

# 一种快速、有效的 LBG 初始码书生成算法

梁彦霞 杨家玮 李 焱

(西安电子科技大学综合业务网国家重点实验室 信息科学研究所 西安 710071)

**摘 要** 针对 LBG 算法依赖初始码书的问题,以贪婪树增长算法与码书间距最大化算法为基础,提出了一种新的 LBG 初始码书生成算法。该算法用贪婪树增长算法生成基础码书,再采用码书间距最大化算法从基础码书中生成初始码书。该算法相对于常用的分裂法降低了计算复杂度,节约了运算时间。与两种基本算法相比,它降低了量化的失真度与平均谱包络失真。

**关键词** 矢量量化, LBG 算法, 贪婪树增长算法, 码书间距最大化算法, 运算时间, 谱包络失真

**中图分类号** TN912.3 **文献标识码** A

## Fast and Effective LBG Initial Codebook Generation Algorithm

LIANG Yan-xia YANG Jia-wei LI Ye

(The State Key Lab. of Integrated Service Networks and Information Science Institute, Xidian University, Xi'an 710071, China)

**Abstract** With consideration of LBG algorithm depending on initial codebook, a new LBG initial codebook generation algorithm was developed, on the basis of Greedy Tree Growing Algorithm (GTGA) and Most Dispersed Codewords in Initialization (MDCI) algorithm. A fundamental codebook was generated by GTGA first, and then an initial codebook was produced from the fundamental codebook by MDCI algorithm. The computation of this new algorithm is lower than usual Split algorithm, and the run time decreases too. Compared with these two basic algorithms, both the Distortion and Average Spectral Distortion are reduced.

**Keywords** Vector quantization, LBG algorithm, Greedy tree growing algorithm (GTGA), Most dispersed codewords in initialization (MDCI) algorithm, Run time, Spectral distortion

## 1 引言

在移动通信中,语音信号的处理非常重要。而在频谱资源匮乏的今天,只用很少的比特对语音参数进行量化且性能又没有大的损失成为合理利用频谱资源的一种手段。矢量量化就是目前比较常用的一种方法,它不仅用于语音编码,而且是语音识别以及图像压缩等的一种重要方法。

在矢量量化的过程中,码书的训练是重要的一环,而码书训练最常用的就是 LBG 算法。它是一种迭代的算法,在理论上很严密,实施过程简便,设计效果好,因而得到了广泛的应用,但是它依赖于初始码书的选取,容易陷入局部最优。目前最常用的初始码书生成算法为随机法、分裂法、乘积码法等,其中分裂法的效果显著。文献[1,2]中提出了一种优于分裂法的码书间距最大化(Most Dispersed Codewords in Initialization, MDCI)算法,这种算法的初始码字全部来自输入矢量;第一个初始码字为输入矢量中离质心最近的矢量,以后每个新产生的初始码字尽可能地远离现有的码字。这样可以避免分裂法导致局部最优,而且计算时间大大缩短。该算法用于图像编码时,峰值信噪比较高,但是得到的初始码字全部来自输入矢量,码字缺乏代表性。为了解决这个问题,先采用贪

婪树算法(Greedy Tree Growing Algorithm, GTGA)得到基础码书,再从基础码书中用 MDCI 算法得到最终的初始码书。

本文提出了一种结合贪婪树算法的码书间距最大化算法(MD-GTGA),用来设计 LBG 算法的初始码书,以克服 LBG 算法的缺点。且将此算法得到的码书用于多带激励线性预测编码<sup>[3]</sup>(Multi-Band Excitation-Linear Predictive Coding)中线谱频率(Line Spectral Frequency)参数的量化。结果显示,用新的初始码书进行 LBG 迭代得到的码书来量化 LSF,译码得到的语音效果好。

## 2 MDCI 的原理及分析

LBG 的初始码书生成算法中最常用的就是分裂法。分裂法的过程如下。

步骤 1 取整个训练序列的质心作为初始码字  $C_0$ ;

步骤 2 将初始码字加上扰动因子  $\alpha$ , 成为新的码字:令  $u=1, v$  在 1 到  $2^u$  之间,即

$$C_{v-1} = (1+\alpha)C_{v-1} \quad (1)$$

$$C_{2^u+v-1} = (1-\alpha)C_{v-1}$$

步骤 3 将这  $2^u$  个码字进行 LBG 迭代,得到新的  $2^u$  个码字;

到稿日期:2010-09-16 返修日期:2010-12-30 本文受国家杰出青年科学基金(60725105),长江学者和创新团队发展计划(IRT0852),863 计划课题和国家自然科学基金项目(2007AA01Z288),国家重点实验室专项基金(ISN02080001),高等学校创新引智计划(B08038)资助。

梁彦霞(1981-),女,博士生,主要研究方向为移动通信、无线通信、语音信号的编解码, E-mail:lx219@sohu.com;杨家玮(1946-),男,教授,主要研究方向为移动通信、无线通信、通信系统架构及性能评估。

步骤4 若  $2^u$  为预定的码书大小,则程序结束;否则  $u = u+1$ ,转向步骤2 继续执行。

分裂法对中间码书的每个码字都要一分为二,然后利用 LBG 算法形成数目是原码书 2 倍的新码书。这样设计的初始码书,不可避免会导致与 LBG 一样的局部最优,码字代表性不强。此外,分裂法的计算工作量也很大。

MDCI 算法与分裂法的主要区别在于采用间距最大的码字作为初始码书。初始码字全部来自输入的训练矢量,第一个码字为输入矢量中离质心最近的矢量,之后的每一个新产生的初始码字尽可能地远离现有的码字,直至产生所需个数的码字。很明显,MDCI 可以避免产生局部最优。

这种方法的复杂度主要在于矢量间欧氏距离的计算。确定第一个初始码字需要计算  $M$  次( $M$  为训练矢量的个数);在循环部分,计算矢量距离的次数为  $\sum_{k=2}^N (M+1-k)$ ,其中,  $N$  为初始码书中矢量的数目,即码本大小。因此,码书初始化过程所需要的矢量距离计算次数为

$$M + \sum_{k=2}^N (M+1-k) = (M - \frac{N}{2} + \frac{1}{2})N \quad (2)$$

即初始化的时间复杂度为  $O(NM)$ ,与 LBG 算法的一次迭代是相当的。

### 3 GTGA 的原理与分析

GTGA<sup>[4,5]</sup> 是基于树状矢量量化器与可变率矢量量化器设计的一种方法。用这种方法设计基础码书,运算量低,避免了空胞腔的产生,提高了码书的性能。

给定训练序列  $X = \{x_0, x_1, \dots, x_{M-1}\}$ ,矢量的维数为  $K$ ,需要设计具有  $N(N \geq 2)$  个矢量的码书,其中  $M \gg N$ 。把训练序列  $X$  看成一个全集  $S$ ,以  $S$  的形心作为根节点  $v_0$ ;以  $v_0$  为父节点,加入扰动因子  $\delta$ ,即由  $v_0(1+\delta)$  和  $v_0(1-\delta)$  得到两个矢量  $v_1$  与  $v_2$ ,称之为子节点;用  $v_1$  和  $v_2$  把  $S$  分裂为两个集  $S_1$  与  $S_2$ , $S_i$  表示  $S$  中距离  $v_i$  最近的矢量的集合;再取  $v_i$  为  $S_i$  的形心,其中  $i=1,2$ 。用矢量  $v_1$  与  $v_2$  构成码书,去量化所有的训练矢量,总失真为

$$D = \sum_{x \in S_1} d(x, v_1) + \sum_{x \in S_2} d(x, v_2) \quad (3)$$

我们称没有子节点的节点为终端节点。贪婪树生长算法开始以  $v_0$  为根节点, $v_1, v_2$  为终端节点(称这次更改为第一次更改),以后每次只更改一个终端节点,即把此终端节点改为父节点,添加其两个子节点为终端节点,保持其它的终端节点不变,从而增加一个终端节点,使得平均失真的减少量与比特率的增量的比值达到最大。这个比值被称为即梯度变量,用式(4)表示。

$$\lambda = \frac{\nabla D}{\nabla l} \quad (4)$$

式中,  $\nabla D$  表示更改一个终端节点以后平均失真的减少量,  $\nabla l$  为增加一个节点后比特率的增量,单位是比特/矢量。

如果终端节点数小于  $N$ ,则可以将所有的终端节点分解,再找出分裂前后失真的减小量最大的终端节点,将该节点作为父节点,添加其两个子节点为终端节点,其他终端节点保持不变。但是这种方法将大部分节点分裂后并没有使用其分裂的结果,为此引入终端节点  $i$  对总失真的贡献  $G_i$ :

$$G_i = \sum_{x \in T_i} d(x, w_i) \quad (5)$$

式中,  $T_i$  为划分的第  $i$  个子集合,  $x$  为此集合内的矢量,  $w_i$  为第  $i$  个终端节点。在每一次更改之后,我们都先求出每一个

终端节点对总失真的贡献,再利用分裂算法对总失真贡献最大的终端节点进行分裂。下一次更改就是把该节点改为父节点,添加其两个子节点为终端节点,而其他的终端节点保持不变。这样更改  $N-1$  次,就可以得到  $N$  个终端节点。

### 4 贪婪树初始码书间距最大化算法(MD-GTGA)

MD-GTGA 先用 GTGA 得到一个较大的基础码书,在此基础码书内,按照码字间距最大原则找到所需个数的码书。这样,既利用了 GTGA 算法相对于分裂法工作量低的优点,又克服了 MDCI 从训练序列中寻找的码字不具有代表性的缺点。MD-GTGA 的具体算法如下。

步骤1 计算整个训练序列的形心,设为  $v_0$ ,将它作为根节点;

步骤2 分裂  $v_0$ ,得到子节点  $v_1$  与  $v_2$ ,以  $v_1$  与  $v_2$  为终端节点,置  $n=2$ ;

步骤3 利用式(5)计算新添加的两个终端节点对总失真的贡献;

步骤4 分裂所有终端节点中对总失真的贡献最大的端节点,将该节点改为父节点,添加其两个子节点为终端节点,而其它的终端节点保持不变,置  $n=n+1$ ;

步骤5 如果  $n < N1$  ( $N1$  为基础码书大小),则返回到步骤3;否则,转向步骤6;

步骤6 设生成的初始码本空间  $R = \emptyset$ ;基础码书的质心为  $C_0$ ;

步骤7 找到基础码书中距离  $C_0$  最近的码字,作为第一个码字  $C_1$ ;且  $R = R \cup C_1, m=1$ ;

步骤8 计算  $R$  的质心  $C_i$ ,从基础码书中找到与  $C_i$  距离最远的码字  $C_{m+1}$ ;  $R = R \cup C_{m+1}$ ,同时  $m=m+1$ ;

步骤9 如果  $m=N$  ( $N$  为最后码书大小),则程序结束;否则转向步骤8 继续计算。

### 5 实验结果与分析

#### 5.1 失真度

为了得到性能优良的码书,用 GTGA 得到大小分别为 1024, 2048, 4096, 8192 和 16384 的基础码书。在此基础上,分别用 MDCI 算法产生 32, 64, 128, 256 和 512 的初始码书。将生成的码书作为 LBG 算法的初始码书进行迭代运算,得到最终码书。平均失真度按下式计算。

$$D = \frac{1}{L} \sum_{r=1}^L d(X_r, Y) \quad (6)$$

式中,  $L$  为测试序列的长度,  $X_r$  为测试矢量,  $Y$  为码书中与  $X_r$  最接近的码字。失真度与基础码书大小的关系如图 1 所示。

从图 1 中可以看出,码书大小为 512 时,基础码书大小为 2048 的失真度最小,16384 次之。码书大小为 256 和 128 时,则基础码书分别为 8192 和 4096 时失真度最小。由数据可见,当基础码书大小为所需码书大小的 32 倍时,失真度相对较低;码书大小为 64 和 32 时(见图 1),基础码书为其 32 倍大小时的失真度为最小或者次小。

图 2 为 GTGA, MDCI 及 MD-GTGA 算法的失真度与码书大小的关系。由图 2 可以看到,新的码书生成算法生成的码书失真度基本低于原码书生成算法,只有码书大小为 32 时,新算法的失真度略高于 MDCI。而 32 大小的码书并不是

(下转第 141 页)

性进行 MARTE 元建模,再利用 ATL 转换工具在元模型层将 MARTE 模型转换为价格时间自动机模型,实质上这是一种基于元模型的模型转换方法。进而通过具体案例进行模型转换,对转换后的结果使用 UPPAAL 工具验证其正确性和可靠性。然而,在嵌入式实时系统中,会涉及硬件的相关信息,MARTE 提供专门的包来对系统硬件如内存、设备等可重用资源建模,而在加入代价后,PTA 也提供了很多算法来解决最小代价问题,这都是本文下一步要研究的内容。

### 参考文献

[1] 王涛,张伟良,葛宁,等. 嵌入式实时系统及其在通信系统中的应用[J]. 电子应用技术,2001,27(11)

[2] 张天,张岩,于笑丰,等. 基于 MDA 的设计模式建模与模型转换[J]. 软件学报,2008,19(09):2203-2217

[3] OMG. Unified modeling language: Superstructure v2.0 [EB/OL]. <http://www.omg.org/docs/formal/05-07-04.pdf>,2005

[4] OMG. UML Profile for MARTE, Beta 2 [EB/OL]. <http://www.omg.org/cgi-bin/doc?ptc/2008-06-08>,2008

[5] Fuentes-Fernández L, Vallecillo-Moreno A. An Introduction to UML Profiles[J]. The European Journal for the Informatics Professional,2004

[6] CEA LIST Team. Papyrus (open source tool for graphical UML2 modeling) [EB/OL]. <http://www.papyrusuml.org/>,2007

[7] Clarke E M, Wing J M. Formal methods: state of the art and future directions[J]. ACM Computing Surveys,1996,28(4):626-643

[8] Del Bianco V, Lavazza L, Mauri M. Model checking UML specifications of real time software[C]//Proceedings of the 8th IEEE

International Conference on Engineering of Complex Computer Systems, 2002:203-212

[9] Bernardi S, Donatelli S, Merseguer J. From UML sequence diagrams and statecharts to analysable petri net models[C]//Proceedings of the 3rd International Workshop on Software and Performance. 2002:35-45

[10] Czarnecki K, Helsen S. Classification of Model Transformation Approaches[J]. Generative Techniques in the Context of Model-driven Architecture,2003

[11] Jouault F, Allilaire F, Bézivina J, et al. ATL: A model transformation tool [J]. Science of Computer Programming, 2008, 72(2):31-39

[12] Jouault F, Bézivina J, Barbero M. Towards an advanced model-driven engineering toolbox[J]. Innovations in Systems and Software Engineering, 2009, 5(1):5-12

[13] Behrmann G, Larsen K G, Rasmussen J I. Optimal Scheduling using Priced Timed Automata[J]. ACM Sigmetrics Performance Evaluation Review, 2005, 32(4):34-40

[14] Behrmann G, David A, Larsen K G. A Tutorial on Uppaal[J]. Formal Methods for the Design of Real-Time Systems, 2004

[15] Object Management Group. OMG [EB/OL]. <http://www.omg.org/>

[16] Jouault F, Kurtev I. Transforming Models with ATL [C]// Satellite Events at the MoDELS 2005 Conference. 2006

[17] Rasmussen J I, Larsen K G, Subramani K. Resource-optimal Scheduling Using Priced Timed Automata[J]. Tools and Algorithms for the Construction and Analysis of Systems, 2004

[18] 刘辉,麻志毅,邵维忠. 元建模技术研究进展[J]. 软件学报, 2008,19(6):1317-1327

(上接第 116 页)

常用码书,因此对码书的优化影响不大。

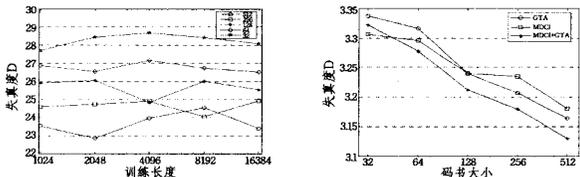


图 1 失真度与不同大小基础码书的关系 图 2 失真度与码书大小关系

### 5.2 平均谱失真

谱包络失真(Spectral Distortion, SD)是比较常用的一种客观评价量化器性能的标准。第  $n$  帧的 SD 表示为

$$SD_n = \frac{1}{\pi} \left\{ \int_0^{\pi} \left[ 20 \log_{10} \left| \frac{H(\omega)}{\hat{H}(\omega)} \right| \right]^2 d\omega \right\}^{1/2} \quad (\text{dB}) \quad (7)$$

式中,  $H(\omega)$  和  $\hat{H}(\omega)$  分别为输入和输出语音的短时线性预测分析谱。

平均谱失真如下式所示。其中,  $M$  为测试矢量的总数。

$$ASD = \frac{1}{M} \sum_{m=1}^M SD_m \quad (8)$$

平均谱失真度与基础码书大小的关系如图 3 所示。虽然各种算法都没有满足语音质量的客观评价标准,但是从文献[6]中可知,这里谱包络失真值及谱泄露值对评价一种码书的好坏也有着直接的影响,其数值的大小仍能反映出码书的质量。

由图 3 可以看到,由所需码书的 32 倍大小的基础码书得到的最终码书,其平均谱失真也是最小或者次小的。图 4 中也可以看到,从平均谱失真的角度看,也是除 32 大小的码书外,采用新算法生成的平均谱失真均较小。

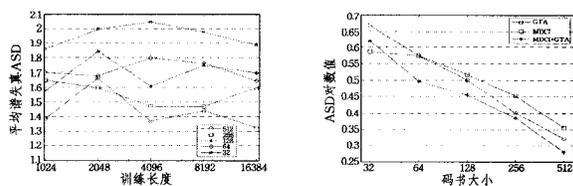


图 3 平均谱失真与不同大小基础码书的关系 图 4 平均谱失真与码书大小关系

**结束语** 本文基于贪婪树及初始码书间距最大算法提出了一种新的算法,该算法结合了贪婪树算法计算量低的特点,并克服了初始码书间距最大化算法不具代表性的缺点。结果显示,用贪婪树算法生成的基础码书大小为最终所需码书大小的 32 倍时,所得码书的失真度及平均谱失真都是最低或次低的,性能相对于原算法均有所改进。

### 参考文献

[1] 李碧,林土胜,廖亮. 初始码字间距最大化的矢量量化码书设计算法 [J]. 小型微型计算机系统, 2009, 30(4): 780-783

[2] Katsavounidis I, Jay Kuo C-C, Zhang Zhen. A New Initialization Technique for Generalized Lloyd Iteration [J]. IEEE Signal Processing Letters, 1994, 1(10): 144-146

[3] 梁彦霞,杨家玮,李焯. 一种低速率多模 MBE 线性预测语音编解码器 [J]. 西安电子科技大学学报, 2010, 37(3): 417-422

[4] Riskin E A, Gray R M. A Greedy Tree Growing Algorithm for the Design of Variable Rate Vector Quantizers [J]. IEEE Trans Signal Proc, 1991, 39(11): 2500-2507

[5] 李朔程,文超,平西建. 两个优于分裂法的初始码书设计方法 [J]. 中国图象图形学报, 2000, 5(A)(1): 48-51

[6] 梁彦霞,杨家玮,李焯. 线谱频率参数的快速、低存储矢量量化 [J]. 吉林大学学报:工学版,录用待发表