

一种矢量量化码书搜索的快速算法

陆哲明¹, 潘正祥^{1,2}, 孙圣和¹

(1. 哈尔滨工业大学自动化测试与控制系, 哈尔滨 150001; 2. 高雄科学技术学院电子工程系, 台湾高雄)

摘 要: 本文提出了一种采用均方误差(MSE)测度的矢量量化码书搜索的快速算法. 该算法在码书设计的每次迭代前预先计算各码字的和值(一个矢量各分量的和)并保存在码书中. 在迭代过程中, 利用输入矢量的和值、各码字的和值以及均方误差三者之间的各种特性排除大部分候选码字而免去许多均方误差计算. 测试结果表明, 相对于穷尽搜索方法, 计算量得到明显的降低, 计算时间减少约 90%, 同时只需要很少的预先计算量和额外存储量.

关键词: 均方误差; 矢量量化

中图分类号: TN919.31 **文献标识码:** A **文章编号:** 0372-2112 (2000) 02-0133-03

A Fast Codebook Search Algorithm for Vector Quantization

LU Zhe-ming¹, PAN Jeng-shyang^{1,2}, SUN Sheng-he¹

(1. Dept. of Automatic Test, Measurement and Control, Harbin Institute of Technology, Harbin 150001, China;

2. Dept. of Electronic Engineering, Kaohsiung Institute of Technology, Kaohsiung, Taiwan)

Abstract: A fast codebook search method based on the mean squared error (MSE) distortion for vector quantization is presented. The sums of the codewords are computed and saved in the codebook before every search. During the search process, a lot of codewords can be rejected by utilizing the relationship among the sum of the input vector, the sums of the codewords and MSE. The test result shows that the reduction of computation is about 90% in the encoding time against full search method, and a little precomputation and extra storage are required.

Key words: mean square error(MSE); vector quantization (VQ)

1 引言

自从 1980 年提出矢量量化码书设计的 LBG 算法^[1]以来, 矢量量化技术已经成功地应用于语音编码和图像压缩系统中. 矢量量化过程可以定义为从 k 维欧几里德空间 R^k 到其一个有限子集 C 的一个映射, 即 $Q: R^k \rightarrow C$, 其中 $C = \{Y_1, Y_2, \dots, Y_n | Y_i \in R^k\}$ 称为码书, n 为码书长度. 该映射应满足: $Q(X \in R^k) = Y_i^*$, 其中 $X = (x_1, x_2, \dots, x_k)$ 为 R^k 中的 k 维矢量, $Y_i^* = (y_{i1}, y_{i2}, \dots, y_{ik})$ 为码书 C 中的码字并满足

$$d(X, Y_i^*) = \min_{1 \leq j \leq n} (d(X, Y_j)) \quad (1)$$

其中 $d(X, Y_i)$ 为矢量 X 与码字 Y_i 之间的失真测度, 本文采用的失真测度为欧氏距离的平方即均方误差(MSE), 其表达式为

$$d(X, Y_i) = \sum_{j=1}^k (x_j - y_{ij})^2 \quad (2)$$

其中 $X = (x_1, x_2, \dots, x_k)$ 为 R^k 中的 k 维矢量, $Y_i = \{y_{i1}, y_{i2}, \dots, y_{ik}\}$ 为码书 C 的码字.

如果把整个码书看成一条由码书中各码字连接而成的基因链, 各码字是其中的基因, 而输入矢量是外部基因, 则传统矢量量化码书设计过程可以形象地描述如下: 在每次迭代中,

每输入一个外部基因 X , 将该基因与码书基因链上的各基因 $Y_i (i = 1, 2, \dots, n)$ 相对比(基因对比)找到与该外部基因最近的一个基因 Y_j (最佳基因), 并把所有与 Y_j 最近的外部基因归为一类(基因归类), 这样一共有 n 类($j = 1, 2, \dots, n$), 然后计算各输入基因与其在码书链上最相近的基因的距离平方之和就得到本次迭代(遗传)的总失真(基因差别). 若本次迭代的总失真与上次迭代的总失真之间的相对误差(两代差别)满足给定要求就停止迭代, 否则求出各类的中心基因作为新的码书链各基因(基因重组)并进行下一次迭代.

矢量量化的优点是可以获得较高的压缩比, 而且解码结构比较简单. 但是, 在码书设计中, 为了找到输入矢量的最佳匹配码字, 传统 LBG 算法^[1]通常要对码书中所有码字作一次穷尽搜索. 对于码书长度为 n , 矢量维数为 k 的穷尽搜索矢量量化编码器, 每一输入矢量的编码计算复杂度为 $O(n \cdot k)$, 随着 k 和 n 的增大计算复杂度和计算时间将显著增加. 这样就限制了矢量量化在高速和实时编码系统中的应用. 为此, 近年来许多学者提出了各种矢量量化快速算法, 如部分失真排除算法^[2]、双测试算法^[3]、 $k-d$ 树搜索算法^[4]、超立方体测试算法^[5]、平均值差法^[6]等.

本文将介绍一种基于均方误差测度的矢量量化码书搜索

收稿日期: 1998-09-07; 修订日期: 1999-04-10

算法. 该算法在码书设计的每次迭代前预先计算各码字的和值并保存在码书中. 在每次迭代过程中, 利用输入矢量的和值、各码字的和值以及均方误差三者的各种特性排除大部分候选码字而免去均方误差计算, 并对码书进行预排序, 选择合适的初始匹配码字, 大大降低码书搜索过程中的计算量.

2 算法

在提出本文算法之前, 先来看看几个重要结论.

2.1 输入矢量与码字间的距离特性

定义一个矢量各分量的和为一个矢量的和值. 设输入矢量 X 的和值为 S_x , 码字 Y_j 的和值为 $S_j (j=1, 2, \dots, n)$, 即

$$S_x = \sum_{l=1}^k x_{il}, \quad S_j = \sum_{l=1}^k y_{jl} \quad (3)$$

可以证明

$$d(X, Y_j) = \sum_{l=1}^k (x_{il} - y_{jl})^2 \geq k \left[\sum_{l=1}^k (x_{il} - y_{jl}) / k \right]^2 = \frac{(S_x - S_j)^2}{k} \quad (4)$$

设当前的最小失真为 d_{\min} , 并令 $MD_1 = k \cdot d_{\min}$. 若

$$(S_x - S_j)^2 \geq MD_1 \quad (5)$$

则根据式(4)可得 $d(X, Y_j) \geq d_{\min}$ (6)

因此, 在每次迭代前预先计算 n 个码字的和值 $S_j (j=1, 2, \dots, n)$ 并保存在码书中, 在迭代过程中预先计算 MD_1 , 然后判断码字 Y_j 的和值 S_j 是否满足式(5), 若满足则码字 Y_j 可以排除而免去 MSE 计算. 实际上, 两个矢量的和值 S_j 相差越大, 往往其间的欧氏距离也越大. 因此, 我们可以在每次迭代之前按 S_j 从小到大的顺序对码书进行排序, 在迭代过程中先用二分搜索法找到与输入矢量 X 和值 S_x 最近的码字, 然后采用往返搜索法在该码字附近搜索与输入矢量 X 距离最近的码字. 在文献[6]提出的算法中, 预先计算各码字的平均值并保存在码书中. 这样, 在预先计算量中就需要 n 次除法. 尽管在编码的场合中, n 次除法计算量并不算多, 但在码书设计的场合中, 假设需要 J 次迭代, 则需要 $n \cdot J$ 次除法. 为此, 本文采用矢量的和值来代替矢量的平均值以避免 $n \cdot J$ 次除法.

2.2 码字间的距离特性

根据距离测度的三角不等式性质, 可以得到

$$\sqrt{d(Y_i, Y_j)} \leq \sqrt{d(Y_i, X)} + \sqrt{d(Y_j, X)} \quad (7)$$

此处, X 为输入矢量, Y_i 和 Y_j 是码书 C 中的任意两个码字. 假设 Y_j 是目前输入矢量 X 最近的码字, 如果码字 Y_i 更接近于输入矢量 X , 则应有

$$\sqrt{d(Y_i, Y_j)} \leq \sqrt{d(Y_i, X)} + \sqrt{d(Y_j, X)} \leq 2 \cdot \sqrt{d(Y_i, X)} \quad (8)$$

即 $d(Y_i, Y_j) \leq 4 \cdot d(Y_i, X)$ (9)

因此, 我们可以在每次迭代前预先计算 $n(n-1)/2$ 个码字间距并保存在码书中, 在迭代过程中如果码字 Y_j 不满足式(9)则该码字就可以排除. 下图 1 是码字排除示意图, 图中 Y_i 是目前输入矢量 X 最近的码字, 则在圆 1 外的所有码字 (例如码字 Y_j) 都不是输入矢量 X 最近的码字. 但是, 在码书长度 n 较大时, 存储 $n(n-1)/2$ 个距离需要很多存储单元. 为了解决

这个问题, 可以牺牲排除量 (即并不把圆 1 外所有码字都排除掉), 以便减少预先计算量和额外存储单元. 容易证明

$$d(Y_i, Y_j) = \sum_{l=1}^k (y_{il} - y_{jl})^2 \geq k \left[\sum_{l=1}^k (y_{il} - y_{jl}) / k \right]^2 = (S_i - S_j)^2 / k \quad (10)$$

此处, S_j 与码字 Y_j 的和值, S_i 为码字 Y_i 的和值. 设当前的最小失真为 $d_{\min} = d(X, Y_i)$, 并令 $MD_2 = 4 \cdot k \cdot d_{\min}$. 如果

$$(S_i - S_j)^2 \geq MD_2 \quad (11)$$

则 $d(Y_i, Y_j) \geq 4 \cdot d_{\min}$ (12)

根据式(9), Y_j 可以排除而无需计算其 MSE. 因此, 可以在每次迭代前预先计算 n 个码字的和值 $S_j (j=1, 2, \dots, n)$ 并保存在码书中, 而在搜索过程中预先计算 MD_2 , 然后根据式(11)来排除圆 1 外的一些码字. 文献[6]中也提到了码字间的这种距离特性, 但在避免存储 $n(n-1)/2$ 个码字间距时所采用的方法增加了较多的预先计算量和额外存储量.

2.3 输入矢量与码字间距离的分量特性和部分特性

由图 1 可 Y_j

见, 圆 1 内圆 2 外的所有码字 (例如码字 Y_m) 也不是与输入矢量 X 最近的码字, 可采用下述方法排除. 定义 t_{ml} 为差矢量 $Y_m - X$ 的第 l

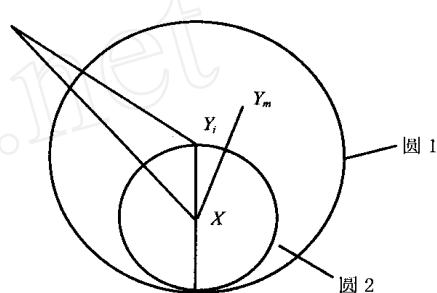


图 1 码字排除示意图

个分量 $y_{ml} - x_{il}$, 并称之为码字 Y_m 的第 l 个差值分量. 令 p_{ml} 为 t_{ml} 的平方, 由于它是失真计算中的一项, 本文称之为码字 Y_m 的第 l 个失真分量. 显然, 只要 $p_{ml} (l=1, 2, \dots, k)$ 中有一项大于当前的最小失真 $d_{\min} = d(X, Y_i)$, 码字 Y_m 就不是与输入矢量 X 距离最近的码字. 另外, 文献[2]提出的部分失真排除算法, 在计算码字 Y_m 与输入矢量 X 的失真过程中, 一旦失真分量的累加 (即部分失真) 超过了当前的最小失真 $d_{\min} = d(X,$

$Y_i)$, 即 $\sum_{l=1}^q p_{ml} \geq d_{\min}, \quad q \leq k \quad (13)$

则码字 Y_m 就可以排除. 因此, 我们可以先找出码字 Y_m 的绝对值最大的差值分量 t_{\max} , 然后判断 t_{\max} 的平方是否大于当前的最小失真 d_{\min} . 若 t_{\max} 的平方大于 d_{\min} 则码字 Y_m 就可以排除, 否则采用文献[2]提出的部分失真排除算法来判断码字 Y_m 是否可以排除. 文献[5]提出的超立方体测试算法中也利用了输入矢量与码字间距离的分量特性, 文中预先计算 d_{\min} 的开方, 然后用码字 Y_m 各差值分量的绝对值与 d_{\min} 的开方进行比较来判断码字 Y_m 是否可以排除, 但开方运算中需要多次乘法和加法.

2.4 快速矢量量化码书搜索算法

根据以上的 1, 2, 3 三条特性, 本文的快速算法描述为:

Step0 初始化: 给定码书长度 n , 输入矢量数 M , 矢量维数 k , 初始码书 C_0 , 失真阈值. 设置迭代计数器 $r=0$, 初始化整体失真 $D_{-1} = +\infty$.

Step1 计算码书 C_r 各码字 Y_i 的和值 $S_i (i = 1, 2, \dots, n)$, 并且根据 S_i 增加的次序对码书 C_r 进行排序, 生成排序码书 $C_{sr} = \{ Y_i | S_i \leq S_{i+1}, 1 \leq i \leq n-1 \}$, 并把排序后的 S_i 也保存在码书中。

Step2 对于每一个输入矢量 $X_i (i = 1, 2, \dots, M)$, 在排序码书 C_{sr} 中搜索最近码字 $Y_{i(t)}$ 并把 X_i 指定到 $i(t)$ 类。这个过程包括以下几个步骤。

Step2.1 输入一个矢量 $X_i = (x_{i1}, x_{i2}, \dots, x_{ik})$, 计算其和值 S_{x_i} 。

Step2.2 采用二分搜索方法寻找与输入矢量 X_i 和值最近的码字 Y_p , 即

$$|S_{x_i} - S_p| \leq |S_{x_i} - S_i|, i = 1, 2, \dots, n, i \neq p$$

这里 S_p 是码字 Y_p 的和值。令 $i(t) = p$, 设置当前最小失真量为 $d_{\min} = d(X_i, Y_p)$, 计算 $MD_1 = k \cdot d_{\min}$ 和 $MD_2 = 4 \cdot MD_1$

Step2.3 在往返搜索过程中, 若某个码字 Y_{p+b} (b 为自然数, $p+b \leq n$) 或 Y_{p-b} ($p-b \geq 1$) 的和值与输入矢量的和值满足式(5)或式(11)则该码字将被排除, 否则根据距离特性 3 判断该码字是否为目前离输入矢量最近的码字, 如果是则把该码字与输入矢量之间的失真作为 d_{\min} 重新计算 MD_1 和 MD_2 , 然后继续搜索下一个码字, 否则直接搜索下一个码字, 直到找到与输入矢量 X_i 最近的码字 $Y_{i(t)}$ 为止, 并把输入矢量 X_i 指定到 $i(t)$ 类。

Step3 计算第 r 次迭代的整体失真 D_r , 这里 $D_r = \frac{1}{M} \sum_{i=1}^M d(X_i, Y_{i(t)})$

Step4 计算各类的中心矢量, 并由这些中心矢量组成新码书。如果 $(D_{r-1} - D_r) / D_r < \epsilon$, 停止迭代得到最终码书, 否则转到 Step5。

Step5 令 $r = r+1$ 并回到 Step1 作下一次迭代。

3 仿真实验

本文用 640×480 的 16 色图像作为训练图像产生码书。测试采用了两种尺寸的码书 ($n = 1024$ 和 $n = 2048$)。为了比较, 本文对穷尽搜索算法, 双测试算法, 部分失真算法也作了相应测试。所有测试在 pentium 计算机上进行。表 1、表 2 分别为 n

表 1 各种快速算法比较 ($n = 2048, k = 4 \times 4$)

算法	乘法次数	加减法次数	比较次数	时间
穷尽搜索算法	629145600	1218969600	39302400	100
双测试算法	226411318	443736731	69769898	64.6
部分失真算法	84179954	168359909	84179954	21.9
本文算法	62895687	312340456	45899132	9.8

$= 2048, k = 4 \times 4$ 和 $n = 1024, k = 4 \times 4$ 两种情况的测试结果。表中的编码时间都按穷尽搜索算法编码时间进行归一化, 所有计算量是指各次迭代的平均计算次数。

表 2 各种快速算法比较 ($n = 1024, k = 4 \times 4$)

算法	乘法次数	加减法次数	比较次数	时间
穷尽搜索算法	314572800	609484800	19651200	100
双测试算法	117611793	229780146	36557185	65.1
部分失真算法	49292158	98584316	49292158	22.2
本文算法	31473434	157645874	26859406	10.1

从测试的结果可以看到, 对于不同大小的码书, 采用本文提出的算法都可以有效地减少编码时间和计算量, 相对于穷尽搜索算法, 每次迭代的平均计算时间减少约 90%。

4 结论

本文介绍了一种基于最小均方误差测度的矢量量化码书搜索的快速算法。该法采用了预排序的码书, 选择合理的初始匹配码字, 并在进行均方误差计算之前, 通过三次过滤排除大部分候选码字。测试结果表明, 相对于穷尽搜索编码方法, 本文提出的算法能减少计算量, 运算时间减少约 90%。

参考文献

- [1] Y. Linde, A. Buzo and R. M. Gray. An algorithm for vector quantizer design. IEEE Trans. Commun., 1980, 28(1): 84~95
- [2] Chang-Da Bei, and R. M. Gray. An improvement on minimum distortion encoding algorithm for vector quantization. IEEE Trans. Commun., 1985, 33(10): 1132~1133
- [3] T. Torres, and J. Huguet. An improvement on codebook search for vector quantization. IEEE Trans. Commun., 1994, 42(2): 208~210
- [4] V. Ranasubramanian, and K. K. Paliwal. Fast k-dimensional tree algorithms for nearest-neighbor search with application to vector quantization encoding. IEEE Trans. Signal Process., 1992, 40(3): 518~531
- [5] M. R. Soleymani, and S. D. Mörgera. An efficient nearest neighbor search method. IEEE Trans. Commun., 1987, 35(4): 677~679
- [6] Chang-Hsing Lee, and Ling-Hwei Chen. High-speed closest codeword search algorithms for vector quantization. Signal Processing 1995, 43: 323~331
- [7] 周汀, 闵昊, 章倩苓. 电子学报, 1997, 25(4): 95~98



陆哲明 1995 年 7 月获得哈尔滨工业大学电磁测量技术及仪器专业学士学位, 1997 年 7 月获得哈尔滨工业大学理论电工专业硕士学位。现在为哈尔滨工业大学自动化测试与控制系电磁测量技术及仪器专业博士研究生。目前主要致力于矢量量化技术及其在图像压缩和语音编码中的应用研究。



潘正祥 1996 年获英国爱丁堡大学博士学位。现任台湾高雄科学技术学院电子工程系副教授, 并被哈尔滨工业大学自动化测试与控制系聘为兼职副导师。他在国际上发表了 30 多篇文章, 大多数被 SCI, EI 检索。目前主要致力于信号处理和微处理机应用研究。

孙圣和 哈尔滨工业大学自动化测试与控制系教授。博士生导师。电子学报编委。获多项国家和省部级奖。发表 3 本专著和 100 多篇文章。目前的研究领域包括计算机自动测试与控制、信号处理与系统辨识等。