢复为 $p_{\tau(j+1)}$, 其中, $p_{\text{latter}} = p_{\tau(j+1)}, p'_{\text{latter}} = p'_{\tau(j+1)}$ 。

2.2.3 数据嵌入与提取样例

本文算法的数据嵌入与提取过程如图6所示。数据嵌入过程首先将原始块分为子块 A 和子块 B, 对于子块 A, 逐行扫描像素, 排序得到像素序列 $p_{\omega} = \{159, 160, 160, 161\}$ 。根据式(1)计算预测误差 $e_{1,1}$ 和 $e_{1,2}$, 根据式(3)和式(4)修改最大和最小像素嵌入 b_1 和 b_2 。对于子块 B, 从上向下和从右到左扫描像素, 排序得到像素序列 $p_{\tau} = \{159, 160, 160, 161, 163\}$, 根据式(5)计算 4 个预测误差为 $e_{2,1}, e_{2,2}, e_{2,3}$ 和 $e_{2,4}$, 根据式(3)和式(4)将 p_8, p_6, p_5, p_9 修改为 p'_8, p'_6, p'_5, p'_9 , 以嵌入 b_3, b_4 和 b_5 。最后将两个子块内所有被修改的像素更新到原始块对应的位置上, 以生成水印块。

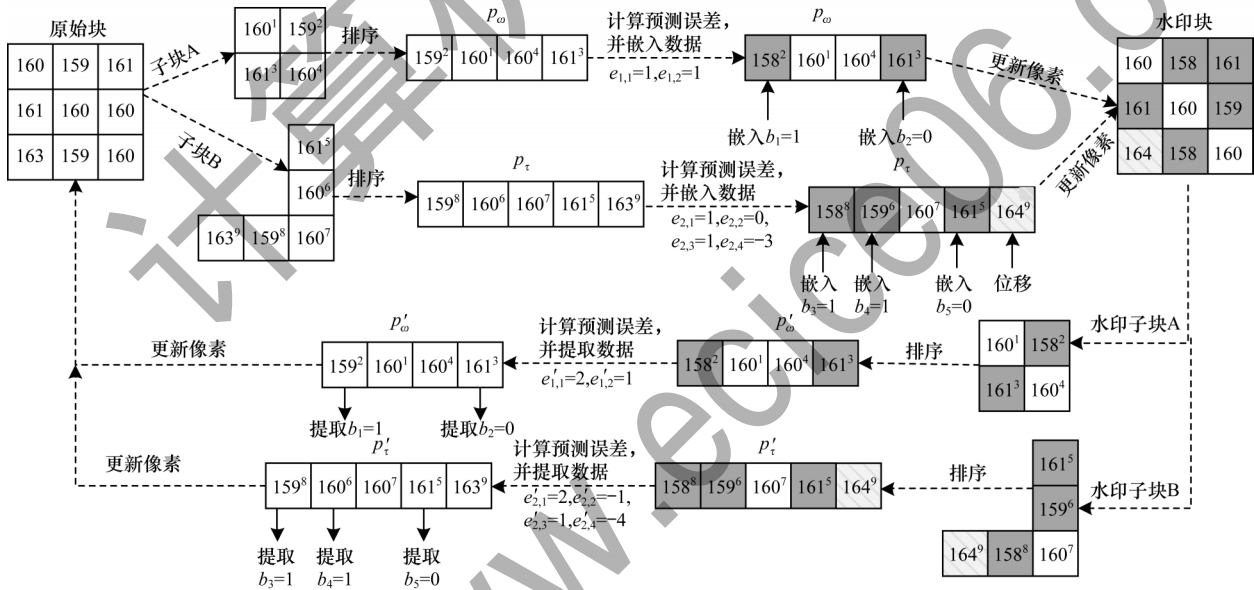


图6 本文算法的数据嵌入与提取过程

Fig.6 Data embedding and extraction process of the proposed algorithm

数据提取过程与数据嵌入过程相同, 首先对水印块进行分块, 得到水印像素序列 p'_{ω} 和 p'_{τ} , 然后分别计算这两个水印序列的预测误差, 对于 p'_{ω} , 计算误差 $e'_{1,1}$ 和 $e'_{1,2}$, 对于 p'_{τ} , 计算误差 $e'_{2,1}, e'_{2,2}, e'_{2,3}$ 和 $e'_{2,4}$, 根据式(6)提取出 b_1, b_2, b_3, b_4 和 b_5 , 根据式(7)和式(8)恢复出原始像素 p_2, p_3, p_8, p_6, p_5 和 p_9 , 最后将所有恢复的像素更新到水印块的相对位置上, 得到原始块。

2.3 局部复杂度

本文考虑到自然图像中包含大量粗糙块, 如果对这些粗糙块进行数据的嵌入操作, 将会产生大量的移位像素, 从而降低图像的嵌入性能。在

文献[17]中, 一个块的局部复杂度定义为当前块的次大像素值与次小像素值的差值。然而, 当分块较小时, 参与排序的像素过少, 难以反映灰度值的变化趋势, 因此, 文献[17]计算的局部复杂度不准确。

由于自然图像的相邻像素呈高度相关关系, 因此可以推断相邻图像块的复杂度也具有高度相关关系。一个光滑块周围的图像块也光滑, 一个粗糙块周围的图像块必定粗糙。本文定义一个图像块的上下文像素为其所有相邻块中距离该块最近的像素。图像块的上下文像素集 C 如式(9)所示:

$$C = \{p_{i-1,j+m}, p_{i+3,j+m}, p_{i+n,j-1}, p_{i+n,j+3} \mid m = -1, 0, 1, 2, 3; n = 0, 1, 2\} \quad (9)$$

上下文像素集 C 的方差被定义为图像块的局部复杂度 N_{NL} , 如式 (10) 所示:

$$N_{NL} = \sqrt{\frac{\sum_{m=-1}^3 \{(p_{i-1,j+m} - \bar{p})^2 + (p_{i+3,j+m} - \bar{p})^2\} + \sum_{n=0}^2 \{(p_{i+n,j-1} - \bar{p})^2 + (p_{i+n,j+3} - \bar{p})^2\}}{16}} \quad (10)$$

其中: \bar{p} 为像素集 C 内像素的平均值。

一个 3×3 图像块的上下文像素示意图如图 7 所示。

$p_{i-1,j-1}$	$p_{i-1,j}$	$p_{i-1,j+1}$	$p_{i-1,j+2}$	$p_{i-1,j+3}$
$p_{i,j-1}$	$p_{i,j}$	$p_{i,j+1}$	$p_{i,j+2}$	$p_{i,j+3}$
$p_{i+1,j-1}$	$p_{i+1,j}$	$p_{i+1,j+1}$	$p_{i+1,j+2}$	$p_{i+1,j+3}$
$p_{i+2,j-1}$	$p_{i+2,j}$	$p_{i+2,j+1}$	$p_{i+2,j+2}$	$p_{i+2,j+3}$
$p_{i+3,j-1}$	$p_{i+3,j}$	$p_{i+3,j+1}$	$p_{i+3,j+2}$	$p_{i+3,j+3}$

图 7 3×3 图像块的上下文像素

Fig.7 Context pixel of 3×3 image block

如果一个图像块的 N_{NL} 小于给定的阈值 T , 则是平滑块。本文按照 2.2 节的方式对平滑块进行块再分, 以嵌入数据。如果一个图像块的 N_{NL} 大于等于给定的阈值 T , 则为复杂块, 对于复杂块不做任何处理。

3 算法流程

3.1 嵌入算法

嵌入算法主要分为以下 3 个步骤:

1) 图像分块与位置地图的构建。首先将原始图像中除第一行像素以外的所有像素分为 3×3 不重叠的块 $\{X_1, X_2, \dots, X_N\}$, 其中, N 为块的总数。如果一个像素为 255 或者 0, 则在嵌入过程中有可能被修改为 256 或者 -1, 会造成上溢或者下溢。为解决该问题, 若 X_i 内含有像素 255 或者 0, 将其标记为 $L_{LM}(i)=1$; 否则标记为 $L_{LM}(i)=0$ 。最后将所有的 L_{LM} , 即图像的位置地图压缩成一个长度为 L_{CLM} 的比特流串。

2) 秘密数据嵌入。在执行数据嵌入的过程中, 嵌入算法只对所有 $L_{LM}(i)=0$ 的图像块进行嵌入。对于每个 $L_{LM}(i)=0$ 的图像块 X_i , 首先计算它的局部复杂度 N_{NL} 。如果 $N_{NL} \geq T$, 则为粗糙块, 不做任何处理; 如果 $N_{NL} < T$, 则为平滑块。按照 2.2.1 节提出的方法进行块再分, 根据式 (1) 和式 (5) 计算误差 $e_{1,j}$ 和 $e_{2,j}$, 根据式 (2) 进行误差的扩展或者移位以嵌入秘密数据。当所有的秘密数据都被嵌入完成后, 该步骤将停止, 最后一个嵌入数据的图像块被记为 X_{end} 。

3) 辅助数据和位置地图的嵌入。为实现数据的可逆恢复, 辅助数据和位置地图被嵌入到图像第一行像素的最低有效位 (Least Significant Bit, LSB) 中。辅助数据为压缩位置地图的长度 L_{CLM} 、阈值 T 、块尺寸以及结束位置 X_{end} 。利用这些数据替换图像第一行的 LSB, 并将替换掉的 LSB 压缩成比特流 S_{LSB} , 最后按照步骤 2 将其嵌入到图像的剩余区域中, 即

$\{X_{end}, X_{end+1}, \dots, X_N\}$ 。

3.2 提取算法

提取算法主要分为以下 3 个步骤:

1) 辅助数据与位置地图的提取。在解码端, 提取算法读取水印图像第一行像素的 LSB 位, 提取 L_{CLM} 、阈值 T 、块尺寸、结束位置 X_{end} 以及压缩的位置地图, 并将其解压得到位置地图。

2) 数据提取。首先将水印图像除第一行以外的所有像素分为 3×3 的块, 然后查找每个块的标记 $L_{LM}(i)$, 计算 $L_{LM}(i)=0$ 的水印块复杂度 N_{NL} , 最后将满足 $L_{LM}(i)=0$ 并且 $N_{NL} < T$ 的水印块按照 2.2 节的方法进行再分块, 计算水印误差 $e'_{1,j}$ 和 $e'_{2,j}$, 并根据式 (6) 提取出秘密数据和压缩的比特流 S_{LSB} 。

3) 图像恢复。首先将提取的秘密数据按照 2.2.2 节的方法恢复出图像的原始块 $\{X_1, X_2, \dots, X_N\}$, 然后解压 S_{LSB} , 将其替换掉图像第一行像素的 LSB 位, 最后恢复出整个图像。

4 实验结果与分析

本文利用 MATLAB R2018a 软件在 6 幅 512×512 像素的灰度图像上, 对比现有的基于像素值排序的可逆数据隐藏算法与本文算法的性能。6 张测试图如图 8 所示, 这些图像都是从 USC-SIPI 数据库中下载获得, 在嵌入过程中数据 b 是随机生成的 0, 1 序列。

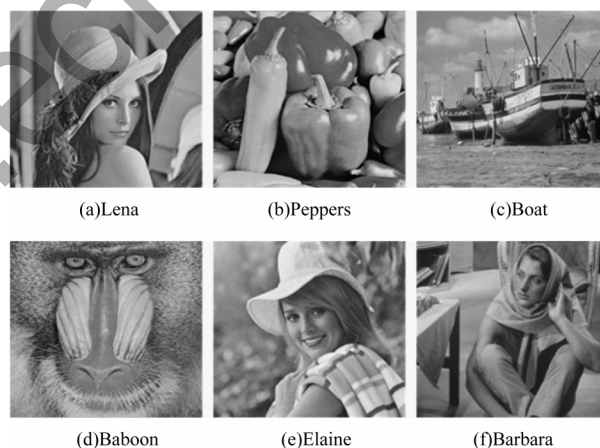


图 8 本文实验 6 张测试图

Fig.8 Six test images of the proposed experiment

本文在 12 种分块模式下对 Lena 图和 Baboon 图进行峰值信噪比 (Peak Signal to Noise Ratio, PSNR) 对比, 结果如图 9 所示。从图 9 可以看出, 当采用分块模式 1 时, 图像的 PSNR 性能最优, 因此, 本文算法的最优模式为模式 1。

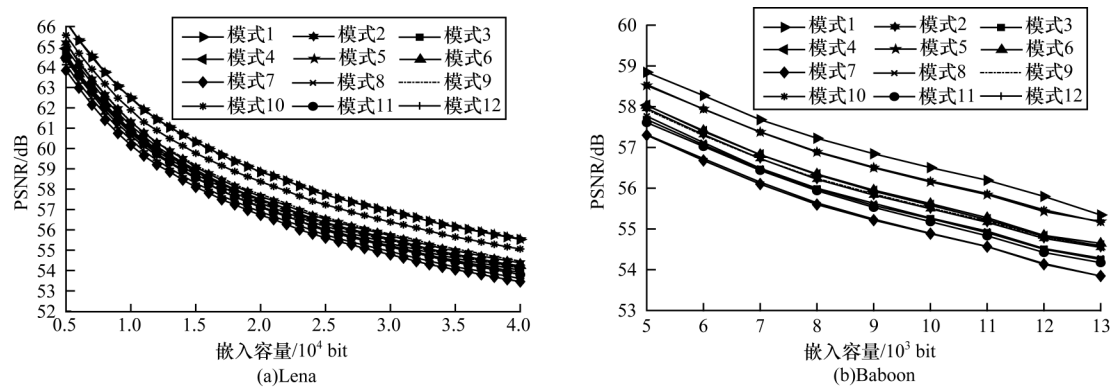


图9 不同分块模式下的PSNR对比

Fig.9 Comparison of PSNR in different block modes

在最优模式下不同算法的PSNR对比如图10所示。从图10可以看出,在大多数情况下,本文算法的PSNR都最优,并且随着嵌入容量的增大,PSNR的增益更明显。因此,嵌入容量越大,本文算法的性能越优。

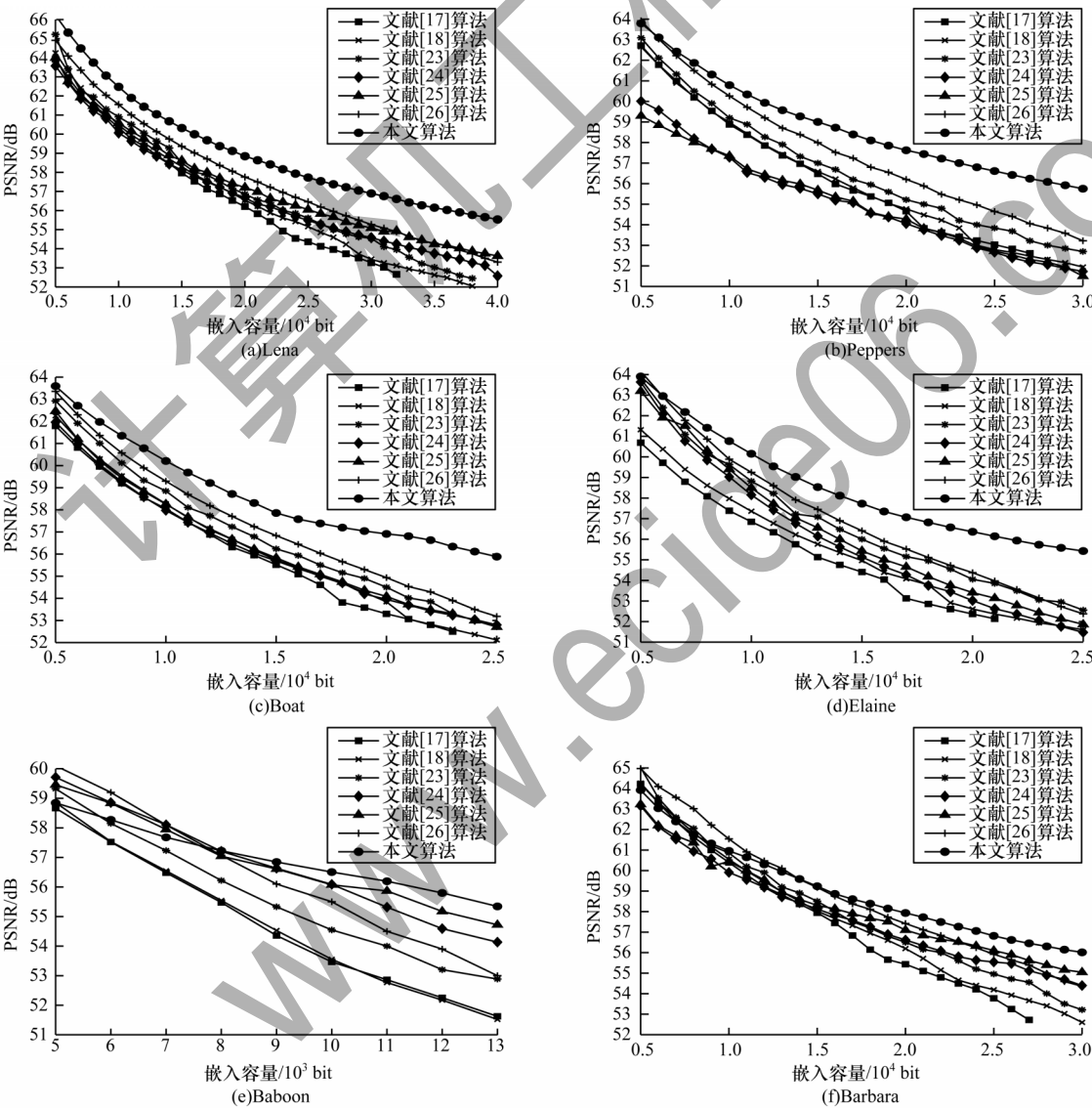


图10 不同算法的PSNR对比

Fig.10 PSNR comparison among different algorithms

从图10可以看出,当Baboon图的嵌入容量小于8 000 bit以及Barbara图的嵌入容量小于12 000 bit时,本文算法的PSNR不是最高,其原因主要有两个:1)本文算法未根据嵌入容量自适应选择图像块大小,即不能在嵌入容量较低时,采用更大的5×5图像块进行再分块,而对于这类复杂图像,相邻像

素联系不紧密,使得对平滑块的选择更为精确,需要更多的上下文像素来计算局部复杂度,因此,本文算法的 PSNR 不是最高;2)以上对比的其他算法在实验过程中都通过穷举搜索法选择最优参数,尽管这种方式可以提升 PSNR,但是也会增大计算复杂度。

表1 当嵌入容量为 1 000 bit时不同算法的 PSNR 对比

Table 1 PSNR comparison among different algorithms when embedding capacity is 1 000 bit							dB
算法	Lena 图像的 PSNR	Peppers 图像的 PSNR	Boat 图像的 PSNR	Elaine 图像的 PSNR	Baboon 图像的 PSNR	Barbara 图像的 PSNR	平均 PSNR
文献[17] 算法	60.49	58.99	58.12	56.99	53.32	60.32	58.04
文献[18] 算法	60.52	59.01	58.20	57.26	53.36	60.37	58.12
文献[23] 算法	60.95	59.11	58.92	58.89	54.59	60.92	58.90
文献[24] 算法	59.92	56.93	57.67	58.33	55.99	59.95	58.13
文献[25] 算法	60.59	57.05	58.28	58.10	56.05	60.54	58.44
文献[26] 算法	61.53	59.97	59.32	59.18	55.19	61.35	59.42
本文算法	62.37	60.88	60.07	60.03	56.41	60.98	60.12

不同算法的最大嵌入容量对比如图 11 所示。分块越小,最大嵌入容量就越大。在文献[17]中,当采用 2×2 分块时,图像的嵌入容量达到了最大值。本文算法是在 3×3 固定大小的分块上实现的。从图 11 可以看出,本文算法的最大嵌入容量在某些图像上是最大的,其原因对于 一个 3×3 大小的图像块,最多可以嵌入 6 bit 数据,充分地利用了图像块内的每一个像素。不同算法的计算复杂度对比如表 2 所示。

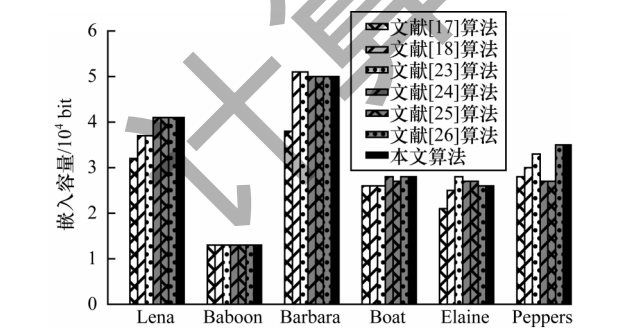


图 11 不同算法的最大嵌入容量对比

Fig.11 Maximum embedding capacity comparison among different algorithms

表 2 不同算法的计算复杂度对比

Table 2 Computational complexity comparison among different algorithms	
算法	计算复杂度
文献[17]算法	$O(16N_L)$
文献[18]算法	$O(16N_p)$
文献[23]算法	$O(9m \times N_w)$
本文算法	$O(N)$

文献[17]通过穷举搜索出三个最优参数,即块尺寸 r_L 、 c_L 和阈值 T_L ,其中 $r_L \in \{2, 3, 4, 5\}$, $c_L \in \{2, 3, 4, 5\}$, T_L 有 N_L 个候选数,因此,文献[17]的计算复杂度近似为 $O(16N_L)$ 。同理,文献[18]的计算复杂度近似为 $O(16N_p)$,其中,16 和 N_p 分别为最优尺寸 $r_p \times c_p$ 和最优阈值 T_p 的候选数。文献[23]需要穷举搜索 4 个最优参

当嵌入容量为 1 000 bit 时,不同算法在 6 张图上的 PSNR 对比如表 1 所示。从表 1 可以看出,相比其他算法,本文算法的平均 PSNR 增益分别为 1.22 dB、2.00 dB、2.08 dB、1.99 dB、1.68 dB、0.70 dB。这表明本文算法在低嵌入容量下能够有效提升嵌入数据后的图像质量。

数,即 vT 、 T_w 、 r_w 和 c_w ,其中, $r_w \in \{3, 4, 5\}$, $c_w \in \{3, 4, 5\}$, T_w 有 m 个候选数, $m = \lfloor (r_w \times c_w - 1)/2 \rfloor$, vT 有 N_w 个候选数。文献[23]的计算复杂度近似为 $O(9m \times N_w)$ 。而本文算法只需要确定一个参数,即阈值 T ,因此计算复杂度仅为 $O(N)$,其中, N 为阈值 T 的候选数。

本文根据不同的嵌入容量,采用不同的最优阈值 T^* 改进本文算法。本文改进的算法与文献[17-18]、文献[23-26]算法在 Barbara 和 Baboon 的 PSNR 对比如图 12 所示。从图 12 可以看出,无论嵌入容量是多少,本文改进算法的 PSNR 都远大于现有算法。

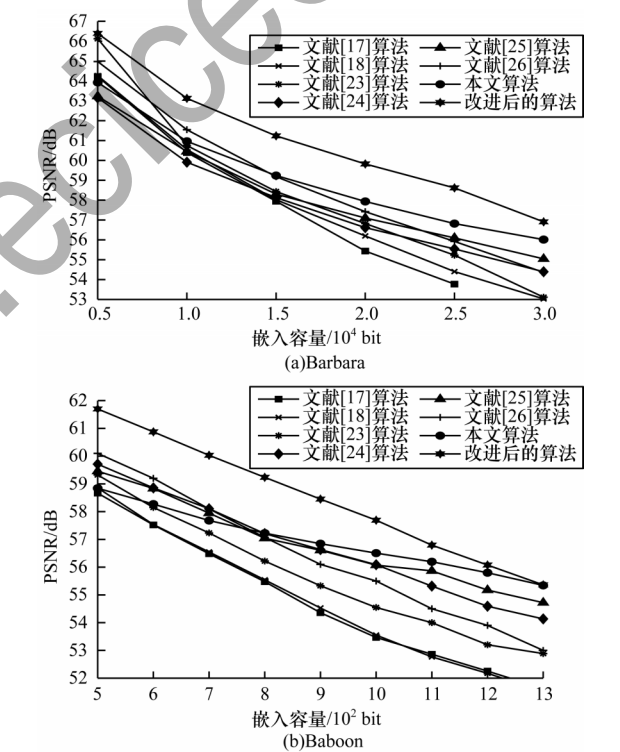


图 12 本文改进算法与其他算法的 PSNR 对比

Fig.12 PSNR comparison among the proposed improved algorithm and other algorithms

5 结束语

本文提出基于像素值排序的可逆数据隐藏算法。通过计算每一个图像块的局部复杂度, 并对比局部复杂度与阈值的大小, 将局部复杂度小于阈值的图像块进行分块, 同时设计12种分块模式, 并通过实验测试不同分块模式下的嵌入性能, 选择能够得到最优嵌入性能的模式进行数据嵌入。此外, 通过改进本文算法, 根据嵌入容量自适应地选择最优阈值 T^* , 以提高嵌入性能。实验结果表明, 本文算法能够充分利用图像块内的每个像素, 具有较优的嵌入性能。后续将在 4×4 和 6×6 图像块上设计分块策略, 进一步提升图像块的嵌入性能。

参考文献

- [1] SHI Y Q, LI X L, ZHANG X P, et al. Reversible data hiding: advances in the past two decades[J]. IEEE Access, 2016, 4: 3210-3237.
- [2] SINGH L, SINGH A K, SINGH P K. Secure data hiding techniques: a survey[J]. Multimedia Tools and Applications, 2020, 79(23/24): 15901-15921.
- [3] COATRIEUX G, GUILLOU C L, CAUVIN J M, et al. Reversible watermarking for knowledge digest embedding and reliability control in medical images[J]. IEEE Transactions on Information Technology in Biomedicine, 2009, 13(2): 158-165.
- [4] 张鸿超. 用于医学图像篡改检测的可逆信息隐藏方法[J]. 现代信息科技, 2020, 4(3): 55-58, 64.
ZHANG H C. Reversible data hiding method for enhancing medical image contrast and tampering detection[J]. Modern Information Technology, 2020, 4(3): 55-58, 64. (in Chinese)
- [5] 姜飞, 李军, 钱振兴. 可提高嵌入容量的密文医学图像可逆数据隐藏方法[J]. 应用科学学报, 2015, 33(6): 644-654.
JIANG F, LI J, QIAN Z X. Reversible data hiding in encrypted medical images with improved embedding capacity[J]. Journal of Applied Sciences, 2015, 33(6): 644-654. (in Chinese)
- [6] MENENDEZ-ORTIZ A, FEREGRINO-URIBE C, HASIMOTO-BELTRAN R, et al. A survey on reversible watermarking for multimedia content: a robustness overview[J]. IEEE Access, 2019, 7: 132662-132681.
- [7] KHAN A, SIDDIQA A, MUNIB S, et al. A recent survey of reversible watermarking techniques[J]. Information Sciences, 2014, 279: 251-272.
- [8] RATHIKA R, KUMARESAN S. Survey on reversible data hiding techniques[C]//Proceedings of the 3rd International Conference on Advanced Computing and Communication Systems. Washington D. C., USA: IEEE Press, 2016: 1-4.
- [9] SINGH P, CHADHA R S. A survey of digital watermarking techniques, applications and attacks[J]. Computer Science, 2013, 2(9): 165-175.
- [10] FRIDRICH J, GOLJAN M, DU R. Invertible authentication[C]//Proceedings of Security and Watermarking of Multimedia Contents III. San Jose, USA: [s. n.], 2001: 197-208.
- [11] CELIK M U, SHARMA G, TEKALP A M. Lossless watermarking for image authentication: a new framework and an implementation[J]. IEEE Transactions on Image Processing, 2006, 15(4): 1042-1049.
- [12] TIAN J. Reversible data embedding using a difference expansion[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(8): 890-896.
- [13] THODI D M, RODRIGUEZ J J. Prediction-error based reversible watermarking[C]//Proceedings of International Conference on Image Processing. Washington D. C., USA: IEEE Press, 2004: 1549-1552.
- [14] SACHNEV V, KIM H J, NAM J, et al. Reversible watermarking algorithm using sorting and prediction[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2009, 19(7): 989-999.
- [15] OU B, LI X L, ZHAO Y, et al. Pairwise prediction-error expansion for efficient reversible data hiding[J]. IEEE Transactions on Image Processing, 2013, 22(12): 5010-5021.
- [16] 黄志强, 王美清. 基于邻域预测差值直方图平移的可逆信息隐藏[J]. 计算机工程, 2014, 40(4): 116-119.
HUANG Z Q, WANG M Q. Reversible information hiding based on neighborhood prediction difference histogram shifting[J]. Computer Engineering, 2014, 40(4): 116-119. (in Chinese)
- [17] LI X L, LI J, LI B, et al. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion[J]. Signal Processing, 2013, 93(1): 198-205.
- [18] PENG F, LI X L, YANG B. Improved PVO-based reversible data hiding[J]. Digital Signal Processing, 2014, 25: 255-265.
- [19] OU B, LI X L, ZHAO Y, et al. Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion[J]. Signal Processing: Image Communication, 2014, 29(7): 760-772.
- [20] QU X C, KIM H J. Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding[J]. Signal Processing, 2015, 111: 249-260.
- [21] OU B, LI X L, WANG J W. High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion[J]. Journal of Visual Communication and Image Representation, 2016, 39: 12-23.
- [22] 张敏倩, 孔咏骏, 彭菓玉, 等. 基于像素值排序的鲁棒可逆信息隐藏方法[J]. 东南大学学报(自然科学版), 2019, 49(5): 873-882.
ZHANG M Q, KONG Y J, PENG G Y, et al. Robust reversible data hiding method based on pixel value ordering[J]. Journal of Southeast University (Natural Science Edition), 2019, 49(5): 873-882. (in Chinese)
- [23] WENG S W, SHI Y Q, HONG W, et al. Dynamic improved pixel value ordering reversible data hiding[J]. Information Sciences, 2019, 489: 136-154.
- [24] KIM S, QU X C, 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.
- [25] OU B, LI X L, ZHANG W M, 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.
- [26] ZHANG T, LI X L, QI W F, et al. Location-based PVO and adaptive pairwise modification for efficient reversible data hiding[J]. IEEE Transactions on Information Forensics and Security, 2020, 15: 2306-2319.