

一种快速的固定码本搜索方法

高建敏, 施海彬, 唐 骏

(厦门大学信息科学与技术学院 ATR 实验室, 福建 厦门 361005)

摘要: 为降低固定码本搜索算法的复杂度, 在脉冲取代法的基础上提出一种码矢分段优化的快速搜索方法。采用码矢分段优化的方法, 在保证语音质量的前提下, 降低计算复杂度。实验结果表明, 与 AMR-WB 采用的深度优先树算法及传统的脉冲取代算法相比, 在不影响语音质量的条件下, 码矢分段优化算法复杂度降低了 70%~80%。

关键词: 语音编码; 固定码本搜索; 码矢分段优化; AMR-WB 标准; MOS 值

Fast Fixed Codebook Search Method

GAO Jian-min, SHI Hai-bin, TANG Jun

(ATR Lab, School of Information Science and Technology, Xiamen University, Xiamen 361005, China)

【Abstract】 In order to reduce complexity of fixed codebook search, a fast algebraic codebook search algorithm based on pulse replacement procedure is proposed in this paper. The proposed method is employed dividing candidate codevector into two sub-codevectors to optimize. It not only lowers the computational complexity sharply, but also ensures the speech quality. Experimental results show that the proposed method reduces the total computational complexity by 70%~80% with speech quality equivalent, compared with that by the depth-first tree search method adopted in AMR-WB and traditional pulse replace method.

【Key words】 speech coding; fixed codebook search; sub-codevector optimization; AMR-WB standard; MOS score

DOI: 10.3969/j.issn.1000-3428.2011.08.092

1 概述

AMR-WB 算法是由第 3 代合作伙伴计划(3GPP)提出的宽带语音编码标准, 其采样率为 16 000 Hz, 采用 14 bit 量化。整个算法的核心部分是 ACELP, 即自适应码本激励线性预测, 包括线性预测编码(Linear Predictive Coding, LPC)、自适应码本搜索以及固定码本搜索。

目前, 固定码本搜索的方法主要有集中搜索、树状搜索以及脉冲取代法 3 种。集中搜索法由嵌套循环搜索法发展而来, 它需要先找出一个临界值, 当前面循环的某一参数绝对值和超过这个临界值, 才会进入最后一个循环进行搜索。深度优先树搜索是成对、有序地按轨道顺序进行脉冲搜索。脉冲取代法包括最不重要脉冲取代、全局脉冲取代以及迭代树取代 3 种。虽然这些次优搜索方法降低了固定码本搜索的复杂度, 但是其仍占整个复杂度的 40%左右^[1]。因此, 在保证语音质量不受影响的前提下, 对固定码本搜索算法的改进依然是研究的重点。本文提出一种基于码矢分段优化的快速码本搜索方法, 在计算复杂度与 ARM 模拟运行时间上与 DFTS 法、PR 法进行了比较并给出结论。

2 AMR-WB 固定码本搜索原理

2.1 固定码本结构

AMR-WB 的固定码本结构采用的是交错正负脉冲设计(Interleaved Single-Pulse Permutation, ISPP), 是具有代数结构的随机码本。这种代数码本无需存储, 可以有效减少编码算法程序和数据的存储量, 解码端可以根据非零脉冲的幅度和位置信息直接得到码矢量, 提高了解码效率, 保证了算法的实时性^[2]。

固定码本码矢量长度为 64, 包含 N_c 个非零脉冲, 每个脉冲的幅度为 +1 或 -1。脉冲可能出现的 64 个位置被划分为

4 个轨道, 不同码率下每个轨道的非零脉冲的数目和位置不同^[3]。现以 12.65 Kb/s 模式为例, 码本结构中包含 8 个非零脉冲, 脉冲可能出现的 64 位置被分为 4 个轨道, 每个轨道含有 2 个非零脉冲, 其码本结构如表 1 所示。

表 1 12.65 Kb/s 模式下固定码本矢量中非零脉冲分布

轨道	脉冲	可能出现的位置
0	i_0, i_4	0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60
1	i_1, i_5	1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61
2	i_2, i_6	2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62
3	i_3, i_7	3, 7, 11, 15, 19, 23, 27, 31, 35, 39, 43, 47, 51, 55, 59, 63

在 12.65 Kb/s 模式下, 初始码矢 $c(n)$ 长度为 64, 包含 8 个带符号的单位脉冲得到, 即:

$$c(n) = s_0\delta(n - m_0) + s_1\delta(n - m_1) + \dots + s_7\delta(n - m_7) \\ n = 0, 1, \dots, 63 \quad (1)$$

2.2 固定码本搜索

固定码本搜索是依据加权输入语音与加权重构语音的均方误差最小的准则找到最优的码本矢量 c_k 及其增益 g_c 。加权均方误差定义为:

$$E_k = \|x - g_c H c_k\|^2 \quad (2)$$

其中, x 为目标矢量; H 为由感知加权合成滤波器的脉冲响应 $h(n)$ 组成的下三角 Toeplitz 卷积矩阵, 其对角线上的元素都为 $h(0)$, 依次向下的对角线上元素分别为 $h(1)$, $h(2)$, \dots , $h(63)$; c_k 是索引为 k 的码矢量。目标矢量 x 由闭环基音搜索目标信号 $x(n)$ 减去自适应码本的贡献值得到, 即: $x_2(n) = x(n) - g_p y(n)$ 。其中, g_p 是自适应码本的增益; $y(n)$

基金项目: 福建省科技厅重大专项基金资助项目(2009HZ0003-1)

作者简介: 高建敏(1985—), 男, 硕士研究生, 主研方向: 嵌入式技术, 语音编码; 施海彬、唐 骏, 博士研究生

收稿日期: 2010-12-16 **E-mail:** gaojianmin.2007@163.com

是滤波后的自适应码本矢量, $y(n) = v(n) * h(n)$ 。

E_k 取最小值时, 式(3)取最大值:

$$Q_k = \frac{(x_2' H c_k)^2}{c_k' H' H c_k} = \frac{(d' c_k)^2}{c_k' \Phi c_k} = \frac{(R_k)^2}{E_k} \quad (3)$$

在式(2)中, $d (= H' x)$ 是反向滤波目标矢量, 它表示 $r(n)$ 和 $h(n)$ 的相关信号, 而 $\Phi (= H' H)$ 是 $h(n)$ 的自相关矩阵。 d 与 Φ 在码本搜索之前即可计算。

d 和 Φ 的各个元素分别表示为:

$$d(n) = \sum_{i=n}^{63} x_2(i)h(i-n) \quad n=0,1,\dots,63 \quad (4)$$

$$\phi(i,j) = \sum_{n=j}^{63} h(n-i)h(n-j) \quad (j \geq i) \quad (5)$$

由于 c_k 仅有 N_c 个非零脉冲, 代数结构的码本存在快速搜索算法。式(3)中的分子和分母可分别简化为:

$$C = \sum_{i=0}^{N_c-1} s_i d(m_i) \quad (6)$$

$$E = \sum_{i=0}^{N_c-1} \phi(m_i, m_i) + 2 \sum_{i=0}^{N_c-2} \sum_{j=i+1}^{N_c-1} s_i s_j \phi(m_i, m_j) \quad (7)$$

其中, m_i 是第 i 个脉冲的索引; s_i 是它的幅度。

为了简化搜索过程, 用适当的参考信号 $b(n)$ 先作脉冲幅度预判决, 即设置某一位置脉冲幅度的符号等于 $b(n)$ 在此位置的符号, 即:

$$s_i = \text{sign}(b(m_i)) \quad (8)$$

其中, 命中函数值 $b(n)$ 可表示为:

$$b(n) = \sqrt{\frac{E_d}{E_r}} r_{LTP}(n) + \alpha d(n) \quad n=0,1,\dots,63 \quad (9)$$

其中, $E_d = d' d$ 是信号 $d(n)$ 的能量; E_r 是经长时预测后残余信号 r_{LTP} 的能量; 扩展因子 α 与码率成反向关系, 码率越高, α 值越大。

固定码本可以看作码本矢量集, 每个码本矢量由位置受限的 N_c 个脉冲组成, AMR-WB 取子帧长度为 5 ms, 每个子帧都需要在固定码本中找到最优的码矢量。但是, 固定码本的全搜索由 N_c 个循环嵌套来完成, 计算复杂度高, 难以得到应用。于是, 搜索码本矢量的目标从最优向次优转变, 如集中搜索、深度优先树搜索及脉冲取代法这些次优算法应运而生。

集中搜索法采用一个阈值限制搜索范围, 这种方法的计算量随着搜索的脉冲个数 N_c 的增加而增大。作为更有效率的方法, 深度优先树搜索法则沿着有局部最大值的树路径进行搜索。AMR-WB 编码端采用了深度优先树搜索算法, 不同编码速率模式下的搜索过程类似, 不同的是脉冲的数目与树的搜索层次^[3]。

在 12.65 Kb/s 模式下, 子帧长度为 64, 码本矢量由 8 个非零脉冲组成, 每个轨道有 2 个脉冲。每次搜索得到处在相邻轨道上的 2 个脉冲。被搜索的 2 个相邻轨道可以从 T_0-T_1 、 T_1-T_2 、 T_2-T_3 、 T_3-T_0 选择。深度优先法的搜索树分为 4 层。第 1 层, 脉冲 P_0 置于轨道 T_0 , 脉冲 P_1 置于轨道 T_1 , 这一层不需要搜索, 脉冲 i_0 置于所有轨道 $b(n)$ 最大的位置。第 2 层, 脉冲 P_2 置于轨道 T_2 , 脉冲 P_3 置于轨道 T_3 , 脉冲 P_2 搜索 4 个位置, 脉冲 P_3 搜索 16 个位置, P_2 的 4 个候选位置是根据轨道 T_2 上 $b(n)$ 最大的 4 个值确定的。第 3 层, 脉冲 P_4 置于轨道 T_1 , 脉冲 P_5 置于轨道 T_2 , 脉冲 P_4 搜索 8 个位置, 脉冲 P_5 搜索 16 个位置。和上一层相似, P_4 的 8 个候选位置是根据轨道 T_1 上 $b(n)$ 最大的 8 个值确定的。第 4 层, 脉冲 P_6 置于轨道

T_3 , 脉冲 P_7 置于轨道 T_0 。脉冲 P_6 搜索 8 个位置, 脉冲 P_7 搜索 16 个位置。因此, 总共的测试组合数目为: $4 \times 16 + 8 \times 16 + 8 \times 16 = 320$ 。经过上述 4 个层次的搜索完成一次迭代, 接着将最先搜索的 2 个脉冲分别置于不同的相邻轨道, 继续下一次迭代。整个迭代过程重复 4 次, 因此全部搜索次数仅为 1 280。

3 新的快速搜索方法

3.1 脉冲取代法思想

最不重要脉冲取代法的基本思想是: 选取一组脉冲组合为默认码矢量, 计算去除一个脉冲后的默认码矢量的 Q_k 值, Q_k 值最大时去除的那个脉冲即最不重要脉冲, 接着用该脉冲所在轨道上的其他脉冲取代这个最不重要脉冲, 如此迭代, 直至 Q_k 不再发生变化为止。但是, 最不重要脉冲取代法不能保证每个子帧均找到最优码本矢量, 于是将取代用的新脉冲从最不重要脉冲所在轨道扩展至所有轨道, 即全局脉冲取代。

3.2 码矢分段优化搜索算法原理

无论最不重要脉冲取代还是全局脉冲取代, 它们的计算复杂度随着脉冲取代次数的增加而迅速增大。然而, 将候选码本矢量拆分成 2 个子码矢量, 分别进行脉冲取代, 然后合并, 这样可以有效减少计算量。现以 12.65 Kb/s 编码模式为例说明码矢分段搜索算法的实现过程, 该模式下码矢量由 8 个非零脉冲组成, 算法描述如下:

(1) 分别对每条轨道上的 16 个脉冲按命中函数值 $b(n)$ 从大到小进行排序, 各轨道取 $b(n)$ 最大的 2 个脉冲, 组合成 8 个脉冲即待拆分的初始码本, 记为 *codevector*0, 计算其 Q_k 值, 记为 Q_{k0} 。

各个轨道的脉冲根据 $b(n)$ 的大小进行排序, 结果如下所示:

T_0 4, 0, 32, 8, 16, 36, 12, 44, 28, 24, 20, 48, 52, 56, 60, 40;

T_1 9, 5, 41, 1, 29, 33, 61, 13, 17, 21, 45, 37, 25, 53, 57, 49;

T_2 2, 6, 10, 42, 22, 30, 58, 26, 14, 46, 38, 62, 54, 50, 18, 34;

T_3 3, 7, 51, 23, 11, 27, 39, 55, 47, 43, 63, 19, 59, 31, 15, 35;

每个轨道各取 $b(n)$ 最大的 2 个脉冲得到(4, 0, 9, 5, 2, 6, 3, 7), 即 *codevector*0。

(2) 将 8 个脉冲按位置索引值大小排序, 将排序后的脉冲组合 *index*[8] 拆分为 *codev*1、*codev*2 2 个子码矢: 索引值为 *index*[0]、*index*[2]、*index*[4]、*index*[6] 的 4 个脉冲组成 *codev*1; *index*[1]、*index*[3]、*index*[5]、*index*[7] 组成 *codev*2。

*codevector*0 中的脉冲按索引排序后为(0, 2, 3, 4, 5, 6, 7, 9), 即 *index*[8]。拆分得到的 *codev*1 为(0, 3, 5, 7), *codev*2 为(2, 4, 6, 9)。

(3) 分别对子码矢 *codev*1、*codev*2 中最不重要的脉冲用其所在轨道的其他脉冲替换, 更新后的子码矢量 Q_k 值局部最大时对应的 4 个脉冲组成 *codev*1'、*codev*2'。

根据 Q_k 最大时去除的那个脉冲为最不重要脉冲, *codev*1、*codev*2 的最不重要脉冲索引分别为 2、7。依次用 T_2 、 T_3 上的 16 个脉冲取代, 若取代后子码矢对应的 Q_k 值增大, 则用当前子码矢 *codev*1'、*codev*2' 分别取代 *codev*1、*codev*2。取代完成后, *codev*1' 为(0, 27, 5, 7), *codev*2' 为(46, 4, 6, 9)。

(4) 将 *codev*1'、*codev*2' 合并得到码本矢量 *codevector*1(0, 4, 5, 6, 7, 9, 27, 46), 计算其 Q_k 值, 假设为 Q_{k1} 。倘若 $Q_{k1} > Q_{k0}$, 则将 *codevector*1 作为新的拆分、替换的码本矢量。

(5) 重复步骤(1)~步骤(4)过程 n 次, 最后的 *codevector* n 即目标码本矢量。步骤(1)~步骤(4)过程如图 1 所示, 以下的实

验 $n = 2$ 。

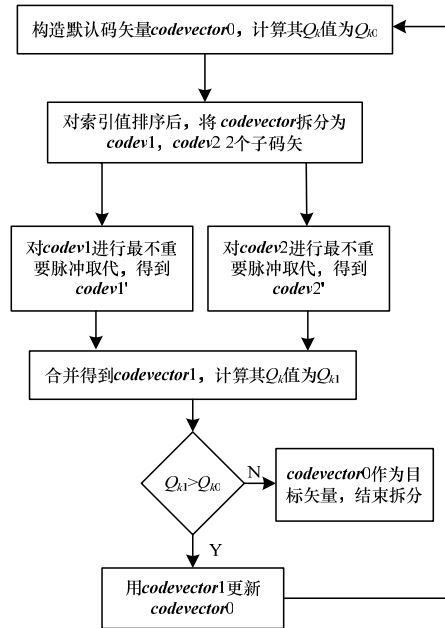


图1 码本矢量搜索流程

4 实验结果分析

用深度优先树搜索(DFTS)、脉冲取代法(PR)及码矢分段优化搜索方法(FS)对干净语音、带噪语音及纯噪声3种测试序列进行测试,利用ITU-T P.862.2 WB_PESQ语音质量估计系统客观评价语音质量^[4-5],表2给出了在12.65 Kb/s模式下3种语音的PESQ平均值。

表2 PESQ测试结果

测试对象	DFTS		PR		FS	
	Raw Mos	Mos-LQ0	Raw Mos	Mos-LQ0	Raw Mos	Mos-LQ0
带噪语音	3.747	3.875	3.621	3.717	3.529	3.593
干净语音	3.517	3.569	3.378	3.372	3.367	3.359
纯噪声	3.497	3.549	3.302	3.272	3.307	3.276

从表2可以看出,总体上FS的语音质量较DFTS以及PR略微下降,其中带噪语音的Mos值超过了3.5,达到了通信质量标准。因此,码矢分段优化算法不仅保证了语音质量,而且复杂度较其他算法具有明显的优势。

固定码本搜索的计算复杂度可以用计算 Q_c 的次数来衡量。从表3可以看出,这种快速搜索算法大幅度降低了搜索次数,码矢分段优化搜索的搜索次数是深度优先树搜索的20%,大大降低了固定码本搜索算法的复杂度。

表3 3种搜索方法的搜索次数比较

深度优先树搜索	裁剪树搜索	码矢分段优化搜索
1 280	704	256

表4给出了针对Testv1、Testv2这2个测试序列进行编码,分别采用DFTS、PR、FS搜索方法的AMR-WB编码算法在ARM仿真器上的模拟运行时间。从表4可以看出,快速搜索算法的仿真运行时间较DFTS、PR分别平均降低了21%、34%。

表4 ARM模拟运行时间

测试序列	Testv1	Testv2
DFTS	1 min14 s	39 s
PR	1 min29 s	47 s
FS	58 s	31 s

5 结束语

AMR-WB算法是一种高品质的宽带语音编码算法,其编码复杂度较高,在ARM上实现时须做必要的优化。本文提出了一种码矢分段优化的固定码本搜索方法,在不影响语音质量的条件下,码矢分段优化算法复杂度较深度优先树搜索下降了70%~80%,在ARM上运行时间平均缩短了21%。

参考文献

- [1] 范锦秀, 赵 欢, 张波涛. AMR-WB 编码算法研究和复杂度分析[J]. 数字信号处理, 2009, 33(7): 68-72.
- [2] 鲍长春. 数字语音编码原理[M]. 西安: 西安电子科技大学出版社, 2007.
- [3] 3GPP Organisational Partners' Publications. TS26.190. Adaptive Multi-rate Wideband Speech Transcoding[EB/OL]. (2000-12-16). <http://www.3gpp.org/ftp/Specs/html-info/26190.htm>.
- [4] ITU-T Study Group. Perceptual Evaluation of Speech Quality[EB/OL]. (2001-06-09). <http://www.itu.int/ITU-T/recommendations/index.aspx?ser=P>.
- [5] 周 凤, 吴 云, 赵 勇. AMR-WB+中帧错误隐藏算法的优化[J]. 计算机工程, 2009, 35(24): 222-224.

编辑 任吉慧

(上接第264页)

网格负载调度算法TD_Load,该算法在资源负载、任务执行时间跨度和时间花费3个方面具有优越性,并且有很好的扩展性,如进一步优化资源负载、减少任务等待时间。下一步的研究方向是在多用户多QoS需求的情况下,实现多个优化目标的调度算法。另外,根据资源的历史状态,采用灰色预测法、Markov链等预测策略对资源的下一个状态进行预测,从而实现动态的任务分配。

参考文献

- [1] 龚红翠, 于 炯, 侯 勇, 等. 用户 QoS 及系统指标指导的计算网络任务调度[J]. 计算机工程, 2009, 35(7): 52-54.
- [2] 姚 军, 马满福. 网络环境下担保约束的信任调度[J]. 计算机工程, 2010, 36(3): 159-161.
- [3] 王新生, 陈敬男, 王伟杰. 基于信任机制的网格资源调度算法[J].

计算机工程, 2010, 36(5): 159-161.

- [4] Ali S, Siegel H J, Maheswaran M, et al. Representing Task and Machine Heterogeneities for Heterogeneous Computing Systems[J]. Journal of Science and Engineering, 2000, 3(3): 195-207.
- [5] 袁禄来, 曾国荪, 姜黎立, 等. 网络环境下基于信任模型的动态级调度[J]. 计算机学报, 2006, 29(7): 1124-1227.
- [6] 黄德才, 张丽君, 郑月峰, 等. TD_Max-min: 信任驱动的网络任务调度新算法[J]. 计算机工程, 2007, 33(24): 80-82.
- [7] 黄德才, 张丽君, 陆亿红, 等. 基于信任驱动的网络任务调度新算法[J]. 计算机应用研究, 2007, 24(11): 26-29.
- [8] 张伟哲, 方滨滨, 胡铭曾, 等. 基于信任 QoS 增强的网格服务调度算法[J]. 计算机学报, 2006, 29(7): 1157-1166.

编辑 张正兴