

中图分类号: TP391 TN919.8 文献标志码: A 文章编号: 1006-8961(2011)09-1583-09

论文索引信息: 邓晓曼, 潘志斌, 乔瑞萍, 高风娟. 面向索引图的大容量信息隐藏方法[J]. 中国图象图形学报, 2011, 16(9): 1583-1591

面向索引图的大容量信息隐藏方法

邓晓曼, 潘志斌, 乔瑞萍, 高风娟

(西安交通大学电子与信息工程学院, 西安 710049)

摘要: 嵌入容量是衡量信息隐藏方法的重要指标之一。为了提高嵌入容量, 提出一种面向矢量量化索引图邻块编码的信息隐藏改进算法, 利用当前块的四邻块的空间相对位置关系和邻块与当前块索引差值的数值范围进行信息隐藏。实验结果表明, 改进后的算法能够显著提高嵌入容量, 在比特率仅增加 5.13% ~ 13.10% 的情况下可将嵌入容量提高到 Chang 方法容量的 2.11 ~ 3.16 倍, 嵌入效率由原来平均 0.10 提高到了 0.26。

关键字: 信息隐藏; 矢量量化; 联合邻域编码; 嵌入容量; 比特率; 嵌入效率

High capacity information embedding scheme in VQ-compressed image index map

Deng Xiaoman, Pan Zhibin, Qiao Ruiping, Gao Fengjuan

(School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049 China)

Abstract: Embedding capacity is one of the important factors that affect the information hiding performance. An improved information hiding scheme for VQ-compressed index map images based-on joint neighboring coding (JNC) technique is proposed to increase the embedding capacity. The proposed method embeds secret data by using the difference values ranging between the current VQ-compressed index and its four neighboring indices. Experimental results show that the proposed method increases the embedding capacity significantly compared with Chang's and Yang's method. The proposed method obtains 2.11 to 3.16 times embedding capacity compared to Chang's method with only a 5.13% to 13.10% increase of the bit rate. As a result, the value of embedding-efficiency is increased on average from 0.10 in Chang's method to 0.26 in the proposed method.

Keywords: information hiding; vector quantization; joint neighboring coding; embedding capacity; bit rate; embedding-efficiency

0 引言

由于互联网信息交换和共享的特性使得传输的信息容易受到攻击和篡改, 因此秘密通信应运

而生。传统上, 秘密信息可以由加密的方法来保护。在加密方法中, 秘密信息传输之前先通过加密技术给信息加密, 将原文通过某种数学映射变为密文。然而加密技术形成的密文与普通信息的明显差异预示了重要信息的存在, 容易引起关注,

收稿日期: 2010-07-20; 修回日期: 2010-11-30

基金项目: 国家自然科学基金项目(60672054); 陕西省科学技术攻关项目(2008K04-01); 德州仪器创新基金(2009W1201); 高校博士点专项基金项目(20100201110030)。

第一作者简介: 邓晓曼(1986—), 女。西安交通大学信息与通信工程专业在读硕士研究生, 主要研究方向为图像处理、信息隐藏和数字水印。E-mail: deng.1986@stu.xjtu.edu.cn。

进而被破解。为了解决这个问题,产生了一种信息隐藏的方法。信息隐藏向信息的载体(即目标)中嵌入信息,使人无察觉的加入秘密比特。信息隐藏通常分为两类,不可逆的(有损)和可逆的(无损)。在不可逆的信息隐藏方法中,只有秘密比特被恢复,而信息载体不被恢复;而在可逆的方法中,二者皆可以被恢复出来。

信息隐藏中有几个重要的衡量指标:嵌入容量、比特率、视觉效果和算法复杂度。对于信息隐藏,嵌入容量是最重要的,大量的嵌入秘密比特,实现秘密通信,是信息隐藏的目的。同时嵌入容量的增加,会带来比特率的增加,而比特率往往受到实际系统的限制,因此,降低比特率是信息隐藏能得到实际应用的重要条件。秘密信息的嵌入,有可能带来图像视觉效果的失真,而失真越大,越容易引起攻击和篡改,安全性越低,因此要尽量减小失真。信息隐藏包括信息的嵌入过程和提取过程。嵌入和提取算法越简单,消耗的时间越短,时间效率越高。

信息隐藏可以在空域、变换域(如 DCT, DFT, DWT)和压缩域实现。空域是直接改变载体数据的值,如 LSB 算法^[1],它利用数据信息的最低有效位(least significant bit)的改变不会明显引起人眼视觉或人耳听觉上的差异这一特点,将秘密信息转化为二值流文件序列依次替换到载体数据的最后一位,以完成秘密信息的隐藏过程。变换域中通过修改变换系数^[2],实现嵌入。但是考虑到减小失真,容量往往受到限制。在压缩域中可将矢量量化^[3](VQ)压缩后的图像作为载体来隐藏信息,它的高压缩率使得传输的数据显著减少,使得其适宜在实际中使用。矢量量化应用在信息隐藏中的一类方法是对码书中码字的排列作处理^[4-6],还有一类是改变索引图中的索引^[7-8]。1999年, Lin 等人^[4]提出将码书分成两个子码书分别嵌入秘密比特 0 和 1。为减小视觉误差 Jo 等人^[5]改进了这一算法,将码书分为 3 个子码书,增加的一个子码书包括了不能被替换的码字。2003年 Du 等人^[6]用聚类方法代替分组来产生子码书进而提高了嵌入容量。2005年 Yang 等人提出 MFCVQ(modified fast correlation VQ)的方法,在 VQ 压缩域插入秘密比特^[7],但是嵌入容量很小;2006年 Lin 等人在 VQ 压缩域用码字配对的方法^[8]改进了这一缺点;2007年 Chang 等人提出 SMVQ(side match VQ)的方法^[9],但是这类算法失真较大。2009年 Chang 等人提出基于索引图建立动态

表的方法,利用索引出现频率进行隐藏^[10],但是计算动态表耗时较多且嵌入容量有限。2009年 Chang 等人提出 JNCVQ(joint neighboring coding VQ)^[11]的方法,利用索引表中当前块的左邻块和上邻块进行一比特的隐藏,不额外引入图像失真,并且与前面的方法^[7,9-10]相比提高了容量。本文提出对 JNCVQ 隐藏方法进一步的改进,增加了使用的邻块数目,使用 4 邻块与当前块的相对位置关系,每个当前块可以嵌入 2bit,同时改进了情况 3 的编码方法,从两个方面增加了嵌入的容量,并在增加容量的基础上省略了情况 4 中 m 个 0 的发送,抑制了比特率的明显增加。

1 已有的工作

1.1 矢量量化

矢量量化^[3]作为一种高效的有损压缩技术,已经广泛应用于图像信号处理和语音信号处理等领域中。矢量量化的编码过程可以描述为在给定的码书

$$C = \{y_0, y_1, \dots, y_{N-1} | y_i \in \mathbf{R}^k, i = 0, 1, \dots, N-1\}$$

中搜索与输入矢量 $\mathbf{x} = (x_0, x_1, \dots, x_{k-1})^T$ 最匹配的码字 $\mathbf{y}_w = (y_{w0}, y_{w1}, \dots, y_{w(k-1)})^T$,从而使 \mathbf{x} 和 \mathbf{y}_w 之间的失真最小。通常失真测度采用欧氏距离,即

$$d(\mathbf{x}, \mathbf{y}_i) = \sum_{l=0}^{k-1} (x_l - y_{il})^2 \quad (1)$$

那么最佳匹配码字为

$$d(\mathbf{x}, \mathbf{y}_w) = \min_{y_i \in C} [d(\mathbf{x}, \mathbf{y}_i)] \quad (2)$$

式中, \mathbf{y}_w 是与输入矢量 \mathbf{x} 最匹配的码字, w 即要传输的索引。

对于图像的矢量量化编码,先把整幅图像按照 $h \times w$ 进行分块,每一小块即构成一个输入矢量,然后对每个输入进行矢量量化,得到整幅图像的索引图,即完成编码。

1.2 基于 JNCVQ 的信息隐藏方法^[11]

1.2.1 生成索引图

索引图生成过程如图 1 所示。信息隐藏是向载体中嵌入秘密信息的过程,在这里载体的形式是灰度图像,即载体图像,记为 \mathbf{I} 。对于大小为 $H \times W$ 的图像 \mathbf{I} ,采用大小为 N 的 k 维码书 \mathbf{y}_i ,计算出每个码字的均值为

$$M_i = \frac{1}{k} \sum_{j=1}^k y_{ij} \quad (3)$$

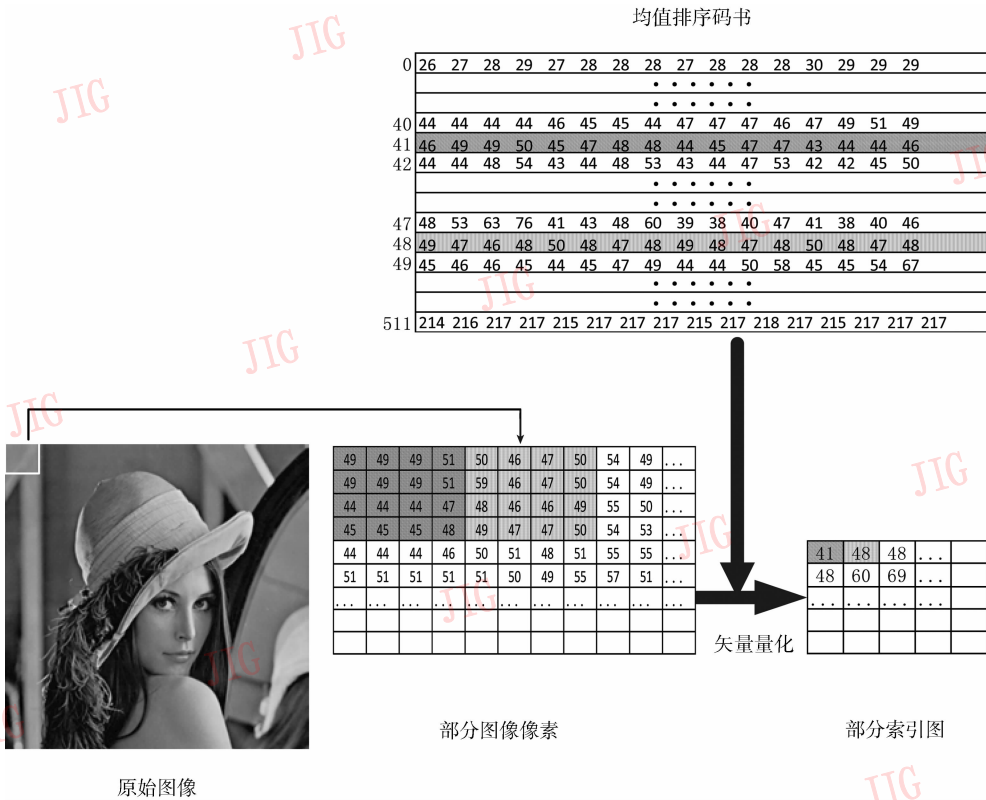


图1 索引图生成过程示意图
Fig. 1 Block diagram of index table generation

将码书按照均值大小进行升序排列 (即 $M_0 < M_1 < \dots < M_{N-1}$)。再用排序后的码书对图像 I 进行矢量量化编码可得到相应的索引图 P , 大小为 $(H/h) \times (W/w)$, 当前索引值为 $P_{r,c}$, 其中 $r=0, 1, \dots, (H/h) - 1, c=0, 1, \dots, (W/w) - 1$ 。

1.2.2 嵌入过程

嵌入过程把秘密比特加到图像的索引图之中, 输出为二进制的码流。秘密信息的嵌入利用不同的相邻块与当前块的索引之差 d 。索引图的第1行第1列不变, 通常第1行和第1列称为种子区域 (seed region), 直接发送, 实际的嵌入从第2行第2列开始。 d 的计算方法随着秘密比特的改变而改变: 当嵌入比特为1时, $d = P_{r,c-1} - P_{r,c}$; 当嵌入比特为0时, $d = P_{r-1,c} - P_{r,c}$, 如图2所示。

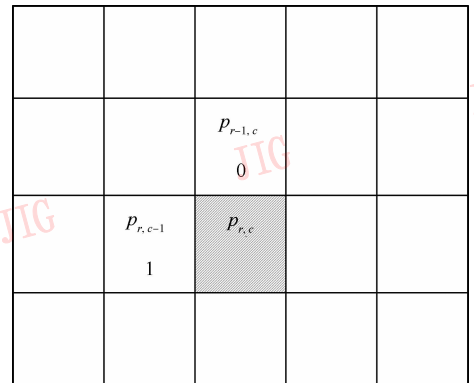


图2 当前块与邻块的位置关系
Fig. 2 Layout of left and upper neighboring blocks

根据 d 的数值范围对当前索引进行编码, 可分为4种情况。当 $0 < d \leq 2^m - 1$ 时为情况1, 当前索引编码为 d 的 m 位二进制表示, $1 \leq m \leq \lceil \log_2 N \rceil$, $\lceil x \rceil$ 表示上取整。当 $-(2^m - 1) \leq d < 0$ 时为情况2, 当前索引编码为 d 的相反数的 m 位二进制表示。当 $d = 0$ 时为情况3, 当前索引编码为 m 个0。当

$|d| > 2^m - 1$ 时为情况4, 当前索引编码为 m 个0和 $P_{r,c}$ 的 $\lceil \log_2 N \rceil$ 位的二进制表示。4种情况和一位秘密比特的不同情况的前缀在表1中给出, 其中 d_2 为对 d 的4种情况的二进制编码结果, “||”表示级联。

重复以上过程直至完成对所有索引的嵌入, 将码流发送给接收方。

表 1 4 种情况和秘密比特的编码方式

Tab. 1 Coding rule for four cases based on 1-bit secret data

秘密比特	情况 1	情况 2	情况 3	情况 4
1	11 d_2	10 d_2	11 $\frac{00\dots0}{m\text{位}}$	10 $\frac{00\dots0}{m\text{位}}$ $(P_{r,c})_2$
0	01 d_2	00 d_2	01 $\frac{00\dots0}{m\text{位}}$	00 $\frac{00\dots0}{m\text{位}}$ $(P_{r,c})_2$

1. 2. 3 提取过程

提取过程可以看做是嵌入过程的逆过程,其结果是恢复出秘密比特和图像的索引图。种子区域不做处理,直接放入索引图中,然后提取从真正有秘密比特嵌入的部分开始。

- 1) 截断码流。从码流中读入 $(2 + m)$ 位记为 s ;
- 2) 提取秘密比特。 s 的第 1 位即秘密比特;
- 3) 提取当前索引 $P_{r,c}$ 。当 s 的末 m 位均为 0, 并且 s 的第 2 位为 1 时(情况 3), 如果秘密比特为 1, 则 $P_{r,c} = P_{r,c-1}$; 如果秘密比特为 0, 则 $P_{r,c} = P_{r-1,c}$ 。

当 s 的末 m 位均为 0, 并且 s 的第 2 位为 0 时, (情况 4), 再从码流中读入 $\lceil \log_2 N \rceil$ 位, 转化为十进制, 记为 q , 则 $P_{r,c} = q$ 。当不符合上述情况的时(情况 1 或 2), 将 s 的末 m 位转化为十进制, 记为 d 。当 s 的第 2 位为 0 时(情况 2), $d = -d$, 当 s 的第 2 位为 1 时(情况 1), d 不变。如果秘密比特为 1, 则 $P_{r,c} = P_{r,c-1} - d$; 如果秘密比特为 0, 则 $P_{r,c} = P_{r-1,c} - d$ 。

重复 1)–3) 直至解码全部码流。至此完成了提取过程, 可无失真的得到秘密信息和索引图。最后再由图像的索引图恢复出原始图像。

1. 3 JNCVQ 算法的局限性

首先, JNCVQ 算法中嵌入过程仅使用当前块的两个邻块嵌入 0 或 1, 除去种子区域的每个索引只能嵌入一比特, 嵌入容量是很有限的。因此可以考虑扩充为使用当前块的 4 个邻块, 使容量成倍提高。其次, JNCVQ 算法编码时情况 3 和情况 4 都先发送 m 个 0, 这 m 个 0 在解码时仅用于区别正常情况(包括情况 1 和情况 2)和其他情况(包括情况 3 和情况 4)。由于存在较大的冗余, 因此这 m 比特增加了比特率, 而且解码时要用 m 个 0 和 s 的第 2 位进行两次比较才能完全区分出 4 种情况, 相应增加了算法的复杂度。实际上, 对情况 3 和情况 4, 完全可以通过更合理的设计编码来避免这部分比特率增加, 并降低解码时的复杂性。

1. 4 基于 MFCVQ 的信息隐藏算法^[7]

Yang 等人提出一种基于改进的快速相关矢量量化(MFCVQ)的信息隐藏算法^[7]。把载体图像按照 4×4 分块, 用码书里的码字给第 1 行和第 1 列的图像块编码, 得到部分索引图, 称为种子区域(seed region), 从第 2 行第 2 列开始, 当前图像块记为 x , 它的 4 个邻块分别记为 CW_e, CW_f, CW_u 和 CW_v , 位置如图 3 所示, 当图像块位于最右一列的时候, 则退化 3 个邻块, 即 CW_e, CW_f 和 CW_u 。

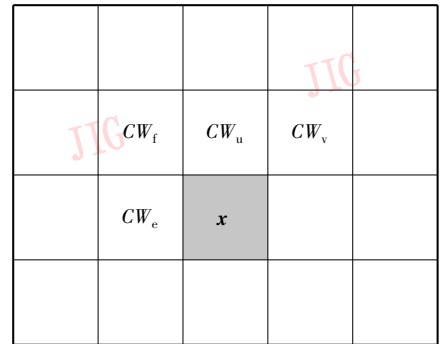


图 3 x 的 4 邻块

Fig. 3 Layout of four neighboring blocks of x

按式(1)计算当前块与 4 邻块的码字距离, 记为

$$d_i = d(x, CW_i) \quad i = e, f, u, v$$

$$d_{\min} = \min\{d_e, d_f, d_u, d_v\}$$

给定一个门限 T , 如果 $d_{\min} < T$, 则当前块可以做信息嵌入; 否则当前块不能做信息嵌入。对于可以进行信息嵌入的块, 计算出

$$CW_{\text{mean}} = \frac{1}{4}(CW_e + CW_f + CW_u + CW_v)$$

再从 $\{CW_e, CW_f, CW_u, CW_v\}$ 的集合中找到与 CW_{mean} 最匹配的码字, 记为 CW_{bm} 。如果当前要嵌入的秘密比特是 0, 则发送标志位“0”和 2bit CW_{bm} 的索引(这里 $\{CW_e, CW_f, CW_u, CW_v\}$ 的索引分别用 $\{00, 01, 10, 11\}$ 表示)。如果当前要嵌入的秘密比特是 1, 则从 $\{CW_e, CW_f, CW_u, CW_v\}$ 集合中找到与当前块 x 最匹配的码字, 记为 CW_x ; 如果 $CW_x = CW_{\text{bm}}$, 则找到与当前块 x 次匹配的码字, 记为 CW'_x , 发送标志位“0”和 2 bit CW_x (或 CW'_x) 的索引。对于不能信息嵌入的图像块, 从码书中找到与当前块 x 最匹配的码字, 记为 CW'_{bm} , 发送标志位“1”和 $\lceil \log_2 N \rceil$ 位 CW'_{bm} 的索引。重复以上过程直至完成所有图像块的编码。

2 本文算法

2.1 生成索引图

生成索引图的过程与1.2.1节中的方法相同,即用均值排序的码书对图像 I 进行矢量量化,得到其索引图 P 。

2.2 嵌入过程

把秘密比特加入索引图中,得到二进制码流。

1) 第1行第1列及最右一列为种子区域,不嵌入,直接传输。

2) 根据两位的秘密比特来确定差值 d 的计算方法,两位的秘密比特即代表了邻块相对于当前块的位置关系,即相对地址指针。邻块的索引用 P_{adj} 表示, $d = P_{adj} - P_{r,c}, P_{adj} \in \{P_{r,c-1}, P_{r-1,c-1}, P_{r-1,c}, P_{r-1,c+1}\}$, 图4为其示意图。

3) 将 d 的分布分为4种情况,用两位的标志位加以区别,根据 d 所在的范围来确定 d 的编码方式。 $0 < d \leq 2^m - 1$ 为情况1,标志位为00, d 编码为 m 位的 d 的二进制表示, $1 \leq m \leq \lceil \log_2 N \rceil$; $-(2^m - 1) \leq d < 0$ 为情况2,标志位为01, d 编

码为 m 位的 d 的相反数的二进制表示; $d = 0$ 为情况3,标志位为10, d 不需要传输, m 位用来直接发送秘密比特; $2^m - 1 < |d|$ 为情况4,标志位为11,无须传输 m 个0, d 直接编码为 $\lceil \log_2 N \rceil$ 位的 $P_{r,c}$ 的二进制表示。

4) 把两位的秘密比特、两位的标志位和 d 的编码打包发送,表2给出其编码规则,其中 d_2 和 $(P_{r,c})_2$ 为步骤3)中对 d 的4种情况的二进制编码结果。

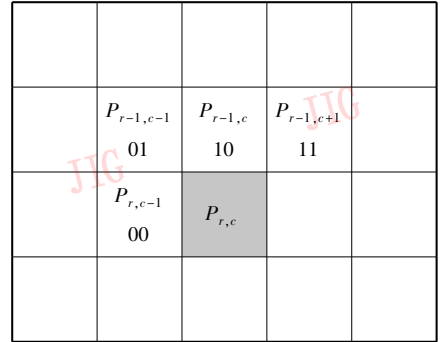


图4 当前块与4邻块的位置关系和相对地址指针

Fig. 4 Layout of four neighboring blocks and relative address pointer

表2 本文的索引编码方式

Tab. 2 Coding rule for four cases based on 2 bit secret data

秘密比特	情况1	情况2	情况3	情况4
00	00 00 d_2	00 01 d_2	00 10 $\underbrace{\times \times \dots \times}_m$ 秘密信息	00 11 $(P_{r,c})_2$
01	01 00 d_2	01 01 d_2	01 10 $\underbrace{\times \times \dots \times}_m$ 秘密信息	01 11 $(P_{r,c})_2$
10	10 00 d_2	10 01 d_2	10 10 $\underbrace{\times \times \dots \times}_m$ 秘密信息	10 11 $(P_{r,c})_2$
11	11 00 d_2	11 01 d_2	11 10 $\underbrace{\times \times \dots \times}_m$ 秘密信息	11 11 $(P_{r,c})_2$

然后返回到步骤2)直至对所有索引的嵌入完成。嵌入过程结束后,将码流发送给接收方。

2.3 提取过程

种子区域的索引直接放入索引表中,没有秘密比特。从嵌入的部分开始,按照以下步骤进行提取:

1) 截断码流。从码流中读入 $(4 + m)$ 位。

2) 提取秘密比特。 $(4 + m)$ 中的前两位为秘密比特,可以直接进行提取。接下来两位为标志位,指示了 d 取值的4种情况;末 m 位记为 s 。

3) 提取当前索引 $P_{r,c}$ 。首先恢复出相邻块之间的差值 d 。根据标志位的指示来处理 s 。标志位为00,

s 转化为十进制记为 d 。标志位为01, s 转化为十进制并取相反数记为 d 。标志位为10, $d = 0$, 则 s 为 m 位的秘密比特,可以直接提取。标志位为11,从码流中再提取 $(\lceil \log_2 N \rceil - m)$ 位与 s 共同构成 $\lceil \log_2 N \rceil$ 位,转化为十进制记为 d 。然后就可以恢复出当前索引 $P_{r,c}$ 。标志位如果是11,则 $P_{r,c} = d$,得到了当前索引;如果不是,按照秘密比特找到相邻块的位置,再与 d 一起计算当前索引,秘密比特的值指示了相邻块相对于当前块的偏移量。如秘密比特为00, $P_{r,c} = P_{r,c-1} - d$; 秘密比特为01, $P_{r,c} = P_{r-1,c-1} - d$; 秘密比特为10, $P_{r,c} = P_{r-1,c} - d$; 秘密比特为11, $P_{r,c} = P_{r-1,c+1} - d$ 。

返回到步骤 1) 直至全部码流处理完。至此完成了提取过程,即可无失真的恢复出秘密比特和索引图。

改进后的算法增加了两个邻块的使用,用 4 邻块的相对位置关系使得每个块可以嵌入 2 bit;同时,情况 3 的时候每块可以省略对 d 编码的部分而直接嵌入 m 位的秘密比特,从两个方面提高了嵌入容量。由于编码方式的改变,用两比特来区分 d 值分布的 4 种情况,简单易行,避免了情况 3 和情况 4 下 m 比特的使用,从而降低了比特率的增加程度和译码复杂度。

3 实验结果

实验在 Intel core2 主频 2.93 GHz、内存 2 GB 的个人计算机上进行;秘密比特采用 rand 函数产生;码书大小为 512,图像采用 4×4 分块;共采用 Lena、Baboon 等 6 幅 512×512 测试图像,如图 5 所示。表 3 中给出了 m 取值不同时(情况 1 和情况 2 合称为正常情况,情况 3 为零情况,情况 4 为特殊情况)3 幅图像 Lena、Baboon 和 Goldhill 的嵌入容量和比特率。其中 m 取值 2~8。比特率 b 依据下式计算

$$b = \frac{c}{H \times W} \quad (4)$$

单位为 bit/pixel,其中 c 为编码后的比特数。

C_p 表示嵌入容量,依据下式计算

$$C_p = 2(c_{n_1} + c_{n_2} + c_{n_3} + c_{n_4}) + m \times c_{n_3} \quad (5)$$

式中, c_{n_1} , c_{n_2} , c_{n_3} 和 c_{n_4} 分别表示属于情况 1,情况 2,情况 3 和情况 4 的图像块数。

表 3 给出了 3 幅图像的嵌入容量、比特率和 4 种情况数目随 m 的变化情况。在本文的方法中,产生的秘密比特的嵌入源自两部分:一部分是除去种子区域的每个索引可以嵌入 2 bit,这部分是固定的;另一部分是零情况的时候每个索引可以嵌入 m 比特,这部分是随图像本身的内容和 m 的取值而变化的。图像本身的平滑程度,以及 m 取值不同时造成零情况的数目不同。然而在实验中发现,零情况也是基本不变的。因此,嵌入的秘密比特数量有随着 m 增长的趋势。 m 取某一个值时,比特率可以达到最小。考虑到不同的图像平滑的程度不同,比特率达到最小时的 m 也是不同的。综合各种情况,本文提出的改进后的算法采用 $m = 5$ 。

表 4 给出了 $m = 5$ 时 6 幅测试图像中各种情况所占的百分比。 $m = 5$ 时正常情况和零情况的总和占大多数,即 d 在正常情况和零情况时编码为 5 位,起到了降低比特率的作用。

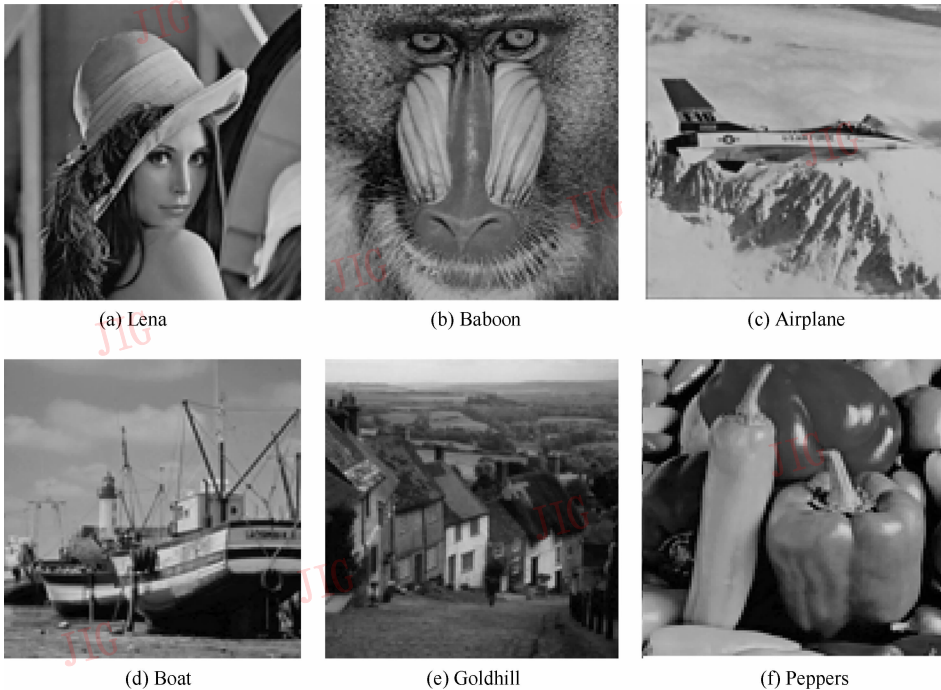


图 5 6 幅载体图像
Fig. 5 Six test images

表3 m 不同取值下的嵌入容量和比特率
Tab.3 Effect of m 's value on the embedding capacity and bit rate

图像		m						
		2	3	4	5	6	7	8
Lena	C_p	35 828	37 596	39 816	41 364	43 584	45 472	48 036
	b	0.723	0.689	0.652	0.642	0.657	0.695	0.747
	正常情况	1 220	3 267	6 168	8 903	11 119	12 771	13 689
	零情况	1 912	1 864	1 953	1 872	1 930	1 924	2 004
	特殊情况	12 870	10 871	7 881	5 227	2 953	1 307	309
	C_p	32 918	33 273	33 728	34 099	34 686	35 091	35 540
Baboon	b	0.775	0.754	0.724	0.697	0.682	0.698	0.746
	正常情况	713	1 868	3 911	6 786	10 464	13 769	15 437
	零情况	457	423	431	419	447	441	442
	特殊情况	14 832	13 711	11 660	8 797	5 091	1 792	123
	C_p	36 126	38 310	40 424	42 674	44 316	46 445	48 428
	b	0.726	0.707	0.680	0.656	0.658	0.692	0.746
Goldhill	正常情况	974	2 262	4 539	7 729	10 953	12 914	13 821
	零情况	2 061	2 102	2 105	2 134	2 052	2 063	2 053
	特殊情况	12 967	11 638	9 358	6 139	2 997	1 025	128

表4 $m=5$ 时6幅测试图像各种情况所占百分比
Tab.4 Percentage of four cases in six test images ($m=5$)

	Lena	Baboon	Airplane	Boat	Goldhill	Peppers	平均
正常情况	55.64	42.41	51.65	51.70	64.42	62.10	54.65
零情况	11.70	2.62	23.75	22.36	13.34	16.44	15.03
特殊情况	32.67	54.97	24.60	29.77	38.36	29.72	35.02

表5 采用了 Lena、Baboon 等6幅图进行了已有算法和本文改进算法的对比实验。表中给出了各种方法的 PSNR,容量,比特率,和嵌入效率。

$$PSNR = 10 \lg \frac{255^2}{MSE} \text{ dB} \quad (6)$$

式中, $MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (I_{ij} - \bar{I}_{ij})^2$ 。这里 I_{ij}, \bar{I}_{ij} 分别表示原始图像和重建图像(i,j)处的像素值。

嵌入效率为

$$e = \frac{C_p}{b \times H \times W} \quad (7)$$

嵌入效率是图像每像素嵌入的秘密比特数和每像素所用的编码比特数的比值,该值表征了传输过程中每比特平均携带秘密信息的数量,数值上越大越好。值越大,表示嵌入方法越有效,即传输较少的比特就可以携带较多的秘密信息。

容量提高倍数为

$$\alpha_c = \frac{C_{p_1}}{C_{p_2}} \quad (8)$$

式中, C_{p_1} 表示改进后算法的嵌入容量, C_{p_2} 表示对比方法的嵌入容量。

比特率增加系数为

$$\alpha_b = \frac{c_1 - c_2}{c_2} \quad (9)$$

式中, c_1 表示改进后算法完成编码的比特数, c_2 表示对比方法完成编码的比特数。

嵌入效率增加系数为

$$\alpha_e = \frac{e_1 - e_2}{e_2} \quad (10)$$

式中, e_1 表示改进后算法的嵌入效率, e_2 表示对比方法的嵌入效率。

表 5 算法性能对比

Tab. 5 Comparison between the proposed method and other methods

方法	指标	Lena	Baboon	Airplane	Boat	Goldhill	Peppers	平均
VQ($b = 0.5625$)	PSNR/dB	33.619	24.251	29.944	29.332	30.293	30.580	29.670
本文($m = 5$)	PSNR/dB	33.619	24.251	29.944	29.332	30.293	30.580	29.670
	容量/bit	41364	34099	51004	49894	42674	45154	44032
	比特率	0.642	0.697	0.623	0.635	0.656	0.635	0.648
	嵌入效率	0.246	0.187	0.312	0.300	0.248	0.271	0.261
	PSNR/dB	33.619	24.251	29.944	29.332	30.293	30.580	29.670
Chang($m = 6$)	容量/bit	16129	16129	16129	16129	16129	16129	16129
	比特率	0.579	0.663	0.572	0.584	0.580	0.571	0.592
	嵌入效率	0.106	0.093	0.108	0.105	0.106	0.108	0.104
本文与 chang 方法比较	α_c	2.565	2.114	3.162	3.093	2.646	2.800	2.730
	$\alpha_b/\%$	10.88	5.13	8.92	8.73	13.10	11.21	9.66
	$\alpha_e/\%$	132.08	101.08	188.89	185.71	133.96	150.93	149.84
Yang($T = 18$)	PSNR/dB	31.285	24.231	28.609	28.611	29.713	29.716	28.694
	容量/bit	8093	990	8895	7746	2753	7186	5944
	比特率	0.408	0.598	0.387	0.417	0.551	0.432	0.465
	嵌入效率	0.076	0.006	0.088	0.071	0.019	0.063	0.054
	α_c	5.111	34.443	5.734	6.441	15.501	6.284	7.408
本文与 Yang 方法比较	$\alpha_b/\%$	57.39	16.63	61.19	52.21	19.16	46.96	39.26
	$\alpha_e/\%$	224.97	2868.25	255.35	323.73	1198.43	327.44	384.06

从表 5 可以看出对于改进后的算法,不同特点的载体图像的嵌入效率是不同的,以 Baboon, Lena 和 Airplane 3 幅典型图像为例,比特率和嵌入容量决定了嵌入效率。对于比特率,正常情况和零情况是 $(4 + m)$ 位编码,特殊情况是 $(4 + \lceil \log_2 N \rceil)$ 位编码,故特殊情况数目的多少决定了比特率的大小。由于 Baboon 的特殊情况数目多, Lena 次之, Airplane 最少,比特率也由大到小依照这个顺序出现;但是由于取了合适的 m 值,尽量减少了不同图像特殊情况的数量差异,使得最终的比特率相差不多。嵌入容量包括每一块嵌入的 2 bit 和零情况额外嵌入的 m 比特,零情况多则嵌入容量大, Airplane 零情况最多,嵌入容量也最多, Lena 次之, Baboon 最少。在比特率相差不多的情况下,嵌入效率主要受嵌入容量的影响。

从表 5 可以看出,改进后的算法在容量上大大提高,这一方面来自于使用了当前块的 4 邻块,使容量较 Chang 的方法使用两个邻块提高一倍;另一方

面情况 3 的时候不必传输 d 的编码,而是直接传输 m 位的秘密比特,由于 $d = 0$ 出现的概率平均在 15% 左右,并且取 $m = 5$,这也显著提高了嵌入容量。同时改进后的算法在编码上用两位的标志位来区别 d 的数值范围的 4 种情况,避免了 Chang 的方法中情况 3 和情况 4 时 m 个 0 的使用,在容量大大提高的情况下比特率并没有提高很多。当 m 值选取合适时, Lena 图中比特率仅增加了 10.88%,容量则提高到原来的 2.565 倍,嵌入效率由原来的 0.106 提高到现在的 0.246;容量提高最小的是 Baboon 图,比特率增加了 5.13%,容量提高到原来的 2.114,嵌入效率由原来的 0.093 提高到现在的 0.187;而容量提高最大的是 Airplane 图,比特率增加了 8.92%,容量提高到原来的 3.162 倍,嵌入效率由原来的 0.108 提高到现在的 0.312。容量提高的程度明显大于比特率增加的程度,嵌入效率与 Chang 的方法^[11]相比平均由 0.10 提高到了 0.26,说明改进后的算法传输每比特能多携带 2 倍以上的秘密

信息。与 Yang 的方法相比,改进后的算法避免了图像的明显失真,而且嵌入容量和嵌入效率都有了很大的提高。

4 结 论

提出的改进算法是在基于 JNCVQ 隐藏方法的基础上,增加使用邻块的数目,即利用当前块的 4 个邻块的相对空间位置关系嵌入 2 bit,并改进编码方式,在情况 3 时省略 d 的编码部分而直接嵌入 m 位的秘密比特,使得嵌入容量得到两方面的增长。同时用两位标志位表示 d 的数值范围的 4 种情况,有效抑制了比特率的提高。从对比实验的结果可以看出,比特率仅增加了 5.13% ~ 13.10%,容量则达到了 Chang 方法的 2.11 ~ 3.16 倍,嵌入效率由原来平均的 0.10 提高到了 0.26。容量的提高远大于比特率的增加,从而使嵌入效率有了显著提高。

参考文献 (References)

[1] Chan C K, Cheng L M. Hiding data in images by simple LSB substitution[J]. Pattern Recognition, 2004, 37(3): 469-474.
[2] Lan T H, Tewfik A H. A novel high-capacity data-embedding system[J]. IEEE Transactions on Image Processing, 2006, 15(8): 2431-2440.

[3] Sun Shenghe, Lu Zheming. Technology and Application of Vector Quantization[M]. Beijing: Science Press, 2002: 31-48. [孙圣和, 陆哲明. 矢量量化技术及应用[M]. 北京: 科学出版社, 2002: 31-48.]
[4] Lin Y C, Wang C C. Digital images watermarking by vector quantization [C]//Proc. National Computer Symp. Taiwan, Taipei, China: Feng Chia University, 1999: 76-87.
[5] Jo M, Kim H. A digital image watermarking scheme based on vector quantization [J]. IEICE Trans. Inf. Syst., 2002, E85-D(6): 1054-1056.
[6] Du W C, Hsu W J. Adaptive data hiding based on VQ compressed images [J]. IEE Proc., Vis. Image Signal Process., 2003, 150(4): 233-238.
[7] Yang B, Lu Z M, Sun S H. Reversible watermarking in the VQ-compressed domain [C]//Proceedings of the Fifth International Conference on Visualization, Imaging, and Image Processing. Calgary: ACTA Press, 2005: 289-303.
[8] Lin C Y, Chang C C. Hiding data in VQ-compressed images using dissimilar pairs[J]. Journal of Computers, 2006, 17(2): 3-10.
[9] Chang C C, Lin C Y. Reversible steganographic method using SMVQ approach based on declustering[J]. Information Sciences, 2007, 177(8): 1796-1805.
[10] Chang C C, Kieu T D, Chou Y C. Reversible information hiding for VQ indices based on locally adaptive coding[J]. Journal of Visual Communication and Image Representation, 2009, 20(1): 57-64.
[11] Chang C C, Kieu T D, Wu W C. A lossless data embedding technique by joint neighboring coding[J]. Pattern Recognition, 2009, 42(7): 1597-1603.