

基于通道线性回归与二维多级预测误差直方图 平移的彩色图像可逆数据隐藏

葛昊立, 陈 妍, 王建军

(复旦大学 电子工程系, 上海 200433)

摘 要: 提出使用通道线性回归来估计当前像素并生成预测误差直方图, 然后按照像素的局部空间复杂性使用二维直方图平移嵌入隐秘信息. 在接收端, 按照和嵌入时完全相反的顺序操作图像, 可以在提取出嵌入信息的同时恢复原图像. 实验结果表明, 与现有的方法相比, 相同嵌入容量下本文的算法具有更高的峰值信噪比.

关键词: 可逆数据隐藏; 彩色图像; 通道线性回归; 局部色度; 预测误差直方图

中图分类号: TN918

文献标识码: A

文章编号: 1000-7180(2018)05-0061-08

Reversible Data Hiding in Color Image Based on Channel Linear Regression and Two-dimensional Multiple Prediction Error Histogram Shifting

GE Hao-li, CHEN Yan, WANG Jian-jun

(Department of Electronic Engineering, Fudan University, Shanghai 200433, China)

Abstract: we propose to generate prediction error histograms using channel linear regression. Then, we embed secret bits using two-dimensional histogram shifting in the order of local space complexity of pixels. At the receiver side, similar processes are applied and additional data is extracted in reverse order to ensure reversibility. The experimental results show that the proposed method achieves better visual quality in terms of PSNR under the same embedding capacity compared to existing methods.

Key words: reversible data hiding; color image; channel linear regression; local chromaticity; prediction error histogram

1 引言

可逆数据隐藏(Reversible Data Hiding, RDH)是一种将信息隐秘地嵌入到对失真零容忍的数字掩体(如图像、视频、音频)中的技术. 该技术保证提取嵌入信息的同时能够完全无误地恢复原始数字掩体, 在军事和医疗领域, 数字图像的失真是不可接受的, 所以研究可逆数据隐藏技术来保证原始掩体和嵌入信息的无失真地恢复是信息安全研究领域^[1-4]的一个重要分支.

人们对以彩色图像为掩体的 RDH 算法进行了

研究. Tseng^[5]等提出了基于聚类的直方图修改 RDH 算法来嵌入隐秘信息. Li^[6]等使用不同通道间相似的边结构信息来提高预测精度, 进而提高嵌入容量和减少图像失真. Ou^[7]着重考虑了通道间的差异性, 提出了自适应的嵌入容量分割的算法. 最近, Yao^[8]利用引导滤波的思想, 提出将参考通道作为引导图来对当前通道像素进行估计的算法.

考虑到彩色图像的三个通道间的相关性和差异性, 本文提出了一种基于通道线性回归与二维多级预测误差直方图平移的彩色图像可逆数据隐藏算法.

2 基于通道线性回归与二维多级预测误差直方图平移的 RDH 算法

如图 1 所示,本文在嵌入过程中将图像每个通道按照棋盘格的模式分割成两个像素集合,对于通道 C ,用 C_1 和 C_2 分别表示灰格像素集与白格像素集,其中 $C \in \{R, G, B\}$.

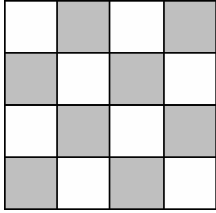


图 1 白块与灰块像素分割

该方法的主要思想是:(1)根据彩色图像三个通道各自的预测误差直方图特性确定嵌入容量的分配和嵌入顺序;(2)根据上一步确定的容量分割和嵌入顺序对每个通道进行嵌入,嵌入过程先在所有通道的灰格像素集上进行,然后在所有通道白格像素集上进行;(3)嵌入过程中,使用通道线性回归预测当前通道,生成多级预测误差直方图,搜索最优直方图分组,最后利用二维直方图平移实现嵌入;(4)在接收端,按照和嵌入过程完全相反的顺序依次在所有通道的白格和灰格像素上提取信息.借助于包含于标记图像中的辅助信息,接收者不仅可以提取嵌入的秘密信息,而且可以准确无误地恢复原彩色图像.下面对提出的算法进行详细叙述.

2.1 确定嵌入顺序与分配嵌入容量

设彩色图像每个通道的大小为 $P \times Q$,对每一个通道,利用其他两个通道对其进行预测,计算预测误差,具体的预测算法见 2.5 节.以 $I_C(i, j)$ 表示通道 C 在像素 (i, j) 处的值, $\hat{I}_C(i, j)$ 为对通道 C 在像素 (i, j) 处的预测值,其中 $C \in \{R, G, B\}$. 则相应的预测误差为

$$e_C(i, j) = I_C(i, j) - \hat{I}_C(i, j) \quad (1)$$

分别统计三个通道的预测误差直方图:

$$h_C(e) = \# \{1 \leq i \leq P, 1 \leq j \leq Q: e_C(i, j) = e\} \quad (2)$$

式中, $\#\{\cdot\}$ 代表集合的基数.若每个通道都是 8 位灰度图像,则 $h_C(e)$ 中 e 的取值范围为 $[-255, 255]$. 由于图像本身的空间相关性和通道间相关性,预测误差直方图的分布通常类似于中心在 $e = 0$ 附近的拉普拉斯分布,在对称中心两侧衰减较快,如图 2 所示.

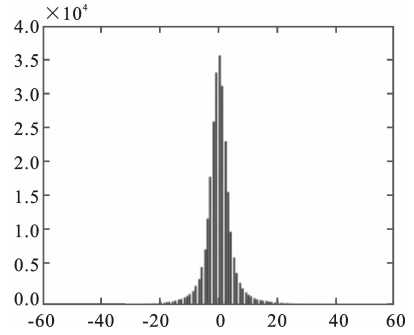


图 2 Lena 红色通道预测误差直方图

而预测误差直方图越尖锐,则表明预测越准确,相应的嵌入容量也越高,嵌入失真也越小.因此,本文算法给预测误差直方图尖锐的通道分配更多的嵌入容量.这里,我们使用预测误差直方图中最高的多个峰值之和来表征预测误差直方图的尖锐程度,该值越大,表明预测误差直方图越尖锐.将通道 C 的直方图 $h_C(e)$ 按照降序排序,取最高的 7 个峰值求和,得到预测误差峰值和 sum_peh_C . 将三个通道的 sum_peh_C 按照降序进行排序,根据该排序结果确定嵌入的顺序.嵌入容量则按照下式进行分配:

$$\begin{cases} EC_R = \left\lfloor \frac{\text{sum_peh}_R}{\text{sum_peh}_R + \text{sum_peh}_G + \text{sum_peh}_B} \times EC + 0.5 \right\rfloor \\ EC_G = \left\lfloor \frac{\text{sum_peh}_G}{\text{sum_peh}_R + \text{sum_peh}_G + \text{sum_peh}_B} \times EC + 0.5 \right\rfloor \\ EC_B = EC - EC_R - EC_G \end{cases} \quad (3)$$

式中, $\lfloor \cdot \rfloor$ 为向下取整函数.对通道 C ,将 $h_C(e)$ 分为 $h_{C_1}(e)$ 与 $h_{C_2}(e)$ 两部分,它们分别代表像素集 C_1 与 C_2 上的预测误差直方图.类似的,分别取 $h_{C_1}(e)$ 和 $h_{C_2}(e)$ 最高的 7 个峰值求和,得到预测误差峰值和 sum_peh_{C_1} 与 sum_peh_{C_2} ,则 C_1 和 C_2 的嵌入容量分别为

$$\begin{cases} EC_{C_1} = \left\lfloor \frac{\text{sum_peh}_{C_1}}{\text{sum_peh}_{C_1} + \text{sum_peh}_{C_2}} \times EC_C + 0.5 \right\rfloor \\ EC_{C_2} = EC_C - EC_{C_1} \end{cases} \quad (C \in \{R, G, B\}) \quad (4)$$

2.2 通道像素嵌入与提取顺序

以按照 RGB 的顺序嵌入为例,先所有的灰格像素上嵌入,然后在所有的白格像素上嵌入,则相应的嵌入和提取顺序如图 3 所示.

当按照 $R_1, G_1, B_1, R_2, G_2, B_2$ 的顺序嵌入信息时,则提取信息的时候要相应地按照 $B_2', G_2', R_2', B_1', G_1', R_1'$ 的顺序反向提取.

每次嵌入信息时,都使用其他五个像素集对当

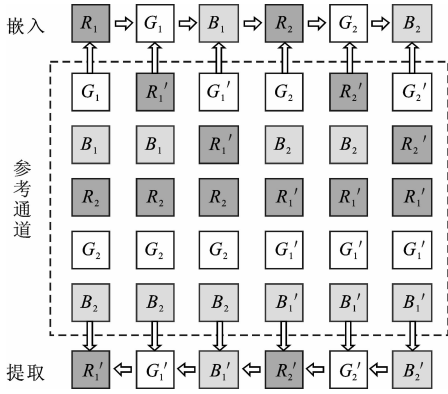


图3 嵌入与提取顺序

前像素集进行预测,进而生成预测误差直方图。当嵌入隐秘信息之后,以 C_1' 和 C_2' 分别表示通道 C 包含隐秘信息的灰格和白格像素集。例如, R_1 像素集嵌入完成后,其变成 R_1' ,当下一步对 G_1 像素集进行嵌入时,为了保证可逆性,必须使用 R_1' 、 B_1 、 R_2 、 G_2 、 B_2 对其进行嵌入。之后四个像素集的嵌入类似。

每个像素集的嵌入和提取信息的流程分别如图4和图5所示。下面将详细描述嵌入和提取流程。

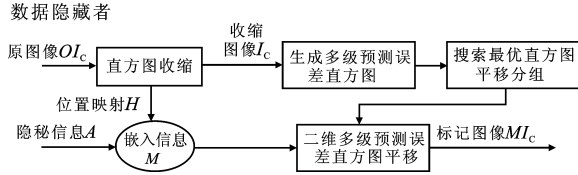


图4 嵌入流程

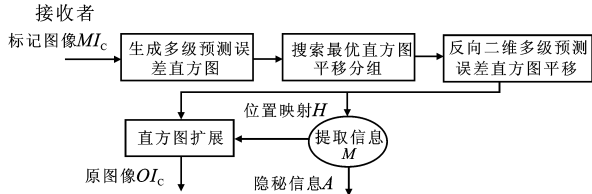


图5 提取流程

2.3 直方图收缩

将嵌入流程中通道 C 的输入原图像用 OI_C 表示。一般而言, OI_C 为 8 位灰度图,其每个像素的取值范围为 $[0, 255]$,因此,数据隐藏者需要将 OI_C 的直方图进行收缩以避免可能的下溢或者上溢。按照从左到右、从上到下的顺序遍历 OI_C 的每个像素,并根据式(5)修改像素值。

$$I_C(i, j) = \begin{cases} 254 & \text{if } OI_C(i, j) = 255 \\ 1 & \text{if } OI_C(i, j) = 0 \\ OI_C(i, j) & \text{otherwise} \end{cases} \quad (5)$$

同时,算法记录一个位置映射 H ,用以区分原始值为 1 或 254 的像素和那些值被修改成 1 或 254

的像素。初始化位置映射 H 为一个 $1 \times N(N = P \times Q)$ 的零向量,如果一个像素 $OI_C(i, j) \in \{0, 255\}$,则将当前 H 向量中索引最低的 0 更新为 1。这样可以得到收缩后的图像 I_C 和位置映射 H 。为了减小 H 的长度,可以使用算术编码或者游程编码对进行压缩 H 。然后,将压缩后的 H 和隐秘信息 A (一个熵为 1 的二进制序列) 串接得到嵌入信息 M ,即 $M = H + A$ 。

2.4 通道线性回归

本节首先分析彩色图像的成像机制,然后利用这个成像机制推导出通道间局部色度相似性。最后,利用通道线性回归和任意两个颜色通道对第三个颜色通道进行预测估计。

假定彩色图像中拍摄的物体表面具有朗伯漫反射的特性,则在物体表面的彩色图像成像可以表示为:

$$I_k(x) = \int Rf(x, \lambda) L(x, \lambda) S_k(\lambda) d\lambda \quad (6)$$

$I_k(x)$ 是相机颜色传感器在像素位置 x 处的输出响应,其中 $k \in \{1, 2, 3\}$ 分别表示对应于 RGB 三种颜色的传感器。 $Rf(x, \lambda)$ 代表物体表面在像素位置 x 处对波长为 λ (单位为纳米) 的光的反射响应。 $S_k(\lambda)$ 表示第 k 个传感器对于波长为 λ 的光的响应特性。 $L(x, \lambda)$ 是彩色图像拍摄时的环境光照,波长 λ 和像素位置 x 的函数。环境光照可能随像素位置 x 的变化而变化。

假定传感器响应特性可以使用 δ 函数来表示,那么 $S_k(\lambda) = \delta(\lambda - \lambda_k)$,进而可以将式(6)简化得到:

$$I_k(x) = \int Rf(x, \lambda) L(x, \lambda) \delta(\lambda - \lambda_k) d\lambda = Rf(x, \lambda_k) L(x, \lambda_k) \quad (7)$$

我们不关心 λ_k 具体的数值为多少,只关心其对应的颜色通道编号,所以可以将 λ_k 省略并且把 k 移到下标位置。简而言之,即 $I_k(x) = Rf_k(x) L_k(x)$ ($k = 1, 2, 3$)。将该式写成 RGB 三个颜色通道,即将像素的值用物体表面响应和环境光照的乘积在长、中、短波三个波段上的分量表示:

$$\begin{cases} R(x) = Rf_1(x) L_1(x) \\ G(x) = Rf_2(x) L_2(x) \\ B(x) = Rf_3(x) L_3(x) \end{cases} \quad (8)$$

将环境光照向量 $L(x) = (L_1(x), L_2(x), L_3(x))$ 表示成光照强度 $\|L(x)\|$ 乘以光照色度 $\alpha(x)$ (也是光照方向):

$$\begin{aligned} L(x) &= \|L(x)\| \alpha(x) \\ \alpha(x) &= (\alpha_1(x), \alpha_2(x), \alpha_3(x)) \end{aligned} \quad (9)$$

式中, $\|L(x)\| = \sqrt{L_1^2(x) + L_2^2(x) + L_3^2(x)}$, $\alpha_k(x) = \frac{L_k(x)}{\|L(x)\|}$. 由于成像过程受到物体 3D 几何表面的影响, 图像局部的光照强度 $\|L(x)\|$ 可能随着空间位置而变化. 为了消除这种局部光照强度带来的影响, 可以考虑将像素从 RGB 颜色空间转换到 rgb 色度空间:

$$\begin{cases} r(x) = \frac{R(x)}{R(x) + G(x) + B(x)} \\ g(x) = \frac{G(x)}{R(x) + G(x) + B(x)} \\ b(x) = \frac{B(x)}{R(x) + G(x) + B(x)} \end{cases} \quad (10)$$

假定物体表面反射 $Rf_k(x)$ 和光照色度 $\alpha(x)$ 具有局部连续一致性. 更准确的讲, 对图像上的任意一个像素, 在其邻域范围内总存在另一个像素, 其物体表面反射和光照色度与该像素相同. 比如, 对两个相邻的像素 x_1 和 x_2 , 如果 $Rf_k(x_1) = Rf_k(x_2)$, $\alpha(x_1) = \alpha(x_2)$, 那么像素 x_1 的色度就和其邻域像素 x_2 的色度相等, 即 $r(x_1) = r(x_2)$, $g(x_1) = g(x_2)$, $b(x_1) = b(x_2)$.

从以上的相机成像机制和推导出, 本文提出了通道线性回归的算法来对嵌入通道进行预测估计. 这里, 我们以 R 通道和 B 通道对 G 通道插值为例来说明估计过程.

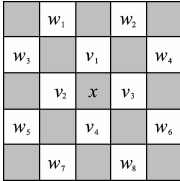


图 6 像素 x 的邻域

当前要嵌入的是 R_1 像素集, 用来估计像素 x 的邻域如图 6 所示. 已知的是中心像素 x 绿色分量与蓝色分量值为 $G(x)$ 、 $B(x)$ 以及它邻域像素的颜色值 $R(v_i)$ 、 $G(v_i)$ 、 $B(v_i)$ 、 $R(w_j)$ 、 $G(w_j)$ 、 $B(w_j)$, 其中 $i = 1, 2, 3, 4, j = 1, 2, \dots, 8$. 目标就是要估计 $R(x)$ 的值. 对于真实世界中拍摄得到的数字图像, 尤其是对于靠近图像中边的像素, 其像素色度很有可能不完全等于其邻域内所有像素的色度. 因此, 有必要对邻域像素进行筛选. 这里引入 gb 色度空间的角度差来刻画像素色度之间的相似程度:

$$\theta(y, x) = \arccos\left(\frac{g(y)g(x) + b(y)b(x)}{\sqrt{g^2(y) + b^2(x)} \sqrt{g^2(x) + b^2(x)}}\right)$$

$$= \arccos\left(\frac{G(y)G(x) + B(y)B(x)}{\sqrt{G^2(y) + B^2(y)} \sqrt{G^2(x) + B^2(x)}}\right) \quad (11)$$

式中, $y \in \{v_i, w_j\} (i = 1, 2, 3, 4, j = 1, 2, \dots, 8)$. 角度越小, 就说明像素间的色度越接近. 根据之前图像表面反射和光照色度局部连续一致性的假设, 总可以在邻域内找到若干像素来估计当前中心像素. 设 $\theta(y, x)$ 中的最小值为下界 $\theta(x)_{\text{low}}$, 并且设 $\theta(x)_{\text{low}} + 3^\circ$ 为上界 $\theta(x)_{\text{high}}$. 如果某一像素和中心像素间的角度差 $\theta(y, x) \in [\theta(x)_{\text{low}}, \theta(x)_{\text{high}}]$, 则该像素成为候选像素.

此时, 得到像素 x 筛选过后的邻域像素, 其包含的像素数目为 U . 我们认为像素三个颜色分量具有线性关系, 如式(12)所示:

$$R(y) = pG(y) + qB(y) + r \quad (12)$$

而在邻域内, 认为参数 p, q, r 的值不变. 因此, 要对像素 x 的邻域求得最优参数, 定义如下损失函数:

$$\sum_{k=1}^U (\mu G(y_k) + qB(y_k) + r - R(y_k))^2 \quad (13)$$

最小化该损失函数, 求得 p, q, r . 最后, 通过式(14)求得通道线性回归对像素 x 红色分量的预测值:

$$\hat{R}(x) = \mu G(x) + qB(x) + r \quad (14)$$

2.5 二维多级预测误差直方图平移

对当前正在嵌入的像素集, 通过上述的通道线性回归计算出像素集内每个像素的预测值. 与此同时, 利用图 6 所示的邻域, 计算像素 x 在通道 $C (C \in \{R, G, B\})$ 的水平和局部复杂性:

$$\begin{aligned} L_{C,h}(x) = & |C(w_1) - C(w_2)| + |C(w_3) - C(v_1)| \\ & + |C(v_1) - C(w_4)| + |C(w_2) - C(v_3)| \\ & + |C(w_5) - C(v_4)| + |C(v_4) - C(w_6)| \\ & + |C(w_7) - C(w_8)| \end{aligned} \quad (15)$$

$$\begin{aligned} L_{C,v}(x) = & |C(w_3) - C(w_5)| + |C(w_1) - C(w_2)| \\ & + |C(w_2) - C(w_7)| + |C(v_1) - C(v_4)| \\ & + |C(w_2) - C(v_3)| + |C(v_3) - C(w_6)| \\ & + |C(w_4) - C(w_6)| \end{aligned} \quad (16)$$

进一步计算像素 x 的在通道 C 的局部复杂性:

$$L_C(x) = L_{C,h}(x) + L_{C,v}(x) \quad (17)$$

一般将像素 x 的局部空间复杂性表示为其在三个通道局部复杂性之和. 这里, 本文对当前正在嵌入的通道分配更多的权重. 例如, 若当前嵌入像素集为 R_1 或者 R_2 , 则像素 x 的局部空间复杂性表示为

$$L^R(x) = 2L_R(x) + L_G(x) + L_B(x) \quad (18)$$

嵌入其他像素集的时候也是类似.

计算像素 x 的预测误差:

$$e_C(x) = C(x) - \hat{C}(x) \quad (19)$$

式中, $C(x)$ 为像素 x 在通道 C 的分量, $\hat{C}(x)$ 为对像素 x 在通道 C 的线性回归预测值。

设当前嵌入像素集为 C_1 ($C_1 \in \{R_1, G_1, B_1\}$), 包含 J 个像, 则 C_1 中的像素表示为 $\{x_1, x_2, \dots, x_J\}$. 将 C_1 中的像素根据其局部空间复杂性从小到大进行排序, 得到新的像素顺序 $\{x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(J)}\}$. 把排序后的像素划分为 K 组, 前 $K-1$ 组每组包含 $\lceil J/K \rceil$ 个像素, 最后一组包含 $J - (K-1) \times \lceil J/K \rceil$ 个像素. 分别统计这 K 组像素的预测误差直方图:

$$h_k(e) = \# \{x \in C_1 : e_C(x) = e\} \quad (1 \leq k \leq K) \quad (20)$$

传统的一维预测误差直方图平移如图 7 所示。

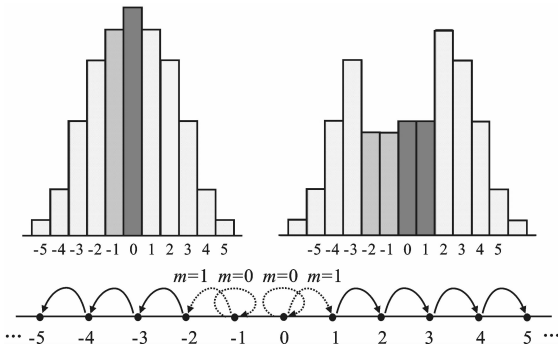


图 7 一维预测误差直方图平移

该嵌入方法导致的图像失真为

$$D = \frac{1}{2} (h(a) + h(b)) + \sum_{p=-\infty}^{a-1} h(p) + \sum_{p=b+1}^{+\infty} h(p) \quad (21)$$

一般而言, 预测误差直方图在 0 附近具有较高的峰值. 当满足一定的嵌入容量要求之后, 我们希望将预测误差直方图平移分组 a 和 b 尽量地向两边扩展以减小嵌入失真. 因此, 在满足给定嵌入容量

$$(\acute{e}_1, \acute{e}_2) = \begin{cases} (e_1, e_2) + (f_k(e_1)m_1, f_k(e_2)m_2) & \text{if } (e_1 = a_k \text{ or } e_1 = b_k) \\ & \text{and } (e_2 = a_k \text{ or } e_2 = b_k) \\ (e_1, e_2) + (f_k(e_1)m, f_k(e_2)m) & \text{if } (e_1 = a_k - 1 \text{ or } e_1 = b_k + 1) \\ & \text{and } (e_2 = a_k - 1 \text{ or } e_2 = b_k + 1) \\ (e_1, e_2) + (f_k(e_1)m, f_k(e_2)) & \text{if } (e_1 = a_k \text{ or } e_1 = b_k) \\ & \text{and } (e_2 = a_k - 1 \text{ or } e_2 = b_k + 1) \\ (e_1, e_2) + (f_k(e_1), f_k(e_2)m) & \text{if } (e_1 = a_k - 1 \text{ or } e_1 = b_k + 1) \\ & \text{and } (e_2 = a_k \text{ or } e_2 = b_k) \end{cases} \quad (25)$$

其中

$$\begin{aligned} f_k(e) &= \begin{cases} -1 & \text{if } e \leq a_k \\ 1 & \text{if } e \geq b_k \end{cases} \\ (e_1, e_2) &= (e(x_s), e(x_{s+1})) \\ (\acute{e}_1, \acute{e}_2) &= (\acute{e}(x_s), \acute{e}(x_{s+1})) \end{aligned} \quad (26)$$

EC(Embedding Capacity) 的前提下, 本文算法搜索这 K 个预测误差直方图 $h_k(e)$ ($1 \leq k \leq K$), 寻找最优的预测误差直方图平移分组 a_k 和 b_k , 即

$$\min \sum_{k=1}^K D_k \quad (22)$$

$$\text{subject to } \sum_{k=1}^K (h_k(a_k) + h_k(b_k)) \geq EC$$

$$\text{式中, } D_k = \frac{1}{2} (h_k(a_k) + h_k(b_k)) + \sum_{p=-\infty}^{a_k-1} h_k(p) + \sum_{p=b_k+1}^{+\infty} h_k(p).$$

在嵌入过程中, 对每一个预测误差直方图 $h_k(e)$ ($1 \leq k \leq K$), 根据对应的 a_k 和 b_k 及每个像素的预测误差 $e(x)$, 将像素 $\{x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(J)}\}$ 分成三类:

$$x \in \begin{cases} X & \text{if } a_k < e(x) < b_k \\ Y & \text{if } e(x) < a_k - 1 \text{ or } e(x) > b_k + 1 \\ Z & \text{otherwise} \end{cases} \quad (23)$$

对属于 X 类的像素, 不修改其像素值; 对属于 Y 类的像素, 按照(24) 修改其像素值:

$$C(x) = \begin{cases} C(x) - 1 & \text{if } e(x) < a_k - 1 \\ C(x) + 1 & \text{if } e(x) > b_k + 1 \end{cases} \quad (24)$$

我们只对属于 Z 类的像素嵌入信息. 利用二维预测误差直方图平移, 按照 $\{x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(J)}\}$ 的顺序(也就是像素局部空间复杂性从小到大的顺序, 对不属于 Z 类的像素则直接跳过) 修改每个像素的预测误差. 每次修改取该顺序中相邻的两个像素 x_s 和 x_{s+1} 进行修改:

$$\begin{aligned} & \text{if } (e_1 = a_k \text{ or } e_1 = b_k) \\ & \text{and } (e_2 = a_k \text{ or } e_2 = b_k) \\ & \text{if } (e_1 = a_k - 1 \text{ or } e_1 = b_k + 1) \\ & \text{and } (e_2 = a_k - 1 \text{ or } e_2 = b_k + 1) \\ & \text{if } (e_1 = a_k \text{ or } e_1 = b_k) \\ & \text{and } (e_2 = a_k - 1 \text{ or } e_2 = b_k + 1) \\ & \text{if } (e_1 = a_k - 1 \text{ or } e_1 = b_k + 1) \\ & \text{and } (e_2 = a_k \text{ or } e_2 = b_k) \end{aligned} \quad (25)$$

x_s 和 x_{s+1} 同属于第 k ($1 \leq k \leq K$) 组, 即 $\lceil J/K \rceil \times (k-1) + 1 \leq s, s+1 \leq \min(\lceil J/K \rceil \times K, J)$. $(m_1, m_2) \in \{(0,0), (0,1), (1,0)\}$, $m \in \{0,1\}$. 最后, Z 类中修改后的像素 x 在通道 C 的分量为

$$C(x) = \acute{e}_C(x) + \hat{C}(x) \quad (27)$$

式中, $e_c(x)$ 即经过式(23) 修改得到的像素 x 在通道 C 的预测误差. 嵌入像素集为 $C_2 (C_2 \in \{R_2, G_2, B_2\})$ 时的嵌入过程也是类似的.

2.6 信息提取与图像恢复

信息的提取流程如图 5 所示, 其中计算多级预测误差直方图和搜索最优直方图平移分组同嵌入过程完全一致. 以对像素集 B_2 的嵌入和提取为例, 嵌入过程中, 使用 G_2, R_2, R_1, G_1, B_1 这五个像素集对 B_2 进行预测, 提取过程中, 也是使用这五个像素集进行预测. 因此, 嵌入和提取过程中生成的多级预测误差直方图完全一致, 进而使用相同的搜索算法搜索得到最优的 a_k 和 b_k 也是完全相同. 这样可以保证本文算法的可逆性.

同嵌入过程类似, 根据 a_k 和 b_k 和此时计算得到的每个像素的预测误差 $e(x)$, 将像素 $\{x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(J)}\}$ 分成三类:

$$x \in \begin{cases} X & \text{if } a_k < e(x) < b_k \\ Y & \text{if } e(x) < a_k - 2 \text{ or } e(x) > b_k + 2 \\ Z & \text{otherwise} \end{cases} \quad (28)$$

$$m = \begin{cases} 0 & \begin{aligned} &\text{if } ((e_1 = a_k - 1 \text{ or } e_1 = b_k + 1) \text{ and } (e_2 = a_k - 1 \text{ or } e_2 = b_k + 1)) \\ &\text{or } ((e_1 = a_k \text{ or } e_1 = b_k) \text{ and } (e_2 = a_k - 2 \text{ or } e_2 = b_k + 2)) \\ &\text{or } ((e_1 = a_k - 2 \text{ or } e_1 = b_k + 2) \text{ and } (e_2 = a_k \text{ or } e_2 = b_k)) \end{aligned} \\ 1 & \begin{aligned} &\text{if } ((e_1 = a_k - 2 \text{ or } e_1 = b_k + 2) \text{ and } (e_2 = a_k - 2 \text{ or } e_2 = b_k + 2)) \\ &\text{or } ((e_1 = a_k - 1 \text{ or } e_1 = b_k + 1) \text{ and } (e_2 = a_k - 2 \text{ or } e_2 = b_k + 2)) \\ &\text{or } ((e_1 = a_k - 2 \text{ or } e_1 = b_k + 2) \text{ and } (e_2 = a_k - 1 \text{ or } e_2 = b_k + 1)) \end{aligned} \end{cases} \quad (31)$$

最后, 根据对应的两种情况, 反向平移二维预测误差直方图, 将 (e_1, e_2) 恢复成原来的 (e_1, e_2) , 如式(32)所示

$$(e_1, e_2) = \begin{cases} (e_1, e_2) - (f_k(e_1)m_1, f_k(e_2)m_2) & \text{if } \Omega \\ \text{if } (e_1, e_2) - (f_k(e_1)m, f_k(e_2)m) & \begin{aligned} &\text{if } ((e_1 = a_k - 1 \text{ or } e_1 = b_k + 1) \\ &\text{and } (e_2 = a_k - 1 \text{ or } e_2 = b_k + 1)) \\ &\text{or } ((e_1 = a_k - 2 \text{ or } e_1 = b_k + 2) \\ &\text{and } (e_2 = a_k - 2 \text{ or } e_2 = b_k + 2)) \end{aligned} \\ (e_1, e_2) - (f_k(e_1)m, f_k(e_2)) & \begin{aligned} &\text{if } ((e_1 = a_k \text{ or } e_1 = b_k) \\ &\text{and } (e_2 = a_k - 2 \text{ or } e_2 = b_k + 2)) \\ &\text{or } ((e_1 = a_k - 1 \text{ or } e_1 = b_k + 1) \\ &\text{and } (e_2 = a_k - 2 \text{ or } e_2 = b_k + 2)) \end{aligned} \\ (e_1, e_2) - (f_k(e_1), f_k(e_2)m) & \begin{aligned} &\text{if } ((e_1 = a_k - 2 \text{ or } e_1 = b_k + 2) \\ &\text{and } (e_2 = a_k \text{ or } e_2 = b_k)) \\ &\text{or } ((e_1 = a_k - 2 \text{ or } e_1 = b_k + 2) \\ &\text{and } (e_2 = a_k - 1 \text{ or } e_2 = b_k + 1)) \end{aligned} \end{cases} \quad (32)$$

式中, Ω 为式(30) 所有情况的并集, $f_k(\cdot)$ 由式(26) 确定. 最终, Z 类中像素在通道 C 的分量可以根据下

式恢复:

$$C(x) = \begin{cases} C(x) + 1 & \text{if } e(x) < a_k - 2 \\ C(x) - 1 & \text{if } e(x) > b_k + 2 \end{cases} \quad (29)$$

对属于 Z 类的像素, 其提取信息和恢复像素值的操作依然是按照 $\{x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(J)}\}$ 的顺序进行, 取同属于第 $k (1 \leq k \leq K)$ 组的相邻两个像素 x_s 和 x_{s+1} , 根据对它们嵌入的比特数分为以下两种情况: (1) 嵌入了两比特 (m_1, m_2) , 如式(30) 所示; (2) 嵌入了一比特 m , 如式(31) 所示.

$$(m_1, m_2) = \begin{cases} (0, 0) & \begin{aligned} &\text{if } (e_1 = a_k \text{ or } e_1 = b_k) \\ &\text{and } (e_2 = a_k \text{ or } e_2 = b_k) \end{aligned} \\ (0, 1) & \begin{aligned} &\text{if } (e_1 = a_k \text{ or } e_1 = b_k) \\ &\text{and } (e_2 = a_k - 1 \text{ or } e_2 = b_k + 1) \end{aligned} \\ (1, 0) & \begin{aligned} &\text{if } (e_1 = a_k - 1 \text{ or } e_1 = b_k + 1) \\ &\text{and } (e_2 = a_k \text{ or } e_2 = b_k) \end{aligned} \end{cases} \quad (30)$$

$$(30)$$

式(32)所示

式恢复:

$$C(x) = e_c(x) + \hat{C}(x) \quad (33)$$

这样,提取出隐秘信息的同时可以恢复图像像素值.再经过下面的直方图扩展,就可以完全无误地恢复原图像.

2.7 直方图扩展

将提取信息 M 中的位置映射 H 解压缩,根据解压缩后的 H 进行直方图扩展以恢复图像原先的纯黑像素(像素值为 0)和纯白像素(像素值为 255),即:

$$OI_C(i,j) = \begin{cases} 255 & \text{if } I_C(i,j) = 254 \text{ and } m = 1 \\ 0 & \text{if } I_C(i,j) = 1 \text{ and } m = 1 \\ I_C(i,j) & \text{otherwise} \end{cases} \quad (34)$$

3 实验结果与讨论

本文算法中的参数 K 的选取会影响算法复杂度和嵌入失真. K 越大,按照像素复杂性的分组后每一组内像素的复杂性也越相近,算法搜索出的最优预测直方图分组也越准确,嵌入失真也会减小.相应的, K 较大时需要搜索的预测误差直方图就越多,算法复杂度也越高.权衡这两者,本文将 K 取为 10.

实验将我们的算法同 Yao^[8]、Ou^[7]、Li^[6] 和 Ou^[1] 的算法进行了比较.

为了验证本文算法的性能,我们在四幅标准测试图像上进行实验.这四幅图像都是 8 位彩色图像,来自 USC-SIPI 数据集^[9],大小均为 512×512 . 它们被广泛地应用于可逆数据隐藏研究中,如图 8 所示.

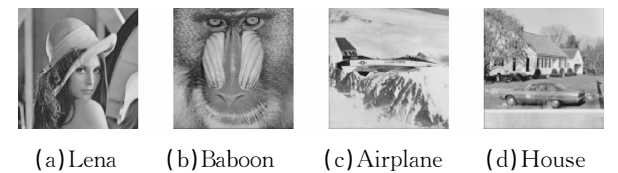


图 8 标准测试图像

根据第二节所述,本文算法根据图像自身各个通道的特性,确定嵌入顺序并分配嵌入容量.嵌入容量为 50 000 比特时,这四幅标准测试图像的嵌入顺序和嵌入容量如表 1 所示.

表 1 嵌入容量 50 000 时嵌入顺序与容量分配

图像	顺序	R_1	G_1	B_1	R_2	G_2	B_2
Lena	RGB	9 138	8 578	7 251	9 159	8 596	7 278
Baboon	GRB	8 275	9 039	7 710	8 246	9 019	7 711
Airplane	GBR	8 147	8 585	8 263	8 156	8 585	8 264
House	GBR	7 194	9 121	8 699	7 203	9 120	8 663

由表 1 可以看出,本文的容量分配策略倾向于将更多的容量分配给图像主色调对应的通道.图像

Airplane 的大部分为白色,所以嵌入容量近似是均匀划分.

同许多低失真的可逆数据隐藏算法一样,实验将嵌入过程中对每个像素每个通道分量的最大修改限制在 1. 因此,嵌入容量从 20 000 开始,步长为 5 000,一直取到最大容量限制(Baboon 这张图步长为 2 000). 本文使用峰值信噪比(Peak Signal-to-Noise Ratio, PSNR)来表征嵌入过程引入的失真,单位为 dB.

用 OI 表示原彩色图像, OI' 表示嵌入信息后的彩色图像,则 OI' 与 OI 间的 PSNR 表示为:

$$\text{PSNR}(OI',OI) = 10 \log_{10} \frac{255^2 \times 3 \times P \times Q}{\sum_C \sum_{j=1}^Q \sum_{i=1}^P (OI'_C(i,j) - OI_C(i,j))^2} \quad (35)$$

式中, P 和 Q 表示每个通道的长和宽, $C \in \{R, G, B\}$.

表 2 和表 3 分别给出了在嵌入容量为 30 000 和 50 000 比特的情况下的峰值信噪比. 表中数据均四舍五入保留小数点后两位.

表 2 峰值信噪比较(30 000 比特,单位: dB)

图像	Yao ^[8]	Ou ^[7]	Li ^[6]	Ou ^[1]	本文
Lena	60.11	58.71	57.04	59.59	60.87
Baboon	56.78	54.55	54.69	54.83	56.85
Airplane	63.28	62.93	60.58	63.03	63.97
House	64.19	64.14	63.03	64.31	64.97
平均	61.09	60.08	58.84	60.44	61.67

表 3 峰值信噪比较(50000 比特,单位: dB)

图像	Yao ^[8]	Ou ^[7]	Li ^[6]	Ou ^[1]	本文
Lena	57.33	56.23	54.63	56.65	57.99
Baboon	52.37	50.83	50.87	49.26	52.89
Airplane	60.64	60.13	58.55	60.17	61.47
House	61.74	61.63	60.71	62.04	62.82
平均	58.02	57.21	56.19	57.03	58.80

由表 2 和表 3 可以看出,当嵌入容量分别为 30 000 和 50 000 比特时,本文算法的 PSNR 在四幅图像上都是最好的. 相比于 Yao^[8]、Ou^[7]、Li^[6] 和 Ou^[1] 的算法,本文的算法在嵌入容量为 30 000 比特时平均 PSNR 分别提高了 0.58 dB、0.59 dB、2.83 dB、1.23 dB; 在嵌入容量为 50 000 时,平均 PSNR 分别提高了 0.78 dB、1.59 dB、2.61 dB、1.77 dB.

为了更好地显示随着嵌入容量增大 PSNR 的变化趋势,图 9 画出了四幅图像的 PSNR-EC 曲线.

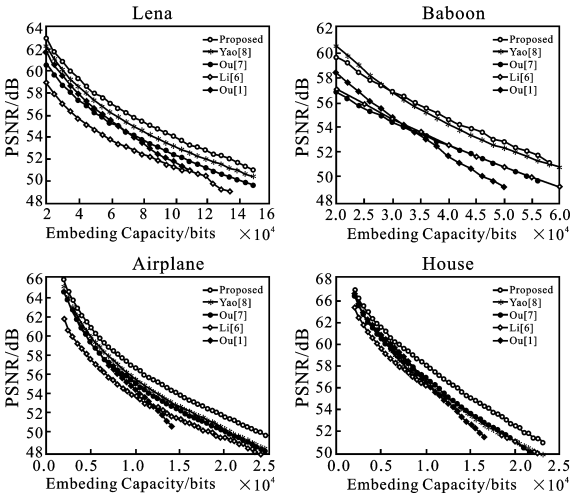


图 9 PSNR-EC 曲线

如图 9 所示,除了在 Baboon 图上嵌入容量小于 30 000 比特时,本文算法的效果不如 Yao^[8] 的算法,其他情况下,本文算法的 PSNR 都高于其他四种算法. 由于 Baboon 这张图纹理性较强,当嵌入容量特别小的时候,本算法对于当前像素复杂性的计算会受到参考通道纹理性的影响,导致对像素的嵌入顺序并不是最优的,进而使得预测误差增大,使得嵌入失真比 Yao^[8] 的算法高.

4 结束语

由于彩色图像三个通道具有高度相关性,我们探讨了利用这种通道间的相关性来对图像通道进行线性回归预测的可逆数据隐藏算法. 基于彩色图像成像过程中朗伯反射和一些局部不变特性的假设,本文验证了当像素强度消除时,局部色度具有相似性. 利用这种局部色度相似性,本文筛选出候选像素以对当前嵌入像素进行通道线性回归估计. 根据生成的预测误差直方图,对局部空间复杂性小的像素优先使用二维预测误差直方图平移算法嵌入. 实验结果表明,相比于现有的算法,在相同嵌入容量时,本文的算法可以获得更高的峰值信噪比.

参考文献:

[1] Ou B, Li X, Wang J. High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion [J]. Elsevier North-Holland, Inc., 2015, 111(C):249-260.

[2] Devi A, ShivaKumar K B. Protection of confidential color image information based on Reversible Data Hiding technique (PCCIRT) [C]//Computing and Network Communications (CoCoNet), 2015 International Conference on. Trivandrum, Kerala, India, IEEE, 2015: 742-747.

[3] Huang H C, Lin J, Lu Y Y. Ownership protection for progressive transmission of color images with reversible data hiding[C]//Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2015 International Conference on. Matsue, Shimane, Japan, IEEE, 2015: 5-8.

[4] Yang H W, Hwang K F. Reversible data hiding for color BMP image based on block difference histogram [C]//Ubi-Media Computing (U-Media), 2011 4th International Conference on. Moscow, Russia, IEEE, 2011: 257-260.

[5] Tseng H W, Ding W B. Reversible data hiding scheme for colour images based on pixel clustering and histogram shifting[J]. The Imaging Science Journal, 2012, 60(1): 47-53.

[6] Li J, Li X, Yang B. Reversible data hiding scheme for color image based on prediction-error expansion and cross-channel correlation[J]. Signal Processing, 2013, 93(9): 2748-2758.

[7] Ou B, Li X, Zhao Y, et al. Efficient color image reversible data hiding based on channel-dependent payload partition and adaptive embedding[J]. Signal Processing, 2015, 108 (C) :642-657.

[8] Yao H, Qin C, Tang Z, et al. Guided filtering based color image reversible data hiding[J]. Journal of Visual Communication and Image Representation, 2017, 43 (C) :152-163.

[9] SIPI USC. The usc-sipi image database [EB/OL]. (2005-07-12) [2009-08-05]. [http:// sipi. usc. edu/ services/ database/ data-base. html](http://sipi.usc.edu/services/database/data-base.htm).

作者简介:

葛昊立 男,(1993-),硕士研究生. 研究方向为信息隐藏和数据加密. E-mail:hlge15@fudan. edu. cn.

陈妍 女,(1994-),硕士研究生. 研究方向为信息隐藏和数据加密.

王建军 男,(1960-),副教授,硕士生导师. 研究方向为信息隐藏、数据加密、图像取证、信息安全等.