

中图分类号: TN919.8 文献标志码: A 文章编号: 1006-8961(2011)09-1592-07

论文索引信息: 吴鑫鹏, 潘志斌, 乔瑞萍, 禹贵辉. 基于改进的参考矢量选取的码字快速搜索算法 [J]. 中国图象图形学报, 2011, 16(9): 1592-1598

基于改进的参考矢量选取的码字快速搜索算法

吴鑫鹏, 潘志斌, 乔瑞萍, 禹贵辉

(西安交通大学电子与信息工程学院, 西安 710049)

摘要: 矢量量化的编码过程计算复杂性极高, 为了减少编码时大量的矢量间距离计算, 许多文献提出利用不等式关系以较少的计算量来估算距离的方法。在 Chang 等人提出的利用双限制三角不等式的快速搜索算法基础上, 通过改进参考矢量的选取方法, 有效提高了码字搜索的效率。实验结果表明, 改进算法的码字排除率可以提高 3.735%~9.976%, 编码时间可以减少 6.03%~35.25%。

关键词: 矢量量化; 码字快速搜索; 参考矢量; 编码性能

Improved fast codeword search algorithm based on reference vector selection

Wu Xinpeng, Pan Zhibin, Qiao Ruiping, Yu Guihui

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

Abstract: The encoding process of vector quantization (VQ) is computational expensive. To reduce the distance computations between vectors, many methods use inequality relations to make distance estimations with less computational cost. This paper aims at extending the method using two-bounds triangle inequality proposed by Chang C C et al by improving reference the vector selection. The experimental results show that our proposed method can increase the ratio of codeword rejection by 3.735%~9.976%, and reduce the encoding time by 6.03%~35.25%.

Keywords: vector quantization; fast codeword search; reference vector; encoding performance

0 引言

矢量量化(VQ)^[1]作为一种有效的有损压缩技术,因其压缩比大且编解码简单等特点,已成为数据压缩编码的重要技术之一,被广泛应用于图像和语音信号处理等领域。矢量量化的编码过程就是在给定的码书 $C = \{u_i \in R^k, i = 0, 1, \dots, N-1\}$ 中寻找与输入矢量 $v = (v_0, v_1, \dots, v_{k-1})^T$ 最匹配的码字 $u_w = (u_{w0}, u_{w1}, \dots, u_{w(k-1)})^T$ (winner codeword), 从而

使 v 和 u_w 之间的失真最小。失真通常采用欧氏距离来衡量,即

$$d(v, u_i) = \sqrt{\sum_{l=0}^{k-1} (v_l - u_{il})^2} \quad (1)$$

那么最佳匹配码字为

$$d(v, u_w) = \min_{u_i \in C} d(v, u_i) \quad (2)$$

式中, u_w 是与输入矢量 v 最匹配的码字, w 即为要传输的索引。

如果采用穷尽搜索算法 (full search, FS), 设码书大小为 N , 那么对每个输入矢量 v 进行编码, 需要

收稿日期: 2010-08-10; 修回日期: 2010-12-15

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); 高校博士点专项基金项目 (20100201110030)。

第一作者简介: 吴鑫鹏 (1987—), 男, 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编码。

E-mail: sinpose@163.com。

计算 Nk 次乘法、 $N(2k - 1)$ 次加法和 $N - 1$ 次比较运算,其计算量巨大,严重限制了矢量量化技术的实际应用和进一步发展,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。

FS 算法的计算复杂度由码书尺寸和矢量维数决定,其复杂度主要集中在欧氏距离的计算上,为了减小 FS 搜索算法的计算量,许多文献提出了各种快速算法^[2-12]。很多快速算法是基于矢量的 1 维特征量的,文献[2]提出使用均值特征量,文献[3]提出使用 2 范数特征量,文献[4]提出使用方差特征量,而角度特征量在文献[5]中开始使用。利用 1 维特征量来删除不匹配码字可以极大减小编码过程中计算欧氏距离的计算负担,但离线计算并保存每个码字的特征量,需要大量的存储空间。此外,文献[6]提出了利用投影的方法,文献[7-8]则提出利用余弦定理的方法,但都只使用了三角不等式的一个边界条件,在文献[9]中则提出一种利用双限制三角不等式的改进方法,进一步有效提高了搜索性能。此外,文献[10]提出利用双测试条件加快图像编码速度的方法,文献[11]通过简化模式减少编码时间,而文献[12]则提出利用变换后的码书的编码算法,这些都有效地提高了搜索性能。

1 已有工作

1.1 利用双限制三角不等式的码字快速搜索算法

文献[9]提出一种利用双限制三角不等式的方法。令 $u_i (i = 0, 1, \dots, N - 1)$ 是码书 C 中的任一码字,其中 N 是码书大小, v 是任一输入矢量。如果直接计算 v, u_i 之间实际的欧氏距离 $d(v, u_i)$ 进行搜索,那么在线计算量很大。因此,可引入参考矢量 x ,以较小的在线计算量间接对 $d(v, u_i)$ 进行估计,那么在由 k 维空间 3 个矢量 x, v, u_i 构成的如图 1 所示的 Δxvu_i 中,存在如下三角不等式关系

$$|d(x, v) - d(x, u_i)| \leq d(v, u_i) \leq d(x, v) + d(x, u_i) \quad (3)$$

可见, v, u_i 之间实际的欧氏距离 $d(v, u_i)$ 分别由上界 $d(x, v) + d(x, u_i)$ 和下界 $|d(x, v) - d(x, u_i)|$ 所限制,为了找到与输入矢量 v 最匹配的码字,这里需要找出 $\min_{u_i \in C} [d(v, u_i)]$, 根据式(3)可得

$$\min_{u_i \in C} |d(x, v) - d(x, u_i)| \leq \min_{u_i \in C} d(v, u_i) \leq \min_{u_i \in C} [d(x, v) + d(x, u_i)] \quad (4)$$

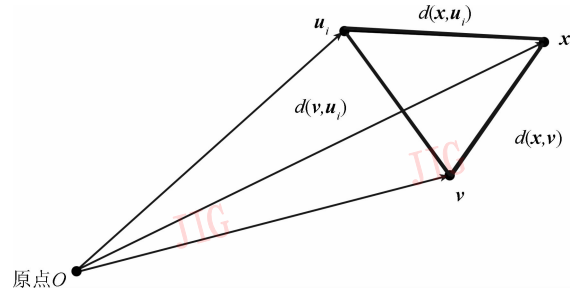


图 1 参考矢量 x 与 v, u_i 构成的 Δxvu_i

Fig. 1 Triangle Δxvu_i formed by reference vector x and v, u_i

为了减少欧氏距离的在线计算量,对欧氏距离下界 $d(v, u_i)$ 按下式估算

$$d_{++}(v, u_i) = |d(x, v) - d(x, u_i)| \quad (5)$$

设码书大小为 N , 选定参考矢量 x 后,对于任一给定的输入矢量 v ,在码书 C 中搜索最佳匹配码字时,式(5)需要计算 N 次 $d(x, u_i)$ 和一次 $d(x, v)$,但所有 $d(x, u_i)$ 可以离线处理,因此对当前码字 u_i 利用下界 $d_{++}(v, u_i)$ 进行能否排除的测试,每个码字平均只需要一次减法和一次绝对值运算。而对于 $d(v, u_i)$ 的直接计算,则每个码字都需要在线计算一次欧氏距离。由此可见,通过引入参考矢量 x ,可大大减少欧氏距离的在线计算,从而降低了编码时的计算复杂度。但离线计算 $d(x, u_i)$ 的结果需要预先保存起来,会增加 N 个单元的存储空间开销。

算法流程如下:

1) 创建大小为 N 的码书 $C = \{u_i \in R^k, i = 0, 1, \dots, N - 1\}$, 并选定一个参考矢量 x 。

2) 对码书 C 中所有码字 u_i , 离线计算 $d(x, u_i)$, 并保存。

3) 对每一个输入矢量 v :

(1) 在线计算 $d(x, v)$

(2) 确定 $d_{\min} = \min_{u_i \in C} [d(x, v) + d(x, u_i)]$ 及初始的 $d_{\min} = \min_{u_i \in C} [d(v, u_{\text{lower}}), d(v, u_{\text{upper}})]$, 其中 u_{lower} 由 $|d(x, v) - d(x, u_{\text{lower}})| = \min_{u_i \in C} |d(x, v) - d(x, u_i)|$ 确定, u_{upper} 由 $d(x, v) + d(x, u_{\text{upper}}) = \min_{u_i \in C} [d(x, v) + d(x, u_i)]$ 确定。

(3) 对码书 C 中每个码字 u_i , 计算下界 $d_{++}(v, u_i) = |d(x, v) - d(x, u_i)|$ 。

①若 $d_{++}(v, u_i) > d_{\min}$, 则排除码字 u_i 。

②若 $d_{++}(v, u_i) > d_{\min}$, 则排除码字 u_i 。

③否则计算实际的欧氏距离 $d(v, u_i)$, 如果

$d(\mathbf{v}, \mathbf{u}_i) < d_{\min}$, 那么进行更新 $d_{\min} \leftarrow d(\mathbf{v}, \mathbf{u}_i)$ 。

上述算法可以大幅度地减少计算量和编码时间,但也存在两个不足之处:1)该算法引入了参考矢量 \mathbf{x} , 但并没有给出 \mathbf{x} 的具体选择方法,文献[9]在实验中只取 $\mathbf{x} = (0, 0, \dots, 0)$, 相当于取坐标原点,即相当于实际上没有选取参考矢量,此时式(5)等效为 $d_{++}(\mathbf{v}, \mathbf{u}_i) = \left| \|\mathbf{v}\| - \|\mathbf{u}_i\| \right|$, 完全退化为等范数最近邻搜索算法^[3];2)该法对初始 d_{\min} 的选取,需要确定 $|d(\mathbf{x}, \mathbf{v}) - d(\mathbf{x}, \mathbf{u}_i)|$ 取最小值时对应的码字 $\mathbf{u}_{\text{lower}}$ 以及 $[d(\mathbf{x}, \mathbf{v}) + d(\mathbf{x}, \mathbf{u}_i)]$ 取最小值时对应的码字 $\mathbf{u}_{\text{upper}}$, 再将 $d(\mathbf{v}, \mathbf{u}_{\text{lower}})$ 与 $d(\mathbf{v}, \mathbf{u}_{\text{upper}})$ 两者之间的较小值作为初始的 d_{\min} 。确定 $\mathbf{u}_{\text{upper}}$ 需要 N 次加法和 $N - 1$ 次比较,而对于 $\mathbf{u}_{\text{lower}}$ 的确定,除 N 次加法和 $N - 1$ 次比较外还需要 N 次绝对值运算。进一步求 $d(\mathbf{v}, \mathbf{u}_{\text{lower}})$ 与 $d(\mathbf{v}, \mathbf{u}_{\text{upper}})$ 之间的较小值则还需要两次欧氏距离计算和一次比较。由于对每个输入矢量 \mathbf{v} 都需要重复这样的步骤,过程非常复杂,明显增加了算法的额外计算量,考虑到码字搜索过程中对 d_{\min} 的不断更新,这种初始化 d_{\min} 的计算实际可用更简单有效的方法实现。

1.2 等均值最近邻搜索算法 (ENNS)^[3]

定理 假定搜索过程中当前已经找到的最小失真为 d_{\min} , m_i 为码字 \mathbf{u}_i 的均值, m_v 为输入矢量 \mathbf{v} 的均值, k 是码字维数。如果

$$|m_i - m_v| \geq \sqrt{d_{\min}/k} \quad (6)$$

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

通常可利用式(6)测试码字 \mathbf{u}_i 是否能被排除,避免欧氏距离 $d(\mathbf{v}, \mathbf{u}_i)$ 的直接计算。

2 本文算法

矢量量化的编码过程计算量巨大,为了减少计算量,通常不直接计算欧氏距离,而采用其他公式进行估算,这里采用式(5)对欧氏距离的下界进行估计。由式(3)可知 $d_{++}(\mathbf{v}, \mathbf{u}_i) \leq d(\mathbf{v}, \mathbf{u}_i)$, 所以当 $d_{++}(\mathbf{v}, \mathbf{u}_i) > d_{\min}$ 成立时,必有 $d(\mathbf{v}, \mathbf{u}_i) > d_{\min}$, 那么不必计算 $d(\mathbf{v}, \mathbf{u}_i)$ 即可排除码字 \mathbf{u}_i 。为了更多的排除不可能的码字,我们希望 $d_{++}(\mathbf{v}, \mathbf{u}_i)$ 的值尽可能大,从而使其更接近真实的欧氏距离 $d(\mathbf{v}, \mathbf{u}_i)$ 。根据式(5),由于 \mathbf{u}_i 和 \mathbf{v} 在 \mathbf{R}^k 空间相互独立且随机分布,因此难以同时兼顾两者达到全局最优的效果。为便于离线操作,本文仅考虑 $d(\mathbf{x}, \mathbf{u}_i)$ 部分,而将

$d(\mathbf{x}, \mathbf{v})$ 作为随机部分,如果能使 $d(\mathbf{x}, \mathbf{u}_i)$ 取值越大或越小,那么从统计意义上分析^[8], $d_{++}(\mathbf{v}, \mathbf{u}_i)$ 的取值也会相应变得越大。

$d_{++}(\mathbf{v}, \mathbf{u}_i)$ 的最大取值是 $d(\mathbf{v}, \mathbf{u}_i)$, 此时, \mathbf{x} 位于线段 $\overline{\mathbf{v}\mathbf{u}_i}$ 的延长线上。两种极限情况是 $\mathbf{x} = \mathbf{v}$ 或者 $\mathbf{x} = \mathbf{u}_i$ 。前者由于输入矢量 \mathbf{v} 是随机且未知的,因此无法据 \mathbf{v} 来设定参考矢量 \mathbf{x} ;于是可令 $\mathbf{x} = \mathbf{u}_i$, 则 $d(\mathbf{x}, \mathbf{u}_i) = 0$, 那么 $d_{++}(\mathbf{v}, \mathbf{u}_i) = d(\mathbf{v}, \mathbf{x}) = d(\mathbf{v}, \mathbf{u}_i)$, 式(5)退化成欧氏距离计算公式,但是这样,对码书 \mathbf{C} 中的每个码字 \mathbf{u}_i , 都需要单独地确定与之对应的参考矢量 \mathbf{x} , 会增加 $d(\mathbf{x}, \mathbf{v})$ 的计算。因此参考矢量选得越多, $d_{++}(\mathbf{v}, \mathbf{u}_i)$ 的值越可能接近于 $d(\mathbf{v}, \mathbf{u}_i)$, 有利于排除更多不可能的码字,但相应的计算复杂度会增大;参考矢量选得较少,计算量会减少,但 $d_{++}(\mathbf{v}, \mathbf{u}_i)$ 的值也会变小,排除的不可能码字也会相应减少。为了折中计算量和使 $d_{++}(\mathbf{v}, \mathbf{u}_i)$ 取值较大的关系,必须选取最合适的参考矢量。因此可以通过对码书进行排序,将空间分布相近的码字划为一段,取各段码书的质心作为对应的参考矢量 \mathbf{x} 。如果假设一段码书中的码字全都相同,取其质心即码字本身作为参考矢量,则 $d(\mathbf{x}, \mathbf{u}_i) = 0$, $d_{++}(\mathbf{v}, \mathbf{u}_i)$ 取到最大值 $d(\mathbf{v}, \mathbf{u}_i)$ 。一般情况,一段码书中的码字并不相等,但经过码书排序以后,可使得相邻码字的空间分布相近,那么质心就可以很好地表征一段码书中邻近码字的整体特征。

如图2所示,这里对码书均匀分段,设码书被分成 l 段,每段包含 m 个码字,则有 $l = N/m$, 其中 N 为码书大小。在每段码书分段中分别设定一个与之对应的参考矢量 \mathbf{x}_k , 由下式给出

$$\mathbf{x}_k = \frac{1}{m} \sum_{i=(k-1)m}^{k \times m - 1} \mathbf{u}_i \quad k = 1, 2, \dots, l \quad (7)$$

搜索采用上下往返搜索算法,即在初始码书段两侧往复搜索,如图2所示。设当前搜索所在的段为第 k 段,则选取 \mathbf{x}_k 作为当前参考矢量,若当前段已搜索完毕,则搜索流程跳转至第 $k - 1$ 段,相应的参考矢量变为 \mathbf{x}_{k-1} 。若 $k - 1$ 段也已搜索完毕,则搜索流程跳转至第 $k + 1$ 段,相应的参考矢量变为 \mathbf{x}_{k+1} , 依次继续,直到某个方向满足式(6)的 ENNS 算法的结束条件,则停止该方向的搜索。

本文算法如下,其中步骤1)2)为离线处理,步骤3)为在线处理:

1)对码书按均值进行升序排序后,将码书等分为 l 段,求每段所含 m 个码字的质心,并保存作为该

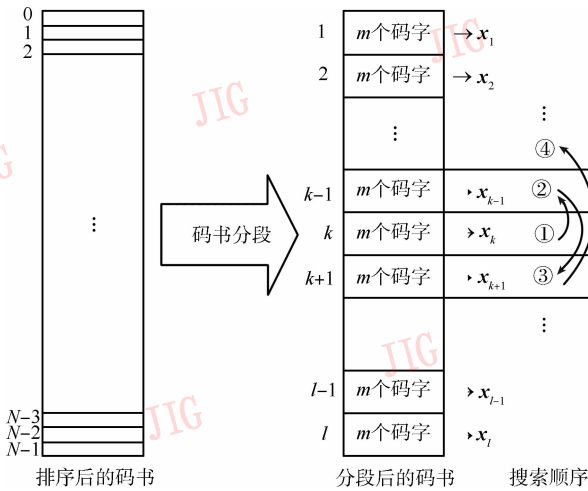


图2 码书分段、对应参考矢量 x_k 及搜索顺序

Fig.2 Segmentation of codebook, corresponding reference vector x_k and search order

段对应的参考矢量 $x_k (k = 1, 2, \dots, l)$ 。

2) 对码书 C 中所有码字 u_i , 根据分段情况找到其对应的参考矢量 x_k , 离线计算 $d(x_k, u_i), i = 0, 1, \dots, N - 1$, 并保存。

3) 对每个输入矢量 v :

(1) 计算 v 的均值, 并在分段后的码书中找到与其均值最接近的码字 u_i 。

(2) 计算 $d(v, u_i)$, 作为初始的 d_{\min} 。

(3) 确定 u_i 所在的码字段及其对应的参考矢量 x_k 。

(4) 计算 $d(x_k, v)$ 。

(5) 先在 u_i 所在码字段内进行搜索, 计算 $d_{++}(v, u_i) = |d(x_k, v) - d(x_k, u_i)|, i = (k - 1) \times m, \dots, k \times m - 1$:

①若 $d_{++}(v, u_i) > d_{\min}$, 则排除码字 u_i 。

②否则计算 $d(v, u_i)$, 如果 $d(v, u_i) < d_{\min}$, 则进行更新 $d_{\min} \leftarrow d(v, u_i)$ 。

(6) 如果搜索范围跳出 u_i 所在的码字段, 则参考矢量 x 也随之切换成跳转到的新的码字段所对应的 x_k , 重复步骤(4)~(5)进行搜索。当满足式(6)的 ENNS 算法结束条件时, 停止搜索。

3 实验结果

本文算法的仿真实验在 Intel Core2 主频 3.00 GHz、内存 2 GB 的个人计算机上进行。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、Peppers、Airplane 和 Baboon 图像, 如图3所示。

本文算法的关键在于码书分段数 l 的选取。 l 取值偏大, 即分的段数较多, 则相应的选择的参考矢量 x 的数目也增多, 计算量及编码时间也相应增加, 极端情况下, 取 $l = N$, 这样对码书 C 中的每个码字 u_i , 都需要单独地确定与之对应的参考矢量 x , 会增加 $d(x, v)$ 的计算; l 取值偏小, 即分的段数较少, 则一些空间分布差距较大的码字也会被分到一段中, 所选参考矢量可能与该码字段中某些矢量差距较大, 从而使 $d(x_i, u_i)$ 偏大, 导致 $d_{++}(v, u_i)$ 偏小, 不利于码字的排除, 极端情况下, 取 $l = 1$, 即将码书 C 的质心作为参考矢量 x , 由于码书 C 中的码字空间分布极广, 单一的参考矢量显然无法很好地表征码书中所有码字的特征。表1—4 分别是码书为 128、256、512 和 1 24 时分段数取不同值, 对 Lena 图的测试结果比较。

由前文的分析知, 若码书分的段数较多, 则相应的选择的参考矢量的数目也增多, 计算量及编码时间也相应增加; 而如果分的段数较少, 则一些空间分布差距较大的码字也会被分到一段中, 所选参考矢量可能与该码字段中某些矢量差距较大, 不利于码字的排除。因此, 对码书分段数目的选取将直接影

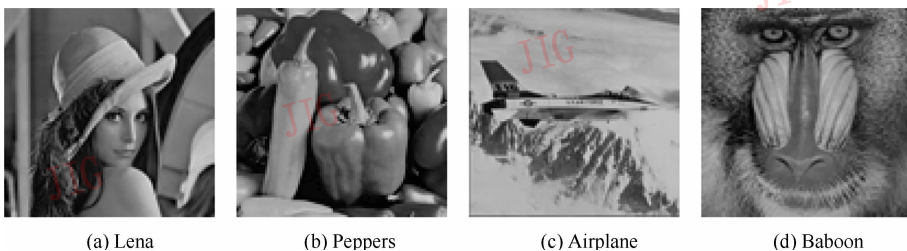


图3 测试图像

Fig.3 Test images

表 1 码书分段数不同时平均每个输入矢量的搜索性能比较 ($N = 128$)Tab. 1 Comparison of search performances of different codebook segmentations ($N = 128$)

码书分段数	加法	乘法	比较	开方	时间/ms	码字排除率/%	搜索段数
1 段	285	138	7	1	18.13	95.526	1
2 段	280	136	7	1.08	17.97	95.714	1.08
4 段	268	129	8	1.21	17.34	96.120	1.21
8 段	266	128	8	1.45	17.50	96.357	1.45
16 段	282	136	8	1.92	18.44	96.328	1.92
32 段	316	154	8	2.94	20.62	96.251	2.94

表 2 码书分段数不同时平均每个输入矢量的搜索性能比较 ($N = 256$)Tab. 2 Comparison of search performances of different codebook segmentations ($N = 256$)

码书分段数	加法	乘法	比较	开方	时间/ms	码字排除率/%	搜索段数
1 段	457	227	10	1	28.91	95.795	1
2 段	441	219	10	1.08	28.12	96.016	0.18
4 段	410	202	11	1.20	26.72	96.465	1.20
8 段	393	194	12	1.44	25.94	96.772	1.44
16 段	402	198	12	1.93	26.57	96.848	1.93
32 段	435	215	12	2.86	28.75	96.791	2.86

表 3 码书分段数不同时平均每个输入矢量的搜索性能比较 ($N = 512$)Tab. 3 Comparison of search performances of different codebook segmentations ($N = 512$)

码书分段数	加法	乘法	比较	开方	时间/ms	码字排除率/%	搜索段数
1 段	756	381	15	1	47.03	96.183	1
2 段	719	362	16	1.07	45.47	96.428	1.07
4 段	654	328	18	1.18	42.19	96.861	1.18
8 段	610	305	20	1.40	40.16	97.183	1.40
16 段	607	304	20	1.85	40.00	97.291	1.85
32 段	632	317	20	2.74	41.87	97.304	2.74
64 段	701	352	20	4.47	45.94	97.212	4.47

表 4 码书分段数不同时平均每个输入矢量的搜索性能比较 ($N = 1\ 024$)Tab. 4 Comparison of search performances of different codebook segmentations ($N = 1\ 024$)

码书分段数	加法	乘法	比较	开方	时间/ms	码字排除率/%	搜索段数
1 段	1 301	661	22	1	80.00	96.524	1
2 段	1 242	631	24	1.06	77.34	96.717	1.06
4 段	1 113	564	29	1.16	71.10	97.132	1.16
8 段	996	504	33	1.36	65.15	97.520	1.36
16 段	960	485	34	1.78	63.29	97.674	1.78
32 段	968	489	35	2.57	64.22	97.726	2.57
64 段	1 027	520	34	4.15	67.81	97.694	4.15

响到改进算法的性能。从表 1—4 的结果可以看出, 128 的码书分成 4 段效果最好, 256 的码书分成 8 段效果最好, 而 512 和 1 024 的码书则在分成 16 段时获得最佳效果。这是由于较大的码书, 所含的细节信息也越多, 只有增大分段数, 才能将不同特征的码

字区分开。在后面的实验中将按以上参数划分码书段落。

表 5 给出的是改进算法与原算法^[9]码字排除率的比较, 表 6 给出的是参考矢量 x 改进前后编码时间的比较。

表5 平均每个输入矢量码字排除率的比较

Tab.5 Comparison of rejection rates of original method and improved method

码书大小	图像	/%		
		文献[9] 算法	改进算法	排除率提高
128	Lena	91.558	96.120	4.562
	Peppers	91.706	96.329	4.623
	Airplane	91.830	95.642	3.812
	Baboon	77.552	85.181	7.629
256	Lena	92.360	96.772	4.412
	Peppers	92.691	96.980	4.289
	Airplane	92.459	96.712	4.253
	Baboon	78.971	87.604	8.633
512	Lena	93.099	97.291	4.192
	Peppers	93.146	97.318	4.172
	Airplane	93.092	97.189	4.097
	Baboon	79.996	89.026	9.03
1 024	Lena	93.939	97.674	3.735
	Peppers	93.843	97.615	3.772
	Airplane	93.429	97.566	4.137
	Baboon	81.007	90.983	9.976

表6 图像编码时间的比较

Tab.6 Comparison of the encoding time

码书大小	图像	编码时间/ms		减少时间/ms	减少百分比/%
		文献[9]算法	改进算法		
128	Lena	18.59	17.34	1.25	6.72
	Peppers	18.43	16.87	1.56	8.46
	Airplane	19.53	18.12	1.41	7.22
	Baboon	49.38	46.40	2.98	6.03
256	Lena	30	25.94	4.06	13.53
	Peppers	30.15	24.84	5.31	17.61
	Airplane	32.81	26.10	6.71	20.45
	Baboon	90.79	77.03	13.76	15.16
512	Lena	50.00	40.00	10.00	20.00
	Peppers	52.5	39.85	12.65	24.10
	Airplane	56.25	41.09	15.16	26.95
	Baboon	169.07	134.22	34.85	20.61
1 024	Lena	85.15	63.29	21.86	25.67
	Peppers	93.44	64.85	28.59	30.60
	Airplane	102.81	66.57	36.24	35.25
	Baboon	316.72	218.75	97.97	30.93

从表5和表6的实验结果看,与原算法^[9]相比,改进算法在码字排除率和编码时间方面均有明显改善。在码字排除率方面,Baboon效果最明显,有7.629%~9.976%,其余测试图像结果较接近,在3.735%~4.623%之间。而在编码时间方面,码书越大,编码时间的减少越显著。码书大小为128时编码时间可减少6.03%~8.46%,而码书大小为1024时,编码时间可减少25.67%~35.25%。实验结果证明了本文算法的有效性。

4 结论

本文是在文献[9]提出算法基础上所做的改进,目的是通过对参考矢量 \mathbf{x} 的最佳选取,进一步提高码字搜索的效率。本文算法的改进主要体现在两个方面:1)对参考矢量的合理选取,通过对码书按均值进行排序后等分成若干段,再将每段码书的质心作为各段对应的参考矢量;2)对初始 d_{\min} 的选取,这里将输入矢量 \mathbf{v} 和与其均值最接近的码字 \mathbf{u}_i 之间的距离 $d(\mathbf{v}, \mathbf{u}_i)$ 作为初始 d_{\min} 。从最终的实验结果看,与原文献算法^[1]相比,本文算法的码字排除率可以提高3.735%~9.976%,编码时间可以减少6.03%~35.25%。

参考文献(References)

- [1] 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.]
- [2] Guan L, Kamel M. Equal-average hyperplane partitioning method for vector quantization of image data [J]. Pattern Recognition Letters, 1992, 13(10): 693-699.
- [3] 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.
- [4] Lee C H, Chen L H. Fast closest codeword search algorithm for vector quantization [J]. IEEE Processings Vision, Image and Signal Processing, 1994, 141(3): 143-148.
- [5] Swilem A, Imamura K, Hashimoto H. A high-speed codebook design algorithm for ECVQ using angular constraint with search space partitioning [C] // International Conference on Multimedia and Expo. Taiwan, China; IEEE Press, 2004: 371-374.
- [6] 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.
- [7] Mielikainen J. A novel full-search vector quantization algorithm based on the law of cosines [J]. IEEE Signal Processing Letters, 2002, 9(6): 175-176.
- [8] Pan Z, Kotani K, Ohmi T, An improved full-search-equivalent vector quantization method using the law of cosines [J]. IEEE Signal Processing Letters, 2004, 11(2): 247-250.
- [9] Chang C C, Chen C C. Full-searching-equivalent vector quantization method using two-bounds triangle inequality [J]. Fundamenta Informaticae, 2007, 76(2007): 25-37.
- [10] Hu Y C, Su B H, Tsou C C. Fast VQ codebook search algorithm for grayscale image coding [J]. Image and Vision Computing, 2008, 26(5): 657-666.
- [11] Lee C Y, Tsai C W, Chiang M C, et al. A fast VQ codebook generation algorithm via pattern reduction [J]. Pattern Recognition Letters, 2009, 30(7): 653-660.
- [12] Lai J Z C, Liaw Y C. A novel encoding algorithm for vector quantization using transformed codebook [J]. Pattern Recognition, 2009, 42(11): 3065-3070.