

# 码书排序对快速码字搜索算法性能影响的分析

乔 阳 潘志斌 乔瑞萍

(西安交通大学电子与信息工程学院, 西安 710049)

**摘 要** 矢量量化快速码字搜索算法中,为了有效地减小搜索范围,必须对原始码书按一定的准则进行重新排序。对现存的两类快速码字搜索算法进行了总结,其中一类是码书按1维顺序关系排序,另一类是码书按2维相邻关系排序。通过实验给出了两类算法的搜索范围和编码时间,并进行了比较和分析,进而提出了在实际编码时如何更好地使用这两种排序关系的准则。

**关键词** 矢量量化 码字快速搜索 码书排序 性能评价

中图法分类号: TN919.8 文献标志码: A 文章编号: 1006-8961(2010)08-1182-07

## Effectiveness Analysis of Codebook Rearrangement on Fast Codeword Search Algorithms

QIAO Yang, PAN Zhibin, QIAO Ruiping

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

**Abstract** In order to reduce search space efficiently, the codebook must be rearranged in the fast codeword search algorithms of vector quantization. Two kinds of algorithms are summarized in this paper. One is the codebook sorted by 1D characteristic value; the other is the codebook mapped into 2D adjacent ripples. Comparisons and analysis are conducted for these two kinds of fast algorithms in terms of the search space and encoding time. The criteria about how to use these two rearrangement method reasonably is also given in this paper.

**Keywords** vector quantization, fast codeword search, codebook rearrangement, performance evaluation

## 0 引 言

矢量量化(VQ)<sup>[1]</sup>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。但是矢量量化编码过程中寻找输入矢量的最佳匹配码字的计算量非常大,严重限制了矢量量化技术的应用和进一步发展,因此研究矢量量化中码字的快速搜索算法具有十分重要的意义。

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

$$Q: R^k \rightarrow C, C = \{y_0, y_1, \dots, y_{N-1}\}$$

其中,  $y_i = (y_{i0}, y_{i1}, \dots, y_{i(k-1)})^T$ ,  $y_i \in R^k$  为码书中的任意一个码字,  $i = 0, 1, \dots, N-1$ ,  $N$  为码书尺寸。矢量量化码字搜索过程即在码书  $C$  中寻找与输入矢量  $x = (x_0, x_1, \dots, x_{k-1})^T$  最匹配的码字  $y_{bm}$  (best match) 的过程。通常输入矢量  $x$  与码字  $y_i$  之间的失真测度采用平方误差测度,即

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

那么达到最小失真的最佳匹配码字  $y_{bm}$  为

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

如果采用穷尽搜索算法(FS),那么对每一个输入矢量  $x$  要获得最佳匹配码字  $y_{bm}$  就需要计算  $Nk$  次

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)

收稿日期:2009-01-08;改回日期:2009-04-01

第一作者简介:乔 阳(1984—),男,2009年于西安交通大学获信息与通信工程专业硕士学位。主要研究方向为图像编码。

E-mail: xjtu.qiaoyang@stu.xjtu.edu.cn

乘法、 $N(2k-1)$ 次加法和  $N-1$ 次比较运算,其计算量非常大,因此为了提高矢量量化的编码效率,许多文献中提出了各种快速码字搜索算法<sup>[2-11]</sup>。

原理上由训练矢量直接生成的原始码书排列是无序的,为了有效减小搜索范围,提高搜索效率,通常必须对原始码书按一定的特征值(标量)进行排序。在传统的快速码字搜索算法中,码书都是按照码字的 1 维特征量进行排序,然后再根据一定的准则在排序后的码书中进行顺序搜索。典型的算法有以均值排序的等均值最近邻搜索算法<sup>[3]</sup>(ENNS)、等均值等方差最近邻搜索算法<sup>[4]</sup>(EENNS)、改进的等均值等方差最近邻搜索算法<sup>[5]</sup>(IEENNS)、等均值等方差等范数最近邻搜索算法<sup>[6]</sup>(EEENNS)以及对 EEENNS 算法的性能分析<sup>[7]</sup>、利用投影的最近邻搜索算法<sup>[8]</sup>、以范数排序的等范数最近邻搜索算法<sup>[9]</sup>和以 K-L 变换后第 1 主成分排序的基于特征向量的算法<sup>[10]</sup>(EVM)等。与基于顺序关系的码书排序的思想不同,2007 年 Chang 等人新提出的 PVDS<sup>[11]</sup>(planar voronoi diagram search method)算法则是根据码字的 2 维相邻关系对码书进行排序,然后按照邻域关系(纹波关系)进行搜索。但迄今为止还未有关于这两类码书排序准则对搜索性能影响的比较,尚不知道在实际编码时如何最合理地选择码书排序的准则。因为 EENNS 算法和 IEENNS 算法是在 ENNS 算法基础上的改进算法,且 ENNS 算法在实际过程中广泛使用,所以本文主要从原理上总结了基于这两类码书排序准则的典型算法 ENNS 算法和 PVDS 算法,通过对两者进行仿真实验和比较分析,提出了如何在快速码字搜索算法中使用这两种关系的准则。

## 1 典型算法

### 1.1 等均值最近邻搜索算法(ENNS)

设  $L$  为  $k$  维空间  $\mathbf{R}^k = (R_0, R_1, \dots, R_{k-1})$  中的一条直线,如果  $L$  上任何一个点  $\mathbf{p} = (p_0, p_1, \dots, p_{k-1})$  满足  $p_0 = p_1 = \dots = p_{k-1}$ ,则称  $L$  为  $\mathbf{R}^k$  的中心线。若  $L$  为  $\mathbf{R}^k$  的中心线,则  $\mathbf{R}^k$  上任一点  $\mathbf{x}$  在直线  $L$  上的投影点为  $\mathbf{L}_x = (m_x, m_x, \dots, m_x)$ ,其中  $m_x$  为矢量  $\mathbf{x}$  的均值。

**定理 1** 假定当前已经找到的最小失真为  $d_{\min}$ ,  $m_i$  为码字  $\mathbf{y}_i$  的均值,  $m_x$  为输入矢量  $\mathbf{x}$  的均值。若

$$|m_i - m_x| \geq \sqrt{d_{\min}/k} \quad (3)$$

则  $d(\mathbf{x}, \mathbf{y}_i) \geq d_{\min}$  成立。

那么  $\mathbf{y}_i$  不可能是最佳匹配码字,所以不必计算  $d(\mathbf{x}, \mathbf{y}_i)$  即可将候选码字  $\mathbf{y}_i$  排除。基于定理 1 的算法称为等均值最近邻搜索算法<sup>[3]</sup>(如 ENNS)。令  $m_{\max} = m_x + \sqrt{d_{\min}/k}$ ,  $m_{\min} = m_x - \sqrt{d_{\min}/k}$ ,则 ENNS 算法将搜索范围限制到两个平行的等均值超平面内,其均值分别为  $m_{\max}$  和  $m_{\min}$  并与中心线  $L$  分别相交于点  $\mathbf{L}_{\max} = (m_{\max}, m_{\max}, \dots, m_{\max})$  和  $\mathbf{L}_{\min} = (m_{\min}, m_{\min}, \dots, m_{\min})$ 。图 1 中阴影部分给出了 2 维情况下减小了的 ENNS 算法搜索范围,它被限制在直线  $L_1$  和  $L_2$  之间,这两条直线分别与中心线  $L$  垂直相交于点  $\mathbf{L}_{\min}$  和  $\mathbf{L}_{\max}$ ,图中  $r_p$  为输入矢量  $\mathbf{x}$  和初始匹配码字  $\mathbf{y}_p$  之间距离的平方  $\sqrt{d_{\min}}$ 。

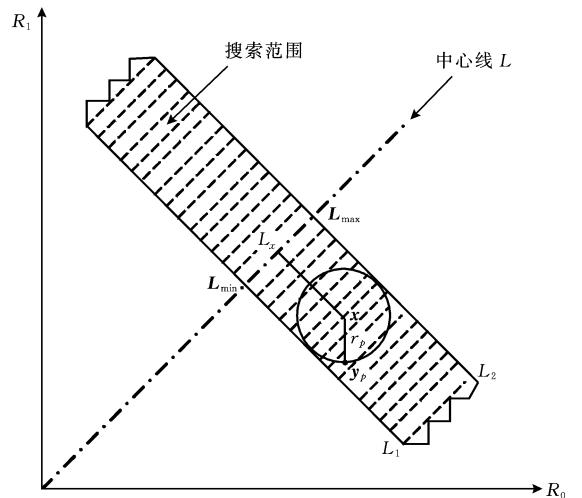
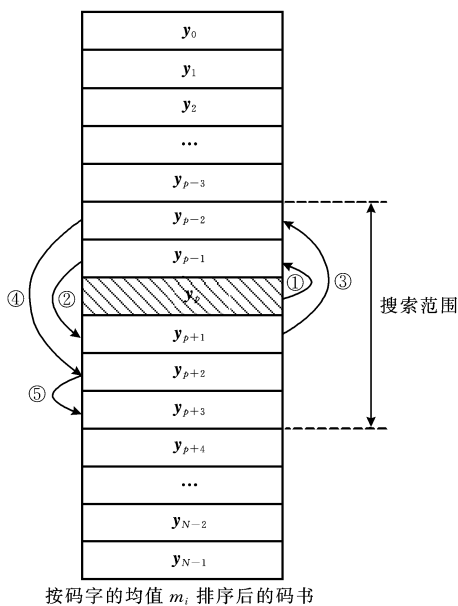


图 1 ENNS 算法搜索范围的 2 维示例

Fig.1 Demonstration of search space for ENNS algorithm in 2-dimensional case

ENNS 算法在开始搜索前需要离线计算各码字的均值  $m_i, i=0, 1, \dots, N-1$ , 然后以均值为特征值, 按照均值的大小对码书进行 1 维的升序排列。在编码阶段需要在线计算输入矢量  $\mathbf{x}$  的均值  $m_x$ , 接着通过二分法找出码书中均值与  $m_x$  最接近的码字  $\mathbf{y}_p$ , 并将该码字作为初始匹配码字。在码字  $\mathbf{y}_p$  附近进行上下交替搜索, 一旦某个方向上码字的均值满足式(3), 则该方向的搜索停止。图 2 给出了 ENNS 算法的搜索过程示意图, 从初始匹配码字  $\mathbf{y}_p$  开始, 先向上搜索  $\mathbf{y}_{p-1}$ , 若  $\mathbf{y}_{p-1}$  的均值不满足式(3), 则向下搜索  $\mathbf{y}_{p+1}$ , 若  $\mathbf{y}_{p+1}$  的均值也不满足式(3), 则又向上搜索  $\mathbf{y}_{p-2}$ , 如果  $\mathbf{y}_{p-2}$  的均值满足式(3), 则从这一时刻开始停止向上搜索而只执行向下搜索, 即从  $\mathbf{y}_{p+2}$  开始向下继续搜索, 直到  $\mathbf{y}_{p+3}$  的均值也满足式(3)



按码字的均值  $m_i$  排序后的码书

图 2 ENNS 算法搜索过程示例

Fig. 2 Demonstration of search flow in ENNS algorithm

时停止整个搜索过程。

### 1.2 PVDS 算法

除了按码字的 1 维特征量对码书进行排序外,码书的另一种排序方法是根据码字之间的 2 维相邻关系来实现的。最近,文献[11]提出了一种新的称为 PVDS 的码书排序算法, PVDS 算法基于对码书的主成分分析(PCA),利用 PCA 分析得到的第 1 主成分轴和第 2 主成分轴,将  $k$  维空间的码字映射到 2 维空间以获得码字之间的相邻关系。PVDS 算法首先利用码书的第 1 主成分和第 2 主成分来构造代表码字之间相邻关系的 Voronoi 平面图,然后根据 Voronoi 图创建每个码字的邻接表,最后由输入矢量初始匹配码字的邻接表来确定搜索范围,进行最佳匹配搜索。

PVDS 算法首先要离线对  $k$  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,得到  $k$  个特征值  $\lambda_1, \lambda_2, \dots, \lambda_k$  和单位特征向量  $D_1, D_2, \dots, D_k$ ,其中  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k$ 。接下来利用  $D_1$  和  $D_2$  两个主成分轴将码字投影到 2 维平面上,一般地,码字  $y_i$  到  $D_j$  的投影值可利用内积表示为  $c_i^j = y_i \cdot D_j, i = 0, 1, \dots, N - 1, j = 1, 2$ 。

若码书投影后的  $N$  个点为  $p_i(c_i^1, c_i^2)$ ,每个点所在的 Voronoi 胞腔为  $V_i$ ,点  $p_i$  所在的胞腔  $V_i$  定义为  $V_i = \bigcup_{i \neq j} \{p_x \in R^2 \mid d(p_x, p_i) < d(p_x, p_j)\}, i = 0, 1, \dots, N - 1$ 。根据以上定义可知,每个胞腔中只包含一个投影点  $p_i$ ,胞腔  $V_i$  中的任意一点  $p_x$  满足  $d(p_x, p_i) < d(p_x, p_j), i \neq j$ 。另外,每个胞腔  $V_i$  为一个凸多边形区域,且由 Voronoi 边界和 Voronoi 顶点组成。实际上, Voronoi 边界由连接 2 个相邻胞腔投影点直线的垂直平分线组成, Voronoi 顶点由这些垂直平分线相交所成,通过这种方法就可以构造出 Voronoi 平面图。图 3 (a) 和图 3 (b) 描述了用上述方法生成 Voronoi 图的过程,图 3 (c) 为生成的 Voronoi 图,阴影为一个 Voronoi 胞腔。

接下来根据已生成的 Voronoi 图为每个码字  $y_i$  创建一个包含  $r$  个纹波(ripple)的邻接表  $T_i^r$  作为码字搜索的范围。图 4 (a) 为码书大小为 128 时码书的 Voronoi 图示例,图 4 (b) 为由图 4 (a) 中虚线区域所示的 10 个投影点构成的局部 Voronoi 图示例。从图 4 (a) 中可以看出任意一个胞腔的相邻胞腔为和它共用一条 Voronoi 边界的那些胞腔,这样利用 Voronoi 图就能很容易地确定邻接胞腔。从图 4 (b) 中可看出,  $V_1$  的邻接胞腔为  $V_2, V_3, V_4, V_5, V_6$  和  $V_7$ ,即码字  $y_1$  的包含一个纹波的邻接表为  $T_1^1 = \{y_1, y_2, y_3, y_4, y_5, y_6, y_7\}$ 。为了得到 2 个纹波的邻接表  $T_1^2$ ,可由  $T_1^1$  内所有码字的邻接表组合而成,即  $T_1^2 = T_1^1 \cup T_2^1 \cup T_3^1 \cup T_4^1 \cup T_5^1 \cup T_6^1 \cup T_7^1$ ,由此步骤可依次得到码字  $y_i$  的多个纹波的邻接表。显然码字  $y_i$  的 0 个纹波的邻接表中只有  $y_i$ ,即为  $T_i^0 = \{y_i\}$ 。

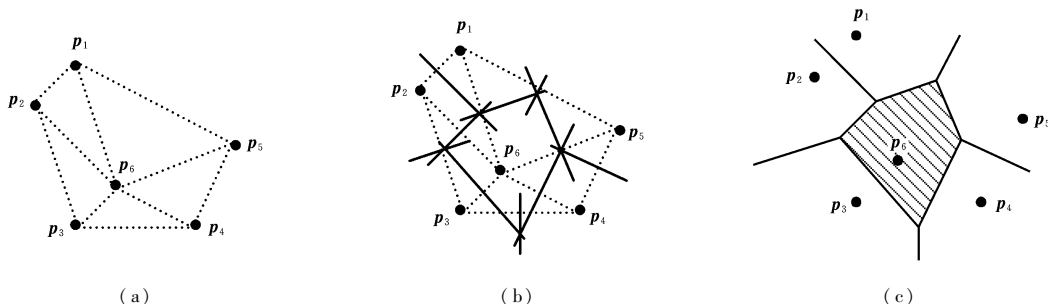


图 3 Voronoi 图的构造过程

Fig. 3 Constructing process of the planar Voronoi diagram

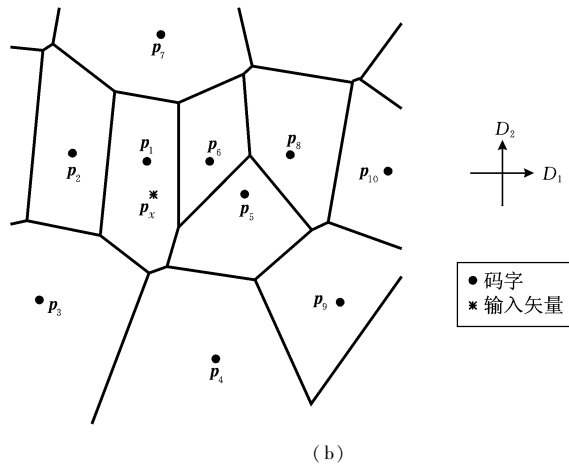
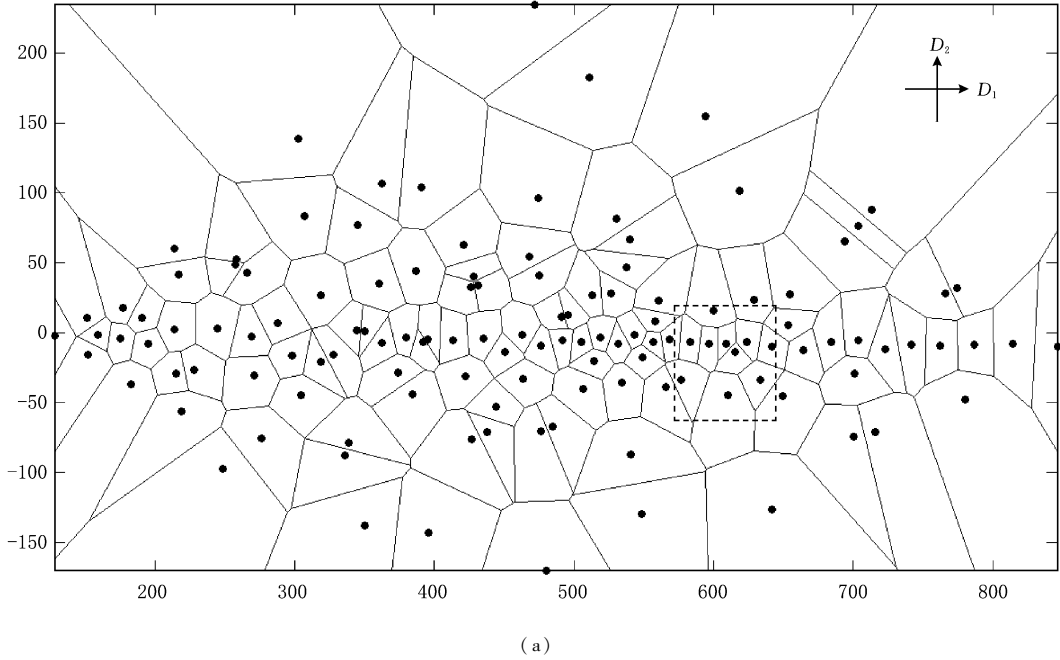


图 4 Voronoi 图示例  
Fig.4 Example of planar Voronoi diagram

下面以图 4(b) 为例来说明 PVDS 算法的码字搜索过程。对于任一输入矢量  $\mathbf{x}$ , 首先利用  $D_1$  轴和  $D_2$  轴将输入矢量  $\mathbf{x}$  投影到点  $\mathbf{p}_x$ , 然后利用快速算法在 2 维 Voronoi 平面上找到与  $\mathbf{p}_x$  最近的点  $\mathbf{p}_1$ , 这样  $\mathbf{p}_1$  对应的码字  $\mathbf{y}_1$  就成为找到的初始匹配码字。对于 0 个纹波,  $\mathbf{y}_1$  就成为找到的最终匹配码字, 对于  $r$  个纹波, 则搜索初始匹配码字  $\mathbf{y}_1$  的  $r$  个纹波的邻接表  $\mathbf{T}_1^r, r \in \{1, 2, \dots, 6\}$  内的所有码字, 找出失真最小的码字作为最终匹配码字, 从而完成对输入矢量  $\mathbf{x}$  的编码过程。

PVDS 算法一般的搜索步骤可描述如下:

离线计算步骤:

- 1) 对码书做 PCA 分析, 得到第 1 主成分轴  $D_1$  和第 2 主成分轴  $D_2$ 。
- 2) 利用  $D_1$  轴和  $D_2$  轴将码字  $\mathbf{y}_i$  投影为 2 维平面的点  $\mathbf{p}_i, i = 0, 1, \dots, N - 1$ 。
- 3) 根据 2 维平面上点  $\mathbf{p}_i$  的分布构造码书的 Voronoi 图。
- 4) 为每个码字  $\mathbf{y}_i$  创建其对应的  $r$  个纹波的邻接表  $\mathbf{T}_i^r$ , 其中  $i = 0, 1, \dots, N - 1, r = 1, 2, \dots, 6$ 。

在线计算步骤:

- 1) 利用  $D_1$  轴和  $D_2$  轴将输入矢量  $\mathbf{x}$  投影为

2 维平面的点  $p_x$ 。

2) 在码书的 Voronoi 图中找到与点  $p_x$  最近的点  $p_{int}$  (initial), 并将  $y_{int}$  作为初始匹配码字, 对于 0 个纹波即将  $y_{int}$  作为最终匹配码字, 然后转向步骤 4), 否则转向步骤 3)。其中  $int \in \{0, 1, \dots, N-1\}$ 。

3) 搜索初始匹配码字  $y_{int}$  邻域表  $T'_{int}$  中的码字, 其中  $r \in \{1, 2, 3, 4, 5, 6\}$ , 并将失真最小的码字作为最终匹配码字, 完成对输入矢量  $x$  的编码过程。

4) 转向下一个输入矢量, 返回步骤 1), 直到完成对所有输入矢量的编码。

## 2 实验比较与分析

为了比较两类码书排序准则对搜索效率的影响, 对 ENNS 算法和 PVDS 算法在各种实验条件下的性能进行了仿真, 并与 IEENNS 算法进行了对比。各类算法的仿真实验均在 Intel P4 主频 2.8 GHz、内存 512 M 的个人计算机上进行。采用  $512 \times 512$  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 和 512, 矢量维数为  $4 \times 4$ 。测试图像为  $512 \times 512$  的 Lena, Boat, Peppers,

Airplane, Baboon 和 GoldHill 图像。

表 1, 表 2 和表 3 分别列出了码书大小在 128, 256 和 512 情况下 FS 算法、ENNS 算法、IEENNS 算法、采用 0 到 6 个纹波的 PVDS 算法 PVDS(0), PVDS(1), ..., PVDS(6) 的搜索范围和编码时间的比较, 其中 0 个纹波表示只以初始匹配码字  $y_{int}$  作为最终匹配码字, 其搜索范围固定为 1。

从表 1—表 3 可以看出, ENNS 算法和 IEENNS 算法的编码质量与 FS 算法是等价的, 而 PVDS 算法中因为只在降维后的 Voronoi 图中采用有限个纹波进行搜索, 最终匹配码字通常不是最佳匹配码字, 因此编码的额外失真不可避免。在极端情况下 PVDS 算法的搜索范围只有覆盖了所有码字时才能保证其编码质量与 FS 算法等价。ENNS 算法和 IEENNS 算法的效率与图像细节有关, 对细节丰富的图像, 搜索范围一般比较大, 编码时间也较长, 而对细节一般的图像搜索范围较小, 编码时间也较少。而 PVDS 算法与图像的细节关系不大, 其搜索范围只取决于所采用的初始匹配码字的纹波数的大小, 对于相同的纹波数, 不同细节的图像将有相似的搜索范围和编码时间。IEENNS 算法是在 ENNS 算

表 1 各算法搜索范围和编码时间的比较 ( $N = 128$ )

Tab.1 Comparison of search space and coding time among the conventional algorithms and the proposed algorithm ( $N = 128$ )

图像	性能	算法									
		FS	ENNS	IEENNS	PVDS(0)	PVDS(1)	PVDS(2)	PVDS(3)	PVDS(4)	PVDS(5)	PVDS(6)
Lena	PSNR/dB	31.529	31.529	31.529	28.201	31.281	31.504	31.518	31.529	31.529	31.529
	搜索范围	128	8.5	4.2	1	6.6	17.6	33.6	54.6	77.6	100.1
	时间/ms	86.3	9.2	9.3	5.2	10.1	15.3	21.9	26.9	32.2	36.3
Boat	PSNR/dB	28.187	28.187	28.187	26.010	27.929	28.153	28.179	28.185	28.187	28.187
	搜索范围	128	11.0	5.6	1	6.6	17.9	34.0	55.5	79.3	102.5
	时间/ms	87.2	11.3	11.5	4.5	9.4	15.0	21.4	28.6	34.5	39.8
Airplane	PSNR/dB	28.960	28.960	28.960	26.750	28.606	28.870	28.923	28.954	28.959	28.960
	搜索范围	128	8.9	4.5	1	6.2	16.5	31.4	54.3	82.0	110.9
	时间/ms	86.9	9.5	9.7	4.5	9.1	14.1	20.0	24.7	31.7	36.9
Peppers	PSNR/dB	29.139	29.139	29.139	26.913	28.837	29.063	29.117	29.126	29.139	29.139
	搜索范围	128	8.5	4.0	1	6.7	18.0	34.3	55.3	79.0	102.9
	时间/ms	87.0	9.2	9.5	5.6	9.6	15.3	22.2	28.0	33.9	38.0
Baboon	PSNR/dB	23.401	23.401	23.401	21.775	23.128	23.349	23.393	23.401	23.401	23.401
	搜索范围	128	25.8	14.0	1	6.9	19.0	36.5	58.391	82.5	104.8
	时间/ms	87.0	23.8	24.0	5.5	11.7	19.5	31.1	42.5	53.1	62.7
GoldHill	PSNR/dB	28.578	28.578	28.578	26.555	28.424	28.558	28.577	28.578	28.578	28.578
	搜索范围	128	12.0	5.9	1	6.6	18.1	34.8	56.1	79.5	103.0
	时间/ms	87.0	11.9	12.0	5.9	11.1	17.0	25.5	32.7	40.3	46.6

表 2 各算法搜索范围和编码时间的比较 ( $N = 256$ )

Tab.2 Comparison of search space and coding time among the conventional algorithms and the proposed algorithm ( $N = 256$ )

图像	性能	算法									
		FS	ENNS	IEENNS	PVDS(0)	PVDS(1)	PVDS(2)	PVDS(3)	PVDS(4)	PVDS(5)	PVDS(6)
Lena	PSNR/dB	32.457	32.457	32.457	28.163	32.040	32.381	32.447	32.451	32.457	32.457
	搜索范围	256	14.9	6.6	1	6.9	18.9	37.2	60.8	88.6	119.1
	时间/ms	173.4	14.3	14.2	5.3	11.4	19.2	27.8	37.2	45.3	53.1
Boat	PSNR/dB	28.835	28.835	28.835	25.932	28.305	28.725	28.811	28.832	28.834	28.835
	搜索范围	256	20.5	9.4	1	6.7	18.7	37.4	61.4	89.0	118.5
	时间/ms	173.1	19.1	19.2	5.3	10.8	18.6	26.9	37.3	47.2	56.6
Airplane	PSNR/dB	29.540	29.540	29.540	26.339	28.904	29.384	29.483	29.517	29.525	29.536
	搜索范围	256	16.2	6.9	1	6.5	17.4	33.0	51.1	73.7	102.2
	时间/ms	173.0	16.4	15.0	4.4	9.7	15.6	23.4	31.4	39.2	46.4
Peppers	PSNR/dB	30.063	30.063	30.063	27.130	29.622	29.952	30.037	30.058	30.062	30.063
	搜索范围	256	14.9	6.3	1	6.9	19.9	39.0	63.0	90.6	121.2
	时间/ms	173.1	15.2	14.5	5.5	10.9	18.9	28.3	38.0	46.4	55.6
Baboon	PSNR/dB	23.797	23.797	23.797	21.679	23.349	23.691	23.774	23.792	23.797	23.797
	搜索范围	256	48.3	24.0	1	7.1	20.3	40.4	66.1	96.0	129.4
	时间/ms	173.4	43.6	41.6	5.9	13.1	23.0	36.4	52.8	70.3	86.7
GoldHill	PSNR/dB	29.655	29.655	29.655	26.590	29.294	29.618	29.648	29.653	29.653	29.655
	搜索范围	256	20.1	9.3	1	7.0	19.5	38.6	62.9	91.4	123.4
	时间/ms	173.4	18.7	18.6	5.9	12.6	20.6	30.9	42.5	54.8	64.1

表 3 各算法搜索范围和编码时间的比较 ( $N = 512$ )

Tab.3 Comparison of search space and coding time among the conventional algorithms and the proposed algorithm ( $N = 512$ )

图像	性能	算法									
		FS	ENNS	IEENNS	PVDS(0)	PVDS(1)	PVDS(2)	PVDS(3)	PVDS(4)	PVDS(5)	PVDS(6)
Lena	PSNR/dB	33.619	33.619	33.619	27.885	32.739	33.371	33.525	33.579	33.613	33.619
	搜索范围	512	26.4	10.6	1	7.1	20.0	40.1	67.6	102.6	144.5
	时间/ms	343.4	24.5	22.3	6.6	13.0	20.8	33.6	49.1	63.4	78.0
Boat	PSNR/dB	29.332	29.332	29.332	25.778	28.540	29.098	29.264	29.301	29.316	29.327
	搜索范围	512	37.4	15.8	1	7.2	20.5	41.1	69.2	104.6	146.4
	时间/ms	343.6	34.5	30.9	6.8	12.8	21.7	34.2	50.6	65.9	81.6
Airplane	PSNR/dB	29.944	29.944	29.944	26.334	29.089	29.645	29.794	29.864	29.889	29.912
	搜索范围	512	29.5	14.4	1	7.0	18.8	36.9	60.9	91.4	130.0
	时间/ms	343.3	27.5	23.3	5.8	12.0	18.6	30.5	42.8	55.2	66.7
Peppers	PSNR/dB	30.580	30.580	30.580	26.936	29.851	30.277	30.435	30.491	30.536	30.555
	搜索范围	512	27.8	10.6	1	7.2	21.4	42.8	71.3	107.8	150.0
	时间/ms	343.5	25.6	22.8	6.4	12.8	22.2	35.6	53.0	68.8	83.8
Baboon	PSNR/dB	24.251	24.251	24.251	21.748	23.505	23.943	24.134	24.204	24.234	24.245
	搜索范围	512	91.3	43.1	1	7.3	21.3	43.4	73.5	111.4	156.6
	时间/ms	343.6	81.7	74.4	7.2	15.5	25.5	44.7	66.9	91.6	119.1
GoldHill	PSNR/dB	30.293	30.293	30.293	26.562	29.700	30.151	30.262	30.287	30.291	30.293
	搜索范围	512	38.3	17.0	1	7.1	20.8	42.6	72.2	109.8	155.0
	时间/ms	343.5	33.7	30.5	8.9	14.4	22.8	37.3	56.4	74.8	93.6

法的基础上引入一个新的方差特征量而得到的改进算法,在码书较大的情况下能进一步提高码字搜索的效率,以 Lena 图像为例,码书大小为 512 时,IEENNS 算法的搜索范围和编码时间分别减小到 ENNS 算法的 40.2% 和 91.0%。

对于细节一般的图像,ENNS 算法和 IEENNS 算法的编码时间只大致对应于 PVDS 算法的 1~3 个纹波,对于细节丰富的 Baboon 图像,对应的纹波数也只为 3~5 个。在离线计算量和存储方面,ENNS 算法只需要计算并存储各码字的均值。PVDS 算法则需要离线对码书做 PCA 分析,构造码书的 Voronoi 图,创建并存储每个码字的邻接表。ENNS 算法和 IEENNS 算法实际表示了码字排列呈 1 维的顺序关系,而 PVDS 算法表示了码字在 2 维平面上呈相邻关系。

综上所述,码书排序呈 1 维顺序关系的算法,码书首先离线按 1 维特征量进行排序,然后根据基于特征量的码字删除准则排除不匹配码字,确定一个较小的搜索范围,然后在减小的搜索范围内寻找最佳匹配码字,所以其编码质量同 FS 算法等且编码时间较少,离线计算简单。在码书排序呈 1 维顺序关系的算法的基础上,引入其他 1 维特征量可以更有效地改进算法,删除更多不匹配码字,进一步提高编码效率。码书排序呈 2 维相邻关系的算法,需要离线得到码书中每个码字的邻接关系,离线计算过程不仅复杂而且需要较大的存储空间,且在码字搜索过程中没有建立一个能够保证获得最佳匹配的码字删除准则,而只是在 2 维平面内搜索初始匹配码字的有限个纹波内的邻接码字,这样为了得到较好的编码质量就要牺牲编码时间;反之,为了使编码时间较少就得损失编码质量。根据以上分析,在实际使用快速搜索算法的过程中建议优先考虑码书排序呈 1 维顺序关系的算法,而基于 2 维相邻关系的码书排序则没有明显的实用意义。

### 3 结 论

总结了两大类码书排序呈 1 维顺序关系与 2 维相邻关系的快速搜索算法,比较了它们的搜索范围和编码时间。总的来说,无论在编码质量还是编码时间方面,码书排序呈 1 维顺序关系的算法要比码书排序呈 2 维相邻关系的算法更加有效,特别是码书

排序呈 2 维相邻关系的算法不仅离线计算非常复杂,还需要很大的额外存储,因此基于 2 维相邻关系的码书排序没有明显的实际意义。在今后开发新的快速码字搜索算法的过程中,码书排序的准则应优先考虑对码书按码字的 1 维特征值进行排序,以便减小系统的复杂度,同时保证重建图像的质量并更有效地减小搜索范围,提高搜索效率。

### 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] Bei C D, Gray R M. An improvement of the minimum distortion encoding algorithm for vector quantization [J]. IEEE Transactions on Communications, 1985, 33(10): 1132-1133.
- [3] Guan L, Kamel M. Equal-average hyperplane partitioning method for vector quantization of image data [J]. Pattern Recognition Letters, 1992, 13(10): 693-699.
- [4] Lee C H, Chen L H. Fast closest codeword search algorithm for vector quantization [J]. IEE Processings-Vision, Image and Signal Processing, 1994, 141(3): 143-148.
- [5] Baek S J, Jeon B K, Sung K M. A fast encoding algorithm for vector quantization [J]. IEEE Signal Processing Letters, 1997, 4 (12): 325-327.
- [6] Lu Z M, Sun S H. Equal-average equal-variance equal-norm nearest neighbor search algorithm for vector quantization [J]. IEICE Transactions on Information and Systems, 2003, E86-D (3): 660-663.
- [7] Pan Z, Kotani K, Ohmi T. Performance comparison between equal-average equal-variance equal-norm nearest neighbor search (EEENNS) method and improved equal-average equal-variance nearest neighbor search (IEENNS) method for fast encoding of vector quantization [J]. IEICE Transactions on Information and Systems, 2005, E88-D(9): 2218-2222.
- [8] Pan Z, Kotani K, Ohmi T. A unified projection method for fast search of vector quantization [J]. IEEE Signal Processing Letters, 2004, 11(7): 637-640.
- [9] Huang C M, Bi Q, Stiles G S, et al. Fast full search equivalent encoding algorithm for image compression using vector quantization [J]. IEEE Transactions on Image Processing, 1992, 1(3): 413-416.
- [10] Lee C H, Chen L H. High-speed closest codeword search algorithms for vector quantization [J]. Signal Process, 1995, 43 (3): 323-331.
- [11] Chang C C, Wu W C. Fast planar-oriented ripple search algorithm for hyperspace VQ codebook [J]. IEEE Transaction on Image Processing, 2007, 16(6): 1538-1547.