

文章编号:1004-9037(2013)02-0155-05

一种基于排序的 JPEG 码流无损水印算法

陈立泽 倪蓉蓉 赵 耀

(北京交通大学信息科学研究所,北京,100044)

摘要:提出了一种基于排序的直接在 JPEG 码流中嵌入水印的算法,该方法在保证嵌完水印后图像无失真的条件下提高了嵌入容量。根据对 JPEG 图像中可变长度码使用情况的统计分析,通过将一幅图像中没有使用过的可变长度码映射为使用过的可变长度码来嵌入水印。在建立映射关系之前,首先对码流中出现过的可变长度码依据其在码流中出现的频率进行排序,然后使用排序后的结果建立映射关系,从而提高可变长度码的利用率,进而提高嵌入容量。在水印嵌入过程中,根据相应的映射关系和水印信息将码流中出现的可变长度码替换为映射的可变长度码。分析和实验结果表明,这种方法能够实现较高容量的水印嵌入,并且嵌完水印后图像无失真。

关键词:数字水印;JPEG 码流;可变长度码;码字映射;排序

中图分类号:TP391

文献标志码:A

Embedded Lossless Watermark in JPEG Bitstream Based on Sorting

Chen Lize, Ni Rongrong, Zhao Yao

(Institute of Information Science, Beijing Jiaotong University, Beijing, 100044, China)

Abstract: A modified scheme of embedding watermark in JPEG bitstream based on sorting is proposed, which ensures that the embedding capacity is improved while the cover image is lossless after embedding the watermark. According to the statistical results of variable length codeing (VLC) usage in an image, watermark is embedded by mapping one unused VLC to one used VLC. Sorting VLC(s) according to their appearing rate in the bitstream before mapping, will improve the VLC utilization rate and increase the embedding capacity after sorting. During the data embedding, the codes appearing in the bitstream are replaced by the mapped codes according to the watermark. The results prove that the proposed scheme has high embedding capacity and can preserve the image with no quality distortion.

Key words: digital watermark; JPEG bitstream; variable length code (VLC); code mapping; sorting

引 言

随着计算机网络和数字媒体的快速发展,数字产品的传播变得广泛。由于数字产品能够被轻易地复制和修改,因此为了保护数字产品的产权,出现了数字水印技术^[1]。在某些特殊的应用领域,如医疗诊断、军事应用和法律诉讼等,对图像的完整性和真实性要求比较高,甚至不允许由于嵌入水印而带来失真,这就要求无损嵌入水印,于是出现了

可逆数字水印。可逆数字水印是指在水印提取后,原始载体能够被无失真地恢复。Tian^[2]提出了扩差法,对每两个相邻像素的差值进行扩展后,在空出的最低位嵌入 1 bit 信息。Alattar^[3]扩展了 Tian 的方法,对 3 个或 4 个相邻像素的差值进行扩展来嵌入多于 1 bit 的水印信息。Ni 等人^[4]提出了一种直方图平移算法,Thodi 等人^[5]提出了一种预测误差扩展算法。

近来在压缩域出现了一些直接在压缩后的数据码流中无损嵌入水印的方法。Liu 等人^[6]提出了一

基金项目:国家杰出青年科学基金(61025013)资助项目;国家自然科学基金(61073159)资助项目;中央高校基本科研业务费专项资金(2012JBM042)资助项目。

收稿日期:2012-05-23;修订日期:2012-06-06

种在 MPEG 码流中嵌入水印的算法,主要是通过修改变长编码(Variable length coding, VLC)的最低有效位(Least significant bit, LSB)位来嵌入水印。Fridrich^[7]提出一种通过修改离散余弦变换(Discrete cosine transform, DCT)系数来嵌入水印的算法。Mobasseri 等人^[8]提出一种通过建立码字映射关系在 JPEG 码流中嵌入数据的算法。之后 Zhang 等人^[9]改进了 Mobasseri 的算法,在压缩后的文件码流中嵌入水印只修改了码流的结构,不影响解码的速度,并且可以保证在嵌入水印后图像的视觉质量不受影响。然而在码流中嵌入水印的缺点就是嵌入的容量有限,因为在码流中有很少的冗余存在。

基于文献[8,9]的码字映射算法,本文提出了一种基于排序的在 JPEG 码流中嵌入无损水印的码字映射算法。该算法根据 VLC 在文件中出现的频率大小不同,对使用过的 VLC 进行排序,优先选择使用出现频率高的 VLC,这既保证嵌入水印前后图像无损和文件大小不变,又提高了嵌入容量。

1 相关理论与算法

1.1 JPEG 结构分析

在 JPEG 标准中,国际电报电话咨询委员会(International telephone and telegraph consultative committee, CCITT)一共为图像亮度分量的 AC 系数定义了 162 种可变长度码,可变长度码即是 run/size 的霍夫曼码,其中 run 表示两个相邻非零系数值之间的 0 的个数, size 表示此交流系数的实际编码值的位数。162 种 VLC 码字的长度从 2~16 不等,在 JPEG 标准中的表 K. 5 给出了 VLC 的对应的码表,在 JPEG 文件的头部一般会存有此码表,供解码时使用。在数据的压缩过程中,每个系数值最后编成的码字形式为 {VLC, 实际码值},例如,某个 AC 系数的值为 -4,它的前面有 3 个连续的 0 值,那么它的 run/size 为 0/3,此系数的最后实际编码值即为 {100 011}。

对于 162 种可变长度码来说,只有一些是经常用到的,而且在某一幅图像中出现的可变长度码的种类也不会太多。图 1 所示为几幅 512×512 大小的图像用到的可变长度码的情况。从图 1 可以看出,一幅图像用到的 VLC 种类一般不超过 70,而一共存在 162 种 VLC 码字,存在极大的冗余,在一幅图像中就可以利用未使用过的 VLC 来嵌入水印。

1.2 相关算法

Mobasseri 等人^[8]提出了一种码字映射算法,该算法是直接 在 JPEG 码流中嵌入水印。通过使用过的 VLC 与未使用过的 VLC 建立映射关系

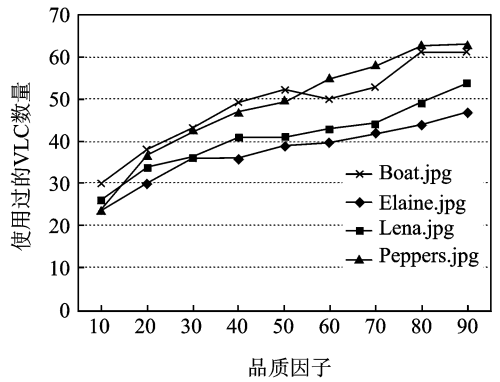


图 1 VLC 使用情况举例

来嵌入水印,该算法不是无损的,并且嵌入的容量并不高。

基于 Mobasseri 的方法,文献[9]提出了一种 VLC 码字映射替换方法。算法主要步骤如下:在嵌入水印前,首先从 JPEG 码流中解析出其用过的 VLC,然后把 162 种 VLC 分成使用过的和未使用过的两类, {使用过,未使用过}。然后对于一个使用过的 VLC 挑选一个码字相同长度但未使用过的 VLC 建立映射关系,此映射关系的实现是通过修改 JPEG 文件 VLC 码表中的相应 VLC 码字的值来实现的。然后根据映射关系和水印信息替换掉码流中出现过的 VLC 来嵌入水印。

文献[9]的算法是一种无损数据嵌入方案,并且嵌完数据后文件大小保持不变。但是该算法在建立映射关系时没有考虑到使用过的 VLC 在文件中出现的频率是不同的。如果在建立映射关系时优先使用出现频率高的 VLC,或者是将出现频率高的 VLC 建立一对多的映射关系,都有可能提高嵌入容量。

2 基于排序的码字映射算法

在嵌入水印前,首先解析 JPEG 文件码流,解析出码流中所有出现过的 VLC^[9]。用 L_i 表示 VLC 码字长度均为 i 的码字的数量,用 C_i 表示码字长度均为 i 的码字的集合,那么 162 种 VLC 码字可分成 16 组 $\{C_1, C_2, \dots, C_{16}\}$ 。对某一长度的码字的集合来说,又可以分成使用过的和未使用过的两类:

$$C_i = \{VLC_{u_i}^1, \dots, VLC_{u_i}^p; VLC_{n_i}^1, \dots, VLC_{n_i}^q\} \quad (1)$$

式中: $VLC_{u_i}^1, \dots, VLC_{u_i}^p$ 表示使用过的 VLC 码字, $VLC_{n_i}^1, \dots, VLC_{n_i}^q$ 表示未使用过的 VLC 码字,并且满足 $p+q=L_i$ ^[9]。

(1) 如果 $p \geq q$ 并且 $q > 0$, 这种情况下,使用过

的 VLC 码字就会有剩余,为了提高嵌入率,映射方法如下:

首先对使用过的 VLC,根据它们在码流中出现的次数按降序排序,假设排序的结果为 $VLC_{u_i}^1, \dots, VLC_{u_i}^p$,其在码流中出现的次数依次为 N_1, \dots, N_p ,然后按顺序与未使用过的 VLC 码建立映射关系

$$\{\{VLC_{u_i}^1 \leftrightarrow VLC_{n_i}^1\}, \dots, \{VLC_{u_i}^q \leftrightarrow VLC_{n_i}^q\}\}$$

经过排序后再建立映射关系就比文献[9]的方法中无序映射嵌入率要高。因为选用的都是出现频率高的 VLC 来建立映射关系,这样就提高了 VLC 的利用率,进而提高了嵌入容量。经过这样映射后,对于 C_i 这组集合来说,因为是一一映射关系,所以其总嵌入容量为

$$sum_1 = N_1 + N_2 + \dots + N_q \quad (2)$$

如果建立一对多的映射关系,嵌入容量可能会更大,假设 $VLC_{u_i}^i$ 分别与 k_i 个未使用过的 VLC 建立映射,那么此时映射关系为

$$\begin{aligned} &\{VLC_{u_i}^1 \leftrightarrow \{VLC_{n_i}^1, \dots, VLC_{n_i}^{k_1}\}\}, \\ &\{VLC_{u_i}^2 \leftrightarrow \{VLC_{n_i}^{q-k_1}, \dots, VLC_{n_i}^{q-k_1+k_2}\}\}, \dots \\ &\{VLC_{u_i}^i \leftrightarrow \{VLC_{n_i}^{q-\sum_{i=1}^{i-1} k_i}, \dots, VLC_{n_i}^{q-\sum_{i=1}^{i-1} k_i+k_i}\}\} \\ &k_i = 2^{\lfloor \log_2 (q - \sum_{j=1}^{i-1} k_j + 1) \rfloor} - 1 \end{aligned}$$

此时是一对多映射,其总嵌入容量为

$$sum_2 = N_1 \times \log_2(k_1 + 1) + N_2 \times \log_2(k_2 + 1) + \dots + N_i \times \log_2(k_i + 1) \quad (3)$$

如果 $sum_1 \geq sum_2$,那么就建立上述一一映射的关系,否则,就建立上述一对多的映射关系,这样能够尽量提高嵌入容量。

(2)如果 $p < q$ 并且 $p > 0$,这种情况下,未使用过的 VLC 就可能会有剩余,为了提高嵌入率,映射方法如下:

首先对使用过的 VLC,根据它们在码流中出现的次数按降序排序,假设排序的结果为 $VLC_{u_i}^1, \dots, VLC_{u_i}^p$,其在码流中出现的次数依次为 $N_1, \dots,$

N_p ,然后按顺序与未使用过的 VLC 码建立映射关系^[9]

$$\begin{aligned} &\{VLC_{u_i}^1 \leftrightarrow \{VLC_{n_i}^1, \dots, VLC_{n_i}^k\}\}, \dots, \\ &\{VLC_{u_i}^p \leftrightarrow \{VLC_{n_i}^{(p-1) \times k + 1}, \dots, VLC_{n_i}^{p \times k}\}\} \end{aligned}$$

式中, $k = 2^{\lfloor \log_2 (q/p + 1) \rfloor} - 1$,这样总的嵌入容量为

$$sum_1 = (N_1 + N_2 + \dots + N_q) \times \log_2(k + 1) \quad (4)$$

在上述这种映射关系中,每个使用过的 VLC 都分别与 k 个未使用过的 VLC 建立了映射关系。如果假设 $VLC_{u_i}^i$ 分别与 k_i 个未使用过的 VLC 建立映射,那么此时映射关系为

$$\begin{aligned} &\{VLC_{u_i}^1 \leftrightarrow \{VLC_{n_i}^1, \dots, VLC_{n_i}^{k_1}\}\} \\ &\{VLC_{u_i}^2 \leftrightarrow \{VLC_{n_i}^{q-k_1}, \dots, VLC_{n_i}^{q-k_1+k_2}\}\}, \dots, \\ &\{VLC_{u_i}^i \leftrightarrow \{VLC_{n_i}^{q-\sum_{i=1}^{i-1} k_i}, \dots, VLC_{n_i}^{q-\sum_{i=1}^{i-1} k_i+k_i}\}\} \end{aligned}$$

其中: $k_i = 2^{\lfloor \log_2 (q - \sum_{j=1}^{i-1} k_j + 1) \rfloor} - 1$,此时总的嵌入容量为

$$sum_2 = N_1 \times \log_2(k_1 + 1) + N_2 \times \log_2(k_2 + 1) + \dots + N_i \times \log_2(k_i + 1) \quad (5)$$

如果 $sum_1 \geq sum_2$,采用第 1 种映射方法,否则采用第 2 种映射方法,这样能尽量提高嵌入容量。

映射关系建立完成后,应该在文件的头部信息中的 VLC 码表中将建立的映射关系一一标示出来,如图 2 所示,映射关系建立完成后可以进行水印的嵌入^[9]。

如图 2 所示,2/5 与 6/3,6/4,6/5 分别建立了映射关系,这是一种一对三的映射关系。那么在文件的码流中当碰见 2/5 的码字(111111110001001)时,如果要嵌入的是“00”,则保持 2/5 的码字不变。如果要嵌入的是“01”,将 2/5 的码字替换为 6/3 的码字(111111110100110),这样就实现了“01”的嵌入。如果要嵌入的“10”或“11”,则将 2/5 的码字替换为 6/4 或 6/5 对应的码字即可。为了保证嵌入水印后的图像文件大小保持不变,映射关系只在 VLC 码字长度相同的码字之间建立映射关系,如 2/5 和

05	06	07	08	09	0A	0B	FF	C4	00	B5	10	00	02	01	03	05	06	07	08	09	0A	0B	FF	C4	00	B5	10	00	02	01	03
03	02	04	03	05	05	04	04	00	00	01	7D	01	02	03	00	03	02	04	03	05	05	04	04	00	00	01	7D	01	02	03	00
04	11	05	12	21	31	41	06	13	51	61	07	22	71	14	32	04	11	05	12	21	31	41	06	13	51	61	07	22	71	14	32
81	91	A1	08	23	42	B1	C1	15	52	D1	F0	24	33	62	72	81	91	A1	08	23	42	B1	C1	15	52	D1	F0	24	33	62	72
82	09	0A	16	17	18	19	1A	25	26	27	28	29	2A	34	35	82	09	0A	16	17	18	19	1A	25	26	27	28	29	2A	34	35
36	37	38	39	3A	43	44	45	46	47	48	49	4A	53	54	55	36	37	38	39	3A	43	44	45	46	47	48	49	4A	53	54	55
56	57	58	59	5A	63	65	65	66	67	68	69	6A	73	74	75	56	57	58	59	5A	63	65	65	66	67	68	69	6A	73	74	75
76	77	78	79	7A	83	84	85	86	87	88	89	8A	92	93	94	76	77	78	79	7A	83	84	85	86	87	88	89	8A	92	93	94

图 2 VLC 映射关系举例

6/3,6/4,6/5 的码字长度都为 16,这样嵌完水印后文件的大小将会保持不变^[9]。

在水印提取端,首先需要读取 JPEG 文件的 AC 码表,然后与标准的霍夫曼码表对照构造出建立的映射关系,根据相应的映射关系提取水印。比如上面的 2/5 与 6/3,6/4,6/5 分别建立了映射关系,那么在读到 2/5 的码字(111111110001001)时提取出的水印信息就是“00”,读到 6/3 的码字(111111110100110)时提取出的水印信息就是“01”。

3 水印的嵌入和提取

水印的嵌入过程可以分成以下几个步骤:

- (1) 从 JPEG 文件码流中解析出使用过的和未使用过的 VLC;
- (2) 按照第 2 节提出的映射算法建立相应的映射关系;
- (3) 将建立的映射关系在文件的头部 AC 码表中标注;
- (4) 根据待嵌入的水印和映射关系,将码流中出现的相应码字依次做替换完成水印的嵌入。

因为水印嵌入后只修改了码流中的 VLC 码字,不影响实际 AC 系数的编码值,并且由于将映

射关系在码表中做了标示,解析码流时将按码表中新的码字对应关系解析,所以不影响图像的质量,实现了真正的无损信息嵌入^[9]。

水印的提取过程分为两步:

- (1) 从文件中读取 AC 码表的信息,建立起修改过的码表并和标准的霍夫曼码表做对比,建立起映射关系;
- (2) 根据得到的映射关系,依次解析码流,提取水印。

由于 JPEG 压缩算法和标准都是公开的,所以在 JPEG 码流中嵌入了水印,安全性不高。所以,在水印的嵌入过程中,根据需要可以进行加密,加密分两部分,一是对原始水印信息的加密,二是对文件的头部附加信息的加密,具体加密算法在这里不作详述。

4 实验分析

为了测试本文提出的算法,使用了和文献[8,9]相同的图像,都来自 USC-SIPI 数据库。这些图像大小为 512×512,用 MATLAB 软件根据不同品质因子压缩成相应的 JPEG 文件。表 1 列出了不同 JPEG 文件不同品质因子下的嵌入容量,从实

表 1 不同品质因子的 JPEG 图像的嵌入容量比较

图像	算法/品质因子	10	20	30	40	50	60	70	80	90
Lena	文献[8]	263	197	235	259	317	94	99	86	152
	文献[9]	552	310	480	261	359	152	248	320	416
	本文算法	586	378	590	354	401	212	298	334	449
Peppers	文献[8]	350	247	338	242	278	357	164	144	378
	文献[9]	522	298	485	351	398	465	372	333	634
	本文算法	587	376	568	465	431	497	511	412	704
Splash	文献[8]	284	369	261	296	378	547	383	359	296
	文献[9]	545	316	765	842	893	1 058	702	691	605
	本文算法	587	396	824	896	954	1 125	831	703	637
Tiffany	文献[8]	226	348	222	310	332	391	534	332	279
	文献[9]	836	820	507	638	468	430	625	629	339
	本文算法	945	936	625	825	512	547	674	734	452
Baboon	文献[8]	792	2 035	1 071	1 342	820	887	953	317	195
	文献[9]	2 398	1 395	104 7	1 100	630	1 081	1 178	306	618
	本文算法	2 475	1 468	1 108	1 185	784	1 149	1 345	501	719
Boat	文献[8]	476	350	457	355	439	514	231	270	528
	文献[9]	975	596	394	541	587	645	513	792	942
	本文算法	1 097	687	512	635	678	843	575	854	1 087
Bridge	文献[8]	719	554	469	583	683	244	283	161	188
	文献[9]	1832	519	305	804	878	497	425	584	455
	本文算法	1 875	603	361	875	965	532	537	671	582
Elaine	文献[8]	111	130	139	200	249	79	92	146	339
	文献[9]	347	270	445	398	465	291	408	535	848
	本文算法	403	278	515	453	517	332	492	625	951

验结果可以看出,本文提出的算法的嵌入容量要高于文献[8,9]算法中的嵌入容量。图 3 给出的是不同品质因子下符合条件的配对的 VLC 组数。从图 3 中可以看出,相比文献[8,9]的算法,本文符合条件的配对组数并无多大提高。但本文中选用的都是经过排序的出现频率较高的 VLC 来建立映射的,这样提高了 VLC 的利用率,所以总的嵌入容量要高一些。

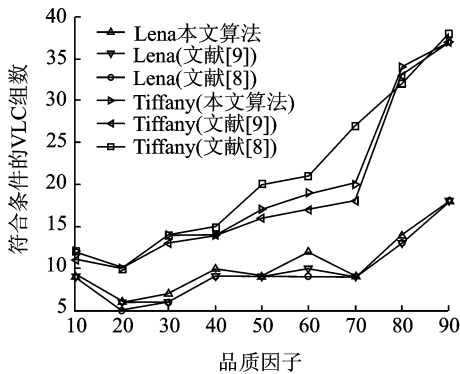


图 3 不同品质因子符合条件的 VLC 组数

5 结束语

本文提出了一种基于排序的码字映射算法,通过将使用过的 VLC 和未使用过的 VLC 建立映射关系直接在 JPEG 码流中嵌入水印。在建立映射关系前,首先对使用过的 VLC 按其在码流中出现的频率大小排序,经过排序选用的都是在码流中出现频率较高的 VLC 来建立映射关系,这样提高了 VLC 的利用率。同时还对出现频率较高的 VLC 采取了一对多的映射关系,也相对提高了 VLC 的利用率。从实验结果可以看出,本文提出的算法相比文献[8,9]中的算法具有更高的嵌入容量。

参考文献:

[1] 周四清,余英林. 数字图像水印技术及其应用[J]. 数据采集与处理, 2001,16(3):353-357.
Zhou Siqing, Yu Yinglin. Digital image watermark-

ing technique and application [J]. Journal of Data Acquisition & Processing, 2001, 16(3):353-357.

[2] Tian Jun. Reversible data embedding using a difference expansion [J]. IEEE Transactions on Circuits and Systems and Video Technology, 2003, 13(8): 890-896.

[3] Alattar A M. Reversible watermark using difference expansion of triplets [C]//2003 International Conference on Image Processing. Barcelona, Spain; IEEE Press, 2003:501-504.

[4] Ni Zhicheng, Shi Yunqing, Ansari N, et al. Reversible data hiding [J]. IEEE Transactions on Circuits and Systems and Video Technology, 2006, 16(3): 354-362.

[5] Thodi D M, Rodriguez J J. Expansion embedding techniques for reversible watermarking [J]. IEEE Transactions on Image Processing, 2007, 16(3):721-730.

[6] Liu Hongmei, Shao Fenglian, Huang Jiwu. A MPEG-2 video watermarking algorithm with compensation in bit stream [C]//2005 Challenges and Systems Lecture Notes in Computer Science, DRMTICS; IEEE Press, 2005:123-134.

[7] Fridrich J, Goljan M, Chen Qing, et al. Lossless data embedding with file size preservation [C]//2004 Security and Watermarking of Multimedia Contents VI. San Jose; IEEE Press, 2004:354-365.

[8] Mobasser B G, Berger R J, Marcinak M P, et al. Data embedding in JPEG bitstream by code mapping [J]. IEEE Transactions on Image Processing, 2010, 19(4):958-966.

[9] Zhang Xinpeng, Qian Zhenxing. Lossless data hiding in JPEG bitstream [J]. Journal of Systems and Software, 2012, 85(2):309-313.

作者简介:陈立泽(1987-),男,硕士研究生,研究方向:数字水印,E-mail:10120379@bjtu.edu.cn;倪蓉蓉(1976-),女,副教授,研究方向:数字水印及信息隐藏;赵耀(1967-),男,教授,研究方向:图像编码、信息检索、数字水印及信息隐藏。

