

中图分类号: TN919.8 文献标志码: A 文章编号: 1006-8961(2011)11-1967-08

论文索引信息: 高风娟, 潘志斌, 乔瑞萍, 邓晓曼. 码书距离排序对索引压缩算法的性能改进 [J]. 中国图象图形学报, 2011, 16(11): 1967-1974

码书距离排序对索引压缩算法的性能改进

高风娟, 潘志斌, 乔瑞萍, 邓晓曼

(西安交通大学电子与信息工程学院, 西安 710049)

摘要: 图像经过矢量量化后得到的索引图具有很强的统计相关性, 从而使得邻近块的索引以较大的概率相等或偏移量较小。按照某种准则对码书进行排序, 可以有效增强索引之间的相关性。基于平方欧几里得距离提出一种新的码书按距离排序方法。与传统的按均值、方差和能量等排序方法相比, 距离排序能大大提高索引图的相关性, 使索引之间的偏移量向值小的方向明显集中。将距离排序后的码书用于 AICS (adaptive index coding scheme) 算法, 实现了更好的压缩性能。

关键词: 矢量量化; 索引图; 距离准则; 码书排序

Improved index coding algorithm using distance-based codebook sorting

Gao Fengjuan, Pan Zhibin, Qiao Ruiping, Deng Xiaoman,

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

Abstract: The index map after vector quantization has a strong statistical correlation. That means the neighboring indices are the same or the offset between them is very small. Codebook sorting can, according to some criteria, enhance the correlation among neighboring indices. Based on the squared Euclidean distance between code words, a new codebook sorting method is proposed. Compared with the conventional mean-ordered codebook, the distance-ordered codebook has a much higher correlations between neighboring indices and the offset become even smaller. As a result, distance-ordered codebook can also significantly improve the compression efficiency of the AICS (adaptive index coding scheme) algorithm.

Keywords: vector quantization; index map; distance; codebook sorting

0 引言

矢量量化 (VQ)^[1] 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。对一幅图像进行矢量量化后可得到其索引图, 由于索引图中存在较多的冗余, 即相邻块的索

引之间具有较强的相关性, 利用这些相关性, 可以对索引图进行进一步压缩, 从而更加提高矢量量化的编码效率。因此, 研究索引图的相关性具有十分重要的意义。

矢量量化可描述为从 k 维欧氏空间 \mathbf{R}^k 到码书 \mathbf{C} 的映射 Q :

$$Q: \mathbf{R}^k \rightarrow \mathbf{C} \quad \mathbf{C} = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{N-1}\}$$

收稿日期: 2010-08-12; 修回日期: 2010-12-09

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); 德州仪器创新基金项目 (2009W1201); 中兴通讯技术开发项目 (20110174)。

第一作者简介: 高风娟 (1986—), 女, 西安交通大学在读硕士研究生, 主要研究方向为图像编码。E-mail: f.j.gao@stu.xitu.edu.cn。

式中, $\mathbf{y}_i = (y_{i0}, y_{i1}, \dots, y_{i(k-1)})^T, \mathbf{y}_i \in \mathbf{R}^k$ 为码书中任意一个码字, $i = 0, 1, \dots, N - 1, N$ 为码书尺寸。

矢量量化的码字搜索过程为在码书 \mathbf{C} 中寻找与输入矢量 $\mathbf{x} = (x_0, x_1, \dots, x_{k-1})^T$ 最匹配 (best match) 的码字 y_{bm} 的过程。通常输入矢量 \mathbf{x} 与码字 \mathbf{y}_i 之间的失真测度采用平方误差测度, 即

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

那么达到最小失真的最佳匹配码字 y_{bm} 为

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

最匹配码字 y_{bm} 对应的索引 I_{bm} 作为矢量量化的结果送入信道进行传输。

在图像压缩中, 通常把输入图像分割成互不重叠的图像块, 再对每一图像块进行矢量量化, 得到相应索引 I_{bm} , 构成一幅索引图。由于图像的相邻块之间具有很强的相关性, 相应的索引之间也保留了这些相关性。很多有效地算法就利用这些相关性对索引图进行再压缩。典型的算法有搜索路径算法 (SOC^[2]), SOC 算法在当前块周围已编码块中进行搜索, 若找到与当前索引相等的块, 则传输相应的搜索路径, 针对该算法已提出了很多改进算法^[3-5]; 低复杂性的索引压缩算法 (LCIC^[6]) 考虑了当前块的索引与上邻块 (左邻块) 索引相等和偏移量较小的情况; 动态树形编码算法 (DTCS^[7]) 利用周围已编

码块的空间关系动态的选择编码树, 实现比特率的降低; ICS (index coding scheme) 算法^[8] 与 DTCS 算法类似, 利用水平和垂直的空间关系动态的选择编码树; 自适应索引编码算法 (AICS^[9]) 是在动态编码树的基础上考虑了当前索引与邻近索引的偏移量问题; 2009 年 Chang 等人提出一种利用状态码书 (state codebook) 进行索引压缩的算法^[10]。

原理上, 通过 LBG^[11] (Linde-Buzo-Gray) 算法训练生成的原始码书是无序排列的, 相邻块索引之间比较分散。如果对原始码书按一定的特征值进行排序, 相邻索引之间的相关性会相应提高。同时码书排序对快速码字搜索算法的性能也有影响^[11]。本文研究了不同的码书排序方法对索引图相关性的影响, 基于平方误差测度准则提出一种新的码书按距离排序方法, 显著改善了索引图的相关性, 有效提高了 AICS 算法的压缩性能。

1 索引图相关性的统计

1.1 邻近块索引相等的统计

索引图的相关性体现在两个方面: 1) 邻近块索引相等; 2) 索引之间相似或偏移量较小。图 1 给出了 Lena 图像索引图的一部分, 由图中可以看出, 邻近块索引具有很强的相关性。

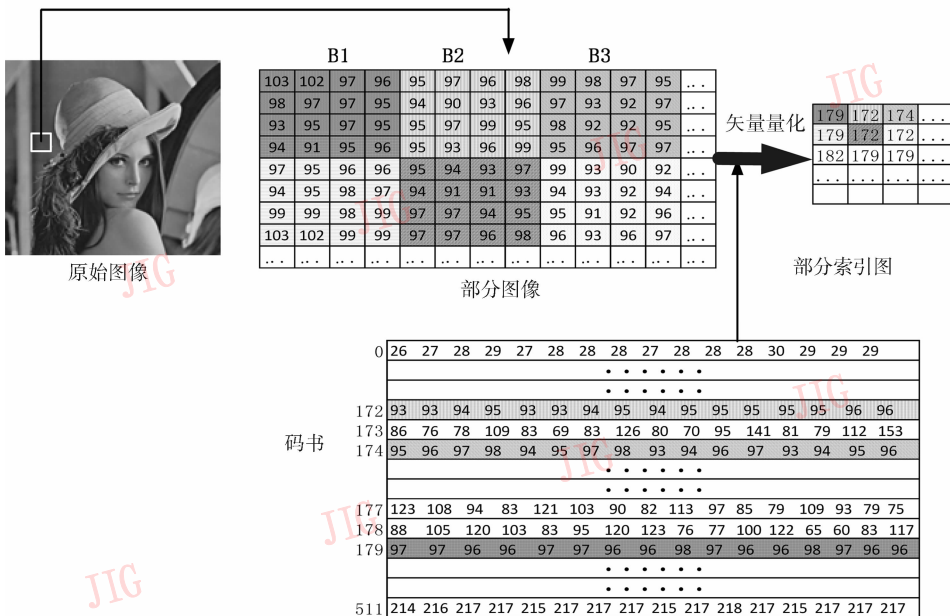


图 1 索引图的一部分

Fig. 1 Portion of index map

研究索引之间的相关性,主要统计当前块的索引与邻近 4 块索引的关系。图 2 规定了当前块 I_c 与其邻近 4 块的位置关系,邻近 4 块指:左边块 LE (left block),左上块 LU (left upper block),上边块 UP (upper block),右上块 RU (right upper block),图中索引已经编码的块用阴影表示。

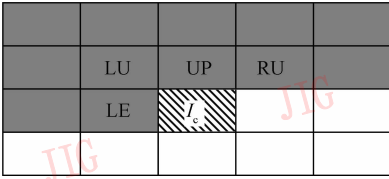


图 2 当前块与邻近四块的位置关系

Fig. 2 The position-relation of the current block and its four neighboring blocks

对一幅测试图像,分别统计当前索引 I_c 与邻近块索引 I_{UP} 、 I_{LE} 、 I_{LU} 、 I_{RU} 的相等次数。实验数据显示,码书越小索引之间的相关性越大。对多数图像,垂直方向的相关性大于水平方向的相关性。然而这种统计方法是有重复的,因为当前索引 I_c 可以同时和多个邻近块的索引相等。为了对索引图进行无重复的统计,可根据相关性设置统计顺序, I_c 依次与 I_{UP} 、 I_{LE} 、 I_{LU} 、 I_{RU} 进行比较。表 1 列出了索引相等的无重复统计结果,其中 N 表示码书大小;SUM_UP 表示在索引图中 I_c 与 I_{UP} 相等的比例(%),同理有 SUM_LE, SUM_LU 和 SUM_RU;SUM 是 SUM_UP、SUM_LE、SUM_LU 与 SUM_RU 的总和。由表中实验数据可以看出,索引图中平均有 45% 左右的索引与邻近四块的索引相等。因此,可以利用这一特性,实现对索引图的进一步压缩。

表 1 当前块与邻近四块索引相等的无重复统计 ($N=512$)

Tab. 1 Statistics without overlapping for the current index and its four neighboring indices equal ($N=512$)

图像	SUM_UP	SUM_LE	SUM_LU	SUM_RU	SUM
Lena	21.53	6.47	4.93	4.23	37.15
F16	25.39	16.02	3.68	3.93	49.01
Lax	10.85	6.99	2.94	3.05	23.83
Peppers	21.84	9.82	4.21	4.31	40.18
Splash	30.01	14.93	5.38	4.98	55.29
Tiffany	46.74	10.67	3.45	5.28	66.14
AVG	26.05	10.82	4.1	4.3	45.27

1.2 邻近块偏移量的统计

偏移量的统计与索引相等情况的统计过程

类似,分别统计当前索引 I_c 与邻近块索引 I_{UP} 、 I_{LE} 、 I_{LU} 、 I_{RU} 的偏移量分布情况,也包括有重复和无重复的统计。当码书按照一定的准则排序后,偏移量将往数值较小的分布范围内集中,详细情况在第 4 节中介绍。

2 码书排序方法

2.1 典型的码书排序方法

典型的码书排序方法有均值排序、方差排序、能量排序和 PCA 排序。

1) 均值排序

码书中的码字按照均值从小到大进行排序,其中 k 维矢量 $p = (p_0, p_1, \dots, p_{k-1})^T$ 的均值定义为

$$m_p = \left(\sum_{l=0}^{k-1} p_l \right) / k \quad (3)$$

2) 方差排序

码书中的码字按照方差从小到大进行排序,其中 k 维矢量的方差定义为

$$v_p = \sqrt{\sum_{l=0}^{k-1} (p_l - m_p)^2} \quad (4)$$

3) 能量排序

码书中的码字按照能量从小到大进行排序,其中能量就是 k 维矢量与原点的距离,定义为

$$e_p = \sqrt{\sum_{l=0}^{k-1} p_l^2} \quad (5)$$

4) PCA 排序

利用主成分分析 (PCA) 方法得到码书的第 1 主成分轴,将 k 维空间的码字映射到 1 维空间,按投影值从小到大对码书排序。

对 k 维空间中大小为 N 的码书进行主成分分析,即计算码书中 N 个码字的协方差矩阵,求出 k 个特征值 $\lambda_1, \lambda_2, \dots, \lambda_k$ 和对应的单位特征向量 D_1, D_2, \dots, D_k ,其中 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k$ 。然后利用第 1 主成分轴 D_1 将码字投影到 1 维坐标上。一般的,码字 y_i 到 D_j 的投影值可利用内积表示为 $c_i^j = y_i \cdot D_j$ ($i = 0, 1, \dots, N-1; j = 1, \dots, k$)。

2.2 距离排序

矢量量化的码字搜索过程中,计算输入矢量 x 与码字 y_i 之间的失真测度采用了平方误差测度,如式(1)所示。平方误差测度实际上是两个矢量之间的平方欧几里得距离,即输入矢量 x 与最匹配码字

y_{bm} 的距离最短。另外,由于图像的相关性,相邻图像块的平方欧几里得距离也比较小,通过矢量量化编码后,最佳匹配码字之间的距离也较小。例如,在图 1 中,图像块 B1、B2 和 B3 对应的索引分别为 179、172 和 174,即最佳匹配码字分别为第 179 个、第 172 个和第 174 个码字。B1 与 B2 之间的平方欧几里得距离为 203,且第 179 个和第 172 个码字之间的距离为 109。B2 与 B3 之间的平方欧几里得距离为 144,且第 172 个和第 174 个码字之间的距离为 67。由此可以看出,B2 与 B3 之间的距离较小,其对应的最佳匹配码字之间的距离也较小。但是由 LBG 算法直接生成的原始码书是无序排列的,距离接近的码字在码书中是分散排列的,导致了邻近块的索引值之间相差较大。因此,可以考虑按照码字之间的距离对码书进行排序,这样就会使距离较小码字的索引值相差不大。对图像进行矢量量化时,相邻图像块的最佳匹配码字在码书中的位置会比较靠近,从而相应的索引值之差也变小,索引图的统计相关性增强。

提出一种新的按照码字之间的距离对码书进行排序的方法。与传统的均值排序、方差排序、能量排序等相比,距离排序可以显著改善邻近块索引之间的偏移量,使偏移量往值小的分布范围内集中。

本文提出的码书按照距离排序的方法如下:选取一个码字作为起始点 begin,在剩余的 $(N - 1)$ 个码字中搜索与 begin 距离最近的码字 w_1 进行排序,然后在 $(N - 2)$ 个码字中再搜索与 w_1 距离最近的码字 w_2 进行排序,依次类推。

在排序时有两个问题需要考虑,一是起始点的选择问题,二是排序的方向问题。本文选择与码书的质心最接近的码字作为起始点。确定起始点 begin 以后,有两种排序方法,一是单向排序,即先排 begin 一侧,排满后,再排另一侧;二是两侧交叉排序。图 3 给出了这两种码书排序方法的示意图。

对于单向排序,确定初始点 begin,在剩余的 $(N - 1)$ 个码字中搜索与 begin 距离最近的码字 w_1 ,将 w_1 排在 begin 的上方,接着在剩余的 $(N - 2)$ 个码字中再搜索与 w_1 距离最近的码字 w_2 ,将 w_2 排在 w_1 的上方,依次类推,直到排到码书的开始位置,然后再进行 begin 下方的排序。如果将起始点放在码书的开始(结尾),则只需依次向下(向上)排序即可。

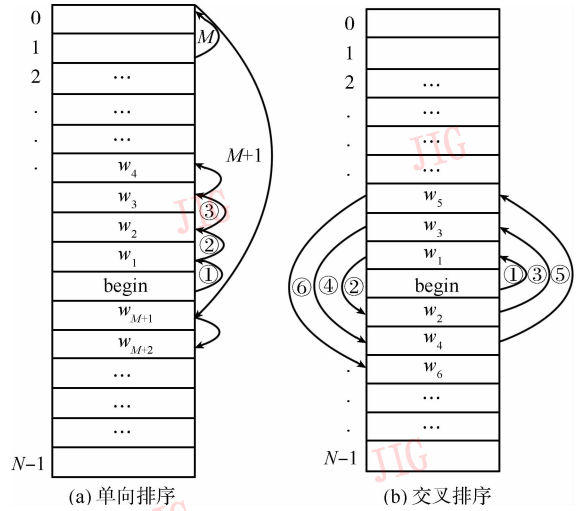


图 3 两种距离排序方法

Fig. 3 Comparison of two distance-based sorting methods

对于交叉排序,确定初始点 begin,在剩余的 $(N - 1)$ 个码字中搜索与 begin 距离最近的码字 w_1 ,将 w_1 排在 begin 的上方,并将与 begin 距离最近的码字 w_2 排在 begin 的下方。接着在剩余的 $(N - 3)$ 个码字中搜索与 w_1 距离最近的码字 w_3 ,将 w_3 排在 w_1 的上方;在剩余的 $(N - 4)$ 个码字中搜索与 w_2 距离最近的码字 w_4 ,将 w_4 排在 w_2 的下方;依次类推,直到最后剩余码字个数为 0。

3 AICS

AICS^[9] (adaptive index coding scheme) 算法是在 LCIC^[6] (low complexity index-compressed) 算法和 DTCS^[7] (dynamic tree-coding scheme) 算法基础上提出的,在动态编码树中考虑了当前索引与邻近索引的偏移量问题。在 AICS 算法中,当前索引值需要与邻近块的已编码索引值进行比较,邻近块的位置如图 4 所示。 I_c 是当前要编码的索引,常用的 5 个邻近块分别为 I_L (left)、 I_{LM} (left most)、 I_T (top)、 I_{TM} (top most) 和 I_{TL} (top left)。

AICS 算法对索引编码的步骤如下:

- 1) 获取当前的索引 I_c , 及其邻近索引 I_L 、 I_{LM} 、 I_T 、 I_{TM} 和 I_{TL} 。
- 2) if ($I_L = I_T$) 按编码表 Tree1 对 I_c 进行编码。
else
if ($I_T = I_{TM}$) 按编码表 Tree2 对 I_c 进行编码。

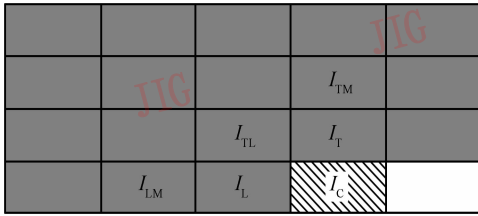


图 4 当前索引与其邻近块的位置关系
Fig. 4 Position assignment of the neighboring indices around current index

else
if($I_L = I_{LM}$)按编码表 Tree3 对 I_c 进行编码。
else
if($I_L = I_{TL}$)按编码表 Tree4 对 I_c 进行编码。
else
if($I_T = I_{TL}$)按编码表 Tree5 对 I_c 进行编码。
else 按编码表 Tree6 对 I_c 进行编码。
重复执行步骤 1), 步骤 2), 直到对所有索引都完成编码。

编码表 Tree1 至编码表 Tree6 分别如表 2 所示。在 AICS 中, 当前块索引 I_c 按范围 ($I_T \pm J$) 或 ($I_L \pm J$) 进行检查, 偏移量为 I_T 或者 I_L 减去当前索引值, 其中 J 是整数。这样, 偏移量就被确定在 $-J$ 到 $+J$ 之间 (含 0), 对它的编码需要 $m = \log_2(2 \cdot J)$ 位比特。为了提高编码效率, J 一般设为 2。

表 2 AICS 算法的编码表

Tab. 2 Coding table for AICS method

状态	$I_c = I_T$	$I_c = I_L$	$I_c \geq I_T - J$ 或 $I_c \leq I_T + J$	$I_c \geq I_L - J$ 或 $I_c \leq I_L + J$	其他
Tree1	1		$00 + m$		$01 + I_c$
Tree2	1	010		$00 + m$	$011 + I_c$
Tree3	010	1		$00 + m$	$011 + I_c$
Tree4	010	1	$00 + m$		$011 + I_c$
Tree5	1	010	$00 + m$		$011 + I_c$
Tree6	11	100	$1010 + m$	$1011 + m$	$0 + I_c$

AICS 算法充分利用了索引图的统计相关性进行编码。首先, AICS 算法对当前索引进行分类, 选择不同的编码表进行编码, 分类的依据是已编码邻近块的相关性, 实际上这是对当前索引的预测。比如当 $I_L = I_{LM}$ 时, 索引在水平方向相等的概率比较大, 只用 1 比特来编码 $I_c = I_L$, 偏移量也选择了 I_L 与 I_c 的差, 如编码表 2 所示; 而当 $I_T = I_{TM}$ 时, 垂直方向

的相关性较强, 用 1 比特编码 $I_c = I_T$, 偏移量为 I_T 与 I_c 的差, 如编码表 2 所示。其次, 在编码时, AICS 算法利用了当前索引与其左边块索引 I_L 和上边块索引 I_T 相等的情况, 并用较少的比特来编码 $I_c = I_L$ 和 $I_c = I_T$ 。另外, AICS 算法中使用了当前索引与其左边块索引 I_L 和上边块索引 I_T 的偏移量, 这在其他索引压缩编码算法中较少使用。同时, 在每一个编码表中, AICS 算法根据不同状态出现的概率, 对标志位采用 Huffman 编码的方式, 有效地降低了编码的比特率。

4 实验结果及分析

4.1 码书排序对索引图相关性的影响

利用 LBG 算法直接生成的原始码书是无序排列的。对码书按照某一准则排序后, 相邻码字之间的相关性增强, 索引图的相关性也增强, 图 5 给出了码书排序前后 Lena 图像的两个不同的索引图, 这里显示的索引图是对索引值利用线性映射 $\frac{I_c}{N-1} \times 255$ 归一化到像素值 $[0, 255]$ 范围内的结果。由图 5(b)(c) 可以看出, 码书按照均值排序后, 索引图与原始图像更加趋于一致, 即此索引图保留了更多的原始图像信息。

码书排序对索引之间的偏移量影响较大, 表 3 和图 6 列出了码书大小为 512 时, 不同排序方法下, Lena 的索引图偏移量的变化。其中距离排序 A 采用码书的质心为起始点, 单向排序的方法; 距离排序 B 采用码书的质心为起始点, 交叉排序的方法。offset1_UP 表示当前索引 I_c 与 I_{UP} 相差为 1 的次数 (%), offset2_UP 表示当前索引 I_c 与 I_{UP} 相差为 2 的次数 (%), Sum1 是 offset1_UP、offset1_LE、offset1_LU 和 offset1_RU 的总和, 其他符号具有类似的含义。

与索引相等的无重复统计类似, 统计偏移量时也可设置统计顺序, 先进行偏移量为 1 的统计, I_c 依次与 I_{UP} 、 I_{LE} 、 I_{LU} 、 I_{RU} 进行比较, 若存在偏移量为 1 的情况, 相应的 offset1 加 1, 否则继续进行偏移量为 2 的统计。表 4 和图 7 给出了不用排序方法时, Lena 索引图偏移量无重复的统计结果。

由实验数据可以看出, 距离排序使索引之间的偏移量有效地向取值较小的情况集中。在对偏移量进行无重复的统计时, 如表 4 所示, 均值排序的偏移

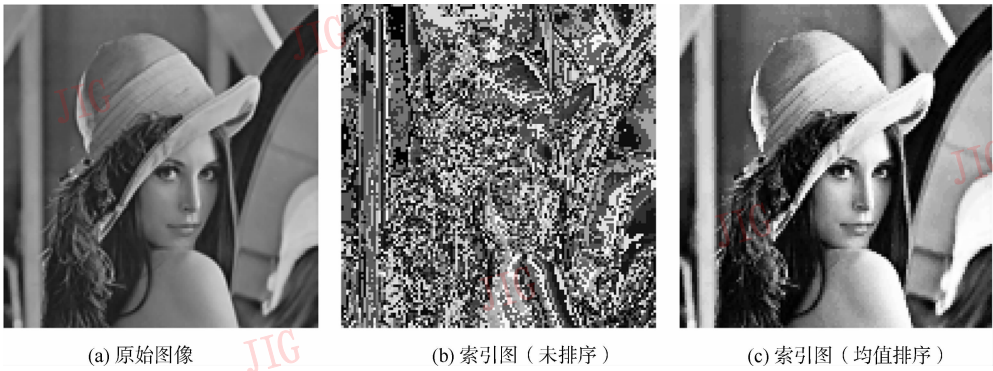


图 5 码书排序对索引图的影响

Fig. 5 Effect of codebook sorting on index map

表 3 不同排序方法,偏移量的有重复统计 (N=512)

Tab. 3 Offset statistics with overlapping among the conventional codebook sorting methods and the proposed method (N=512)

图像		均值	方差	能量	PCA	距离	距离
Lena	未排序	排序	排序	排序	排序	排序 A	排序 B
offset1_UP	0.19	2.80	0.47	2.40	2.60	15.18	15.09
offset1_LE	0.29	2.10	0.47	1.76	2.20	12.77	12.49
offset1_LU	0.28	2.17	0.63	1.91	2.25	12.48	12.38
offset1_RU	0.32	2.03	0.55	1.82	2.02	11.72	11.62
offset2_UP	0.31	3.02	0.86	3.54	3.54	8.25	8.20
offset2_LE	0.35	2.99	0.93	3.58	3.35	7.84	7.78
offset2_LU	0.36	2.64	0.91	3.33	3.01	7.31	7.20
offset2_RU	0.40	2.54	1.04	3.10	3.02	7.55	7.42
Sum1	1.08	9.10	2.12	7.89	9.07	52.15	51.58

表 4 不同排序方法,偏移量的无重复统计 (N=512)

Tab. 4 Offset statistics without overlapping among the conventional codebook sorting methods and the proposed method (N=512)

图像		均值	方差	能量	PCA	距离	距离
Lena	未排序	排序	排序	排序	排序	排序 A	排序 B
offset1_UP	0.09	2.11	0.44	1.91	1.89	9.52	9.50
offset1_LE	0.16	1.10	0.29	0.99	1.16	3.99	3.85
offset1_LU	0.12	0.88	0.37	0.99	0.98	2.77	2.85
offset1_RU	0.10	0.89	0.33	1.02	0.85	2.35	2.36
offset2_UP	0.12	1.94	0.62	2.04	2.33	3.48	3.53
offset2_LE	0.10	1.42	0.48	1.54	1.45	1.72	1.74
offset2_LU	0.09	1.04	0.33	1.02	0.97	1.25	1.25
offset2_RU	0.11	0.92	0.50	1.01	1.09	1.17	1.17
Sum1	0.47	4.99	1.44	4.91	4.88	18.62	18.55

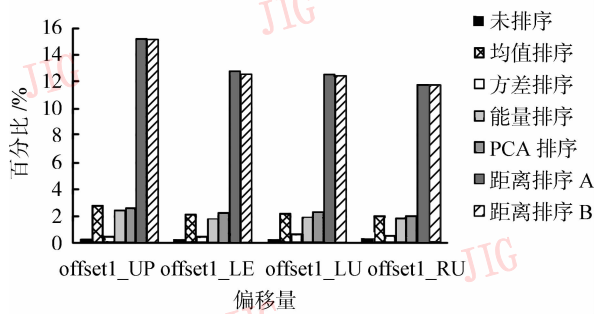


图 6 有重复统计时,不同排序方法的比较

Fig. 6 Comparison of the conventional codebook sorting methods and the proposed method with overlapping

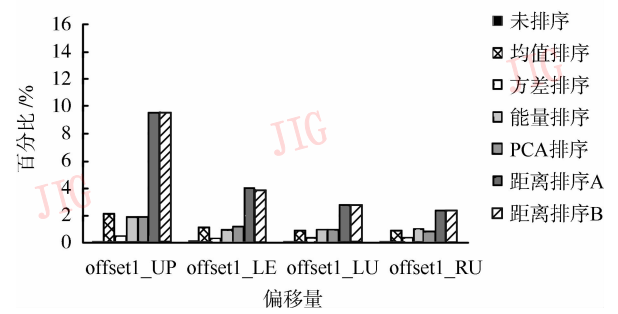


图 7 无重复统计时,不同排序方法的比较

Fig. 7 Comparison of the conventional codebook sorting methods and the proposed method without overlapping

量为 1 的总和 Sum1 等于 4.99%,而距离排序 A 的 Sum1 为 18.62%,改善了约 3.7 倍。距离排序的效果明显优于其他排序方法。均值排序和 PCA 排序的效果相近,稍微优于能量排序,方差排序的效果最差。

A 和 B 两种距离排序方法的计算量和复杂度相差不大。但两种方法对不同索引图偏移量的统计结果有略微差异,如表 4 中,对 Lena 图,A 方法的 Sum1 为 18.62%,高于 B 方法的 18.55%;而对 Splash 图,A 方法的 Sum1 为 19.40%,低于 B 方法的 19.43%。

4.2 基于码书距离排序的索引压缩编码算法改进

在索引压缩算法中,AICS^[9]算法考虑了当前索

引与邻近索引的偏移量问题,因此码书排序可以有效改进 AICS 算法的压缩率。

为了验证码书排序的效果,本文采用 512 × 512 的 Lena 图像作为训练集,利用 LBG 算法生成码书,码书大小分别为 128、256、512 和 1 024,矢量维数为 4 × 4。测试图像为 512 × 512 的 Lena、Lake、Goldhill、Bridge、Toys、Splash、Couple、F16、Tiffany 和 Truck,分别编号 I_1 至 I_10。仿真实验在主频 2.93 GHz 的 Intel Core 2 处理器,内存 2 GB 的个人计算机上进行。

表 5 列出了在不同码书大小下,AICS 算法使用未排序码书和距离排序 A 码书的比特率。表 6 列出了 AICS 算法的平均压缩率。

表 5 不同码书下,AICS 算法的比特率

Tab. 5 Bit rates of AICS method using different codebooks

		/(bit/像素)										
码书	算法	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_10	AVG
128	AICS	0.329	0.338	0.367	0.432	0.196	0.244	0.361	0.278	0.225	0.384	0.315
	本文算法	0.268	0.295	0.304	0.393	0.174	0.187	0.302	0.234	0.170	0.317	0.264
256	AICS	0.396	0.401	0.426	0.493	0.222	0.282	0.430	0.337	0.279	0.479	0.374
	本文算法	0.327	0.344	0.368	0.456	0.196	0.226	0.372	0.278	0.210	0.416	0.319
512	AICS	0.508	0.495	0.519	0.569	0.281	0.430	0.523	0.439	0.344	0.576	0.468
	本文算法	0.427	0.431	0.473	0.541	0.253	0.338	0.467	0.352	0.284	0.529	0.409
1 024	AICS	0.608	0.581	0.590	0.638	0.341	0.523	0.603	0.554	0.438	0.653	0.553
	本文算法	0.532	0.519	0.554	0.617	0.300	0.449	0.558	0.479	0.371	0.621	0.500

表 6 不同码书下,AICS 算法的平均压缩率

Tab. 6 Average bit rate reductions of AICS method using different codebooks

RED	码书 128	码书 256	码书 512	码书 1 024
AICS	27.91	25.02	16.64	11.45
本文算法	39.55	36.05	27.12	19.92
提高幅度	11.64	11.03	10.48	8.47

压缩率的定义为

$$RED = \frac{b_0 - b}{b_0} \times 100(\%) \quad (6)$$

式中, b_0 为索引压缩前传输每像素所需比特数, b 为索引压缩后传输每像素所需比特数。码书大小为 128、256、512 和 1 024 时, b_0 的值分别为 0.437 5、0.5、0.562 5 和 0.625 (bit/像素)。

由实验数据可以看出,码书按照距离排序后,AICS 算法的比特率可进一步下降 10% 左右。在

码书大小为 128 时,使用距离排序的码书,AICS 算法的比特率下降了 11.6%,而在码书大小为 1 024 时,AICS 算法的比特率只下降了 8.5%。造成此现象的原因在于,在小码书的情况下,矢量量化的量化台阶比较少,索引之间的相关性比较高,对码书按照码字之间的距离进行排序后,邻近块索引值之间的偏移量进一步集中,有利于 AICS 算法的编码。而随着码书的增大,量化台阶增多,邻近块索引值之间相等或者相似的概率降低,AICS 算法的压缩率降低。

5 结 论

对矢量量化编码后的索引图进行了统计相关性研究,实验结果表明,索引图垂直方向的相关性强于水平方向,一幅索引图中约 45% 以上的索引与其邻近四块的索引值相等。按照不同准则排序之后的码

书可以进一步改善索引图的相关性,使索引之间的偏移量往值小的方向有效集中。基于搜索最佳匹配码字的平方误差测度准则,提出一种新的码书按照距离排序的方法。与传统的均值排序相比,偏移量为 1 的情况由 4.99% 增加到 18.62%,距离排序有约 3.7 倍的改进。将距离排序后的码书用于索引压缩编码算法,特别是使用了偏移量的 AICS 算法,可以进一步显著降低比特率,对不同的码书大小,AICS 算法的比特率可下降 10% 左右。

参考文献 (References)

- [1] Linde Y, Buzo A, Gray R M. An algorithm for vector quantizer design [J]. IEEE Transactions on Communications, 1980, 28 (1) : 84-95.
- [2] Hsieh C H, Tsai J C. Lossless compression of VQ index with search-order coding [J]. IEEE Transactions on Image Processing, 1995, 41 (2) : 327-31.
- [3] Shie S C, Chen L T. Image compression based on side-match VQ and SOC [C] // Digital Image Computing: Techniques and Applications. Melbourne, Australia: IEEE Computer Society, 2009: 369-373.
- [4] Ho C Y, Hsieh C H, Chao C W. Modified search order coding for vector quantization indexes [J]. Tamkang Journal of Science and Engineering, 1999, 2 (3) : 143-148.
- [5] Chang C C, Chen G M, Hu Y C. A novel index coding scheme for vector quantization [J]. Fundamenta Informaticae, 2006, 71 (2/3) : 215-227.
- [6] Hu Y C, Chang C C. Low complexity index-compressed vector quantization for image compression [J]. IEEE Transactions on Consumer Electronics, 1999, 45 (1) : 219-224.
- [7] Chen P Y, Chen R D. An index coding algorithm for image vector quantization [J]. IEEE Transactions on Consumer Electronics, 2003, 49 (4) : 1513-1520.
- [8] Chen P Y, Yu C T. Lossless vector-quantised index coding design and implementation [J]. IEE Proceedings Circuits, Devices & Systems, 2005, 152 (2) : 109-117.
- [9] Somasundaram K, Dornic S. Adaptive index coding scheme for VQ indices compression [C] // Proceedings of the International Conference on Computing: Theory and Applications. Kolkata, India: IEEE Computer Society, 2007: 639-644.
- [10] Chang C C, Chen G M, Lin C C. Lossless compression schemes of vector quantization indices using state codebook [J]. Journal of Software, 2009, 4 (4) : 274-282.
- [11] Qiao Yang, Pan Zhibin, Qiao Ruiping. Effectiveness analysis of codebook rearrangement on fast codeword search algorithms [J]. Journal of Image and Graphics, 2010, 15 (8) : 1182-1188. [乔阳, 潘志斌, 乔瑞萍. 码书排序对快速码字搜索算法性能影响的分析 [J]. 中国图象图形学报, 2010, 15 (8) : 1182-1188.]