

中图法分类号: TP391 文献标识码: A 文章编号: 1006-8961(2023)03-0734-15

论文引用格式: Wang K, Wu S W, Yin X L, Fu J Q, Chen B and Lu W. 2023. General variable-length code mapping relevant reversible data hiding in JPEG bitstream. Journal of Image and Graphics, 28(03): 0734-0748 (王可, 吴绍武, 尹晓琳, 付婧巧, 陈兵, 卢伟. 2023. 全体变长编码映射的 JPEG 可逆信息隐藏. 中国图象图形学报, 28(03): 0734-0748 [DOI:10.11834/jig.220343])

全体变长编码映射的 JPEG 可逆信息隐藏

王可, 吴绍武, 尹晓琳, 付婧巧, 陈兵, 卢伟*

1. 中山大学计算机学院, 广州 510006; 2. 广东省信息安全技术重点实验室, 广州 510006;
3. 机器智能与先进计算教育部重点实验室, 广州 510006

摘要: **目的** JPEG(joint photographic experts group)作为一种广泛使用的图像格式,对其进行可逆信息隐藏有实际应用价值。大多数以 JPEG 图像为载体的可逆信息隐藏方案会导致图像质量降低和文件膨胀。由于在 JPEG 比特流中,并没有使用较多的变长编码(variable-length code, VLC),因此以未使用的 VLC 替换已使用的 VLC 可以实现秘密信息的无损嵌入,但这类方法的嵌入容量有限,且会导致文件膨胀。为了实现更大的嵌入容量并更好地控制文件膨胀,本文提出了新的全体 VLC 映射方案。**方法** 首先,重排序原始比特流中的行程长度/幅值大小(run/size value, RSV),获得去除编码冗余的载体;然后,引入中间 VLC 的概念,根据 VLC 频数分布,确定使文件膨胀大小相对于嵌入载荷最小的最优中间 VLC 映射模型;接着,通过计算模拟嵌入下的文件膨胀大小,确定构建最优中间 VLC 映射所需的嵌入 VLC 和中间 VLC 集合;最后,根据最优中间 VLC 映射关系修改定义哈夫曼表和 JPEG 比特流实现秘密信息的嵌入。**结果** 实验在 USC-SIPI 数据库上与基于 DCT(discrete cosine transform)系数修改和基于 VLC 映射的方法进行对比。相较于基于限长 VLC 映射的方法,对不同的载体图像,本文方法的嵌入容量有 5~40 倍较大提升。与现有的典型可逆信息隐藏方法相比,在相同的嵌入容量下,本文方法嵌入造成的文件膨胀大小明显降低。对 QF(quality factor)为 90 的载体图像嵌入 1.8×10^4 比特的秘密信息,本文方法造成的文件膨胀最高可降低 46%。实验结果验证了 RSV 重排序的有效性,且提出的最优中间 VLC 映射可以在提供较大嵌入容量的同时保持较小的文件膨胀。**结论** 本文提出全体变长编码映射的 JPEG 可逆信息隐藏方法,通过构建最优的中间 VLC 映射,可以保持载体图像不失真并提供较大嵌入容量,且使文件膨胀更小。

关键词: 可逆信息隐藏(RDH);变长编码(VLC);定义哈夫曼表(DHT);JPEG 比特流;文件膨胀;嵌入容量

General variable-length code mapping relevant reversible data hiding in JPEG bitstream

Wang Ke, Wu Shaowu, Yin Xiaolin, Fu Jingqiao, Chen Bing, Lu Wei*

1. School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China;
2. Guangdong Province Key Laboratory of Information Security Technology, Guangzhou 510006, China;
3. Ministry of Education Key Laboratory of Machine Intelligence and Advanced Computing, Guangzhou 510006, China

Abstract: **Objective** Data hiding technique based secret data can be embedded into the cover images, and the certified receiver can extract it from the marked image. However, most of data hiding methods cannot restore the cover image after

收稿日期: 2022-04-27; 修回日期: 2022-06-29; 预印本日期: 2022-07-05

* 通信作者: 卢伟 luwei3@mail.sysu.edu.cn

基金项目: 国家自然科学基金项目(62261160653, 62102101); 阿里巴巴创新研究计划

Supported by: National Natural Science Foundation of China (62261160653, 62102101); Alibaba Innovative Research (AIR) Program

data extraction. Reversible data hiding (RDH) can be as a special sort of data hiding techniques to ensure the complete recovery of the cover image and the secret data. Nowadays, JPEG is the most popular image format, which is widely used in diverse photography devices and the internet. JPEG images are commonly-used as the cover images of RDH. For the RDH schemes in joint photographic experts group (JPEG) images, the performance evaluations are focused on embedding capacity, visual quality, and file size increment of the marked image. One category of RDH in JPEG images is based on the variable-length code (VLC) mapping, which can preserve the visual quality of the marked image unchanged. Therefore, only the optimizations for embedding capacity and file size increment are required. Most previous methods based on VLC mapping prefer to improve the embedding capacity while keeping the file size unchanged. Although both the visual quality and the file size of the marked image will not be changed, the embedding capacity is rather limited. To achieve sufficient embedding capacity in terms of slight file size increment, we develop a new RDH method on the basis of general VLC mapping, which provides larger embedding capacity with smaller file size increment. **Method** To embed secret data, VLC mapping-based RDH methods replace the used VLCs with the unused VLCs in the bitstream. To get larger embedding capacity and smaller file size increment, we propose the mechanism of intermediate VLC mapping. First, the default defined Huffman table (DHT) is used to encode the DCT coefficients during JPEG compression. The run/size value (RSV) with higher frequency may be assigned to the VLC with longer code lengths. So before data embedding, RSV reordering is conducted first, and VLCs with shorter code lengths are allocated to RSVs with higher frequencies. After RSV reordering, coding redundancy is reduced and the VLC frequency distribution can be used to measure the performance of VLC mapping relationships. Next, we analyze the effect of intermediate VLC mapping and direct VLC mapping on the file size of the marked image. The intermediate VLC mapping will perform better than direct VLC mapping in terms of keeping the file size when the frequency of the selected intermediate VLC is less than the number of bits equal to 1 in the secret data. The optimal intermediate VLC mapping model is then proposed. To construct a map from the used VLC to the unused VLC and minimize the file size increment, the VLC with the smallest frequency is selected as the intermediate VLC for each code length. Finally, our method is employed to construct the optimal intermediate VLC mapping for the given embedding capacity and the JPEG bitstream. For secret data extraction, the certified receiver can reconstruct the VLC mapping without auxiliary information. **Result** The experiments are carried out on the USC-SIPI database. The embedding capacity is greatly improved by 5 to 40 for multiple cover images compared to previous VLC mapping-related RDH methods. When 1.8×10^4 bits of secret data are embedded in the cover JPEG images with quality factor 90, our file size increment can be reduced by up to 42%. The experiments have verified the effectiveness of RSV reordering as well. The proposed mapping can reduce the file size increment while a large embedding capacity is provided. **Conclusion** A new strategy is implemented to construct VLC mapping, called optimal intermediate VLC mapping. To realize RDH in JPEG bitstream, our method can keep the cover image unchanged, provide larger embedding capacity, and optimize smaller file size increment further.

Key words: reversible data hiding (RDH); variable-length code (VLC); defined Huffman table (DHT); JPEG bitstream; file size increment; embedding capacity

0 引言

信息隐藏技术利用图像的相关性,可以将秘密信息隐形地嵌入到载体数据中,获得高视觉质量的载密图像,指定接收方可以从载密图像中提取出秘密信息。大多数的信息隐藏技术在数据嵌入过程中会对载体图像造成不可逆的破坏,可逆信息隐藏(reversible data hiding, RDH)是一种特殊的信息隐藏技术,可以保证载体数据和秘密信息的完整恢复。因此 RDH 被认为是遥感、医学诊断和司法图像处理

(Hou 等,2019;欧博 等,2022)等敏感场景下用于版权认证和完整性保护的有效方法。目前已有许多对无压缩图像提出的 RDH 方法,根据使用技术分为无损压缩(Fridrich 等,2002;Celik 等,2005)、差分扩展(Tian,2003;Dragoi 和 Coltuc,2014)和直方图平移(Ni 等,2006;Yin 等,2021;Zhang 等,2013)。

JPEG(joint photographic experts group)是目前最流行的图像格式,广泛应用于多种摄影设备和互联网传播,也是信息隐藏的常用载体(尹晓琳 等,2022;Lu 等,2021),以 JPEG 图像为载体的 RDH 方法也得到了研究人员的关注。但是由于 JPEG 压缩

需要减少高频分量以实现更好的压缩效果,上文提到的以无压缩图像为载体的 RDH 方法并不能直接应用于 JPEG 图像。目前 JPEG 图像的 RDH 方法根据使用的技术分为基于量化 DCT (discrete cosine transform) 系数修改的方法 (Huang 等, 2016; Hou 等, 2018)、基于量化表修改的方法 (Fridrich 等, 2002; Wang 等, 2013) 和基于变长编码 (variable-length code, VLC) 映射的方法 (Qian 和 Zhang, 2012; Hu 等, 2013)。

基于量化 DCT 系数修改的方法直接在 JPEG 文件解码出的量化 DCT 系数上修改,实现秘密信息的嵌入。Xuan 等人(2007)方法利用量化 DCT 系数的频数统计结果符合以 0 为中心的拉普拉斯分布的特点,将量化 DCT 系数分为 3 部分,即适合用于信息嵌入的部分、绝对值小于阈值而无需修改的部分和绝对值大于阈值需要平移为信息嵌入创造空间的部分,应用直方图平移技术实现 JPEG 图像的 RDH。之后,在 Xuan 等人(2007)方法的基础上提出了一系列方法,进一步优化了系数选择策略,如 Sakai 等人(2008)的方法。这些方法降低了信息嵌入导致的失真,但是以 JPEG 图像为载体的 RDH 方法的评价指标除了方法可提供的嵌入容量和载密图像的视觉质量之外,还包括载密图像相较于载体图像的文件膨胀大小。Huang 等人(2016)提出了新的量化 DCT 系数选择策略,即保持 0 系数不变,只扩展绝对值为 1 的系数用于嵌入信息,绝对值大于 1 的系数平移 1 位提供用于信息嵌入的空间,嵌入后所有量化 DCT 系数改变的幅值最大为 1,不仅保证了视觉质量,并且避免了由于修改 0 系数导致的文件膨胀。Hou 等人(2018)又在该方法基础上进一步优化了块和频率的选择策略,实现了更好的性能。基于量化 DCT 系数修改的 RDH 方法,在提供较大的嵌入容量的同时,造成的文件膨胀可接受并能获得视觉质量较高的载密图像,在 3 个评价指标下做到了较好的平衡,也因此得到了许多的关注研究。

基于量化表修改的 RDH 方法首先由 Fridrich 等人(2002)提出。在信息嵌入过程中,首先将量化表中为偶数的量化因子除以 2,再将载体图像中对应位置的量化 DCT 系数乘以 2,从而无损且可逆地提供可用于信息嵌入的冗余,然后将秘密信息嵌入到修改后的量化 DCT 系数的最小显著位 (least significant bits, LSB)。Chen 等人(2010)使用修改前后

的量化表将量化 DCT 系数映射到更大的表上,从而实现信息嵌入,可提供更大的嵌入容量和更好的视觉质量,但导致的文件膨胀较大。JPEG 标准默认的量化表提供了压缩图像在文件大小和视觉质量之间的平衡,而基于量化表修改的方法不可避免地破坏了压缩图像的平衡,因此这类方法很难同时在嵌入容量、视觉质量和文件膨胀上达到高性能,在数据嵌入后, JPEG 图像的存储大小会显著增加。

基于 VLC 映射的方法通过修改定义哈夫曼表和 JPEG 比特流实现信息嵌入。Mobasseri 等人(2010)发现在 JPEG 压缩过程中,只有小部分 VLC 用于编码量化 DCT 系数,即已使用的 VLC;而定义哈夫曼表中存在大量 VLC 未在 JPEG 比特流中出现,即未使用的 VLC。该方法构建了从已使用的 VLC 到未使用的 VLC 的映射,并在 JPEG 比特流中以未使用的 VLC 替换已使用的 VLC 实现秘密信息的嵌入,最后根据 VLC 映射关系修改 JPEG 文件头中的定义哈夫曼表实现原载体图像的可逆恢复。Qian 和 Zhang(2012)的方法和 Hu 等人(2013)的方法提出了新的构建 VLC 映射的方法,可以将已使用的 VLC 映射到多个未使用的 VLC,从而提供更大的嵌入容量。基于 VLC 映射的方法直接在 JPEG 比特流上修改实现信息嵌入,不需要编码和解码过程。这类方法直接修改了 VLC 与量化 DCT 系数的对应关系,因此修改前后的 JPEG 比特流解码出的量化 DCT 系数完全一致,即基于 VLC 映射的方法不会导致图像视觉质量的降低。另一方面,通过限定 VLC 映射关系中已使用 VLC 和未使用 VLC 的编码长度,基于 VLC 映射的方法可以很好地控制文件膨胀,部分方法能达到文件膨胀为 0 甚至是负数的性能。但目前提出的大部分基于 VLC 映射的方法相比于基于量化 DCT 系数和量化表修改的方法,所能提供的嵌入容量十分有限,为了解决这一问题,还需要构建更优的 VLC 映射规则。

总体来说,对于 JPEG 图像的 RDH 方案,基本的评价指标为嵌入容量、视觉质量和文件膨胀大小。嵌入容量与视觉质量之间、嵌入容量与文件膨胀大小之间都存在权衡。基于量化 DCT 系数和基于量化表修改的方法,需要同时对 3 个评价指标优化,但是基于 VLC 映射的方法并不会造成视觉质量的降低,因此只需要优化嵌入容量和文件膨胀大小。另一方面,基于 VLC 映射的方法也是相对高效的信息

嵌入方法,因为秘密信息直接嵌入到 JPEG 比特流中,不需要解压缩 JPEG 文件。但是以往的基于 VLC 映射的方法总是倾向于在不导致文件膨胀的前提下提高嵌入容量,虽然可以使载密图像的视觉质量和文件大小都与载体图像相同,但所能提供的嵌入容量过低,极大限制了这类方法的应用。而若基于 VLC 映射的方法能提供足够的嵌入容量,并且造成的文件膨胀是可接受的,可以更好地在大多数场景下应用。

本文提出了一种新的基于全体 VLC 映射的 JPEG 图像的 RDH 方法,在可接受的文件膨胀大小下,可提供较大的嵌入容量。首先,针对 JPEG 压缩过程中存在的编码冗余,优化了原始图像的定义哈夫曼表,根据新的编码表修改 JPEG 比特流后获得去除了编码冗余的载体;其次,分析了 VLC 映射导致的文件膨胀,进一步提出了中间 VLC 的概念,并论证了使用中间 VLC 映射相比于直接 VLC 映射,造成的文件膨胀更小;然后,基于 JPEG 比特流中 VLC 频数的统计特征,构建了最优中间 VLC 映射模型;最后,计算模拟嵌入下文件膨胀大小,确定最优的中间 VLC 映射。实验结果表明,本文方法相较于之前的 JPEG 图像的 RDH 方法,可以更好地平衡嵌入容量和文件膨胀。

1 JPEG 变长编码

1.1 JPEG 比特流结构

在 JPEG 压缩过程中,空域图像经过 DCT 变换和量化,像素值转化为量化直流(direct current, DC)系数和量化交流(alternating current, AC)系数。接下来,在熵编码阶段,首先获得量化 AC 和量化 DC 系数的中间形式。量化 AC 系数采用行程长度编码预压缩,被转换为行程长度/幅值大小(run size value, RSV)的中间形式,其中,行程长度为两个非零量化 AC 系数之间 0 系数的个数,幅值大小为编码下一个非零量化 AC 系数所需的编码长度。量化 DC 系数的中间形式记录了相邻量化 DC 系数的差值。量化 AC 系数的中间形式 RSV 经过哈夫曼编码实验 VLC 编码,量化 DC 系数的中间形式使用 VLI 编码。最后使用 2 进制编码 VLC 和 VLI 替换量化 AC 和 DC 系数的中间形式得到 JPEG 比特流的压缩数据段。编码和中间形式的对应关系分别存储在

AC 和 DC 系数的定义哈夫曼表(defined Huffman table, DHT)中,定义哈夫曼表以及量化表存储在 JPEG 文件头中,JPEG 文件头和压缩数据段组成了 JPEG 比特流。

JPEG 文件头中的 DHT 字段中定义了构造哈夫曼表所需要的信息,量化 AC 系数的 DHT 结构如图 1 所示。根据编码长度将 VLC 分为 16 类,每个类中的 VLC 个数和每个 VLC 对应的 RSV 都记录在 DHT 中。图 1 中, N_i 表示长度为 i 的 VLC 的个数, $R_{i,j}$ 表示对应 VLC 的编码长度为 i 的第 j 个 RSV。由于 JPEG 标准中使用的 VLC 是固定的,只需将 $R_{1,1}$ 到 $R_{16,N_{16}}$ 按顺序对应到每个 VLC 便可实现 JPEG 文件的编码和解码。

DHT文件头		N_1	N_2	...	N_{15}	N_{16}
$R_{1,1}$	...	R_{1,N_1}	...	$R_{16,1}$	...	$R_{16,N_{16}}$

图 1 DHT 的结构

Fig. 1 The structure of DHT

1.2 变长编码映射

JPEG 标准的 DHT 中定义了 162 个 VLC,但 JPEG 比特流中只使用了小部分编码。通过修改量化 AC 系数的 DHT,改变 RSV 对应的 VLC,构建从已使用的 VLC 到未使用的 VLC 的映射关系,并根据该映射关系将压缩数据段中已使用的 VLC 替换为未使用的 VLC,从而实现信息嵌入。

图 2 给出了一个秘密信息嵌入到 JPEG 比特流上的例子。给定 JPEG 比特流,解码出部分的 RSV 和 AC 系数为 $\{(1,6), 20, (1,6), -16\}$ 。其中, $(1,6)$ 为行程长度为 0、幅值为 6 的 RSV, 20 和 -16 为非零量化 AC 系数的幅值。比特流中并未出现 $(4,5)$ 对应的 VLC,因此可以构建 $(1,6) \rightarrow (4,5)$ 的映射关系用于嵌入秘密信息。为了保证信息嵌入不破坏图像质量,需要修改 DHT 中 $(4,5)$ 为 $(1,6)$,使原 $(4,5)$ 对应的 VLC 和原 $(1,6)$ 对应的 VLC 解码出的 RSV 一致。为了嵌入 1 比特的秘密信息“1”,以原始 DHT 中 $(4,5)$ 对应的 VLC 替换 $(1,6)$ 对应的 VLC,若嵌入 1 比特的秘密信息“0”,则不需要修改。嵌入信息后,JPEG 比特流解码出的 RSV 和 AC 系数仍然为 $\{(1,6), 20, (1,6), -16\}$ 。总的来说,基于 VLC 映射实现信息嵌入改变了 JPEG 比特流,同时也修改了 DHT 中 RSV 和 VLC 的对应关系,

因此不会影响解码出的 DCT 系数,从而实现无损可逆的信息隐藏。

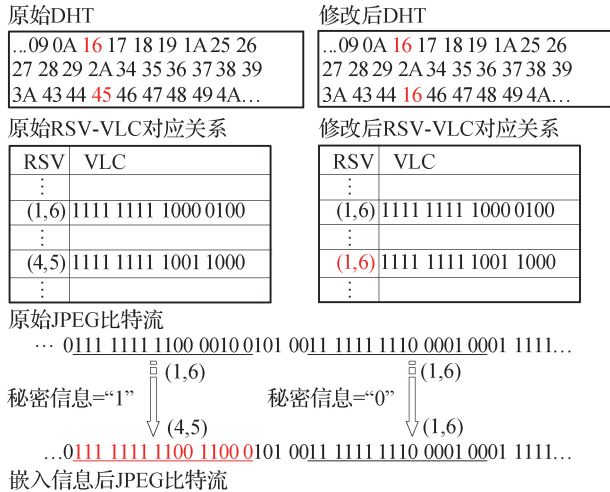


图2 VLC映射实现 JPEG 可逆信息隐藏的示例

Fig. 2 An example of RDH by VLC mapping in JPEG bitstream

基于 VLC 映射的 RDH 方法不会降低图像质量,因此 VLC 映射的构造需要考虑如何提供更大的嵌入容量并使文件膨胀更小。

JPEG 图像 Baboon 中不同长度类的 VLC 的平均使用次数和其中已使用和未使用的 VLC 的个数的统计结果如图 3 所示。根据统计结果,VLC 的平均使用次数与其编码长度为负相关;编码长度较短的 VLC 基本已被使用,而未使用的 VLC 编码长度较长。本文定义限长 VLC 映射,其构建映射的规则为:编码长度等于 a 的 VLC 只能被映射到编码长度小于或等于 a 的 VLC。使用限长 VLC 映射做信息嵌入,文件膨胀必小于等于 0。根据 VLC 的统计结果,在编码长度较短的类中很难构建符合要求的映射关系;在编码长度较长的类中,可以构建映射关系,但是 VLC 的使用次数较少,即比特流中可被替换的 VLC 个数较少,因此嵌入容量受到很大限制。

构建全体 VLC 映射的规则为:不需限制 VLC 映射关系中已使用 VLC 和未使用 VLC 的编码长度,编码长度等于 a 的 VLC 可以被映射到编码长度等于 b 的 VLC。构建全体 VLC 映射时,可以选择出现频数高而编码长度较短的 VLC,映射到编码长度较长的未使用的 VLC,在一定程度上解决了嵌入容量不足的问题。但是使用全体 VLC 映射做信息嵌入,在嵌入载荷较大时,易造成严重的文件膨胀。如图 3 所示,若被选取用做载体 VLC 的已使用的 VLC

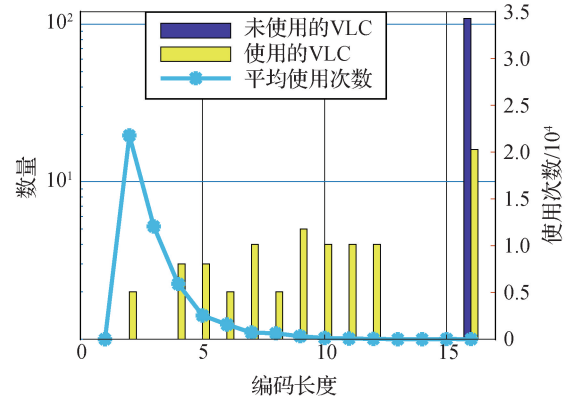


图3 16个类别中使用与未使用的 VLC 个数和平均使用次数

Fig. 3 The number of the used and unused VLCs and the average occurrence in each category

需要满足大容量的嵌入需求,则其编码长度会较短;但未使用的 VLC 只存在于编码长度为 16 的类中。因此在嵌入秘密信息时会导致大量编码长度短的 VLC 被编码长度长的 VLC 替换,这显然会导致较大的文件膨胀。

2 本文方法

2.1 文件膨胀大小分析

依据是否在 JPEG 比特流的压缩数据段使用,将 VLC 分为已使用的 VLC 集合 $S = \{v_1^{So}, v_2^{So}, \dots, v_n^{So}\}$ 和未使用的 VLC 集合 $U = \{v_1^{Uo}, v_2^{Uo}, \dots, v_m^{Uo}\}$ 。其中, n 为已使用的 VLC 个数, m 为未使用的 VLC 个数。 S 中元素在压缩数据段中出现频数的集合为 $H = \{h_1^o, h_2^o, \dots, h_n^o\}$, S 和 U 中元素编码长度的集合为 $L = \{l_1^o, l_2^o, \dots, l_n^o, l_{n+1}^o, \dots, l_{n+m}^o\}$, 集合中元素的下标索引 1 ~ (n + m) 对应 R_1 到 R_{16} 的顺序。原始压缩数据段的大小为

$$F = \sum_{i=1}^n h_i^o \times l_i^o \quad (1)$$

为了更清晰地分析造成文件膨胀的因素,先考虑单对单 VLC 映射的简化情况。设定秘密信息是长度为 P 的比特串,其中,“1”的个数为 h_s 。构建从已使用的 v_e^{So} 到未使用的 v_u^{Uo} 的映射, $v_e^{So} \rightarrow v_u^{Uo}$, 其中, v_e^{So} 为用于嵌入信息的 VLC,称为载体 VLC。显然为了实现秘密信息的完整嵌入,载体 VLC 的频数 h_e^o 需要大于等于 P 。

定义 $v_e^{So} \rightarrow v_u^{Uo}$ 为直接 VLC 映射。频数直方图

改变如图4(a)所示, 嵌入信息后压缩数据段中的 h_s 个 v_e^{So} 被 v_u^{Uo} 替换, v_e^{So} 的频数由 h_e^o 变为 $h_e^o - h_s$, v_u^{Uo} 的频数由 0 变为 h_s 。因此, 嵌入信息引起的文件膨胀大小为

$$I_1 = (l_{u+n}^o - l_e^o) \times h_s \quad (2)$$

若 $l_e^o \geq l_{u+n}^o$, 则该映射为 1.2 节介绍的限长 VLC 映射, 虽然 $I_1 \leq 0$ 不会造成文件膨胀, 但并不能提供较大的嵌入容量; 若 $l_e^o \leq l_{u+n}^o$, 则该映射为全体 VLC 映射, 虽然可以满足较大嵌入容量的需求, 但为了保证 $h_e > P$, 可能出现 $l_e^o = 2$ 而 $l_{u+n}^o = 16$ 的极端情况, 文件膨胀会较大。

由于直接 VLC 映射无法同时满足对嵌入容量和文件膨胀的需求, 考虑通过引入中间 v_c^{So} 完成从 S 中元素到 U 中元素的映射, 映射变为 $v_e^{So} \rightarrow v_c^{So} \rightarrow v_u^{Uo}$, 称为中间 VLC 映射。频数直方图改变如图4(b)所示, 比特流中 h_s 个 v_e^{So} 被 v_c^{So} 替换, 而 v_c^{So} 全部被 v_u^{Uo} 替换, v_e^{So} 的频数改变和直接 VLC 映射相同, 中间 VLC 的频数由 h_e^o 变为 h_s , v_u^{Uo} 的频数由 0 变为 h_e^o , 嵌入信息造成的文件膨胀大小为

$$I_2 = (l_c^o - l_e^o) h_s + (l_{u+n}^o - l_c^o) h_e^o \quad (3)$$

只需满足 $h_c^o < h_s$ 和 $l_e^o < l_c^o < l_{u+n}^o$, 便有 $I_2 < I_1$, 则中间 VLC 映射相对于直接 VLC 映射, 可以在提供相同嵌入容量的前提下, 减小文件膨胀。本文从影响 JPEG 比特流大小的因素出发, 通过优化 DHT 去除编码冗余, 确定最优的中间 VLC 映射, 以在提供较大嵌入容量的同时尽可能缩小文件膨胀。

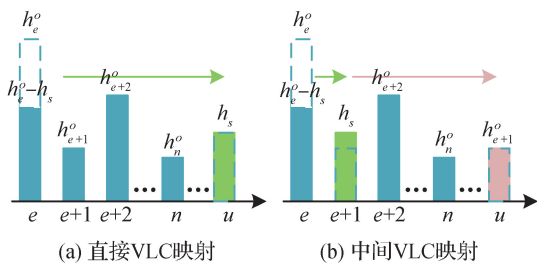


图4 直接和中间 VLC 映射示例

Fig.4 The examples of direct and intermediate VLC mapping
((a) direct VLC mapping; (b) intermediate VLC mapping)

2.2 RSV 重排序

根据标准的哈夫曼编码, 频数越高的字符会被分配长度越短的编码, 以使得平均编码长度最短。JPEG 压缩过程中的熵编码也利用该思想进一步压缩数据, 但为了节省运算开销和能够快速地重建出

哈夫曼编码树, JPEG 标准使用的是范式哈夫曼编码。RSV 和 VLC 的对应信息存储在 DHT 中, 因此可以直接编码 RSV 得到比特流。所以在实际情况中, 频数高的 RSV 对应的 VLC 的编码长度并不一定小于频数低的 RSV 对应的 VLC 的编码长度。RSV 重排序前的 VLC 频数如图5(a)所示, 其中, 横坐标为从 R_1 到 R_{16} 的编码长度增长, 但对应的 VLC 频数并不是单调下降。根据式(1), 更大的 h_i 对应更小的 l_i 能取得更小的 F , 因此 JPEG 标准提供的 DHT 对于图像压缩来说并不是最优的, 会造成编码的冗余。

在进行正式的信息嵌入之前, 先对载体图像做 RSV 重排序, 为频数更高的 RSV 分配编码长度更短的 VLC。RSV 重排序后, 已被使用和未被使用的 VLC 集合分别为 $S' = \{v_1^S, v_2^S, \dots, v_n^S\}$ 和 $U' = \{v_1^U, v_2^U, \dots, v_m^U\}$, S' 中元素的频数集合为 $H' = \{h_1, h_2, \dots, h_n\}$, S' 和 U' 中元素的长度集合为 $L' = \{l_1, l_2, \dots, l_n, l_{n+1}, l_{n+2}, l_{n+m}\}$, H' 和 L' 中的元素满足的条件为

$$\begin{cases} h_i \geq h_j & i < j \\ l_i \geq l_j & i > j \end{cases} \quad (4)$$

RSV 重排序操作所缩小的文件大小为 $F_R =$

$\sum_{i=1}^n (h_i^o \times l_i^o) - (h_i \times l_i)$ 。RSV 重排序修改了 RSV 和 VLC 的对应关系, 并相应更新了压缩数据段中的 VLC, 并不会修改解码后得到的载体图像。

RSV 重排序后再次统计从 $R_{1,1}$ 到 $R_{16,N_{16}}$ 对应的 VLC 的频数, 如图5(b)所示, 呈单调下降。RSV 重排序不仅去除了原始数据流的编码冗余, 也有助于下一步寻找使文件膨胀最小的中间 VLC 映射, 即最优中间 VLC 映射。

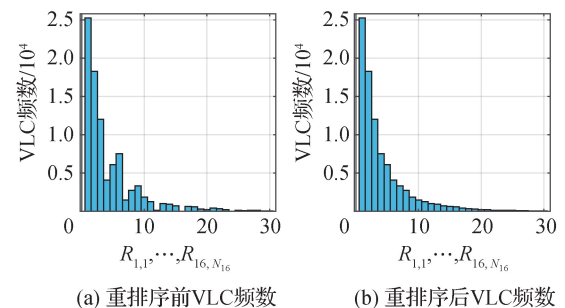


图5 RSV 重排序前后 VLC 频数

Fig.5 The frequency of VLCs before and after
RSV reordering ((a) before RSV reordering;
(b) after RSV reordering)

2.3 最优中间 VLC 映射

在 RSV 重排序后得到单调下降的 VLC 频数分布的基础上,构建最优中间 VLC 映射。根据式(3),形如 $v_e^S \rightarrow v_c^S \rightarrow v_u^U$ 的中间 VLC 映射,导致的文件膨胀可以分为嵌入映射膨胀和中间映射膨胀两部分。嵌入映射膨胀为使用 v_c^S 替换比特流中的 v_e^S 导致的膨胀,记为 $I_E = (l_c - l_e)h_s$; 影响因素有秘密信息中“1”的个数 h_s , v_e^S 与 v_c^S 的编码长度差、频数差。其中, h_s 只由秘密信息决定,无法在嵌入过程中优化这一影响因素。中间映射膨胀为使用 v_u^U 替换比特流中的 v_c^S 导致的膨胀,记为 $I_I = (l_{n+u} - l_c) \times h_c$, 影响因素有中间 VLC 的频数 h_c 以及 v_c^S 和 v_u^U 的编码长度差。

首先,考虑确定最优 v_u^U 。为了缩小 v_c^S 和 v_u^U 的编码长度差,选择 U' 中编码长度最小的 VLC 作为 v_u^U 可以使 l_{n+u} 取到最小值,而在 RSV 重排序后 v_1^U 有着 U' 中 VLC 中最小的编码长度,因此可确定 $u=1$ 。

其次,确定最优的 v_c^S 。固定 l_c , 使 h_c 最小,则令 v_c^S 为 S' 中编码长度等于 l_c 的 VLC 中频数最小的 VLC。

至此,已初步确定 v_u^U 和 v_c^S 的选择策略,若进一步缩小中间映射膨胀,可将 $v_c^S \rightarrow v_u^U$ 视为一个直接 VLC 映射,通过再引入一个中间 VLC 来减小中间映射膨胀。

再引入新的中间 v_{c1}^S , 使 $v_e^S \rightarrow v_c^S \rightarrow v_u^U$ 转为 $v_e^S \rightarrow v_{c1}^S \rightarrow v_u^U$, 称 v_{c1}^S 在新的映射中为起始中间 VLC。使用新的 VLC 映射嵌入秘密信息后,比特流中 h_c 个 v_c^S 被 v_{c1}^S 替换,而 v_{c1}^S 全部被 v_u^U 替换, v_c^S 的频数仍为从 h_c 变为 h_s , v_{c1}^S 的频数由 h_{c1} 变为 h_c , v_u^U 的频数由 0 变为 h_{c1} 。中间映射膨胀的计算式为

$$I_I^1 = (l_{c1} - l_c)h_c + (l_{n+u} - l_{c1})h_{c1} \quad (5)$$

因此,只需满足 $h_{c1} < h_c$ 和 $l_c < l_{c1} < l_{n+u}$, 便有 $I_I^1 < I_I$, 即相对于 $v_c^S \rightarrow v_u^U$, 使用新的映射可以进一步缩小中间映射膨胀。因为 U' 中的元素对应的频数和编码长度在 H' 和 L' 中的索引必大于 S' , 经过 RSV 重排序,根据式(4),必有 $h_{c1} < h_c$ 和 $l_{c1} < l_{n+u}$ 。同样,先固定 l_{c1} , 使 h_{c1} 最小,则令 v_{c1}^S 为 S' 中编码长度等于 l_{c1} 的 VLC 中频数最小的 VLC。由于直接增大或减小 l_{c1} 对 I_I^1 中 $(l_{c1} - l_c)h_c$ 和 $(l_{n+u} - l_{c1})h_{c1}$ 两项造成的大小改变是相反的,因此若想进一步缩小 I_I^1 , 需要再将 $v_{c1}^S \rightarrow v_u^U$ 视为直接映射,即再通过引入

新的中间 VLC 来减小中间映射膨胀。

根据上述分析,可以通过不断引入新的 VLC 降低中间映射膨胀。使中间映射膨胀最小而引入的所有中间 VLC 组成集合 $VI = \{v_{c0}^S, v_{c1}^S, \dots, v_{cx}^S\}$ 。值得注意的是,必有 $l_{cx} = l_{n+u} - 1$ 和 $l_{ci} - l_{c(i-1)} = 1$, 因为若 $l_{n+u} - l_{cx} \geq 2$ 或 $l_{ci} - l_{c(i-1)} \geq 2$, 则可以通过对映射 $v_{cx}^S \rightarrow v_u^U$ 或 $v_{c(i-1)}^S \rightarrow v_{ci}^S$ 引入新的中间 VLC 来进一步缩小中间映射膨胀,这与 VI 是使 $v_{c0}^S \rightarrow v_u^U$ 导致的文件膨胀最小而引入的所有中间 VLC 组成的集合矛盾。综上所述,集合 VI 由编码长度等于 $l_{c0} \sim l_{n+u} - 1$ 的类中频数最小的 VLC 组成,其中所有元素的频数和编码长度满足

$$\begin{cases} l_{ci} < l_{cj} & 0 \leq i < j \leq x \\ h_{ci} > h_{cj} & 0 \leq i < j \leq x \end{cases} \quad (6)$$

新的中间映射导致文件膨胀的计算式为

$$I_{VI}^2 = (l_{n+u} - l_{cx}) \times h_{cx} + \sum_{i=1}^x [(l_{ci} - l_{c(i-1)}) \times (h_{c(i-1)} - h_{ci})] \quad (7)$$

引入 $x+1$ 个中间 VLC 后,中间映射膨胀达到最小,实现了最优中间 VLC 映射。在最优中间 VLC 映射模型下,从 v_e^S 到 v_u^U 的映射过程为 $v_e^S \rightarrow v_{c0}^S \rightarrow v_{c1}^S \rightarrow \dots \rightarrow v_{cx}^S \rightarrow v_u^U$ 。嵌入秘密信息后 VLC 频数的变化统计如图 6 所示。嵌入映射为 $v_e^S \rightarrow v_{c0}^S$ 时, v_e^S 的频率由 h_e 变为 $h_e - h_s$, 中间映射为 $v_{c0}^S \rightarrow \dots \rightarrow v_{cx}^S \rightarrow v_u^U$, v_{c0}^S 的频率由 h_{c0} 变为 h_s , 之后, v_{ci}^S 的频率由 h_{ci} 变为 $h_{c(i-1)}$, v_u^U 的频率由 0 变为 h_{cx} 。

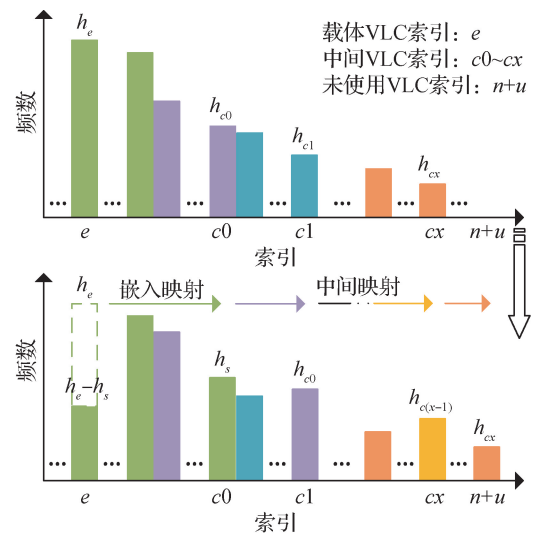


图 6 最优中间 VLC 映射模型

Fig. 6 Optimal intermediate VLC mapping model

下面给出对载体图像和嵌入载荷构建最优中间 VLC 映射的方法。

若只使用一个载体 VLC 构建映射,能提供的嵌入容量至多为 JPEG 比特流中 VLC 的最大频数 h_1 ,为了更好地应对大嵌入载荷的需求,需要再增加一个载体 VLC,进一步增大可提供的嵌入容量。将嵌入载荷分配到两个载体 VLC 上,可以使选择的载体 VLC 的长度与起始中间 VLC 的长度更接近,从而进一步降低嵌入映射膨胀。第 1 个映射为 $v_{e1}^S \rightarrow v_{u1}^U$,使用 v_{e1}^S 作为载体 VLC 映射到 v_{u1}^U ,有 $u1 = 1$;第 2 个映射为 $v_{e2}^S \rightarrow v_{u2}^U$,使用 v_{e2}^S 作为载体 VLC 映射到 v_{u2}^U ,由于 v_{e1}^S 在第 1 个映射中已被使用,因此选择在 U' 中编码长度仅大于等于 v_{e1}^S 的 v_{u2}^U ,有 $u2 = 2$ 。为了简化运算,本文方法设置约束如下:在第 1 个映射达到最大嵌入容量后,再通过第 2 个映射嵌入信息。

嵌入映射和中间映射导致的文件膨胀都与 v_{e0}^S 的编码长度有关,若 v_{e0}^S 的编码长度偏短而与 v_e^S 的编码长度接近,则会减小嵌入映射膨胀而增大中间映射膨胀;反之,则会增大嵌入映射膨胀而减小中间映射膨胀。因此起始中间 VLC 长度 l_{e0} 并不能被直接确定。在满足嵌入容量要求的前提下,需要尽可能地选择编码长度与 l_{e0} 接近的 v_e^S 以缩小嵌入映射膨胀。假设给定的秘密信息中“1”的个数为 h_s ,分配给第 1 个映射的嵌入载荷为 h_{s1} ,分配给第 2 个映射

的嵌入载荷为 h_{s2} 。对载体比特流,使用不同的载体 VLC 和中间 VLC 集合 VI 构建映射做模拟嵌入,计算文件膨胀大小。可得到在最优中间 VLC 映射模型下,使文件膨胀最小的映射及其使用的载体 v_{e1}^S , v_{e2}^S 和中间 VLC 集合 VI_1 , VI_2 。

确定载体 v_{e1}^S , v_{e2}^S 和中间 VLC 集合 VI_1 , VI_2 的过程可描述为

$$\begin{aligned} v_{e1}^S, v_{e2}^S, VI_1, VI_2 &= \arg \min(FSI) \\ \text{s. t. } VI_1 &\subseteq \text{Min}_1, VI_2 \subseteq \text{Min}_2, h_{e1} + h_{e2} \geq P \end{aligned} \quad (8)$$

式中, Min_1 和 Min_2 分别为各编码长度类中频数最小和次小的 VLC 集合,总文件膨胀 FSI 的计算式为 $FSI = (l_{n+u1} - l_{e1})h_{s1} + (l_{n+u2} - l_{e2})h_{s2} + I_{VI_1}^2 + I_{VI_2}^2$ (9)

下面给出一个实例,对 RSV 重排序后的载体图像计算式(8)确定的载体 VLC 为 v_3^S 和 v_2^S ; VI_1 由编码长度等于 5~15 的类中频数最小的 VLC 组成, VI_2 由编码长度等于 5~12 的类中频数最小的 VLC 组成,总映射过程如图 7 所示。可以看出,构成映射的 VLC 的编码长度并不完全满足 $l_{ex} = l_{n+u} - 1$ 和 $l_{ci} - l_{c(i-1)} = 1$,因为根据 JPEG 标准的 DHT,各编码长度类中 VLC 个数并不完全满足需求,如编码长度为 13、14 比特的 VLC 个数为 0,而编码长度为 15 比特的 VLC 个数为 1。即使如此,使用本文的最优中间 VLC 映射模型,仍能使中间映射膨胀最小。

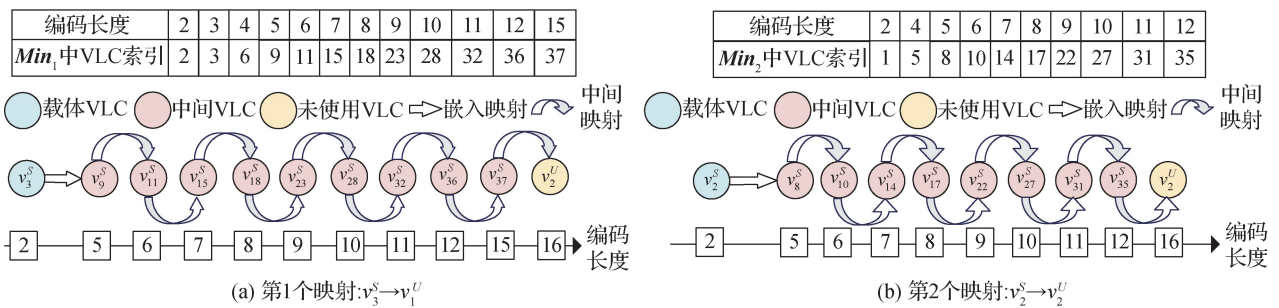


图7 最优中间 VLC 映射过程

Fig. 7 The optimal intermediate mapping process ((a) the first mapping: $v_3^S \rightarrow v_1^U$; (b) the second mapping: $v_2^S \rightarrow v_2^U$)

下面通过一个示例说明由接收到的 JPEG 比特流中的哈夫曼表倒推出嵌入信息时所用 VLC 映射的过程。

如图 8 所示,有两个相同的 RSV 字符“11”,嵌入映射为 $(1,1)_v^1 \rightarrow (1,1)_v^2$ 。 $(1,1)_v^1$ 为第 1 个“11”

对应的 VLC,即载体 v_e^S ; $(1,1)_v^2$ 为第 2 个“11”对应的 VLC,即起始中间 v_{e0}^S 。第 2 个“11”为 DHT 中第 9 个 RSV,结合 DHT 表中记录的每个编码长度类中 VLC 个数,其编码长度为 5,则根据最优中间 VLC 映射模型,可确定 VI 由编码长度等于 5~15 的类

中频数最小的 VLC 构成。按每个编码长度类中频数最小的 VLC 索引取得所有中间 VLC 组成集合 VI , 其中元素为 $(1,1)_v^2$, $(3,1)_v$, $(4,1)_v$, $(2,2)_v$, $(9,1)_v$, $(15,0)_v$, $(3,3)_v$, $(7,2)_v$ 。若嵌入信息时使用了两个映射, 则 DHT 中会有两对相同的 RSV, 可同样倒推出嵌入时使用的映射。因此使用本文方法接收方并不需要辅助信息就可以构建出映射。但是需要在秘密信息前加上 16 位的辅助信息用于存储秘密信息的长度, 并与秘密信息一起嵌入到经过 RSV 重排序的 JPEG 比特流中, 用于指示接收端何时停止秘密信息的提取。

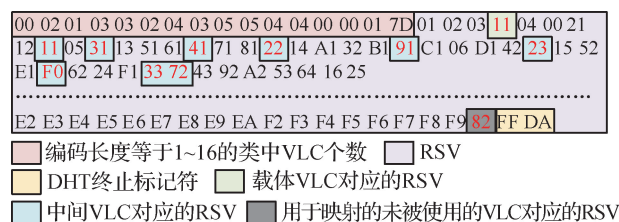


图8 修改后的 DHT

Fig. 8 The DHT after embedding secrets

2.4 秘密信息的嵌入和提取过程

对 JPEG 比特流 J 嵌入总长度为 P 的包含 16 位辅助信息的秘密信息, 详细过程如下:

1) 解析原始的 JPEG 比特流 J , 获得文件头中量化 AC 系数的 DHT。重排序操作确定新的 RSV 和 VLC 的对应关系, 根据新的对应关系修改原始 DHT 得到 D , 使用 D 编码 J 的压缩数据段并替换文件头中的 DHT 得到 J' 。

2) 统计 J' 的压缩数据段中 VLC 频数, 根据式(8)获得构建最优中间 VLC 映射需要的载体 v_{e1}^s 和 v_{e2}^s 、中间 VLC 集合 VI_1 和 VI_2 。

3) 根据最优中间 VLC 映射关系修改 D , 得到 D' 。

4) 先通过 v_{e1}^s 和 VI_1 构建的第 1 个映射修改 J' 的压缩数据段嵌入信息。若秘密信息未嵌入完成, 再通过 v_{e2}^s 和 VI_2 构建的第 2 个映射嵌入信息。

5) 以 D' 替换 J' 文件头中的 D , 获得载密比特流 J^s 。

从 JPEG 比特流 J^s 中提取秘密信息并恢复载体图像的详细过程如下:

1) 解码接收到的 JPEG 比特流 J^s , 获得文件头中量化 AC 系数的哈夫曼表 D' , 根据 D' 倒推出嵌

入信息时的 VLC 映射。

2) 先提取出前 16 位数据为嵌入秘密信息的长度, 再提取出相应长度的秘密信息, 并根据映射关系恢复原始的压缩数据段。

3) 使用 DHT 替换 J^s 文件头中的 D' , 以恢复的原始压缩数据段替换 J^s 的压缩数据段组成原始载体图像 J 。

3 实验结果

为了验证本文方法的性能, 对嵌入容量和文件膨胀性能进行对比实验。实验从 USC-SIPI (Weber, 1997) 数据库中获得 tiff 格式的测试图像, 并使用 MATLAB R2017a 将测试图像转化为 $512 \times 512 \times 8$ bit 的 JPEG 格式的灰度载体图像, 与基于 VLC 映射的 JPEG 图像的 RDH 方法 (Hu 等, 2013; Qiu 等, 2018; Du 等人, 2022) 和基于量化 DCT 系数修改的 JPEG 图像的 RDH 方法 (Huang 等, 2016; Hou 等, 2018) 进行对比。

3.1 基于 VLC 映射的方法对比

从 USC-SIPI 数据库中挑选 Baboon, Boat, Bridge, Lena, Peppers 和 Splash 为测试图像, 以 10 ~ 90 的 9 个质量因子 (quality factor, QF) 对测试图像进行 JPEG 压缩获得载体图像。统计 Hu 等人 (2013) 方法、Qiu 等人 (2018) 方法和本文方法对 6 幅载体图像所能嵌入秘密信息的最大比特数, 结果如表 1 所示。3 种方法所能提供的嵌入容量都依赖于所选择的载体 VLC 的频数, 因此对不同图像应用同一种方法所能提供的嵌入容量也差距较大。正如前文分析, Hu 等人 (2013) 方法和 Qiu 等人 (2018) 方法提供的嵌入容量十分有限, 这是由于需要保证 VLC 映射中已使用的 VLC 和未使用的 VLC 的编码长度相同, 二者都只能选择频数较低的 VLC 作为载体 VLC。本文方法能提供的嵌入容量相对于 Hu 等人 (2013) 方法和 Qiu 等人 (2018) 方法有着绝对优势, 对不同载体图像, 本文方法能提供的嵌入容量是 Hu 等人 (2013) 方法和 Qiu 等人 (2018) 方法的 5 倍以上, 且差距随着质量因子 QF 的增长而扩大。对于 QF = 90 的载体图像, Hu 等人 (2013) 方法和 Qiu 等人 (2018) 方法仅提供数千比特的嵌入容量, 而本文方法能提供数万比特的嵌入容量。值得注意的是, 本文方法能提供的嵌入容量随着 QF 的增大有

明显的上升趋势,但使用限长 VLC 映射的方法能提供的嵌入容量可能随着 QF 的增大而减小。表 1 第 1 行为 Baboon 图像的测试结果。在 $QF = 10$ 时, Hu 等人(2013)方法和 Qiu 等人(2018)方法能提供 7×10^3 bit 左右的嵌入容量,但在 $QF = 90$ 时,Hu 等

人(2013)方法和 Qiu 等人(2018)方法能提供的嵌入容量缩减为不到 1×10^3 bit。与之相反的是,本文方法对 Baboon 图所能提供的嵌入容量随着 QF 的增加由 1×10^4 增加至 6×10^4 bit,这也说明本文方法能够更稳定地满足嵌入载荷的需求。

表 1 不同质量因子下嵌入容量的对比
Table 1 Comparison of embedding capacity with different quality factors

图像	方法	质量因子(QF)								
		/bit								
		10	20	30	40	50	60	70	80	90
Baboon	Hu 等人(2013)	6 576	3 478	1 758	1 790	1 268	1 304	1 353	757	681
	Qiu 等人(2018)	7 169	3 767	1 970	2 077	1 482	1 582	1 634	1 033	939
	本文	11 661	19 762	26 279	30 879	34 687	38 636	43 548	51 541	67 540
Boat	Hu 等人(2013)	950	551	691	571	687	811	679	1 000	1 929
	Qiu 等人(2018)	1 635	1 185	1 242	1 020	1 079	1 139	962	1 249	2 172
	本文	7 786	10 871	13 671	16 465	18 835	21 304	24 758	30 127	42 438
Bridge	Hu 等人(2013)	1 802	999	713	860	1 035	714	817	596	615
	Qiu 等人(2018)	2 473	1 514	1 099	1 193	1 269	926	1 002	782	858
	本文	10 473	17 149	23 619	28 002	31 703	35 279	39 875	46 887	62 558
Lena	Hu 等人(2013)	565	326	259	296	369	204	286	351	576
	Qiu 等人(2018)	1 256	1 004	913	938	999	767	760	695	842
	本文	6 780	8 661	10 183	11 298	12 923	14 951	17 952	23 083	35 282
Peppers	Hu 等人(2013)	567	517	717	493	557	652	455	374	1 562
	Qiu 等人(2018)	1 263	989	1 078	923	928	964	774	695	1 775
	本文	6 950	8 667	9 989	11 136	12 981	15 098	18 711	24 636	38 818
Splash	Hu 等人(2013)	497	597	1 064	1 345	1 382	1 547	1 180	1 147	710
	Qiu 等人(2018)	1 193	1 169	1 257	1 545	1 655	1 821	1 474	1 407	1 037
	本文	5 372	6 428	7 681	8 855	10 225	11 789	14 412	20 512	33 093

注:加粗字体表示各图像对应的最优结果。

全体 VLC 映射在能提供较大嵌入容量的同时,也会导致所得到的载密图像的存储大小的膨胀。本文通过 RSV 重排序和构建最优的中间 VLC 映射来减小文件膨胀。由于 Hu 等人(2013)方法和 Qiu 等人(2018)方法使用的是限长 VLC 映射,而 Du 等人(2022)方法和本文方法使用的是全体 VLC 映射,两类方法所能提供的嵌入容量差距较大,实验分别使用较小和较大的嵌入载荷进行测试,并统计所导致

的文件膨胀,实验结果如表 2 所示。在嵌入载荷为 800 bit 时,Hu 等人(2013)方法和 Qiu 等人(2018)方法基本可以完成秘密信息的嵌入,且不会造成文件大小的改变,即文件膨胀为 0。Du 等人(2022)方法和本文方法都对载体图像的编码冗余做了一定处理,在嵌入载荷较小时,信息嵌入导致的文件膨胀可被编码冗余完全抵消,因此文件膨胀为负数。如 $QF = 20$ 的载体图像嵌入 800 bit 的秘密信息的实验

表2 本文方法与基于 VLC 映射的方法在不同质量因子下嵌入不同长度秘密信息时文件膨胀大小的对比
Table 2 Comparison of file size increment between previous VLC-based methods and ours with different quality factors and embedded with different lengths of secret data

图像	方法	嵌入载荷 (QF = 20)		嵌入载荷 (QF = 30)		嵌入载荷 (QF = 40)		嵌入载荷 (QF = 70)		嵌入载荷 (QF = 80)		嵌入载荷 (QF = 90)	
		800	4 000	800	6 000	800	8 000	800	14 000	800	16 000	800	18 000
Baboon	Hu 等人(2013)	0	-	0	-	0	-	0	-	-	-	-	-
	Qiu 等人(2018)	0	-	0	-	0	-	-	-	0	-	0	-
	Du 等人(2022)	-8 392	-6 416	-6 088	-952	-4 008	4 384	-944	21 648	240	21 488	-7 520	20 280
	本文	-8 456	-6 656	-6 296	-184	-4 304	3 728	-208	16 456	984	19 416	-7 896	11 672
Boat	Hu 等人(2013)	-	-	-	-	-	-	-	-	0	-	0	-
	Qiu 等人(2018)	0	-	0	-	0	-	0	-	0	-	0	-
	Du 等人(2022)	-5 200	-1 608	-2 904	3 512	-1 744	7 704	208	20 784	-1 456	23 064	-5 056	27 136
	本文	-5 456	-2 024	-2 920	2 808	-1 968	7 682	240	17 488	-1 032	20 784	-5 124	23 104
Bridge	Hu 等人(2013)	0	-	-	-	0	-	0	-	-	-	-	-
	Qiu 等人(2018)	0	-	0	-	0	-	0	-	-	-	0	-
	Du 等人(2022)	-6 136	-4 736	-4 608	1 992	-2 584	5 544	-408	20 480	512	20 456	-4 440	18 216
	本文	-6 344	-4 672	-4 504	1 264	-2 720	5 240	-112	18 136	1 328	20 312	-3 992	16 448
Lena	Hu 等人(2013)	-	-	-	-	-	-	-	-	-	-	-	-
	Qiu 等人(2018)	0	-	0	-	0	-	-	-	-	-	0	-
	Du 等人(2022)	-5 488	-1 816	-3 656	2 888	-2 216	6 192	888	18 688	1 264	24 496	464	32 767
	本文	-5 592	-1 943	-3 816	2 712	-2 336	7 120	464	17 568	1 216	20 808	1 240	25 472
Peppers	Hu 等人(2013)	-	-	-	-	-	-	-	-	-	-	0	-
	Qiu 等人(2018)	0	-	0	-	0	-	-	-	-	-	0	-
	Du 等人(2022)	-5 360	-1 472	-3 088	2 560	-1 976	6 256	904	17 984	480	23 096	-8 304	30 048
	本文	-5 384	-1 376	-3 448	2 320	-2 232	6 234	248	16 560	176	18 496	-8 510	18 136
Splash	Hu 等人(2013)	-	-	0	-	0	-	0	-	0	-	-	-
	Qiu 等人(2018)	0	-	0	-	0	-	0	-	0	-	0	-
	Du 等人(2022)	-7 752	-4 816	-7 864	-1 376	-7 136	2 936	-3 656	12 568	-3 216	19 904	576	24 504
	本文	-7 968	-5 120	-8 000	-872	-7 424	2 464	-3 832	10 824	-2 904	15 472	824	21 464

注:“-”表示嵌入载荷大于方法所能提供的嵌入容量,加粗字体为各图像对应的最优结果。

结果,Du 等人(2022)方法和本文方法的文件膨胀为负数,相较于限长 VLC 映射方法存在较大优势。

在嵌入载荷较大时,Hu 等人(2013)方法和 Qiu 等人(2018)方法并不能完成秘密信息的嵌入。从实验结果可以看出,对于 QF 较小的载体图像,Du 等人(2022)方法和本文方法去除编码冗余的操作对控制文件膨胀有更大影响;而对于 QF 较大的载体图像,编码冗余并不能再抵消信息嵌入导致的文件膨胀,Du 等人(2022)方法和本文方法导致的文件膨胀也随之增大。根据实验结果,在嵌入载荷较大时,本文方法相对于 Du 等人(2022)方法更具优

势。如对于 QF = 90 的 6 幅载体图像,嵌入 1.8×10^4 bit 的秘密信息,本文方法导致的文件膨胀相比于 Du 等人(2022)方法最少降低 9%,最多降低 42%。两种方法在不同载体图像上的性能差距不同是因为不同载体图像的 VLC 频数的统计结果相差较大。为了更好地比较这两种方法控制文件膨胀的性能,使用两种方法对 USC-SIPI 数据库中的 210 幅图像进行测试并统计文件膨胀。首先,210 幅原始数据格式为 tiff 的测试图像,经过与前文一样的处理转为 $512 \times 512 \times 8$ bit 的 JPEG 压缩图像,QF 取 30、50、70、90。然后,使用 Du 等人(2022)方法和本

文方法对 4 种 QF 的载体图像嵌入秘密信息, 计算文件膨胀的平均值, 结果如图 9 所示。实验结果表明, 本文方法导致的平均文件膨胀略低于 Du 等人 (2022) 方法。图 9(d) 中本文方法的优势体现得更加明显, 如在嵌入载荷为 1.8×10^4 bit 时, 本文方法导致的平均文件膨胀相较于 Du 等人 (2022) 方法缩

小了接近 5×10^3 bit。进一步验证了本文的最优中间 VLC 映射模型的有效性, 可以稳定地满足较大嵌入载荷的需求, 有效利用 VLC 频数的分布特性, 使信息嵌入导致的文件膨胀被有效控制。JPEG 图像的 RDH 方法对文件膨胀和嵌入容量的要求在最优中间 VLC 映射模型下同时得到了满足。

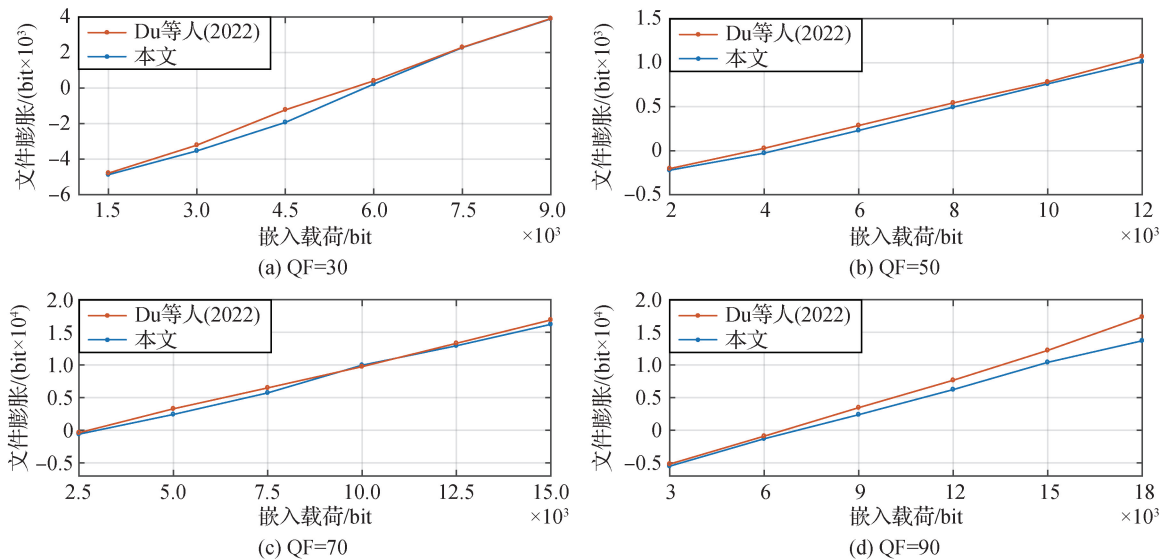


图 9 本文方法与 Du 等人 (2022) 方法在不同质量因子下嵌入不同长度秘密信息时文件膨胀大小的对比

Fig. 9 Comparison of file size increment between Du et al. (2020) and ours with different quality factors and embedded with different lengths of secret data((a) QF = 30; (b) QF = 50; (c) QF = 70; (d) QF = 90)

3.2 基于量化 DCT 系数修改的方法对比

为进一步验证本文方法的性能, 与基于量化 DCT 系数修改的方法进行对比。首先, 分析两种方法对视觉质量的影响。基于量化 DCT 系数修改的方法在嵌入过程中, DCT 系数会被修改, 势必造成视觉质量的降低, 如 Huang 等人 (2016) 和 Hou 等人 (2018) 两种先进的基于量化 DCT 系数修改的方法得到的载密图像的峰值信噪比 (peak signal to noise ratio, PSNR) 会随着嵌入载荷的升高而降低。而本文方法基于 VLC 映射实现, 不会修改 DCT 系数, 因此不会降低载密图像的视觉质量, 与载体图像计算的 PSNR 所得结果为正无穷。然后, 对比本文方法与基于量化 DCT 系数修改的方法所导致的文件膨胀。对 QF = 30, 50, 70, 90 的载体 JPEG 图像, 分别使用 Huang 等人 (2016) 方法、Hou 等人 (2018) 方法和本文方法以 3 种嵌入载荷进行测试, 并统计文件膨胀。结果如表 3 所示。可以看出, 本文方法的文件膨胀总是最小的。但也有特殊的图像, 如 Splash

图像, 在 QF 较小时, Hou 等人 (2018) 方法和本文方法并不能提供足够的嵌入载荷。这种情况是因为个别图像的 VLC 频数统计结果并不够集中, 因此本文方法不能满足过大的嵌入需求, 但根据实验结果, 本文方法对大部分图像能提供足够的嵌入容量。

本文方法的文件膨胀包括信息嵌入导致的文件膨胀和 RSV 重排序操作缩减的编码冗余, 因此实际文件膨胀要小于信息嵌入导致的文件膨胀, 这一点在 3.1 节的实验结果中也有体现。为了更好地与基于量化 DCT 系数修改的方法对比, 将信息嵌入导致的文件膨胀记为 EFI (embedding-file size increment), 将实际文件膨胀记为 PFI (pure file size increment), 则有 PFI 必小于 EFI。根据实验结果, 在 QF 越大, 嵌入载荷越大时, EFI 和 PFI 的差距越不明显, 即优化编码冗余对控制文件膨胀所能起到的作用越不明显。这一点也与理论分析吻合, 在嵌入载荷较大时, EFI 会明显增大, 而编码冗余只与图像本身有关, 并不会随着嵌入载荷的增大而变化。

表3 本文方法与基于量化 DCT 系数修改的方法在不同质量因子下嵌入不同长度秘密信息时文件膨胀大小的对比
Table 3 Comparison of file size increment between previous DCT-based methods and ours with different quality factors and embedded with different lengths of secret data

图像	方法	嵌入载荷 (QF = 30)			嵌入载荷 (QF = 50)			嵌入载荷 (QF = 70)			嵌入载荷 (QF = 90)		
		3 000	6 000	9 000	4 000	8 000	12 000	5 000	10 000	15 000	6 000	12 000	18 000
Baboon	Huang 等人(2016)	4 232	8 393	12 876	5 616	11 040	16 920	7 288	14 704	22 496	10 160	20 376	30 776
	Hou 等人(2018)	3 872	7 880	12 480	<u>5 336</u>	10 480	<u>16 536</u>	<u>7 128</u>	14 440	22 384	<u>9 672</u>	19 960	30 496
	本文(EFI)	<u>3 768</u>	<u>7 856</u>	<u>11 848</u>	5 704	<u>9 496</u>	16 640	7 656	<u>13 144</u>	<u>19 960</u>	9 920	<u>15 464</u>	<u>21 192</u>
	本文(PFI)	-4 264	-384	3 808	408	4 344	11 336	5 312	10 496	17 480	304	6 032	11 488
Boat	Huang 等人(2016)	4 344	8 328	12 376	5 944	11 808	17 544	7 496	15 240	22 824	9 624	20 160	30 640
	Hou 等人(2018)	3 872	7 664	11 968	5 688	<u>10 600</u>	16 976	7 328	<u>14 528</u>	22 216	<u>8 976</u>	<u>19 896</u>	30 672
	本文(EFI)	<u>3 168</u>	<u>7 464</u>	<u>11 360</u>	<u>4 480</u>	11 784	<u>14 384</u>	<u>7 144</u>	15 368	<u>20 352</u>	11 824	20 240	30 824
	本文(PFI)	-1 504	2 832	6 880	2 560	9 600	12 408	5 616	13 440	18456	4 120	12 888	23 192
Bridge	Huang 等人(2016)	4 704	9 072	13 448	6 360	12 016	17 720	8 120	15 704	23 856	10 520	21 152	31 584
	Hou 等人(2018)	4 312	8 616	<u>12 816</u>	5 816	11 456	16 456	7 152	15 016	23 160	<u>9 456</u>	20 704	31 008
	本文(EFI)	<u>3 576</u>	<u>7 472</u>	13 168	<u>5 408</u>	<u>8 776</u>	<u>15 848</u>	<u>7 064</u>	<u>12 728</u>	<u>21 528</u>	9 464	<u>14 928</u>	<u>22 960</u>
	本文(PFI)	-2 608	1 160	6 976	1 712	5 200	12 400	4 760	10 536	19 152	2 984	8 424	16 320
Lena	Huang 等人(2016)	3 680	8 160	12 520	5 360	11 056	17 376	7 152	14 296	21 896	9 016	17 216	25 384
	Hou 等人(2018)	3 840	8 040	12 584	5 480	10 968	16 984	7 352	13 896	21 712	<u>7 728</u>	16 424	<u>25 272</u>
	本文(EFI)	<u>3 352</u>	<u>7 736</u>	<u>12 040</u>	<u>4 840</u>	<u>10 720</u>	<u>12 872</u>	<u>6 416</u>	<u>13 536</u>	<u>19 144</u>	8776	18 056	26 992
	本文(PFI)	-1 736	2 536	7 000	2 088	8 160	10 120	5 416	12 488	18 152	7 264	16 680	25 712
Peppers	Huang 等人(2016)	3 680	8 096	12 624	5 544	11 192	17 528	7 592	14 352	21 608	8 992	17 576	26 944
	Hou 等人(2018)	4 272	<u>7 632</u>	12 528	5 784	10 944	17 096	6 992	14 136	20 688	<u>7 808</u>	<u>17 168</u>	<u>26 544</u>
	本文(EFI)	<u>3 232</u>	7 768	<u>11 936</u>	<u>4 984</u>	<u>10 904</u>	<u>13 096</u>	<u>5 816</u>	<u>13 752</u>	<u>19 120</u>	9 704	19 672	28 032
	本文(PFI)	-1 740	2 350	6 984	2 456	8 352	10 288	4 792	12 680	18 264	24	9 920	18 168
Splash	Huang 等人(2016)	4 328	9 656	10 584	5 824	11 664	16 904	6 936	13 608	19 888	9 104	17 448	25 986
	Hou 等人(2018)	<u>3 944</u>	8 432	-	5 880	11 160	-	7 080	13 064	19 192	8 272	<u>16 560</u>	25 160
	本文(EFI)	4 208	<u>8 376</u>	-	<u>5 880</u>	<u>10 416</u>	-	<u>6 272</u>	<u>12 056</u>	-	<u>7 344</u>	18 552	<u>22 552</u>
	本文(PFI)	-4 992	-976	-	-2 560	2 096	-	944	6 576	-	6 064	17 304	21 504

注:“-”表示嵌入载荷大于方法所能提供的嵌入容量,加粗字体为各图像对应的每列 PFI 最优值,下划线为各图像对应的每列 EFI 最优值。

表 3 中以下划线标识出了 PFI 与 Huang 等人(2016)方法和 Hou 等人(2018)方法的文件膨胀对比结果。可以看到,即使去除了 RSV 重排序操作的影响,本文的信息嵌入方法导致的文件膨胀仍小于 Huang 等人(2016)方法和 Hou 等人(2018)方法。说明本文提出的最优中间 VLC 映射可以很好地控制由于信息嵌入导致的文件膨胀。与基于量化 DCT 系数修改的方法相比,不仅保有基于 VLC 映射方法的优势,即无损嵌入、不需编解码等,并且能很好地平衡嵌入容量和文件膨胀。

4 结 论

本文提出了新的基于全体 VLC 映射的 JPEG 图像的 RDH 方法,称为最优中间 VLC 映射模型,能够提供较大嵌入容量并有效控制文件膨胀。由于现有 VLC 映射模型不能同时满足对嵌入容量和文件膨胀的需求,本文提出了中间 VLC 映射模型。相对于直接 VLC 映射,中间 VLC 映射既可以避免嵌入容量不足问题,又可以有效减小文件膨胀。同时,为了

进一步减小文件膨胀,利用 RSV 重排序后 VLC 的频数分布,构建了最优中间 VLC 映射模型。在该模型下,直接 VLC 映射被分解为多个中间 VLC 映射,从而最小化 VLC 替换造成的文件膨胀。而对于给定的载体图像和嵌入载荷,可以通过模拟嵌入确定构建最优 VLC 映射所需的嵌入 VLC 和中间 VLC 集合。

实验结果表明,与基于量化 DCT 修改的方法相比,本文方法在保证无损嵌入的同时,能够提供较大的嵌入容量,并且使用最优中间 VLC 映射可以有效减少文件膨胀。本文方法对使用非标准的哈夫曼表的压缩图像,接收端并不能构建信息嵌入时所用的 VLC 映射。因此在未来的研究中,需要设计新的算法构建更通用的 VLC 映射模型。

参考文献 (References)

- Celik M U, Sharma G, Tekalp A M and Saber E. 2005. Lossless generalized-LSB data embedding. *IEEE Transactions on Image Processing*, 14(2): 253-266 [DOI: 10.1109/TIP.2004.840686]
- Chen L S T, Lin S J and Lin J C. 2010. Reversible JPEG-based hiding method with high hiding-ratio. *International Journal of Pattern Recognition and Artificial Intelligence*, 24(3): 433-456 [DOI: 10.1142/S0218001410007968]
- Dragoi I C and Coltuc D. 2014. Local-prediction-based difference expansion reversible watermarking. *IEEE Transactions on Image Processing*, 23(4): 1779-1790 [DOI: 10.1109/TIP.2014.2307482]
- Du Y, Yin Z X and Zhang X P. 2022. High capacity lossless data hiding in JPEG bitstream based on general VLC mapping. *IEEE Transactions on Dependable and Secure Computing*, 19(2): 1420-1433 [DOI: 10.1109/TDSC.2020.3013326]
- Fridrich J, Goljan M and Du R. 2002. Lossless data embedding—new paradigm in digital watermarking. *EURASIP Journal on Advances in Signal Processing*, 2002(2): # 986842 [DOI: 10.1155/S1110865702000537]
- Hou D D, Wang H Q, Zhang W M and Yu N H. 2018. Reversible data hiding in JPEG image based on DCT frequency and block selection. *Signal Processing*, 148: 41-47 [DOI: 10.1016/j.sigpro.2018.02.002]
- Hou D D, Zhang W M, Liu J Y, Zhou S Y, Chen D D and Yu N H. 2019. Emerging applications of reversible data hiding//*Proceedings of the 2nd International Conference on Image and Graphics Processing*. Singapore, Singapore: ACM: 105-109 [DOI: 10.1145/3313950.3313952]
- Hu Y J, Wang K and Lu Z M. 2013. An improved VLC-based lossless data hiding scheme for JPEG images. *Journal of Systems and Software*, 86(8): 2166-2173 [DOI: 10.1016/j.jss.2013.03.102]
- Huang F J, Qu X C, Kim H J and Huang J W. 2016. Reversible data hiding in JPEG images. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(9): 1610-1621 [DOI: 10.1109/TCSVT.2015.2473235]
- Lu W, Zhang J H, Zhao X F, Zhang W M and Huang J W. 2021. Secure robust JPEG steganography based on AutoEncoder with adaptive BCH encoding. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(7): 2909-2922 [DOI: 10.1109/TCSVT.2020.3027843]
- Mobasser B G, Berger R J, Marcinak M P and NaikRaikar Y J. 2010. Data embedding in JPEG bitstream by code mapping. *IEEE Transactions on Image Processing*, 19(4): 958-966 [DOI: 10.1109/TIP.2009.2035227]
- Ni Z C, Shi Y Q, Ansari N and Su W. 2006. Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(3): 354-362 [DOI: 10.1109/TCSVT.2006.869964]
- Ou B, Yin Z X and Xiang S J. 2022. Overview of reversible data hiding in plaintext image. *Journal of Image and Graphics*, 27(1): 111-124 (欧博, 殷赵霞, 项世军. 2022. 明文图像可逆信息隐藏综述. *中国图象图形学报*, 27(1): 111-124) [DOI: 10.11834/jig.210384]
- Qian Z X and Zhang X P. 2012. Lossless data hiding in JPEG bitstream. *Journal of Systems and Software*, 85(2): 309-313 [DOI: 10.1016/j.jss.2011.08.015]
- Qiu Y Q, He H, Qian Z X, Li S and Zhang X P. 2018. Lossless data hiding in JPEG bitstream using alternative embedding. *Journal of Visual Communication and Image Representation*, 52: 86-91 [DOI: 10.1016/j.jvcir.2018.02.005]
- Sakai H, Kuribayashi M and Morii M. 2008 Adaptive reversible data hiding for JPEG images//*Proceedings of 2008 International Symposium on Information Theory and Its Applications*. Auckland, New Zealand; IEEE: 7-10. [DOI: 10.1109/ISITA.2008.4895529]
- Tian J. 2003. Reversible data embedding using a difference expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(8): 890-896 [DOI: 10.1109/TCSVT.2003.815962]
- Wang K, Lu Z M and Hu Y J. 2013. A high capacity lossless data hiding scheme for JPEG images. *Journal of Systems and Software*, 86(7): 1965-1975 [DOI: 10.1016/j.jss.2013.03.083]
- Weber A G. 1997. The USC-SIPI image database: version 5 [EB/OL]. [2022-04-12]. <http://sipi.usc.edu/database/>
- Xuan G R, Shi Y Q, Ni Z C, Chai P Q, Cui X and Tong X F. 2007. Reversible data hiding for JPEG images based on histogram pairs//*Proceedings of the 4th International Conference Image Analysis and Recognition*. Montreal, Canada: Springer: 715-727 [DOI: 10.1007/978-3-540-74260-9_64]
- Yin X L, Lu W, Liu W T, Guo J M, Huang J W and Shi Y Q. 2021. Reversible data hiding in halftone images based on dynamic embedding states group. *IEEE Transactions on Circuits and Systems for*

Video Technology, 31(7): 2631-2645 [DOI: 10.1109/TCSVT.2020.3032685]

Yin X L, Lu W, Zhang J H and Luo X Y. 2022. Robust JPEG steganography based on lossless carrier and robust cost. Journal of Image and Graphics, 27(1): 238-251 (尹晓琳, 卢伟, 张俊鸿, 罗向阳. 2022. 无损载体和鲁棒代价结合的 JPEG 图像鲁棒隐写. 中国图象图形学报, 27(1): 238-251) [DOI: 10.11834/jig.210406]

Zhang W M, Hu X C, Li X L and Yu N H. 2013. Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression. IEEE Transactions on Image Processing, 22(7): 2775-2785 [DOI: 10.1109/TIP.2013.2257814]

作者简介

王可,女,硕士研究生,主要研究方向为多媒体安全和信息隐

藏。E-mail:wangk237@mail2.sysu.edu.cn

卢伟,通信作者,男,教授,博士生导师,主要研究方向为多媒体信息安全、数字取证、信息隐藏、隐私保护和多媒体大数据智能。E-mail:luwei3@mail.sysu.edu.cn

吴绍武,男,博士研究生,主要研究方向为多媒体安全和信息隐藏。E-mail:wushw25@mail2.sysu.edu.cn

尹晓琳,女,博士研究生,主要研究方向为多媒体安全和信息隐藏。E-mail:yinxl6@mail2.sysu.edu.cn

付婧巧,女,硕士研究生,主要研究方向为自然语言处理。

E-mail:fujq3@mail.sysu.edu.cn

陈兵,男,博士,讲师,主要研究方向为多媒体信息安全和信息隐藏。E-mail:chenbing@gpnu.edu.cn