

新颖的差值扩展可逆数据隐藏算法

罗剑高¹, 韩国强², 沃焱²

(1. 广东农工商职业技术学院计算机系, 广东 广州 510507; 2. 华南理工大学计算机科学与工程学院, 广东 广州 510006)

摘要: 提出一种变换方向自适应的差值扩展可逆数据隐藏算法(ACD), ACD 通过扩展像素对的差值嵌入数据。像素对中一为参照像素, 另一为变换像素, ACD 根据像素对所处区域类型的估测, 自适应选择变换像素, 暗区选大值像素, 亮区选小值像素, 否则选默认方向像素。理论分析及实验结果表明, ACD 解决了预测误差扩展可逆数据隐藏算法(PEE)在暗或亮载体时灰度溢出急剧增加的问题, 与采用现有灰度溢出解决方案的 PEE 相比, 改善了载体普适性, 性能更稳定。

关键词: 可逆数据隐藏; 差值扩展; 灰度溢出; 自适应数据嵌入

中图分类号: TP391

文献标识码: A

Novel reversible data hiding based on difference expansion

LUO Jian-gao¹, HAN Guo-qiang², WO Yan²

(1. Department of Computer, Guangdong AIB Polytechnic College, Guangzhou 510507, China;

2. School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China)

Abstract: A novel reversible data hiding algorithm based on difference expansion called ACD was proposed, which used the so-called adaptive change direction technology. ACD embeds data by expanding the differences of neighboring pixel values. In a pair of pixels, a pixel was for reference and the other was for transform. ACD first estimates the type of the pixels area, then chooses transform pixels by the type that it chooses the large value pixel in dark areas and the small value pixel in bright areas, or else selects the pixels by default direction. Theoretical analysis and experimental results show that ACD solves the problem of gray overflow increased dramatically in the reversible data hiding based on prediction error expansion (PEE) when the carrier is very dark or bright. Compared with the PEE with the existing gray overflow solution, ACD is suitable for wider types of carrier and has more stable performance.

Key words: reversible data hiding, difference expansion, gray overflow, adaptive data embedding

1 引言

可逆数据隐藏^[1](RDH, reversible data hiding)技术是一种以某种可逆方式把待隐藏数据嵌入到载体, 生成相对载体而言感知变化不明显的含隐载体, 在需要时能从该含隐载体中提取被嵌入的数据, 且能完全还原载体的数据隐藏技术。RDH 适用于对载体失真特别敏感(如遥感、医学图像)的数据隐藏应用。从 20 世纪 90 年代末至今, 相关文献已刊发了一系列的 RDH 具体算法^[2~10], 与此同时,

RDH 相关应用技术也不断取得进步^[11,12]。现有的 RDH 大致可归为以下 3 类: 基于无损压缩^[1]; 扩展变换^[2~10]; 直方图变换^[1]。近年来, 基于预测误差扩展的 RDH 算法(PEE, prediction error expansion)得到了研究人员的广泛关注, 因为其具有更好利用载体空间冗余的能力。一般地, PEE 相对其他类型的算法具有更优的容量—失真性能。

本文的研究以 8 位灰度图像为载体。由于对载体直接执行扩展变换无法避免灰度溢出, 即在扩展变换某特征值的过程中, 有可能使参与变换的某些

收稿日期: 2015-01-11; 修回日期: 2015-03-15

基金项目: 国家自然科学基金资助项目(No.61472145); 国家自然科学基金广东省联合基金资助项目(No.U1035004)

Foundation Items: The National Natural Science Foundation of China (No.61472145), The National Natural Science Foundation of China—Guangdong Province (No.U1035004)

像素值小于 0 或大于 255, 导致无法正确保存这些变换后的像素值, 因此, 有效的 PEE 必须包含能解决灰度溢出的处理过程。目前, PEE 采用的灰度溢出解决方法^[1-10]所需付出的代价一般与载体的直方图分布密切相关, 比如当载体像素值在 0 或 255 附近占比偏高时, 解决灰度溢出问题所需的辅助比特数急剧增加。现有大多数有关 PEE 的文献在防灰度溢出处理方面讨论的不够充分, 没有明确指出所提算法因灰度溢出问题, 有效容量—失真性能除与载体图数据冗余有关外, 还与像素直方图边缘分布有密切联系, 没有指出载体图过暗或过亮时算法有效容量—失真性能急剧恶化的缺陷, 其仿真实验一般也较少涉及较暗或较亮的载体图。由于现有灰度溢出解决技术影响了 PEE 算法的实际适用范围, 因此研究更有效防溢出的技术已成为一个重要的课题^[13]。

针对现有 PEE 在灰度溢出问题处理上存在的不足, 本文提出一种新颖的变换方向自适应的差值扩展 (ACD, adaptive change direction) 算法。ACD 中的预测算子采用相邻像素预测, 在不明显增加计算复杂度的前提下, 自适应处理有效解决了暗或亮载体时, 有效数据嵌入能力大大下降的问题, 使载体直方图边缘分布的大小不再是影响载体数据嵌入能力的重要因素, 与现有 PEE 算法相比, 较为显著地改善了载体的普适性。

2 相关工作及分析

扩展嵌入是一种把数据比特嵌入到扩展特征值最低有效位 (LSB) 的可逆变换技术, 由 Tian^[2] 在 2003 年首先提出。Tian 在像素对上定义了一组称之为差值扩展 (DE) 的可逆变换, 2007 年, Thodi 等^[3]把 DE 推广到预测误差上, 定义了一套通用的预测误差扩展可逆变换 (PEE), 用 MED (median edge detector) 预测误差算子来计算预测误差, 随后的 PEE 研究者先后提出了几种更有效的预测误差算子^[4-10], 用于获取更适合数据嵌入的预测误差值。在阐述本文 ACD 之前, 先分析 PEE 变换和与之相关的灰度溢出解决方法。

2.1 PEE 变换

在数据嵌入端, 首先通过某种像素预测算子, 对宿主 I 的当前像素值进行预测, 设当前像素值为 x , 其预测值为 \hat{x} , 则像素预测误差 $e = x - \hat{x}$ 。基于 e 位移扩展和两倍扩展变换的数据嵌入规则为

$$e' = \begin{cases} e + \Delta + 1, e > \Delta \\ e - \Delta - 1, e < -\Delta - 1 \\ 2e + b, \text{其他} \end{cases} \quad (1)$$

或者

$$e' = \begin{cases} e + \Delta + 1, e > \Delta + 1 \\ e - \Delta - 1, e < -\Delta \\ 2e - b, \text{其他} \end{cases} \quad (2)$$

其中, e' 为扩展后的像素误差, Δ 为大于等于 0 的嵌入容量参数, 用于控制 RDH 的数据嵌入容量和载体的嵌入失真, RDH 可嵌入的数据总量等于 e 处在区间 $[-\Delta - 1, \Delta]$ 或 $[-\Delta, \Delta + 1]$ 的像素量。 b 为嵌入的数据比特 (0 或 1)。由扩展后的预测误差 e' 可得到嵌入数据后的像素值 $x' = \hat{x} + e'$ 。

在数据提取端, 设含隐图像为 I , 当前像素值为 x' , 先用同一预测算子计算出与嵌入端 \hat{x} 相一致的 x' 预测值 \hat{x}' , 令 $e' = x' - \hat{x}'$, 执行式 (1) 或式 (2) 的逆操作, 可求得对应的 e

$$e = \begin{cases} e' - \Delta - 1, e' > 2\Delta + 1 \\ e' + \Delta + 1, e' < -2\Delta - 2 \\ \left\lfloor \frac{e'}{2} \right\rfloor, \text{其他} \end{cases} \quad (3)$$

或者

$$e = \begin{cases} e' - \Delta - 1, e' > 2\Delta + 2 \\ e' + \Delta + 1, e' < -2\Delta - 1 \\ \left\lceil \frac{e'}{2} \right\rceil, \text{其他} \end{cases} \quad (4)$$

原始载体对应的 $x = e + \hat{x}$, 式 (5) 依次对 $[-2\Delta - 2, 2\Delta + 1]$ 或 $[-2\Delta - 1, 2\Delta + 2]$ 中的 e' 进行模 2 运算 (除以 2 求余), 提取被嵌入的数据为

$$b = e' \bmod 2 \quad (5)$$

需要注意的是, 式 (1) 或式 (2) 可能会使 x' 超出有效区间 $[0, 255]$, 致使 x' 无法有效保存, 从而导致数据提取端无法正确获取式 (1) 或式 (2) 对应的值, 破坏变换的可逆, 即所谓的灰度溢出问题。为确保 PEE 变换可逆, 必须解决灰度溢出问题。

2.2 灰度溢出解决方法

解决灰度溢出的策略现在主要有 2 种: 1) 令可能出现溢出的像素保持不变, 并把与它们位置有关的信息记录下来作为辅助信息之一, 传送到数据提取端, 以便数据提取时做相应的逆处理^[2-5,8,9]; 2) 压缩载体图的灰度直方图, 保证扩展变换后, 像素

的灰度值都处在有效范围内，记录灰度直方图压缩的相关信息作为辅助信息的一部分，传送到数据提取端，供直方图做相应的解压缩变换^[6,7,10]。2 种策略中的灰度溢出信息记录或采用位置图压缩方式或采用标识比特的方式，位置图压缩方式的灰度溢出辅助信息量与压缩算子的效率密切相关，不适合小尺寸的图像载体，且相对标识比特方式，数据压缩处理过程会较明显地增加 RDH 算法复杂度，一般不予以采用。接下来，本文以 2 个具体的典型灰度溢出解决方法为例来分析现有灰度溢出解决方法存在的不足。

2.2.1 2 次检测标识比特法

2 次检测标识比特法 (FB, flag bit) 是策略 1 的经典方法之一，由 Thodi 等^[3]首先提出。本文以 Vasilij 等^[4]采用的一种 2 次检测标识比特法为例说明其实现过程。一般地，大多数载体的像素可以执行多次不溢出的扩展变换，2 次变换后溢出和 1 次变换后溢出的像素 (如图 1 样本 1、2、3) 比较少，Vasilij 等在扩展变换前用 2 次溢出测试过程记录下这些像素：如图 1 所示，逐一对像素执行扩展变换距离最大的测试 (预测误差 $e \geq 0$ ，令 b 为 1，否则为 0)，如果 1 次变换就溢出，用比特 1 标识，2 次

变换溢出的用比特 0 标识，其他的不用标识。找出 2 次变换或 1 次变换溢出的像素后，将标识比特组成的比特串与待嵌入数据合并后的结果更新待嵌入数据比特，然后开始执行数据的嵌入。在数据嵌入过程中，被标识的像素不嵌入有效数据，其中，1 次变换溢出的像素保持不变，2 次变换溢出的像素执行 1 次变换距离最大的变换 (嵌入相应的填充比特 0 或 1)，除此之外的像素执行常规的扩展变换。在数据提取端，先对待逆变换的像素做 1 次变换距离最大的嵌入变换，如果灰度溢出，则借助溢出标识比特进一步区分该像素在嵌入端是保持不变还是执行了 1 次变换距离最大的填充变换，否则可确定该像素在嵌入端执行了常规的扩展变换。

2.2.2 直方图边缘压缩法

图 2 是属于策略 2 的直方图边缘压缩法 (HM, histogram modification) 示意，早在 2006 年宣国荣等^[6]就提出了 HM 处理方法。本文以罗剑高等^[10]防灰度溢出方法为例来说明 HM 的实现过程。在扩展变换前，先对载体进行预处理，将 $[0, \Delta]$ 和 $[255-\Delta, 255]$ 区间的灰度值加或减 $\Delta+1$ ，把载体灰度值的区间压缩成 $[\Delta+1, 254-\Delta]$ ，然后在预处理后的载体上执行灰度变化量不大于 $\Delta+1$ 的扩展变换，灰度溢出

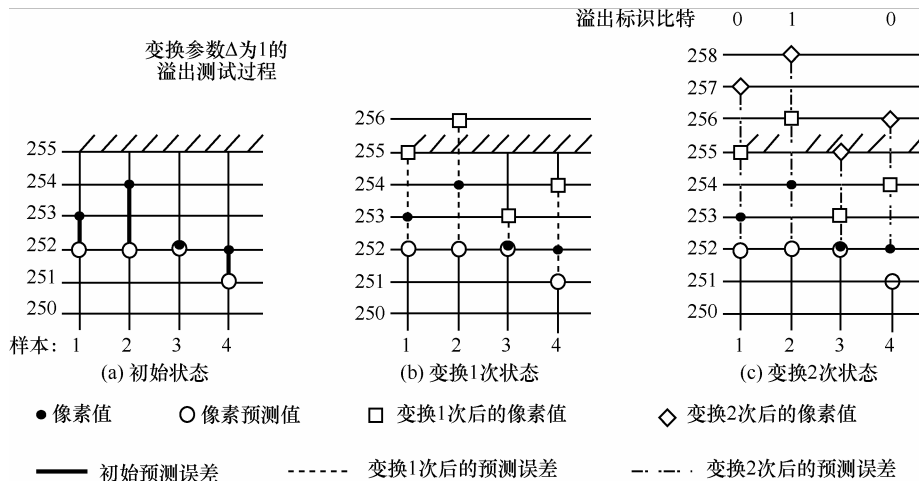


图 1 式(1)变换下的 2 次检测标识比特法示例

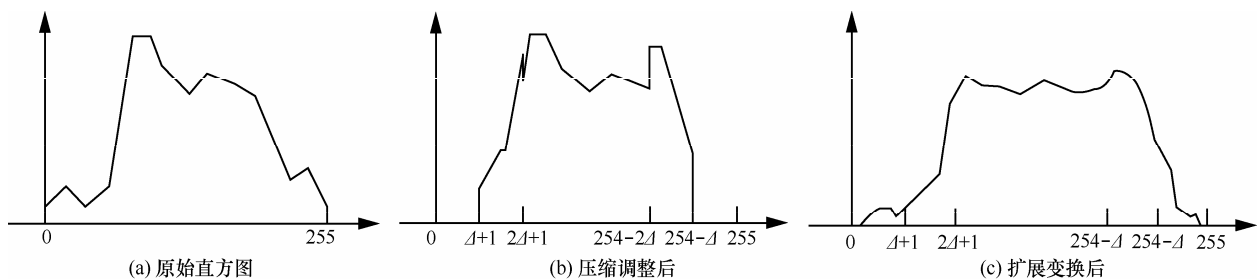


图 2 灰度直方图调整示意

即可避免。预处理过程具体如下，灰度压缩变换信息用标识比特串 F 记录：令 F 初值为空，按某个次序逐一扫描载体像素，当前灰度值 x 小于 $\Delta+1$ 或大于 $254-\Delta$ ，则令 x 加或减 $\Delta+1$ ，同时在 F 末尾追加标识比特“1”；当前灰度值 $x \in [\Delta+1, 2\Delta+1]$ 或 $[254-2\Delta, 254-\Delta]$ ，则 x 不变，同时在 F 末尾追加标识比特“0”；否则不做处理。 F 作为辅助信息的一部分传送给提取端，提取端根据 F 可执行上述预处理的相应逆变换，即可最终恢复载体图。

可以看出，FB 方法产生的灰度溢出辅助信息量等于溢出测试时二次变换后溢出和一次变换后溢出的像素数量，HM 方法产生的灰度溢出辅助信息量等于像素值处在 $[0, 2\Delta+1]$ 和 $[254-2\Delta, 255]$ 区间的像素数量，一般的自然图像，这个量都较小，甚至为 0，但当载体图很亮或很暗时，这个量将急剧增加，最终导致有效嵌入数据量急剧减少。

3 变换方向自适应差值扩展嵌入方案

扩展变换是定义在一对值（不失一般性，记为 (x, y) ）差值上的可逆变换。在 Tian^[2] 的 DE 中， x, y 分别为某相邻的 2 个像素值，变换操作同时修改 x, y 值， x, y 可能增加，也可能减小；在 PEE 中， y 为某像素的灰度值， x 为该像素灰度值的预测值，变换操作只增加或减小 y ， x 仅作为参照值，变换操作中保持不变。图 3 是 DE 灰度溢出和 PEE 灰度溢出的示例。

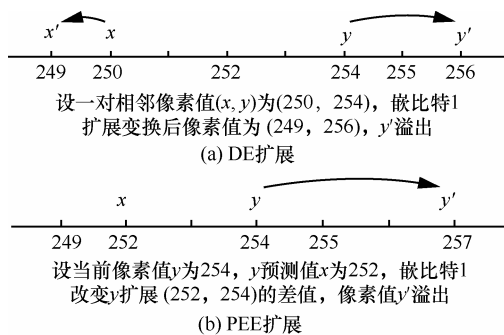


图 3 灰度溢出示例

本文注意到，图 3(b) 中 (x, y) 值的变换方向是固定的，即只变换 y ，当 y 大于 x 时， y 往增加的方向改变，当 y 小于 x 时， y 往减小的方向改变，如果本文可以调整 (x, y) 值的变换方向，比如，在 (x, y) 为 $(252, 254)$ 时，令 y 为参照值，修改 x 扩展 (x, y) 的差值，就可以避免图 3(b) 中灰度溢出，基于这个思路，本文提出一种改善灰度溢出问题的差值扩展嵌入方案-变换方向自适应的差值扩展，

并实现了一个称为 ACD 的具体算法。变换方向自适应差值扩展方案可显著地减少灰度溢出的发生，降低相应的灰度溢出辅助信息量，其原理如图 4 所示，当 (x, y) 在暗区时，修改大值；在亮区时，修改小值；否则修改默认位置上的 y 值。为避免值对 $(255, 255)$ 的变换溢出，亮区选用式(2)变换；为避免值对 $(0, 0)$ 的变换溢出，暗区选用式(1)变换；中间区选用式(1)或式(2)变换。

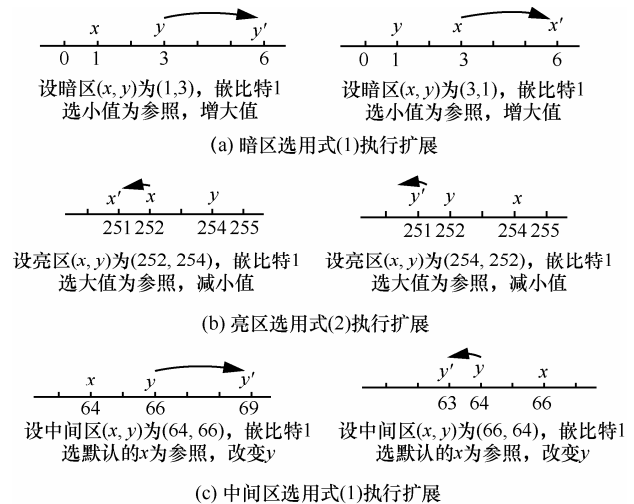


图 4 变换方向自适应的差值扩展原理

4 变换方向自适应差值扩展 RDH

文中变化方向自适应的差值扩展 RDH 算法(简称 ACD)除了变化方向问题外，其他需要考虑的问题与常规 PEE 算法类似，本节对 ACD 算法的实现进行具体说明。

4.1 预测算子及变化方向的选择

ACD 是一种预测误差扩展 RDH 算法，预测算子采用相邻像素预测。

设载体图像 I 的分辨率为 $u \times v$ ，对载体图像执行图 5 所示的像素扫描，把 I 转换成长度为 $u \times v$ 的像素序列 O 。

记 O 中下标为 i (令第一个像素的 i 为 0) 的像素为 O_i ，ACD 依次对像素对 (O_i, O_{i+1}) 进行数据嵌入处理。设当前待处理像素对 (O_i, O_{i+1}) 的值为 (x_i, x_{i+1}) ，执行变换时的变换方向确定过程用伪码表示如下。

伪码 1 确定差值扩展的变换方向

记 (x_i, x_{i+1}) 中 x_i 为变换值， x_{i+1} 为 x_i 的预测值时的嵌入方向为 $r1$ ；

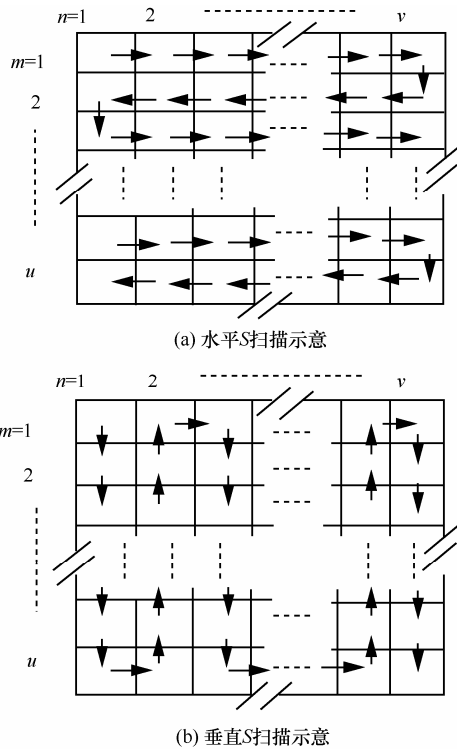


图 5 像素扫描示意

记 (x_i, x_{i+1}) 中大值为变换值, 小值作为大值预测值的嵌入方向为 r2;

记 (x_i, x_{i+1}) 中小值为变换值, 大值作为小值预测值的嵌入方向为 r3;

嵌入方向为 r2 或 r3 时, 如果 $x_i = x_{i+1}$, 变换值选 x_i, x_{i+1} 作为 x_i 的预测值。

令嵌入方向 $r=r1, i$ 为 0;

While ($i < u \times v - 1$) {

if ($i > 1$) {

if ($2 \lfloor \frac{x_{i-1}}{2} \rfloor \leq \Delta$ & $2 \lfloor \frac{x_{i-2}}{2} \rfloor \leq \Delta$);

估测像素对处在暗区, 用 r2 更新 r;

else if

$(2 \lfloor \frac{x_{i-1}}{2} \rfloor \geq (255 - \Delta))$ & $(2 \lfloor \frac{x_{i-2}}{2} \rfloor \geq (255 - \Delta))$

估测像素对处在亮区, 用 r3 更新 r;

else if

$(\Delta < 2 \lfloor \frac{x_{i-1}}{2} \rfloor < (255 - \Delta))$ & $(\Delta < 2 \lfloor \frac{x_{i-1}}{2} \rfloor < (255 - \Delta))$

估测像素对处在中间区, 用 r1 更新 r;

else

估测像素对处在亮区—中间区或暗区—中间区的过渡区, r 保持不变;

```
}
    i=i+1;
}
```

注意, 上述 (x_i, x_{i+1}) 变换方向是由已完成变换且 LSB 置 0 后的前 2 个像素值确定的, 所以, 在提取端, 在对应的扩展逆变换前, 由上述伪码, 按与嵌入端相同的次序可以得到嵌入时各像素对的扩展变换方向。

4.2 溢出处理

灰度溢出是扩展变换无法避免的问题。ACD 采用变换方向自适应的处理虽然大大减少了灰度溢出的发生, 但并不能绝对避免灰度溢出问题, 例如: $(0, 255)$ 像素对的变换。

为此, 本文在 2 次检测标识比特法的基础上, 设计了一种 1 次检测标识比特法, 其处理过程如下: 令溢出信息标识比特串 F 初值为空, 逐一对像素对执行扩展变换, 如果出现灰度溢出, 则取消该变换, 保持像素对不变, 同时在 F 末尾追加标识比特“1”; 如果没有出现灰度溢出, 则执行扩展变换, 之后在变换的结果上再执行一次变换量尽可能大 (预测误差 $e \geq 0$, 令 b 为 1, 否则为 0) 的灰度溢出测试变换, 如果出现溢出, 在 F 末尾追加标识比特“0”。 F 作为辅助信息的一部分传送给提取端, 数据提取时, 根据 F 执行如下处理, 即可辨识嵌入时因灰度溢出问题而保持原值的像素对: 以倒序的次序取 F 的当前标识比特, 按与嵌入端像素处理的倒序逐一对含隐像素对执行逆变换, 在扩展逆变换前, 先执行一次变换量尽可能大的灰度溢出测试变换, 如果溢出, 则可观察当前标识比特, 为“1”表示当前像素对在嵌入端存在灰度溢出问题, 未执行扩展, 数据提取时亦不做变换, 保持原值。除此之外, 其他像素对在数据提取端需执行相应的扩展嵌入逆处理。

4.3 嵌入参数优化及多层嵌入

为了限制像素值最大变化的幅度, ACD 约定 Δ 在 $[0, 15]$ 中取值, Δ 用 4 bit 表示。ACD 单层 (单次) 嵌入的数据量不大于 1bpp (bit per pixel 每像素比特), 当单层嵌入的数据量无法满足需要时, 可执行多层嵌入。为了简化 ACD 表述, 文中多层嵌入采用相对独立的方式, 即每层的嵌入与相邻层的嵌入不发生数据交互, 各自分别依次嵌入数据。执行多层嵌入时, 各层像素扫描方向在水平 S 扫描、垂直 S 扫描两者间交替变化, 非末层嵌入的 Δ 取 15, 末层 Δ 在 $[0, 15]$ 中取值, 对 Δ 值进行优化, 就是要找

到完整嵌入待隐藏数据的前提下末层 Δ 的最小值, 它可使含隐载体失真最小。ACD 采用折半迭代的方法优化 Δ 值, 具体步骤详见第 4.5 节的数据嵌入过程。由于允许多层嵌入, 在数据提取端执行对应的多层提取时, 就要求能识别数据提取操作的最末层 (对应数据嵌入阶段的首层), 用于结束数据提取操作。ACD 每层分别用 1 比特辅助信息作为层标识位来实现这个要求, 具体见第 4.4 节。

4.4 辅助信息传送及结构

RDH 算法在数据提取/载体恢复端, 需要数据嵌入时所用的, 称之为辅助信息的一些参数。为实现数据的盲提取和载体的盲恢复, 要求该信息包含在载体内, 随载体传送。为此, ACD 提出一种新颖的 LSB 置换法来实现辅助信息随载体的传送, 具体过程如下: 取每层的层标识比特作为该层待嵌入的第一个数据比特, 然后依次取已完成扩展变换像素的 LSB 作为待嵌入比特。设该层扩展嵌入完毕时已嵌入比特数为 n , 也就是说该层前 $n-1$ 个像素的 LSB 被嵌入到载体中, 在扩展嵌入完毕后, ACD 需隐藏的数据及相应的辅助信息就可依次保存在这 $n-1$ 个 LSB 中。

ACD 各层需嵌入的数据比特串结构规定如下: 先构造该层扩展嵌入完毕后除层标识之外的辅助信息比特串 A , 然后用 A 与当前待嵌入的数据比特串 D 合并的结果更新当前 D 为

$$D=A+D \quad (6)$$

如果 D 中比特数大于 $n-1$, 则其前 $n-1$ 个比特填充到 $n-1$ 个 LSB 中后, 用截除前 $n-1$ 比特的 D 更新 D 。如果 D 中比特数不大于 $n-1$, 则 LSB 中不足被 D 比特填充的填充比特 “1”, 然后置 D 为空。为在提取端能分辨出填充比特, 令 ACD 需嵌入 D 的初始值为 ACD 待隐藏的数据比特加一个作为填充比特起始标识的 “0” 比特。数据提取端, 丢弃提取比特串最末 “0” 及之后的 “1” 比特, 可去除填充比特起始标识和填充比特。

ACD 算法中的辅助信息包括: 末层 Δ 、末层的像素扫描方式、各层的灰度溢出信息、各层的层标识位。在多层嵌入过程中, 每层各自处理自己的辅助信息, 其中, 首层嵌入的层标识比特规定为 “1”, 其他层规定为 “0”, ACD 末层嵌入的辅助信息除了层标识之外结构为

$$A=S+L_b+\Delta_2+F \quad (7)$$

非末层嵌入的辅助信息除层标识之外的辅助信息结构为

$$A=L_b+F \quad (8)$$

其中, F 为本层溢出信息比特串, Δ_2 为表示末层嵌入参数的 4 比特, S 表示扫描方向 (0 表示水平 S 型扫描, 1 表示垂直 S 型扫描), L_b 是长度为 $\lceil \lg(u \times v) \rceil$ 的比特串, 用于表示当前层参与 LSB 置换的像素数 $n-1$ 。由于非末层的 Δ 固定为 15, S 可由式(7)的 S 推导, 故相比式(7), 式(8)中的 A 要少用 5 比特。

4.5 ACD 算法实现过程

ACD 算法的实现包括 2 部分, 数据的嵌入和数据提取/载体恢复, 后一部分是前一部分的逆过程。设待隐藏的数据比特串为 D , 载体图像为 I , 含隐载体最大可接收的失真为 σ , ACD 的实现过程描述如下。

1) 数据嵌入过程

①令 $\Delta=15$, 扫描方向 $S=0$,

// $S=0$ 水平 S 扫描, $S=1$ 垂直 S 扫描

②依次扫描像素, 把图像 I 转换成像素序列 O 。

③自适应确定变换方向后, 对当前像素对进行数据嵌入处理: 确定待嵌比特、执行扩展变换、收集溢出信息。

④扩展嵌入处理完毕后, 用式(8)按非末层嵌入模式构造辅助信息, 更新待嵌入的 D , 设 D 的比特数为 m , 如果 $m+5$ 不大于第 4.4 节所述的 LSB 置换数 $n-1$, 则进入末层嵌入模式, 用式(7)重新构造辅助信息和 D , 把 D 中比特依次填入 LSB 中, 不足部分填入填充比特, 保存当前含隐图像, 进入第⑤步的 Δ 迭代求优; 否则把含式(8)辅助信息的 D 中按 4.4 节所述处理, 然后用当前含隐载体更新载体, 计算当前含隐载体的失真, 如果大于 σ , 退出算法, 提示嵌入失败, 否则改变扫描方向 S , 转第②步执行新一层的数据嵌入。

⑤末层 Δ 优化阶段执行 4 次折半迭代, 在末层嵌入迭代阶段, 如前一过程完整嵌入了 D , 令 $\Delta=\Delta-\left\lfloor \frac{\Delta}{2} \right\rfloor$, 否则令 $\Delta=\Delta+\left\lfloor \frac{\Delta}{2} \right\rfloor$, 重复执行最后一层的嵌入。输出能满足载荷数据完整嵌入且 Δ 最小时的含隐图像。

2) 数据提取/载体恢复过程

①设提取数据比特串 W 为空, 令扫描方向 S 等于第一个像素的 LSB。

②按 S 依次扫描像素, 把 I 转换成像素序列 O' 。顺序扫描 O' , 确定各 (O'_i, O'_{i+1}) 变换方向 r 。

③提取 O' 第 2 到第 $\lceil \lg(u \times v) \rceil + 1$ (首层提取)

或第 1 到第 $\lceil \lg(u \times v) \rceil$ (非首层) 个像素 LSB, 换算成在整数, 设为 m 。用 O' 第 1 到第 m 个像素的 LSB 添加在 W 头部。

④从 O' 末尾位置开始, 根据 W 中提供的辅助信息, 进行相应的逆变换, 提取数据, 同步依次恢复 m 个被 LSB 置换的像素值。最后在 W 中清除掉当前层的辅助比特, 得到更新后的 W 。

⑤如果当前层标识为“0”, 用当前层输出图像更新 I , 用 15 更新 Δ , 改变扫描方向 S , 然后转到第 2 步; 否则舍弃 W 末尾的填充比特, 得到所求的目标数据 W 和 I , 过程结束。

为了便于理解 ACD 实现过程, 图 6 给出了单层数据嵌入和提取的示例。数据嵌入过程如图 6(a) 所示, 从载体的左上角开始, 依次在 (O_i, O_{i+1}) 的

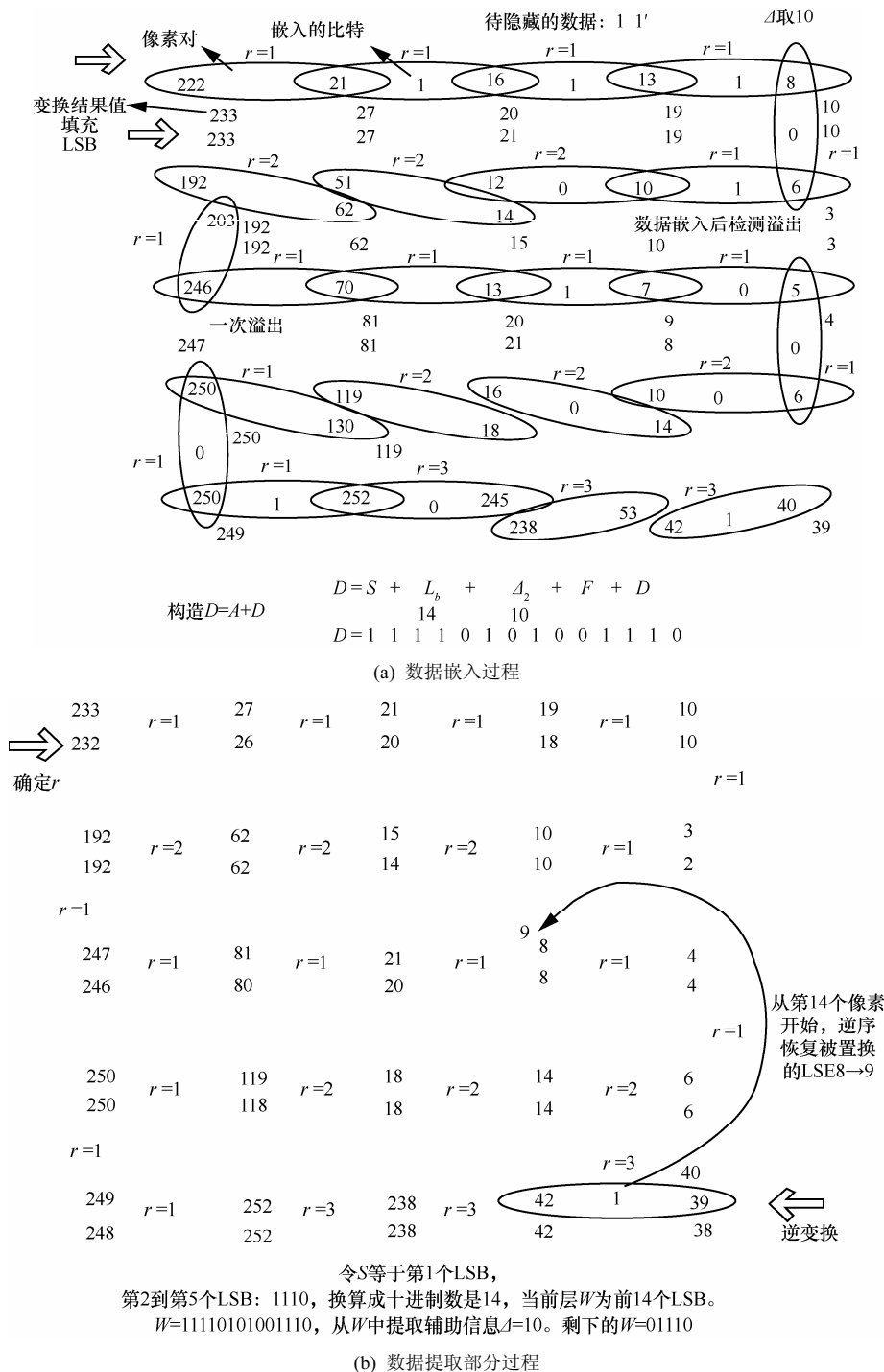


图 6 ACD 单层数据嵌入和提取过程示例

当前像素值上执行扩展变换。第一个要嵌入的比特是层标识比特“1”，然后依次为已完成扩展变换像素（如 233、27、20...）的 LSB。扩展变换结束后，构建 D ，最后把 D 次填充到对应的 14 个 LSB 中。图 6(b)是数据提取的部分过程，首先令 S 等于含隐载体的第一个 LSB，然后按扫描次序取前 1~5 个像素的 LSB，换算成十进制数 14，取前 14 个 LSB 为当前提取的数据 W ，从 W 提取出 $\Delta=10$ ；令初始 2 个变换对的变换方向为 $r1$ ，接着用当前像素对的 LSB 置 0 后的前 2 个像素值来修改变换方向；最后从最后一个像素对开始，逆序进行相应的扩展变换的逆处理，用提取的比特恢复相应像素的 LSB，如图中的 8 被恢复为 9。限于篇幅，图 6(b)示例中有关数据提取剩下未表述的部分请读者根据文中 ACD 实现过程的描述自行完成。

5 实验结果及算法分析

为验证 ACD 的有效性，本文用 C++实现 ACD 和 3 个比较算法，执行程序可在文献[14]参阅。

ACD 创新主要体现在有效改善现有 PEE 算法中灰度溢出解决方法的不足，为便于验证这个创新的贡献，3 个比较算法采用与 ACD 相同的预测算子。3 个比较算法根据各自所采用的灰度溢出解决方法，分别被简称为 DE-FB、DE-HM1、DE-HM2。

1) DE-FB: 除了变换方向固定外（更改像素对的左边像素），其他处理方法与 ACD 相同。

2) DE-HM1: 除了预测算子改用相邻像素预测外，其他处理方法与罗剑高^[10]等 RDH 算法相同。

3) DE-HM2: 与 DE-HM1 相似，不同的是在直方图调整时，DE-HM1 把直方图上 $[0, \Delta]$ 和 $[255-\Delta, 255]$ 值分别调整到 $[\Delta+1, 2\Delta+1]$ 和 $[254-2\Delta, 254-\Delta]$ ，而 DE-HM2 加入了简单的自适应处理，在暗区，根据 $[0, \Delta]$ ， $[2\Delta+2, 3\Delta+2]$ 像素数量的大小决定合并区间，如果前者数小，则把 $[0, \Delta]$ 的像素值加 $\Delta+1$ 合并到区间 $[\Delta+1, 2\Delta+1]$ ，否则把 $[0, 2\Delta+1]$ 区间的像素值加 $\Delta+1$ ， $[\Delta+1, 2\Delta+1]$ 像素值合并到 $[2\Delta+2, 3\Delta+2]$ ，亮区的合并与暗区类似。本文需记录的溢出信息比特数是合并后区间的像素数量，显然，DE-HM2 所需的溢出信息比特量小于或等于 DE-HM1 所需的量，是 DE-HM1 灰度溢出法的一种改进，其付出代价是溢出比特量的减少意味着参与调整的像素数增加，这有可能相应地增加失真。

实验载体图来自文献[15]，待嵌入数据为随机比特流。大量的实验结果表明，ACD 具有非常优良的防灰度溢出能力。图 7 是实验中具有代表性的 4 幅测试图，分别代表灰度边界分布少、溢出问题可忽略的 Lena，灰度分布均匀的 Texmos1，灰度分布相对偏暗的 Man，灰度分布集中在 0 和 255 的 Ruler。图 8 显示的是作用在以上载体的 4 个算法的容量—失真性能实验结果对比。

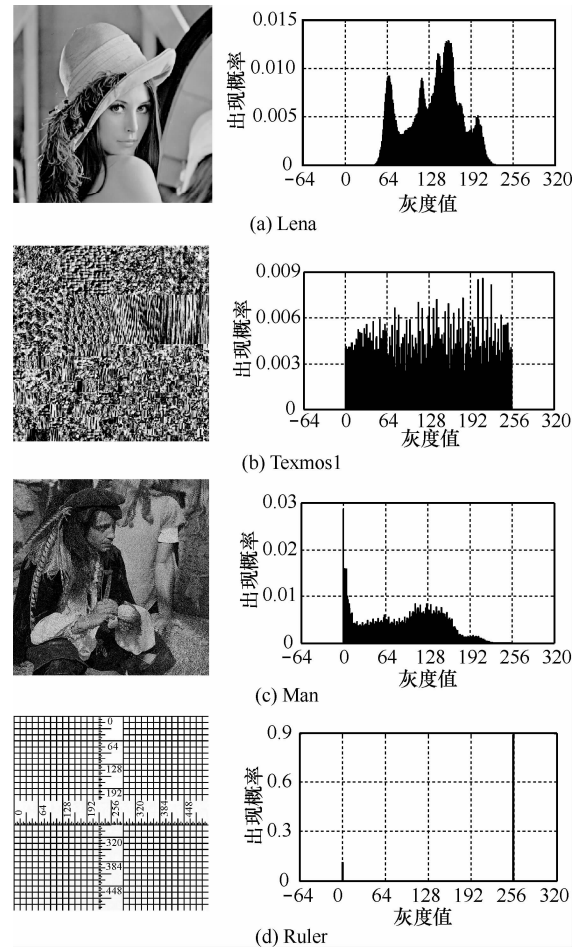


图 7 有代表性的测试图

图 8(a)中 4 条容量—失真曲线基本重合在一起，这是因为 Lena 在 0 或 255 附近的像素数量可忽略不计，用于处理溢出所付出的性能损失也可忽略不计。Texmos1 纹理丰富，灰度值均匀，0 或 255 附近的像素极易导致溢出问题，ACD 自适应处理能在不增加溢出信息量的情况下减少较大比例的溢出问题，不过这有可能会增加相关像素的变化值，图 8(b)表明 ACD 采用的自适应方法利大于弊，其算法性能较明显优于其他 3 种算法。Man 的实验对比结果与 Texmos1 的实验对比结果类似，只是图 8(c)

的数据嵌入容量相对较大, ACD 在减少灰度溢出信息方面所获的贡献相对总嵌入量的比例没有图 8(b) 那么明显。Ruler 的灰度值只有 0 和 255 这 2 种, 扩展变换极易导致溢出, 由于溢出辅助信息的急剧增加, DE-FB 和 DE-HM1 已无法在该图嵌入有效数据, DE-HM2 在这种载体场合表现优于 DE-HM1, 但总体上不如 ACD。由图 8 可看出, ACD 相对 3 种比较算法, 具有非常好的灰度溢出问题的处理效果。4 种算法比较而言, ACD 的优越性较为明显, 特别是在失真较大时 (此时 Δ 较大, 溢出问题不容忽视)。

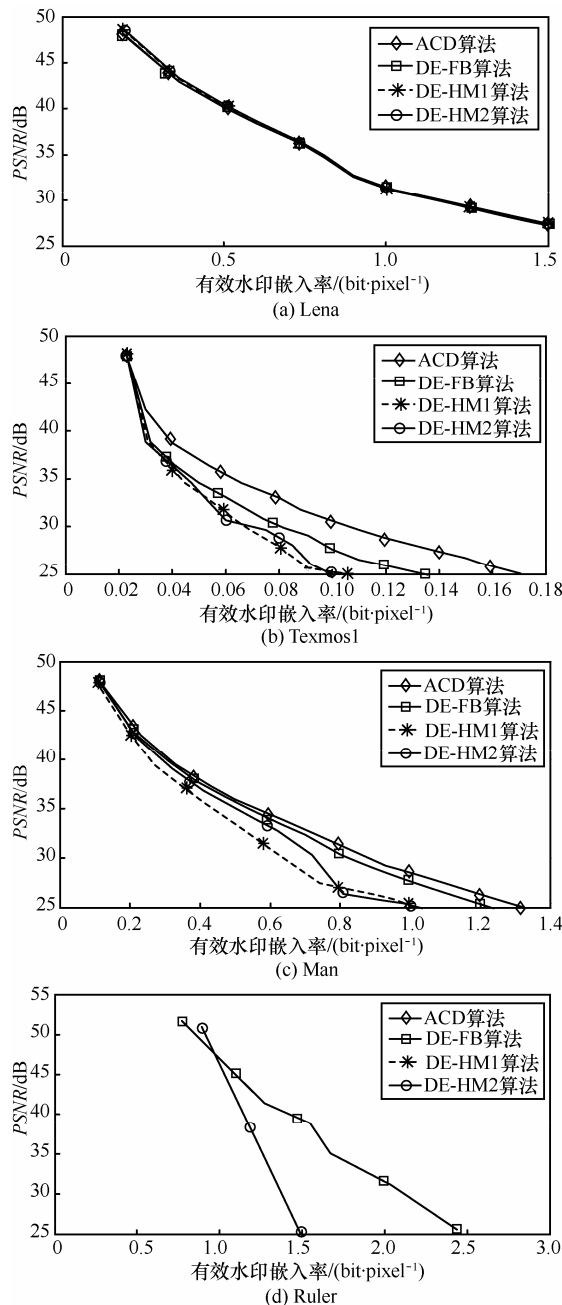


图8 算法容量—失真性能对比

6 结束语

RDH 技术经过十多年的研究, 虽取得了较大进步, 但总体还处在发展中, 更为有效的算法及 RDH 容量的理论边际、RDH 技术的实际应用等有待于科研人员做进一步的研究。本文从有效控制灰度溢出的角度提出了一种变换方向自适应的 RDH 算法——ACD。ACD 在像素对中执行像素差值扩展变换, 当估测像素对处在暗区 (灰度值靠近 0) 时, 以小值作为大值的预测值, 更改大值, 当估测像素对处在亮区 (灰度值靠近 255) 时, 以大值作为小值的预测值, 更改小值, 否则以像素对右值作为左值的预测值, 更改左值。理论分析及实验结果表明, ACD 有效利用局部特征自适应调整像素对的变换方向, 明显地改善了灰度溢出问题, 解决了现有灰度溢出解决方案使载体在偏暗或偏亮时因灰度溢出问题导致有效嵌入容量不稳定的缺陷, ACD 稳定的容量—失真性能, 适用于各种灰度直方图分布的图像载体。ACD 的优越性验证了变换方向自适应思想的有效性, 把这个思想用于其他预测误差算子的 PEE 是下一步的研究方向。

参考文献:

- [1] ASIFULLAH K, AYESHA S, SUMMUYA M, et al. A recent survey of reversible watermarking techniques[J]. Information Sciences 2014,279: 251-272.
- [2] TIAN J. Reversible data embedding using a difference expansion[J]. IEEE Transactions on Circuits and Systems for Video Technology. 2003,13(8):890-896.
- [3] THODI D M, RODRIGUEZ J J. Expansion embedding techniques for reversible watermarking[J]. IEEE Transactions on Image Processing, 2007,16(3):721-730.
- [4] VASILYIY S, HYOUNG J K, JEHO N, et al. Reversible watermarking algorithm using sorting and prediction[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2009,19(7):989-999.
- [5] CHEN M, CHEN ZY, ZENG X, et al. Model order selection in reversible image watermarking[J]. IEEE Journal of Selected Topics in Signal Processing, 2010, 4(3):592-604.
- [6] 宣国荣, 姚秋明, 柴佩琪, 等. 基于整数小波阈值嵌入的无损数据隐藏[J]. 计算机应用, 2006, 26(12): 2890-2893.
- [7] XUAN G L, YAO Q M, CHAI P Q, et al. Lossless data hiding using integer wavelet transform and threshold embedding technique[J]. Computer Applications, 2006, 26(12):2089-209.
- [8] LEE C F, CHEN H L, TSO H K. Embedding capacity raising in reversible data hiding based on prediction of difference expansion[J]. Journal of Systems and Software, 2010,83(10):1864-1872.
- [9] PENG F, LI X L, YANG B. Adaptive reversible data hiding scheme based on integer transform[J]. Signal Processing, 2012, 92(1):54-62.
- [10] WANG J X, NI J Q, PAN J W. A high capacity reversible data hiding

scheme based on generalized prediction-error expansion and adaptive embedding[J]. *Signal Processing*, 2014, 98(5): 370-380.

- [10] 罗剑高, 韩国强. 基于完整上下文预测的可逆数据隐藏[J]. *计算机科学*, 2013, 40(11A): 219-223.

LUO J G, HAN G Q. Reversible data hiding based on full context prediction[J]. *Computer Science*, 2013, 40(11A): 219-223.

- [11] 罗剑高, 韩国强, 沃焱, 等. 篡改定位精度可动态调整的无损图像认证算法[J]. *华南理工大学学报(自然科学版)*, 2011, 39(7): 121-126.

LUO J G, HAN G Q, WO Y, et al. Lossless image authentication algorithm with dynamic adjustable tamper localization accuracy[J]. *Journal of South China University of Technology (Natural Science Edition)*, 2011, 39(7): 121-126.

- [12] 罗剑高, 韩国强, 沃焱, 等. 基于自适应图像块组合的无损图像认证算法[J]. *通信学报*, 2012, 33(6): 64-72.

LUO J G, HAN G Q, WO Y, et al. Lossless image authentication algorithm based on adaptive combinations of image basic blocks[J]. *Journal on Communications*, 2012, 33(6): 64-72.

- [13] 邓小鸿, 陈志刚, 邓惠文, 等. 医学图像可逆数字水印防溢出处理研究[J]. *计算机工程与应用*, 2013, 49(23): 162-165.

DENG X H, CHEN Z G, LIU H W, et al. Study on preventing overflow and underflow of reversible digital watermarking for medical images[J]. *Computer Engineering and Applications*, 2013, 49(23): 162-165.

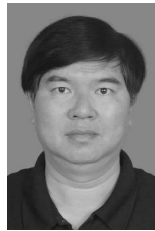
- [14] Miecuc R&D center, ACD Demo software [EB/OL]. [2015-1-11] <http://www.miecuc.net/research/rdh/>.

- [15] Signal and Image Processing Institute, University Southern California, Los Angeles. Image database[EB/OL]. [2015-1-11] <http://sipi.usc.edu/database/>.

作者简介:



罗剑高 (1971-), 男, 江西金溪人, 博士, 广东农工商职业技术学院副教授, 主要研究方向为图像处理、数字水印、信息安全等。



韩国强 (1962-), 男, 江西临川人, 博士, 华南理工大学教授、博士生导师, 主要研究方向为图像处理、多媒体技术等。



沃焱 (1975-), 女, 云南昆明人, 博士, 华南理工大学教授, 主要研究方向为图像处理、数字水印、信息安全等。