高效 $\pm k$ 自适应图像隐写术

陈嘉勇 [1, 2] 张卫明 [2, 3] 韩 涛 [2] 祝跃飞 [2] 郭东辉 [1]

摘 要 自适应隐写可避免修改载体的敏感区域, 有效提高隐写术的安全性. 总结分析了现有自适应隐写的边信息同步方法. 通过推广单调纹理函数思想, 构造一种基于自由度的纹理保序函数, 实现了自适应隐写边信息的快速同步. 结合湿纸编码和多层嵌入思想, 提出自适应 ± 1 和 ± 2 隐写算法. 新算法既保证通信双方能快速同步边信息, 同时具有很高的抗检测性能. 理论分析和实验结果均表明了算法的有效性.

关键词 信息隐藏, 隐写术, 自适应隐写, 湿纸编码, 边信息

引用格式 陈嘉勇, 张卫明, 韩涛, 祝跃飞, 郭东辉. 高效 $\pm k$ 自适应图像隐写术. 自动化学报, 2013, **39**(10): 1594−1601

DOI 10.3724/SP.J.1004.2013.01594

An Efficient Adaptive Image Steganographic Method for $\pm k$ Embedding

CHEN Jia-Yong[1, 2] ZHANG Wei-Ming[2, 3] HAN Tao[2] ZHU Yue-Fei[2] GUO Dong-Hui[1]

Abstract Adaptive steganography which avoids changing the sensitive area in carrier, can effectively improve the security of steganography. Firstly, the schemes of side information synchronization in adaptive steganography are summarized and analyzed. Secondly, using the freedom degree which is extented by the previous one-way texture functions, a new order-preserving texture function as side information of adaptive steganography is constructed. Finally, by combining wet paper codes and the method of multi-layered embedding, ± 1 algorithm and ± 2 algorithm are proposed. The proposed scheme can efficiently enhance the security of adaptive steganography while maintaining fast side information synchronization between communicators. Theoretical analysis and experimental results show the efficiency of the proposed scheme.

Key words Information hiding, steganography, adaptive steganography, wet paper coding, side information

Citation Chen Jia-Yong, Zhang Wei-Ming, Han Tao, Zhu Yue-Fei, Guo Dong-Hui. An efficient adaptive image steganographic method for $\pm k$ embedding. *Acta Automatica Sinica*, 2013, **39**(10): 1594−1601

信息隐藏是将消息嵌入多媒体数据中的一种技术, 可用于隐蔽传输或版权保护等目的. 隐写术[1]是信息隐藏的主要分支, 主要考虑如何实现隐蔽通信.

自适应隐写[2] 可以根据载体纹理特性, 在信息嵌入过程中, 尽量避免对敏感区域的修改, 从而提高抗检测能力. 对空域图像自适应隐写研究表明[3]: 通过修改纹理复杂的区域嵌入消息可以有效抵抗检测. 但是由于区域纹理复杂性度量在嵌入消息后会发生变化, 导致接收者无法定位嵌入区域提取消息. 解决方法之一是采用湿纸编码[4], 接收方无需知道嵌入位置, 但计算复杂度高. 湿纸编码是由 Fridrich 提出的, 发送者可借助湿纸编码在载体的某些位置被限制修改的情况下嵌入信息, 而接收者无需知道哪些位置是嵌入位置即可提取信息.

Filler 等[5] 提出的 STC 编码是一种更灵活的适用于自适应隐写的编码方法, 它允许发送方对载体自由定义嵌入失真, 通过 STC 编码可以用最小的失真代价嵌入消息, 接收者不需要关于失真定义的任何信息即可提取消息. 其指出有两种失真定义方式: 一是加性失真; 二是非加性失真. 所谓加性失真即对每个像素独立定义失真, 全局失真以被修改像素的失真和计算; 而非加性失真的定义要考虑相邻像素之间的关系. 但是, 无论采用哪种方式, 如何合理定义失真才能有效反映抗检测性能, 仍是一个困难的问题.

无论是湿纸编码还是 STC 编码, 计算复杂度都比较高, 影响嵌入速度. 一种快速的自适应隐写方法是设计巧妙的消息嵌入策略, 使消息嵌入后接收方

收稿日期 2012-01-12 录用日期 2012-08-02
Manuscript received January 12, 2012; accepted August 2, 2012
国家自然科学基金 (60803155, 61170234, 60970141, 60902102, 61274133), 郑州市科技创新团队项目 (10CXTD150), 中国科学院战略性先导专项课题 (XDA06030601), 国家重大科技专项 (2010ZX03004-003) 资助
Supported by National Natural Science Foundation of China (60803155, 61170234, 60970141, 60902102, 61274133), Science and Technology Innovation Team of Zhengzhou (10CXTD150), Strategic and Piloted Project of Chinese Academy of Sciences (XDA06030601), National Science and Technology Major Project of China (2010ZX03004-003)
本文责任编委 刘一军
Recommended by Associate Editor LIU Yi-Jun
1. 厦门大学信息科学与技术学院 厦门 361005 2. 解放军信息工程大学信息工程学院 郑州 450002 3. 中国科学技术大学信息科学技术学院 合肥 230026
1. School of Information Science and Technology, Xiamen University, Xiamen 361005 2. School of Information Engineering, PLA Information Engineering University, Zhengzhou 450002 3. School of Information Science and Technology, University of Science and Technology of China, Hefei 230026

仍可定位消息嵌入位置. 例如: 文献 [6−8] 使用特定修改方式保持嵌入区域像素的纹理复杂度单调上升, 但这种同步方式导致每个像素修改模式受限, 使像素负载消息的能力降低; 文献 [9−10] 在嵌入消息后引入额外修改调整纹理复杂度, 但这种同步方式增加了修改量和修改幅度. 文献 [11] 使用 2 × 4 分块作为一个单元计算纹理复杂度, 并根据不同的纹理复杂度选择不同的编码参数; 纹理复杂度越高, 选择的编码嵌入率越大, 充分利用纹理复杂区域嵌入信息; 在同步边信息时, 利用计算纹理复杂度公式中的模 2 取整, 通过控制像素的修改方向, 保证了嵌入前后纹理复杂度一致.

针对现有自适应隐写算法的不足, 本文推广了文献 [6] 等提出的单调纹理函数, 构造了一种新的基于纹理的加性失真度量函数, 并结合湿纸编码和多层嵌入思想, 设计高效的自适应图像隐写算法.

本文内容安排如下: 第 1 节作为预备知识简介湿纸编码算法; 第 2 节讨论边信息同步方法并提出 ±1 算法、±2 算法; 第 3 节给出实验结果; 第 4 节总结与讨论.

1 预备知识

1.1 基本定义

下面以灰度图像为例描述本文方法. 假设载体为 $M \times N$ 的 256 级灰度图像: $\boldsymbol{x} = \{x_{i,j}\}$, $1 \le i \le M$, $1 \le j \le N$, $x_{ij} \in [0, 255]$. 记嵌入率为 r, 嵌入消息后得到的载密图像为 $\boldsymbol{y} = \{y_{i,j}\}$. 在隐写术中为了安全, 载体信号的修改幅度通常要受到一定限制. 设最大允许的修改幅度为 f, 则修改幅度值的集合为 $Z = \{f, f+1, \cdots, f\}$. 在实际应用中, 考虑到修改幅度太大会降低其隐蔽安全性, 通常考虑 $f = 1$ 和 2 的情况, 此时分别对应 ± 1 和 ± 2 隐写. 嵌入率定义为所嵌入的消息比特数与载体长度的比, 记为 $e = n/N$, 表示每个载体符号承载的信息量.

1.2 湿纸编码

湿纸编码可在载体的某些位置被限制修改的情况下嵌入信息, 而接收者无需知道哪些位置是受限的即可提取信息. 与一般隐写码的不同之处在于, 发送方只能通过修改载体中 “干” 的位置嵌入 $\boldsymbol{m}$.

假设某段载体 $\boldsymbol{s} = (s_1, \cdots, s_n)$ 中有 l 个位置允许修改 (即 “干” 的位置), 其余 $n-l$ 比特不允许修改 (即 “湿” 的位置). 记允许修改载体位为 $\{s\}_j$, $j \in L \subset \{1, 2, \cdots, n\}$, $|L| = l$. 发送方修改 $\boldsymbol{s}$ 中 q 个 “干” 的位置, 利用嵌入矩阵 K 将消息 $\boldsymbol{m}$ 嵌入 $\boldsymbol{s}$, 得到 $\boldsymbol{c}$. 其中, $q \le l \le n$, 且满足

$$K\boldsymbol{c} = \boldsymbol{m} \tag{1}$$

取 $\boldsymbol{v} = \boldsymbol{c} - \boldsymbol{s}$, 则

$$K\boldsymbol{c} = \boldsymbol{m} - K\boldsymbol{s} \tag{2}$$

由于 $\boldsymbol{s}$ 中有 $n-l$ 个位置不允许修改, 故 $\boldsymbol{v}$ 对应的 $n-l$ 个分量只能取 0, 即 $\boldsymbol{v}$ 中只有 l 个不确定分量, 其余的 $n-l$ 个分量为 0. 故对所有 v_i $(i \notin L)$, 从 K 中去掉第 i 列, 共去掉 $n-l$ 列, 所得矩阵记为 H, 相应的在 $\boldsymbol{v}$ 中删去 $n-l$ 个列向量 v_i, $i \notin L$, 记为 $\boldsymbol{u}$. 从而 (2) 可改写为

$$H\boldsymbol{u} = \boldsymbol{m} - K\boldsymbol{s} \tag{3}$$

这里, H 是 $q \times l$ 维矩阵. 通过求解式 (3) 即完成湿纸编码的编码.

载体湿率记为 $a_{\text{wet}} = \frac{n-l}{n}$, 其中, $0 < a_{\text{wet}} < 1$. 事实上, 隐写码也可视为一种特殊的湿纸编码, 即 $a_{\text{wet}} = 0$ 的湿纸编码. 湿纸编码的嵌入率记为 $r_{\text{wet}} = q/l$, 特别的, 隐写码的嵌入率为 q/n.

2 自适应隐写算法

首先, 提出一种新的自适应隐写边信息同步方法; 其次, 提出自适应隐写算法; 最后, 进一步探讨自适应隐写的边信息同步问题.

2.1 边信息同步

边信息同步问题是自适应隐写的关键问题. 对自适应图像隐写术而言, 所谓边信息同步是指隐写接收方根据载密图像的纹理信息, 获得与发送方相同的关于消息嵌入位置的信息. 为了提高计算速度, 自适应隐写术通常应尽量避免采用计算复杂度较高的湿纸编码来同步边信息. 目前快速同步边信息算法的基本原理为: 发送方构造具有保序性质的纹理复杂度刻画函数 (简称 “纹理保序函数”), 使消息嵌入前后关于隐写区域的纹理度量保持某种次序, 从而使得消息接收方能正确同步消息的嵌入区域, 提取嵌入信息.

设载体最大允许的修改幅度为 Δ, 则修改幅度值集合为 $d(\Delta) = \{-\Delta, -\Delta+1, \cdots, 0, \cdots, \Delta\}$.

定义模化函数 $\rho(\cdot)$ 为

$$\rho(i) = i - i \bmod (2t+1) \tag{4}$$

其中, $t \ge \Delta$ 为模参数. 定义像素 $x_{i,j}$ 的自由度函数 $FR(\cdot)$ 为

$$FR(x_{i,j}, \Delta, t) = \min\{\Delta, t - |x_{i,j} \bmod (2t+1) - t|\} \tag{5}$$

显然, 有 $0 \le FR(\cdot) \le \Delta$. 对 $x_{i,j} \in |x$, 定义纹理复

杂度函数 $TX(\cdot)$ 如下:

$$\begin{aligned} TX\left(x_{i,j}\right) = |&8\rho\left(x_{i,j}\right) - \\ &\rho\left(x_{i-1,j-1}\right) - \rho\left(x_{i-1,j}\right) - \\ &\rho\left(x_{i-1,j+1}\right) - \rho\left(x_{i,j-1}\right) - \\ &\rho\left(x_{i,j+1}\right) - \rho\left(x_{i+1,j-1}\right) - \\ &\rho\left(x_{i+1,j}\right) - \rho\left(x_{i+1,j+1}\right)| \end{aligned} \tag{6}$$

对像素 $x_{i,j} \in \boldsymbol{x}$ 而言, 若对 $x_{i,j}$ 的任意修改方式 $a \in d(\Delta)$, 有 $TX\left(\Omega\left(y_{i,j}\right)\right) \geq TX\left(\Omega\left(x_{i,j}\right)\right)$, 则称 $x_{i,j}$ 为自由点 (F 点), 否则称 $x_{i,j}$ 为受限点 (R 点). 特别的, F 点可视为自由度为 Δ 的 R 点. 记 F 点的集合为 C_F, R 点的集合为 C_R, 则 $\boldsymbol{x} = C_F \cup C_R$. 根据模化函数的定义, 为使 $\rho(x_{i,j}) = \rho(y_{i,j})$, 对 $x_{i,j}$ 的修改幅度不能超过其自由度.

取 $\Delta = 1$, $t = 2$, 把图像 "lena.bmp" 纹理复杂度最高的 βN 个点作为消息嵌入区域, 则 β 取不同值情况下, 由上述纹理复杂度定义得到的嵌入区域如图 1 所示.

(a) lena.bmp (b) $\beta = 0.1$ (c) $\beta = 0.3$

图 1 消息嵌入位置

Fig. 1 The position of embedding messages

由图 1 可知, $TX(\cdot)$ 可较准确地刻画载体的纹理复杂度, 利用 $TX(\cdot)$ 选取消息嵌入区域具有较高的安全性.

以 $TX(\cdot)$ 作为自适应隐写的纹理保序函数, 对接收方而言, 若所有载体像素的修改幅度均不超出其自由度, 则对任意 $x_{i,j}$, 有 $TX(y_{i,j}) = TX(x_{i,j})$. 此时, 载体嵌入信息后不会影响接收者对嵌入位置的识别, 即通信双方可完成边信息同步.

下面举例说明如何对像素进行嵌入修改, 使边信息保持同步.

例如, 当 $\Delta = 2$, $t = 2$ 时, 对嵌入区域中的像素点 $x_{i,j}$, 有 $\rho(x_{i,j}) = x_{i,j} - x_{i,j} \bmod 5$, $FR(x_{i,j}) = \min\{2, 2 - |x_{i,j} \bmod 5 - 2|\}$. 为使 $TX(y_{i,j}) \geq TX(x_{i,j})$, 对 $x_{i,j}$ 的修改需满足以下规则: 若 $x_{i,j} \bmod 5 = 2$, 则 $x_{i,j}$ 为自由度为 2 的 F 点, 嵌入可选择不变、+1、−1、+2 或 −2 共 5 种修改方式; 若 $x_{i,j} \bmod 5 = 1$ 或 3, 则 $x_{i,j}$ 为自由度为 1 的 R 点, 嵌入可选择不变、+1 或 −1 三种修改方式; 若 $x_{i,j} \bmod 5 = 0$ 或 4, 则 $x_{i,j}$ 为自由度为 0 的 R 点. 嵌入可选择不变、+1 (或 −1) 两种修改方式中的一种.

需要注意的是, 在实际应用中, 若遇到像素灰度值饱和的情况, 像素的自由度需要特殊处理. 例如, 当 $\Delta = 1$, $t = 3$ 时, 若 $x_{i,j} = 255$, 则 $FR(x_{i,j}) = 1$, 但为避免修改值溢出, 255 只能减 1, 故应将其自由度定义为 0. 若饱和像素的比例较小, 此特殊情况发生的概率很小, 故对整体嵌入性能的影响可忽略不计.

下面给出具体的自适应隐写算法. 首先, 结合湿纸编码和双层编码思想, 提出自适应 ± 1 隐写算法. 然后, 结合湿纸编码和三层编码思想, 提出自适应 ± 2 隐写算法.

2.2 ± 1 隐写

为便于说明嵌入过程, 下面把载体序列视为一维数组. 记灰阶载体 $\boldsymbol{x} = (x_1, \cdots, x_N)$, $x_i \in [0, 255]$, 用 $L(\boldsymbol{x}) = (L(x_1), \cdots, L(x_N))$ 表示 LSB 层, $S(\boldsymbol{x}) = (S(x_1), \cdots, S(x_N))$ 表示次 LSB 层. 不妨记嵌入区域的像素占比为 β, 取 $\Delta = 1$, $t = 2$.

首先, 以 $\rho(\boldsymbol{x})$ 为边信息, 选择纹理复杂度 $\rho(x_{i,j})$ 最大的 βN 个象素作为消息嵌入区域. 其次, 在载体的 LSB 层进行嵌入, 在得到 LSB 层的预期修改位置后, 仅对自由度为 0 的像素进行修改, 并不直接修改自由度为 1 的载体像素值. 把需要修改且自由度为 1 的像素标记为 "干", 其余像素点标记为 "湿", 从而在次 LSB 层构造了一条新的湿纸隐写信道. 采用文献 [12] 中的 LT 方法, 在次 LSB 层的湿纸信道上完成消息的嵌入. 接收方以 $\rho(\boldsymbol{y})$ 为边信息, 定位消息嵌入区域, 对嵌入区域的 LSB 序列和次 LSB 层分别执行 LSB 提取和湿纸编码的解码算法即可提取消息.

记嵌入区域中载体中自由度为 0、1 的像素的比例分别为 p_0 和 p_1. 综上, 载体的 LSB 层可嵌入 βN 比特信息, 载体的次 LSB 层可嵌入 $\frac{p_1 \beta N}{2}$ 比特信息, 两层最多可嵌入 $\left(1 + \frac{p_1}{2}\right)\beta N$ 比特信息. 若给定消息嵌入率 r, 则发送方只需先计算 $\beta = \frac{r}{1 + 0.5 p_1}$, 然后采用上述算法, 即可在 βN 个像素中嵌入 rN 比特信息.

考虑模化函数的定义, 在平均意义下, 有 $p_0 = 20\,\%$, $p_1 = 80\,\%$. 相较于文献 [6] 提出的嵌入方法, 本文 ± 1 算法对像素的修改数量大约可减少 $28.57\,\%$.

下面用一个例子说明上述 ± 1 隐写的嵌入过程.

例 1. 如图 2 所示, 设载体的嵌入区域包含 8 个像素 $\boldsymbol{x} = (25, 26, 25, 33, 48, 49, 53, 32)$, 待嵌消息序列为 $\boldsymbol{m} = (1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1)$.

步骤 1. 在 $\boldsymbol{x}$ 的 LSB 层 $L(\boldsymbol{x})$ 嵌入前 8 比特消息, 即 $\boldsymbol{m}_1^8 = (1, 0, 0, 0, 0, 1, 0, 1)$, 并设嵌入过程需修改第 3、第 4、第 7 和第 8 个像素的 LSB 位. 分两种情况考虑, 对自由度为 0 的第 3 像素只有 +1

这种修改方式, 即直接把第 3 个像素加 1. 自由度为 1 的像素可有 $+1$ 和 -1 两种修改方式, 如何修改由第二步的嵌入过程决定.

步骤 2. 将第一层修改后载体的次 LSB 层 $S(\boldsymbol{x})$ 的第 4、7 和 8 位设置为 "干" 的位置, 其余位置作为 "湿" 的位置 (即阴影位置), 然后用湿纸编码在 $S(\boldsymbol{x})$ 上嵌入消息, 假设这一步可以嵌入 3 比特消息, 即 $\boldsymbol{m}_9^{11}=(0,0,1)$, 并设嵌入过程需要修改 $S(\boldsymbol{x})$ 的第 4 和第 8 个比特. 为完成前两步的嵌入, 对第 4 和第 8 个像素需要同时修改 LSB 位和次 LSB 位, 第 7 个像素只需要修改 LSB 位. 注意到奇数加 1, 或偶数减 1 可以实现同时修改 LSB 位与次 LSB 位, 所以将第 4 个像素的灰度值加 1, 第 8 个像素的灰度值减 1; 第 7 个像素的灰度值为奇数, 令其减 1, 则只修改 LSB 位, 而次 LSB 位不动.

综上, 上述算法通过对 4 个像素作加 1 或减 1 修改嵌入了 11 比特消息. 接收方提取消息时, 先根据边信息定位消息嵌入区域, 提取载密图像嵌入区域的 LSB 层, 再用湿纸编码的解码算法对嵌入区域的 LSB 层进行提取.

注意, 湿纸编码在相对长的载体上才能够实现, 此例中只采用 8 个像素是为说明双层嵌入构造的修改过程.

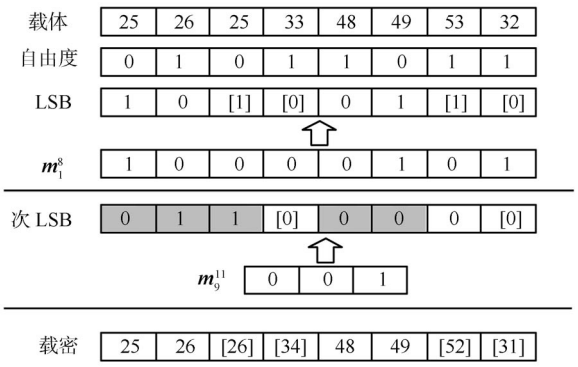


图 2　± 1 自适应隐写流程

Fig. 2　The adaptive embedding procedure of ± 1

2.3　± 2 隐写

记灰阶载体 $\boldsymbol{x}=(x_1,\cdots,x_N)$, $x_i\in[0,255]$, 用 $L(\boldsymbol{x})=(L(x_1),\cdots,L(x_N))$ 表示其 LSB 层, $S(\boldsymbol{x})=(S(x_1),\cdots,S(x_N))$ 表示次 LSB 层, $T(\boldsymbol{x})=(T(x_1),\cdots,T(x_N))$ 表示第三 LSB 层.

取 $\Delta=2$, $t=2$. 首先以 $\rho(\boldsymbol{x})$ 为边信息, 选取纹理度量 $\rho(x_{i,j})$ 最大的 βN 个载体像素作为嵌入区域, 其中 β 为关于嵌入率 r 的函数. 记嵌入区域中自由度为 0、1 和 2 的点的比例分别为 p_0、p_1 和 p_2.

第一层嵌入在嵌入区域的 LSB 层 $L(\boldsymbol{x})=(L(x_1),\cdots,L(x_N))$ 嵌入信息. 其中, 得到所有需要修改的像素点, 对自由度等于 0 的点采用加 1 或减 1 的修改方式直接修改, 对自由度大于 0 的点只记录其预期修改目标, 并不直接修改. 在载体 LSB 层可嵌入 βN 比特信息.

第二层嵌入在第一层修改后得到的载体的次 LSB 层 $S(\boldsymbol{x})=(S(x_1),\cdots,S(x_N))$ 嵌入信息. 将第一层的 $p_0\beta N$ 个自由度等于 0 位置记为 "湿" 的位置, $\frac{(1-p_0)\beta N}{2}$ 个待定修改点的位置记为 "干" 的位置. 对其余 $\frac{(1-p_0)\beta N}{2}$ 个位置, 若像素自由度大于 1, 则记为 "干" 的位置, 共 $\frac{p_2(1-p_0)\beta N}{2(p_1+p_2)}$ 个; 否则记为 "湿" 的位置, 共 $\frac{p_1(1-p_0)\beta N}{2(p_1+p_2)}$ 个. 由于第一层嵌入后平均需要对 $\frac{(1-p_0)\beta N}{2}$ 个 LSB 位修改, 不失一般性, 假设 $\frac{(1-p_0)\beta N}{2}$ 为整数, 且刚好有 $\frac{(1-p_0)\beta N}{2}$ 个位置需修改, 通过在修改时选择加 1 或减 1, 发送方可以自由控制这个像素的次 LSB 位. $S(\boldsymbol{x})$ 中的另外 $\frac{p_2(1-p_0)\beta N}{2(p_1+p_2)}$ 个 "干" 的位置如需要修改, 可通过对灰度值加 2 或减 2 来实现. 因此, 采用湿纸编码可在次 LSB 层嵌入 $\left(1+\frac{p_2}{p_1+p_2}\right)\frac{(1-p_0)\beta N}{2}$ 比特信息.

第三层嵌入在第二层修改后得到的载体的第三 LSB 层 $T(\boldsymbol{x})$=$(T(x_1),\cdots,T(x_N))$ 嵌入信息. 第二层嵌入平均需要对 $\frac{p_2(1-p_0)\beta N}{4(p_1+p_2)}$ 个灰度值进行加 2、减 2 操作, 仍假设 $\frac{p_2(1-p_0)\beta N}{4(p_1+p_2)}$ 为整数, 且刚好有 $\frac{p_2(1-p_0)\beta N}{4(p_1+p_2)}$ 个位置需要加 2、减 2. 通过选择加 2 或减 2, 发送方可以自由控制这 $\frac{p_2(1-p_0)\beta N}{4(p_1+p_2)}$ 个灰度值的第三 LSB 位. 限定载体中其他 $\frac{p_1(1-p_0)\beta N}{4(p_1+p_2)}$ 个位置不能变动, 则用湿纸编码可在 $T(\boldsymbol{x})$ 上嵌入信息. 因此, 采用湿纸编码可在载体的第三 LSB 层嵌入 $\frac{p_2(1-p_0)\beta N}{4(p_1+p_2)}$ 比特信息.

综上, 上述三层嵌入方法的嵌入容量为 $\left(1+\frac{(1-p_0)(p_1+2.5p_2)}{2(p_1+p_2)}\right)\beta N$. 给定消息嵌入率 r, 则发送方只需计算 $\beta=\frac{r}{1+\frac{(1-p_0)(p_1+2.5p_2)}{2(p_1+p_2)}}$, 然后采用上述 ± 2 算法, 即可在 βN 个像素中嵌入 rN 比特信息. 其中, 采用文献 [12] 中的 LT 方法, 在次 LSB 层和第三 LSB 层湿纸信道上进行消息嵌入.

考虑模化函数的定义, 在平均意义下, 有 $p_0=40\,\%$、$p_1=40\,\%$ 和 $p_2=20\,\%$. 因此, 本文提出的 ± 2 算法可在 βN 个像素点中嵌入约 $1.45\beta N$ 比特信息, 本文算法对像素的修改数量大约可减少 $31.03\,\%$.

接收方提取消息时, 先定位载密图像的消息嵌入区域, 提取嵌入区域的 LSB 层, 再用湿纸编码的解码算法对嵌入区域的次 LSB 层和第三 LSB 层分别进行提取即可.

下面用一个例子说明上述隐写的消息嵌入过程.

例 2. 如图 3 所示, 设载体的嵌入区域包含 8 个像素 $\boldsymbol{x} = (25, 26, 25, 33, 48, 49, 53, 32)$, 待嵌消息序列为 $\boldsymbol{m} = (1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1)$.

步骤 1. 在 $L(\boldsymbol{x})$ 的 LSB 层嵌入前 8 比特消息, 即 $\boldsymbol{m}_1^8 = (1, 0, 0, 0, 0, 1, 0, 1)$, 并设嵌入过程需修改第 2、第 3、第 4 和第 7 个像素的 LSB 位. 分两种情况考虑, 对自由度为 0 的第 3 个像素只有加 1 这种修改方式, 即直接把第 3 像素加 1; 自由度大等于 1 的像素至少有 $+1$ 和 -1 两种修改方式, 如何修改由步骤 2 的嵌入过程决定.

步骤 2. 将经步骤 1 修改后的载体的次 LSB 层 $S(\boldsymbol{x})$ 的第 2、4、7 和 8 位设置为 "干" 的位置, 其余位置作为 "湿" 的位置 (即阴影位置). 其中, 第 8 位定义为 "干" 位置的原因是其自由度为 2, 而第 2、4 和 7 位是因为其 LSB 位需要修改且自由度大等于 1. 然后, 用湿纸编码在 $S(\boldsymbol{x})$ 上嵌入消息, 假设这一步可以嵌入 4 比特消息, 即 $\boldsymbol{m}_9^{12} = (0, 0, 1, 1)$, 并设嵌入过程需要修改 $S(\boldsymbol{x})$ 的第 2、7 和 8 位. 为完成第一、二步的嵌入, 对第 2 和第 7 个像素需要同时修改 LSB 位和次 LSB 位, 而第 4 个像素只需要修改 LSB 位, 对第 8 个像素只需要修改次 LSB 位. 注意到奇数加 1, 或偶数减 1 可以实现同时修改 LSB 位与次 LSB 位, 所以发送方将第 2 和第 7 个像素的灰度值加 1, 第 4 个像素的灰度值减 1; 第 8 个像素暂不修改.

步骤 3. 将经步骤 2 修改后的载体的第三 LSB 层 $T(\boldsymbol{x})$ 的第 8 位设置为 "干" 的位置, 其余位置作为 "湿" 的位置 (即阴影位置). 假设这一步可以嵌入 1 比特消息, 即 $\boldsymbol{m}_{13}^{13} = (1)$, 并设嵌入过程需要修改 $T(\boldsymbol{x})$ 的第 8 位. 为完成第二、三步的嵌入, 对第 8 个像素需同时修改其次 LSB 层和第三 LSB 层. 对其进行减 2 操作可完成同时修改这两层, 且不影响 LSB 层.

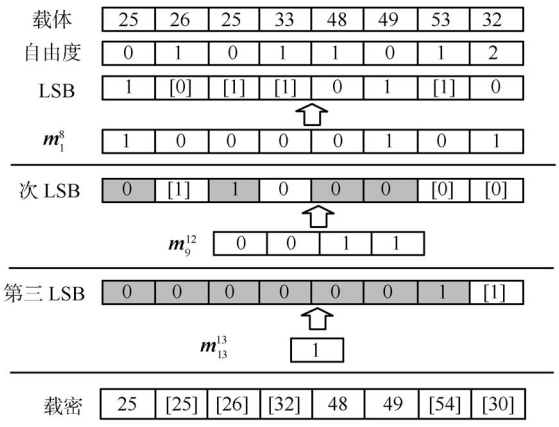


图 3 ± 2 自适应隐写流程

Fig. 3 The adaptive embedding procedure of ± 2

综上, 上述过程通过对 5 个像素作加减 1 或加减 2 修改嵌入 13 比特消息.

2.4 边信息同步问题的讨论

自适应隐写通常将载体中纹理最复杂的区域选为隐写区域. 为实现快速边信息同步, 通信双方通常会定义某种关于图像局部区域纹理信息的函数, 对相同图像区域而言, 该函数在消息嵌入前后保持偏序性, 本文称该类函数为纹理保序函数. 一般地, 对任意像素 $x_{i,j} \in \boldsymbol{x}$ 和嵌入修改方式 $a \in d(\Delta)$, 纹理保序函数是关于 $x_{i,j}$ 邻域像素集合的函数. 其中, 隐写区域对应的纹理保序函数在消息嵌入后的取值增加或不变, 而非隐写区域对应的纹理保序函数在隐写后的取值不变或减少. 因此, 通信双方利用上述纹理保序函数即可实现对消息嵌入区域的同步.

在限失真隐写信道中, 根据修改方式是否受到限制, 所有载体像素可分为两种类型: 自由点 (F 点) 和受限点 (R 点). 限失真 (像素修改幅度 $\leq \Delta$) 自适应隐写信道本质上是由若干 R 点和 F 点组成的混合进制隐写信道. R 点的处理策略影响自适应隐写算法的性能. 考察现有的自适应隐写术, 目前对 R 点的处理方法可分为两种:

1) 把 F 点视为 R 点, 所有点的嵌入策略都按 R 点的嵌入策略执行. 例如: 文献 [8] 提出的边缘区域隐写算法和文献 [6] 提出的 NRE 隐写算法均采用该方法. 文献 [8] 采用载体上互不重叠的 3×3 窗口定义像素失真, 并选择嵌入失真最小的区域进行嵌入. 但只在中间点上嵌入消息, 其余八个点作为对其纹理复杂度 D 的刻画. 为同步嵌入消息, 其在嵌入时首先取门限值 θ, 当 D 不小于 θ 和 D 不大于 $-\theta$ 时才进行嵌入. 嵌入过程中对每个载密像素通过选择 $+1$ 或 -1 操作嵌入 1 比特信息, 通过选择载体像素的修改方向保证不会将新的 D 值置于范围 $-\theta < D < \theta$. 文献 [6] 则采用载体上互不重叠的 2×2 窗口定义像素纹理噪声 NL (Nosiy level), 并选择纹理最复杂的区域进行嵌入. 为实现嵌入消息同步, 其在嵌入时通过设置嵌入策略, 使得在消息嵌入后, 嵌入区域对应的纹理复杂度不降低, 而非嵌入区域纹理保持不变. 该方法主要是通过牺牲载体的嵌入容量换取嵌入区域的同步.

2) 把 R 点视为 F 点, 所有点的嵌入策略都按 F 点的嵌入策略执行, 在嵌入完成后再通过某种修正方式进行调整. 例如: 文献 [9] 根据相邻像素的差值确定分块的 3 个纹理层次, 差值分别为 [0, 15], [16, 31], [32, 255]. 为避免消息嵌入后嵌入区域的纹理层次发生改变, 文献 [9] 在嵌入完成后对嵌入前后纹理复杂度不在同一层次的像素对进行了调整. 其通过调整嵌入位平面的更高位来保证不改变嵌入消息的前提下, 调整像素对的差值. 类似的, 文献 [10] 也采取了先嵌入再调整的策略来解决边信息的同步

问题. 该方法会导致部分像素点的隐写失真过大, 故不适用于限失真隐写信道.

本文在限失真隐写信道下对文献 [6] 等的单调纹理函数进行推广, 构造了自由度函数, 并根据像素自由度设定修改状态数. 因此, 本文方法可在保持纹理复杂度单调升的条件下, 充分发挥每个像素负载消息的能力.

3　实验结果

本节从抗检测性能和计算效率两个方面出发, 考察本文提出的自适应隐写算法的性能.

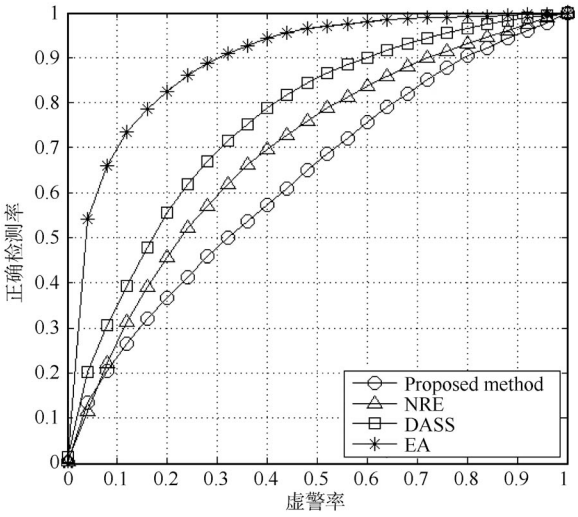
3.1　抗检测性能

Fridrich 等[13] 指出: 在所有多元编码中, 三元隐写编码 (即 ± 1 嵌入) 可以达到最小嵌入失真. 下面采用本文提出的 ± 1 算法 ($\delta = 1, t = 2$ 时) 和近年来三个典型的 ± 1 算法进行性能比较. 三个对比算法分别为: 2009 年 Li 等提出的 NRE 算法[6]、2010 年 Huang 等提出的 EA 算法[10] 和 2011 年 Filler 等提出的 DASS 算法[14]. 实验用图采用 UCID (Uncompressed color image database) 图像库[15] 中的 1 338 幅 384×512 规格图像转换得到的灰度图像. 为考察上述四种算法在不同嵌入率条件下的抗检测性能, 采用 2010 年 Cai 等提出的 RHF 检测特征[16] 和 2010 年 Pevny 等提出的 SPAM 检测特征[17], 并根据文献 [16−17] 中实验部分的描述, 在 RHF 和 SPAM 检测算法中分别使用 Fisher 线性分类器和 SVM (Support vector machine) 分类器进行分类.

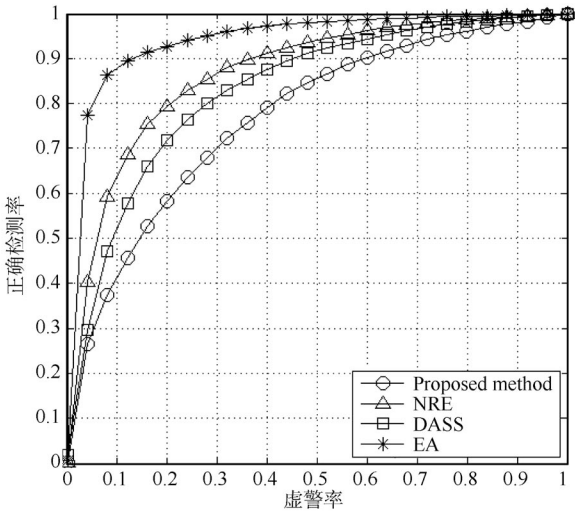
下面分别使用 ROC 曲线和最小平均分类误差 (P_E) 来展示检测结果, 其中 $P_E = \min_{P_{FA}}[(P_{FA} + P_{MD}(P_{FA}))/2]$, P_{FA} 和 P_{MD} 分别是虚警率和漏检率, (P_E) 值越小说明隐写分析算法检测效果越好, 即隐写算法安全性越差.

在两种检测算法中均采用 50 % 的载体图像及其对应的载密图像进行训练, 用剩余的 50 % 的图像进行测试. 消息嵌入率取 a =0.5, 0.6, 0.7, 0.8, 0.9, 1.0. 限于篇幅, RHF 算法的检测结果只列出了 a =0.5 和 0.7 的 ROC 曲线, 如图 4 所示. 其余嵌入率情况下, 本文算法与其他算法的对比效果与 a = 0.5 和 0.7 的情况一致. 类似的, SPAM 算法的检测结果如图 5 所示.

由实验结果可知, 对 RHF 检测算法而言, 本文算法在不同嵌入率下的性能均优于其他算法; 对 SPAM 检测算法而言, 本文算法在不同嵌入率下也明显优于其他算法. 从综合抗检测性能角度衡量, 本文提出的算法在上述四个算法中是性能最优的.



(a) $a = 0.5$



(b) $a = 0.7$

图 4　RHF 算法的检测结果

Fig. 4　The detection results of RHF algorithm

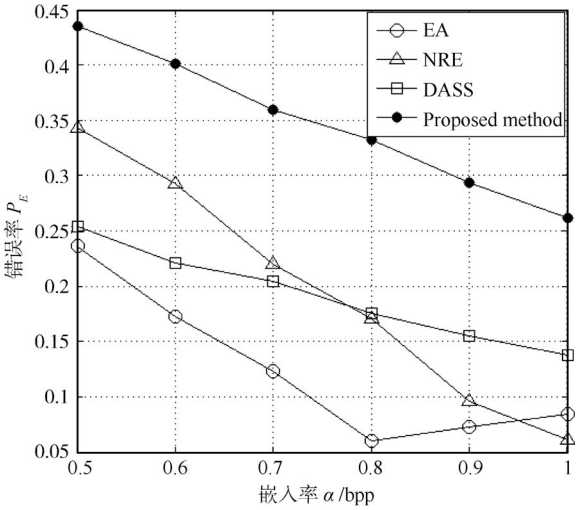


图 5　SPAM 算法的检测结果

Fig. 5　The detection results of SPAM algorithm

3.2 计算效率

采用文献 [18] 提出的双层嵌入思想, 在 LSB 层和次 LSB 分别采用双层湿纸编码也可实现自适应隐写. 类似地, 采用文献 [19] 提出的三层嵌入思想, 在载体的低三层中分别采用湿纸编码, 也可实现自适应隐写.

相较于 "多层湿纸编码", 本文提出的隐写具有如下优点:

1) LSB 层避免了使用湿纸码的带来的复杂度. LSB 层不需要湿纸码, 只用最平凡的嵌入即可;

2) 次 LSB 层降低了湿纸码的计算复杂度. 次 LSB 层只在嵌入区域使用湿纸码, 由于载体长度变短, 湿率变低, 故可有效降低湿纸码的计算复杂度. 事实上, 无论是用 LT 算法还是 STC 算法, 载体长度都是影响湿纸码计算复杂度的最主要因素; 对于 LT 算法, 湿率高, 找到解的成功概率低, 需要多次执行; 对于 STC 算法用于湿纸, 湿率高, 湿像素被修改的比率高.

分别采用本文自适应 ± 1 隐写算法和 "双层湿纸编码" 在 UCID 图像库的 1 338 幅图像中以嵌入率嵌入信息. 其中, 湿纸编码采用 LT 算法实现, 消息嵌入率分别取 $a = 0.5, 0.7$. 当 $a = 0.5$ 时, 本文算法完成一次消息嵌入平均耗时比双层嵌入方法减少 65.24 %; 当 $a = 0.7$ 时, 前者完成一次消息嵌入平均耗时比后者减少 71.63 %.

综上, 与三种不引入湿纸编码的自适应隐写术相比, 本文算法具有更好抗检测性能; 与 "多层湿纸编码" 相比, 本文算法在抗检测性能相当的前提下, 具有更高的计算效率.

4 总结

边信息同步是自适应隐写术的核心问题. 本文在分析现有自适应隐写算法本质的基础上, 构造了一种新的纹理保序函数, 提出 ± 1 隐写和 ± 2 隐写算法. 新算法既保证了通信双方可实现边信息的快速同步, 又通过引入湿纸编码和多层嵌入方法提高嵌入效率, 增强其抗隐写检测的性能. 如何降低新算法的计算复杂度是下一步研究的问题.

References

1 Cheng Wei-Dong, Huang Ji-Wu, Liu Hong-Mei. A 3D-DCT-Based information hiding algorithm for color images. *Acta Automatica Sinica*, 2003, **29** (2): 258−266
(程卫东, 黄继武, 刘红梅. 一种基于 3D-DCT 的彩色图像信息隐藏算法. 自动化学报, 2003, **29**(2): 258−266)

2 Filler T, Fridrich J. Design of adaptive steganographic schemes for digital images. In: Proceedings of the 8th Electronic Imaging, Media Watermarking, Security, and Forensics. San Francisco, CA: SPIE, 2011. 1−14

3 Fridrich J, Kodovský J, Goljan M, Holub V. Steganalysis of content-adaptive steganography in spatial domain. In: Proceedings of the 13th International Conference on Information Hiding. Prague, Czech Republic: LNCS, 2011. 102−117

4 Fridrich J, Goljan M, Lisonek P, Soukal D. Writing on wet paper. *IEEE Transactions on Signal Processing*, 2005, **53**(10): 3923−3935

5 Filler T, Judas J, Fridrich J. Minimizing embedding impact in steganography using trellis-coded quantization. In: Proceedings of the 7th Electronic Imaging, Media Forensics and Security. San Jose, USA: SPIE, 2009. 1−14

6 Lu Y F, Li X L, Yang B. A secure steganography: noisy region embedding. In: Proceedings of 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing. Kyoto, Japan: IEEE, 2009. 1046−1051

7 Zhang X P, Wang S Z. Steganography using multiple-base notational system and human vision sensitivity. *IEEE Signal Processing Letters*, 2005, **12(1)**: 67−70

8 Singh Kh M, Singh L S, Singh A B, Devi Kh S. Hiding secret message in edges of the image. In: Proceedings of the 2007 International Conference on Information and Communication Technology. Dhaka, Bangladesh: IEEE, 2007. 238−241

9 Yang C H, Weng C Y, Wang S J, Sun H M. Adaptive data hiding in edge areas of images with spatial LSB domain systems. *IEEE Transactions on Information Forensics and Security*, 2008, **3**(3): 488−497

10 Luo W Q, Huang F J, Huang J W. Edge adaptive image steganography based on LSB matching revisited. *IEEE Transactions on Information Forensics and Security*, 2010, **5**(2): 201−214

11 Liu G J, Liu W W, Dai Y W, Lian S G. An adaptive matrix embedding for image steganography. In: Proceedings of the 3rd International Conference on Multimedia Information Networking and Security. Shanghai, China: IEEE, 2011. 642−646

12 Fridrich J, Goljan M, Soukal D. Efficient wet paper codes. In: Proceedings of the 7th International Workshop on Information Hiding. Barcelona, Spain: LNCS, 2005. 204−218

13 Fridrich J. Minimizing the embedding impact in steganography. In: Proceedings of the 8th Workshop on ACM Multimedia and Security. Geneva, Switzerland: ACM, 2006. 2−10

14 Filler T, Fridrich J. Design of adaptive steganographic schemes for digital images. In: Proceedings of the 8th Electronic Imaging, Media Watermarking, Security, and Forensics. San Francisco, CA: SPIE, 2011. 24−26

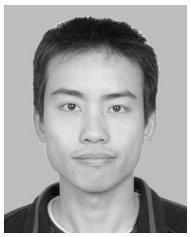
15 Schaefer G, Stich M. UCID: an uncompressed color image database. In: Proceedings of the 2nd Storage and Retrieval Methods and Applications for Multimedia. San Jose, USA: SPIE, 2003. 472−480

16 Cai K W, Li X L, Zeng T Y, Yang B, Lu X Q. Reliable histogram features for detecting LSB matching. In: Proceedings of the 17th International Conference on Image Processing. Hong Kong, China: IEEE, 2010. 1761−1764

17 Pevny T, Bas P, Fridrich J. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, 2010, **5**(2): 215−224

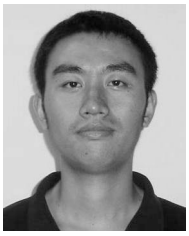
18 Zhang X, Zhang W, Wang S. Efficient double-layered steganographic embedding. *IET Electronics Letters*, 2007, **43**(8): 482−483

19 Zhang W M, Zhang X P, Wang S Z. Near-optimal codes for information embedding in gray-scale signals. *IEEE Transactions on Information Theory*, 2010, **55**(3): 1262−1270



陈嘉勇 厦门大学博士后. 2005 年获解放军信息工程大学信息研究系学士学位, 2008 年获解放军信息工程大学应用数学专业硕士学位, 2012 年获解放军信息工程大学信息工程学院工学博士学位. 主要研究方向为信息隐藏和网络安全. 本文通信作者. E-mail: cjy1003@sina.com
(**CHEN Jia-Yong** Postdoctor at Xiamen University. He received his bachelor degree in engineering in 2005, master degree in science in 2008, and Ph. D. degree in engineering in 2012 from PLA Information Engineering University. His research interest covers information hiding and network security. Corresponding author of this paper.)

张卫明 解放军信息工程大学信息工程学院副教授. 主要研究方向为信息隐藏和密码学.
E-mail: weimingzhang@shu.edu.cn
(**ZHANG Wei-Ming** Associate professor at the School of Information Engineering, PLA Information Engineering University. His research interest covers information hiding and cryptography.)



韩 涛 解放军信息工程大学信息工程学院博士研究生. 主要研究方向为信息隐藏和网络安全.
E-mail: lhstslhsts@163.com
(**HAN Tao** Ph. D. candidate at the School of Information Engineering, PLA Information Engineering University. His research interest covers information hiding and network security.)

祝跃飞 解放军信息工程大学信息工程学院教授. 主要研究方向为密码学和网络安全. E-mail: zhu_yuefei@163.com
(**ZHU Yue-Fei** Professor at the School of Information Engineering, PLA Information Engineering University. His research interest covers information security and cryptography.)

郭东辉 厦门大学教授. 主要研究方向为人工智能, 网络通讯和集成电路设计.
E-mail: zhu_yuefei@163.com
(**GUO Dong-Hui** Professor at the School of Xiamen University. His research interest covers artificial intelligence, network communication, and integrated circuit design.)