

H/PC 语音压缩与重放的实现^{*}

李锦宇, 王仁华

(中国科学技术大学 电子工程与信息科学系, 安徽 合肥 230027)

E-mail: jyli@mail.ustc.edu.cn

http://www.ustc.edu.cn

摘要: 提出了一种适合于纯软件 H/PC 语音压缩与重放的高音质、低复杂度的 5kbps 语音编码器——PS-ACELP (pre-search algebraic-code-excited linear-prediction). 这个编码器的语音质量与 ITU-T 的 G. 723.1 的 5.3kbps 编码器的语音质量相当, 算法复杂度约为 G. 723.1 的三分之一, 约 7MIPS. 详细分析了所提出算法取得低运算复杂度的原因和过程. 由于具有高音质、低复杂度、低码速率的特点, PS-ACELP 算法可以广泛地应用于纯软件实现的语音压缩与重放、Voice mail、Internet Phone 等方面.

关键词: H/PC; PS-ACELP; 低复杂度; 预搜索; 开环基音预测; 自适应码本搜索; 固定码本搜索

中图分类号: TP391 文献标识码: A

H/PC(手持电脑)是一种简化的电脑,它的存储器和可用资源非常少,为了实现录放音和收发 Voice mail 等功能,就需要尽可能地对语音进行压缩,使得存储的语音尽量少占用空间资源.为此,必须提供一种高效率、高音质的语音编码器.

当今语音编码的一个发展方向是更低的码速率和更低的复杂度,从近两年 ITU-T 公布的语音编码标准 G. 729 和 G. 723.1 中我们可以看出这个趋势. G. 729 是 8kbps 的共轭结构代数码激励线性预测编码器(CS-ACELP). 该算法延迟小、语音质量好,但是算法复杂度较高,在 20MIPS 以上. ITU-T 的 G. 723.1 标准^[1]具有两种码速率,分别为 5.3kbps 和 6.4kbps,算法复杂度为 18MIPS,语音 MOS 分为 3.9. 它的 6.4kbps 的编码器使用的是多脉冲最大似然量化(MP-MLQ)激励,而 5.3kbps 编码器使用的是代数码(ACELP)激励. 由于具有高音质和低码速率的特点, G. 723.1 被作为 PSTN 网上的可视电话的语音通信标准,并广泛应用于其他通信领域.

但是,无论是 G. 723.1 还是 G. 729 算法都远不能满足我们所承担的由国家 863 资助的 H/PC 语音压缩与重放的项目的要求. 联想 H/PC 采用的是国产的 RISC 芯片,其速度相当与 486DX66,而且只能支持定点运算. 另外,联想 II/PC 只有 16 兆 RAM,存储空间相当少. 这就要求我们在保证语音质量的前提下,研究一种低复杂度的语音编码器,而且为了节省录音后的语音的存储空间,该编码器需要有尽可能高的压缩比.

G. 723.1 算法远不能够在 H/PC 上实时地实现语音的压缩与重放,为此,我们提出了一种高音质、低复杂度的 5kbps 语音编码器——PS-ACELP (pre-search algebraic-code-excited linear-prediction; 预搜索代数码激励线性预测编码器). 该 5kbps 编码器的语音质量与 G. 723.1 的 5.3kbps 编码器的语音质量相当,算法复杂度仅为 7MIPS,约是 G. 723.1 算法复杂度的三分之一. 本

• 收稿日期: 1999-06-29; 修改日期: 2000-03-17

基金项目: 国家 863 高科技发展计划资助项目(863-306-03-02-2)

作者简介: 李锦宇(1974-),男,江西人,硕士生,主要研究领域为低比特率语音编码,语音信号处理,数字信号处理;王仁华(1943-),男,上海人,教授,博士生导师,主要研究领域为数字信号处理,语音通信,多媒体通信.

文将详细讨论 PS-ACELP 算法的低复杂度策略。

在 PS-ACELP 算法中,由于在非常耗时的开环基音预测、自适应码本搜索和固定码本搜索等方面采取了大量的预搜索算法,使得搜索局限在一个很小的范围内极大地降低了算法的复杂度。另外,PS-ACELP 算法还在自适应码本搜索时采用了递推算法,也使算法复杂度得到了降低。

PS-ACELP 算法已经成功地应用在 H/PC 上进行纯软件实现的语音压缩与重放方面,获得了非常好的实时录放效果。不仅如此,PS-ACELP 算法还可以用于 Voice mail 和 Internet Phone 等方面,具有广阔的应用前景。

1 PS-ACELP 编码器的低复杂度策略

高音质的 CELP 编码器的运算量主要集中在自适应码本搜索和固定码本搜索以及开环基音预测等处。下面,我们通过和 G. 723. 1 的 5. 3kbps 编码器的算法进行对比,介绍 PS-ACELP 编码器的低复杂度策略,并分析如何在降低复杂度的同时仍然保持较高的语音音质。

1.1 开环基音预测

为了减小自适应码本搜索的复杂度,首先要进行开环基音预测,获得开环基音 T_{op} ,使自适应码本搜索在开环基音值 T_{op} 附近进行。

基音搜索是将使
$$C_{ol}(j) = \frac{\left(\sum_{n=0}^{n=L} f(n)f(n-j) \right)^2}{\sum_{n=0}^{n=L} f(n-j)f(n-j)}$$
 最大的 j 值作为开环基音值 ($f(n)$ 是感觉加权

后的语音信号)。G. 723. 1 采用的搜索是对 $j \in [18, 142]$ 逐个进行搜索,这样的搜索算法是 CELP 基音预测的经典算法,但是运算量相当大。

1.1.1 PS-ACELP 对开环基音周期采取预搜索的算法

PS-ACELP 首先对感觉加权后的语音信号 $f(n)$ 进行预处理,然后再获得开环基音。具体算法如下:

(1) 将感觉加权后的语音信号 $f(n)$ 进行截止频率为 1KHZ 的低通滤波,然后将滤波后的信号进行 4:1 的抽样,获得信号 $f_s(n)$,采取低通滤波的目的是防止抽样后的信号混叠。

(2) 针对 4:1 抽样后获得的信号 $f_s(n)$,在 $j \in [5, 35]$ 范围内进行逐个搜索,选择使

$$C_{ol}(j) = \frac{\left(\sum_{n=0}^{n=L/4} f_s(n)f_s(n-j) \right)^2}{\sum_{n=0}^{n=L/4} f_s(n-j)f_s(n-j)}$$

最大的 j 作为预搜索的 index。

(3) 针对原始的感觉加权后的语音信号 $f(n)$,在 $j \in [4 * \text{index} - 3, 4 * \text{index} + 3]$ 范围内进行搜索,选择使

$$C_{ol}(j) = \frac{\left(\sum_{n=i}^{n=L} f(n)f(n-j) \right)^2}{\sum_{n=i}^{n=L} f(n-j)f(n-j)}$$

最大的 j 作为开环基音 T_{op} 。

由于进行了抽样,PS-ACELP 在第(2)步预搜索的运算量只是 G. 723. 1 开环基音搜索算法的

1/16. 这是因为需要搜索的点数由于 4:1 的抽样降为原来需要搜索点数的 1/4, 而每次需要计算判距的乘加数也由于 4:1 的抽样降为原来需要的 1/4, 因此总共需要的运算量为 G. 723. 1 算法的 1/16. PS-ACELP 的第(1)步的低通滤波和第(3)步的 7 次搜索的运算量也很小, 因此, PS-ACELP 的开环基音搜索具有比 G. 723. 1 算法低得多的运算量, 大约为 G. 723. 1 开环基音预测算法的 18%.

1.1.2 PS-ACELP 限制第 3 子帧开环基音搜索范围的算法

PS ACELP 算法与 G. 723. 1 算法一样, 将语音分成 4 个 7.5ms 的子帧, 只对第 1 和第 3 子帧进行开环基音预测. 由于基音频率的最大变化速率是 1%/ms, 第 3 子帧与第 1 子帧相差 15ms, 则基音频率最大相差 15%, 所以我们在 $[0.8T_{op}, 1.2T_{op}]$ (T_{op} 为第 1 子帧的开环基音) 的范围内进行第 3 子帧的开环基音搜索.

采用将第 3 子帧的开环基音的搜索限制在第 1 子帧开环基音的一个范围之内, 不但可以保证一定的语音基音周期的连续性, 还可以降低开环基音搜索的复杂度, 使得运算量最多为原来开环基音搜索运算量的 $(1+40\%)/2 * 100\% = 70\%$.

将开环基音预搜索和限制第 3 子帧开环基音搜索范围这两种快速算法结合起来, 则 PS-ACELP 在开环基音预测的算法复杂度为 G. 723. 1 相应算法的 $18\% * 70\% = 13\%$.

经过大量测听实验表明, 采用上述快速算法后合成的语音与 G. 723. 1 编解码后的语音没有什么质量上的差别.

1.2 自适应码本搜索

PS-ACELP 编码器的自适应码本搜索的基本思想与 G. 723. 1 是一致的, 根据开环基音周期 T_{op} 的数值, 在 $[T_1, T_1+3]$ ($T_1 - T_{op} - 1$) 之间进行自适应码本搜索. 自适应码本搜索采用 5 级基音预测^[2], 设 $x(n)$ 为自适应码本搜索的目标信号, $r(n)$ 为自适应码本贡献, $h(n)$ 为合成滤波器的冲激响应, M 为候选的基音周期, g_0, g_1, g_2, g_3, g_4 为自适应预测系数, 则 5 级基音预测的差错测度为

$$e(n) = x(n) - \sum_{k=0}^{k=n} h(k) \sum_{i=0}^{i=4} g_i r(n - (M + 2 - i) - k), 0 \leq n \leq L - 1,$$

其中 L 是进行自适应码本搜索的子帧长度. 在 G. 723. 1 和 PS-ACELP 算法中取 $L = 60$.

写成向量形式为

$$e = x - g_0 H r_0 - g_1 H r_1 - g_2 H r_2 - g_3 H r_3 - g_4 H r_4,$$

则平方误差为

$$E = e^T e = x^T x - 2C_M^T g.$$

其中

$$g = \begin{bmatrix} g_0, g_1, g_2, g_3, g_4, -0.5g_0^2, -0.5g_1^2, -0.5g_2^2, -0.5g_3^2, -0.5g_4^2, -g_0g_1, \\ -g_0g_2, -g_0g_3, -g_0g_4, -g_1g_2, -g_1g_3, -g_1g_4, -g_2g_3, -g_2g_4, -g_3g_4 \end{bmatrix},$$

$$C_M = \begin{bmatrix} x^T H r_0, x^T H r_1, x^T H r_2, x^T H r_3, x^T H r_4, r_0^T H^T H r_0, r_1^T H^T H r_1, \\ r_2^T H^T H r_2, r_3^T H^T H r_3, r_4^T H^T H r_4, r_0^T H^T H r_1, r_0^T H^T H r_2, \\ r_0^T H^T H r_3, r_0^T H^T H r_4, r_1^T H^T H r_2, r_1^T H^T H r_3, \\ r_1^T H^T H r_4, r_2^T H^T H r_3, r_2^T H^T H r_4, r_3^T H^T H r_4 \end{bmatrix}. \quad (1)$$

最小化平方误差 E 等价于最大化 $C_M^T g$. G. 723. 1 自适应码本搜索时, 将 5 级自适应预测系数进行量化. 量化码本拥有 N 个 20 维码矢量 (G. 723. 1 算法为 170 个, PS-ACELP 算法为 256 个), 这

个 20 维矢量就是上面的 g . 编码算法针对 $[T_1, T_1+3]$ 之间的每个基音值, 计算出矢量 C_M , 然后对码本中的每一个码矢 g , 计算出 $C_M^T g$, 选取相应于最大 $C_M^T g$ 的码字 g 和 M , 传送到解码端. 可以看出, 1 次自适应码本搜索需要计算 4 次矢量 C_M (从 T_1 到 T_1+3 各计算 1 次) 以及计算和比较 $4 * N$ 次 $C_M^T g$, 它们的计算量都极大. 自适应码本搜索占了 G. 723. 1 运算量中非常大的一部分.

1.2.1 PS-ACELP 计算 C_M 的递推求取

C_M 是一个 20 维矢量. 它可以看成由 3 部分组成: 第 1 部分是 60 维目标矢量 x 和 HR_i (60 维矢量, $i \in [0, 4]$) 的点积; 第 2 部分是 HR_i 的自相关点积; 第 3 部分是 HR_i 的互相关点积. 所以, 计算 C_M 的关键是要计算 60 维矢量:

$$Z_i = HR_i = \sum_{k=0}^{k=n} h(k)r(n-(M+2-i)-k), \quad 0 \leq i \leq 4, \quad 0 \leq n \leq L-1, \quad T_1 \leq M \leq T_2.$$

PS-ACELP 在计算 C_M 时采用了两种递推算法.

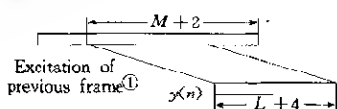
令 $y(n) = r(n-(M+2))$, 则 $Z_i = \sum_{k=0}^{k=n} h(k)y(n-k+i)$.

递推算法 1. 由于 $Z_i(n) = \sum_{k=0}^{k=n} h(k)y(n-k+i)$, 则

$$\begin{aligned} Z_{i-1}(n) &= \sum_{k=0}^{k=n} h(k)y(n-k+i-1) \\ &= \sum_{k=0}^{k=n-1} h(k)y(n-1-k+i) + h(n)y(i-1) \\ &= Z_i(n-1) + h(n)y(i-1). \end{aligned} \quad (2)$$

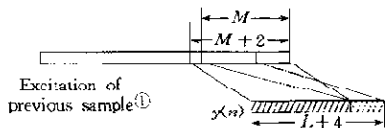
其中 $1 \leq n \leq L-1$. 如果已经获得了 $Z_i(n)$, $i=4, 0 \leq n \leq L-1$, 则对于每一个 $Z_i(n)$ ($i \in [0, 3]$) 的元素都可以通过式(2)简单地由一次乘法依次获得. 这样, 计算一次 $Z_4(n)$ 需要 1 830 次乘法运算, 计算 $Z_3(n), Z_2(n), Z_1(n), Z_0(n)$ 总共需要 $4 * 60 = 240$ 次乘法运算. 所以, 对于 $M \in [T_1, T_1+3]$, 每一子帧语音需要 $4 * (1830 + 240) = 8280$ 次乘法运算来计算 HR_i . 得到 HR_i 后, 接着利用式(1)来计算 C_M , 所以总共需要 $8280 + 4 * 20 * 60 = 13080$ 次乘法运算来计算进行一个子帧的自适应码本搜索所需要的 C_M .

递推算法 2. 设在闭环基音搜索时, 当候选周期为 M 时, 自适应码本贡献为 $r(n)$, 当候选周期为 $M+1$ 时, 自适应码本贡献为 $r'(n)$. 矢量 $y(n) = r(n-(M+2))$ 和矢量 $y'(n) = r'(n-(M-2))$ 是由上一帧的激励样本点构成的. 当 $M > L+2$ ($L=60$) 时, $y(n)$ 简单地复制距离上一帧尾部 ($M+2$) 样本点开始的 $L+4$ 个样本点的激励信号; 当 $M \leq L+2$ 时, $y(n)$ 以候选周期 M 为模, 复制距离上一帧尾部 ($M+2$) 样本点开始的样本点的激励信号. 图 1 说明了当 $M > L+2$ 时, 矢量 $y(n)$ 的获取. 图 2 说明了当 $M \leq L+2$ 时, 矢量 $y(n)$ 的获取.



①前一帧的激励信号.

Fig. 1 Obtainment of vector $y(n)$ when $M > L+2$
图 1 $M > L+2$ 时矢量 $y(n)$ 的获取



①前一帧的激励信号.

Fig. 2 Obtainment of vector $y(n)$ when $M \leq L+2$
图 2 $M \leq L+2$ 时矢量 $y(n)$ 的获取

所以, 当 $T_1 > 62$ 时, 对于候选周期为 M 的矢量 $y(n)$ 和候选周期为 $M+1$ 的矢量 $y'(n)$, 有 $y'(n+1) = y(n)$, 则递推算法为

$$\begin{aligned}
 Z'_i(n) &= \sum_{k=0}^{k=n} h(k)y'(n-k+i) = \sum_{k=0}^{k=n} h(k)y(n-1-k+i) \\
 &= \sum_{k=0}^{k=n-1} h(k)y(n-1-k+i) + h(n) * y(i-1) \\
 &= Z_i(n-1) + h(n) * y(i-1).
 \end{aligned} \tag{3}$$

其中 $1 \leq n \leq L-1$.

当 $T_1 \leq 62$ 时, 递推算法为

(a) $n \leq T_1 + 2$ 时,

$$Z'_i(n) = Z_i(n-1) + h(n) * y(i-1), \quad 1 \leq n \leq T_1 + 2. \tag{4}$$

(b) $n > T_1 + 2$ 时,

$$\begin{aligned}
 Z'_i(n) &= \sum_{k=0}^{k=n} h(k)y'(n-k+i) = \sum_{k=0}^{k=T_1+2} h(k)y'(n-k+i) + \sum_{k=T_1+3}^{k=n} h(k)y'(n-k+i) \\
 &= \sum_{k=0}^{k=T_1-2} h(k)y(n-1-k+i) + \sum_{k=T_1+3}^{k=n} h(k)y(n-1-k+i) + \\
 &\quad \sum_{k=T_1+3}^{k=n} h(k)y'(n-k+i) - \sum_{k=T_1+3}^{k=n} h(k)y(n-1-k+i) \\
 &= \sum_{k=0}^{k=n} h(k)y(n-1-k+i) - \sum_{k=T_1+3}^{k=n} h(k)y'(n-k+i) - \sum_{k=T_1+3}^{k=n} h(k)y(n-1-k+i) \\
 &= \sum_{k=0}^{k=n} h(k)y(n-1-k+i) + \sum_{k=0}^{k=n-T_1-3} h(n-k)y'(k+i) - \sum_{k=0}^{k=n-T_1-3} h(n-k)y(k+i-1) \\
 &= Z_i(n-1) + h(n) * y(i-1) + Z'_i(n-T_1-3) - Z_{i-1}(n-T_1-3),
 \end{aligned} \tag{5}$$

其中 $T_1 + 3 \leq n \leq L-1$.

利用式(3)~(5), 在求得 $M=T_1$ 时的 Hr_i 后, 可以递推地求出 $M \in [T_1+1, T_1+3]$ 时的 Hr_i . 再结合式(2), 只需要计算 $M=T_1$ 时的 Hr_i , 其他 Hr_i 均可以通过递推式(2)~(5)递推地获得.

现在考虑递推算法 2 运算量最大的情况. 从式(3)~(5)中可以看出, 当 T_1 为最小候选基音周期时, 计算矢量 Hr_i 的运算量最大. 在进行自适应码本搜索时, 最小候选基音周期为 18, 所以, 当 $T_1=18$ 时, 在获得相应的 Hr_i 后, $T_1=19, 20, 21$ 时的 Hr_i 可以分作两部分来计算:

(a) $0 \leq n \leq 20$ 时, 按照式(4), 每一个元素只需要 1 次乘加运算就可以获得.

(b) $21 \leq n \leq 59$ 时, 按照式(5), 每一个元素只需要 4 次乘加运算就可以获得.

因此, 总共需要 $21 + 39 * 4 = 177$ 次乘加运算来计算每个矢量 Hr_i .

所以, 一个子帧计算矢量 $Hr_i (0 \leq i \leq 4, 0 \leq n \leq L-1, T_1 \leq M \leq T_2)$ 最多需要 $(1830 + 60 * 4) + 3 * (177 + 60 * 4) = 3321$ 次乘加运算. 得到 Hr_i 后, 接着利用式(1)来计算 C_M , 所以总共最多需要 $3321 + 4 * 20 * 60 = 8121$ 次乘加运算来计算进行一个子帧的闭环基音搜索所需要的 C_M .

以上的 PS-ACELP 采用的递推算法是严格的数学推导, 这样做不会给算法的数值带来任何改动, 只会大大加快运算的速度. G. 723. 1 只采用了步骤 1 的递推算法, 所以 PS-ACELP 算法和 G. 723. 1 算法在计算 C_M 上的运算量之比 $\leq 8121 / 13080 * 100\% = 62\%$ (在递推算法 2 中运算量最大的情况下等号成立, 一般情况小于 62%).

1.2.2 PS-ACELP 自适应码本搜索的预搜索策略

PS-ACELP 算法将矢量 g 划分成 3 个矢量 g_1, g_2, g_{12} 之和: $g = g_1 + g_2 + g_{12}$.

$$g_1 = [0, g_1, g_2, g_3, 0, 0, -0.5g_1^2, -0.5g_2^2, -0.5g_3^2, 0, 0, 0, 0, -g_1g_2, -g_1g_3, 0, -g_2g_3, 0, 0]_j,$$

$$g_2 = [g_0, 0, 0, 0, g_4, -0.5g_0^2, 0, 0, 0, -0.5g_4^2, 0, 0, 0, -g_0g_4, 0, 0, 0, 0, 0],$$

$$g_{12} = [0, 0, 0, 0, 0, 0, 0, 0, 0, -g_0g_1, -g_0g_2, -g_0g_3, 0, 0, 0, -g_1g_4, 0, -g_2g_4, -g_3g_4].$$

在范围 $i \in [0, N_1], j \in [0, N_2], M \in [T_{op} - 1, T_{op} + 2]$ 内, 对 $g_{ij} = g_{1i} + g_{2j} + g_{12ij}$ 进行搜索, 将对应于 $C_M^T g_{ij}$ 最大的 i, j, M 作为搜索的结果.

如果我们对每一个 i 和 j 都进行搜索, 即采用全搜索的方法, 那么这个方法和 G. 723.1 的算法是一样的, 没有任何改进. 实际上, PS-ACELP 采用的搜索算法是先预搜索再进行精细搜索, 方法如下:

(1) 对所有候选的 $M \in [T_{op} - 1, T_{op} + 2]$ 进行预搜索, 寻找相应的 $i = \text{Index}(M), i \in [0, N_1 - 1]$, 使得 $C_M^T g_{1i}$ 最大.

(2) 对于每一对 M 和 $\text{Index}(M)$, 寻找相应的 $j \in [0, N_2]$, 获得

$$g_{\text{index}(M)j} = g_{1\text{index}(M)} + g_{2j} + g_{12\text{index}(M)j},$$

使得 $C_M^T g_{\text{index}(M)j}$ 最大.

(3) 通过比较, 获得对应于 $C_M^T g_{ij}$ 最大的 i, j, M 的值.

这是一种次最优的搜索算法, 它的搜索策略使得运算量下降为 $4 * (N_1 + N_2)$. PS-ACELP 算法中 $N_1 = 32, N_2 = 8$. 这样, 计算和比较 $C_M^T g$ 的次数为 $4 * (32 + 8) = 160$ 次, 不到 G. 723.1 算法的 1/4. 而且值得注意的是, 在本算法的第(1)步: 计算和比较 $C_M^T g_{1i}$ 时, 由于矢量 g_1 只有 9 个位置是非 0 值, 其实际的计算只是 9 维矢量的相乘而不像 G. 723.1 算法那样是 20 维矢量的相乘. 所以, PS-ACELP 算法的自适应码本搜索的运算量与 G. 723.1 算法的运算量之比为 $\frac{4 * (32 * 9 + 8 * 20)}{4 * 170 * 20} * 100\% = 13\%$, 而且由 g_1, g_2, g_{12} 构成的矢量 g 的码字数为 $32 * 8 = 256$ 个, 大于 G. 723.1 算法的 170 个, 因此为矢量的量化提供了更多的码字.

为了使预搜索算法取得更好的效果, 我们在设计码本时采用的是分级设计的策略. 首先设计一个 3 级基音预测器的量化码本, 这个码本用于 3 级基音预测, 这个码本有 $N_1 = 32$ 个码矢量. 然后在这个 3 级基音预测器的量化码本的基础上设计出 5 级基音预测器的量化码本, 这个码本有 $N_1 * N_2 = 256$ 个码矢量.

由以上分析可以明显地看出: 通过采用预搜索码字 g_1 , 再精细搜索码字 g_2 的矢量量化, PS-ACELP 的自适应码本搜索方法远小于 G. 723.1 的自适应码本搜索方法的复杂度. 我们在 PS-ACELP 编码器中分别采用该自适应码本搜索算法和 G. 723.1 的搜索算法, 发现重构的语音音质没有感觉上的区别.

综合 PS-ACELP 的递推算法和预搜索算法, PS-ACELP 的整个自适应码本搜索算法与 G. 723.1 算法相应部分的运算量之比大致为 32%.

1.3 固定码本搜索

PS-ACELP 的固定码本搜索算法也采取了预搜索算法, 使得固定码本搜索需要的运算量是 G. 723.1 算法相应部分运算量的 19%, 但这两种算法的语音质量没有听觉上的差别. 具体的预搜索算法我们在参考文献[3]中已经进行了详细的论述.

2 算法性能

我们不仅利用 PS-ACELP 算法在 H/PC 上实时地实现了语音的压缩与重放,而且将它 ADSP21060 上也实时地实现. 经过测试,它在 40MIPS 的 ADSP21060 上仅占用了约 7MIPS 的运算量. 我们在单片 ADSP21060 上设计了一个多路语音编解码系统,在实现 4 路 PS-ACELP 算法的同时,另外留下 12MIPS 的芯片处理余量可以灵活地用于实现自动回波抵消和加密解密等实时算法以及一些控制、通信功能.

我们组织了大量的人员对 PS-ACELP 算法和 5.3kbps 的 G.723.1 算法进行了 MOS 分的评定. 我们的测试数据是 8 男 8 女的 8KHZ 采样 16bit 量化的语音数据. 测试的结果是:PS-ACELP 算法的 MOS 分为 3.86,G.723.1 算法的 MOS 分为 3.90,两个算法的语音质量是相当的.

表 1 是 PS-ACELP 算法和 5.3kbps 的 G.723.1 算法的性能比较.

Table 1 Performance comparison of PS-ACELP and 5.3kbps G.723.1

表 1 PS-ACELP 算法和 5.3kbps 的 G.723.1 算法的性能比较

Algorithm ^①	Complexity ^②	Bit rate ^③	MOS ^④
PS-ACELP	7MIPS	5kbps	3.86
5.3kbps G.723.1	18MIPS	5.3kbps	3.90

①算法,②复杂度,③码速率,④MOS 分.

3 结束语

本文提出了一种高品质、低复杂度的 5kbps 语音编码器——PS-ACELP.

PS-ACELP 算法在 CELP 算法运算量极大的开环基音预测、自适应码本搜索和固定码本搜索等方面都采用了预搜索的搜索算法:首先进行预搜索,把最终的搜索局限在一个小的范围内,然后进行精细的搜索,这样就大大降低了搜索的运算量. 这种预搜索的方法是一种次最优的搜索算法,它虽然不能像 G.723.1 算法中的全搜索算法那样获得非常高的精度,但是大量测听实验表明,PS-ACELP 能够取得与 G.723.1 算法大致相当的语音质量.

另外,PS-ACELP 算法还在自适应码本搜索计算 C_M 时采用了两个递推算法,由于递推算法是严格的数学推导,这样做不会给算法的数值方面带来任何改动,只会大大降低运算的复杂度.

我们将 PS-ACELP 算法的自适应码本搜索、固定码本搜索、开环基音预测的运算量占 G.723.1 算法相应步骤的运算量的百分比列在表 2 中. 从表中可以看出,正是由于在 CELP 算法这些高复杂度的关键步骤方面,PS-ACELP 采取了预搜索快速算法,获得比 G.723.1 算法相应步骤低得多的运算量,才使得整个 PS-ACELP 算法的运算复杂度约为 7MIPS,远低于 G.723.1 算法的复杂度.

Table 2 Comparison of key step complexity

表 2 关键步骤运算量比较

Corresponding step complexity ratio of PS-ACELP to G.723.1 ^①	Open-Loop pitch estimation ^②	Adaptive codebook search ^③	Fixed codebook search ^④
	13%	32%	19%

①PS-ACELP 相对于 G.723.1 算法相应步骤运算量的百分比,②开环基音预测,③自适应码本搜索,④固定码本搜索.

由于 PS-ACELP 编码器具有高品质、低码率和低复杂度的特点,它不仅可以适用在 hpc 这种低档 CPU 上进行语音的纯软件实时压缩与重放,而且可以用于 Voice mail 和 Internet Phone 等方面,具有广阔的应用前景.

References:

- [1] ITU-T Recommendation G. 723. Dual rate speech coder for multimedia communications transmitting at 5.3&6.3 kbit/s. International Telecommunication Union, 1995.
- [2] Kleijn, W. B., Paliwal, K. K. Speech Coding and Synthesis. Amsterdam: Elsevier Science B. V., 1995. 95~102.
- [3] Li, Jin-yu, Wang, Ren-hua, Ge, Yong. An algorithm suitable for compressing and replaying speech on H/PC. In: Gao, Wen, *et al.*, eds. Proceedings of the 4th National Computer Intelligence Interface and Intelligence Application Conference. Beijing: Publishing House of Electronics Industry, 1999. 57~61 (in Chinese).

附中文参考文献:

- [3] 李锦宇, 王仁华, 葛勇. 一种适合于 H/PC 语音压缩与重放的编解码算法. 见: 高文, 等编. 第 4 届中国计算机智能接口与智能应用学术会议论文集. 北京: 电子工业出版社, 1999. 57~61.

Realization of Compressing and Replaying Speech on H/PC*

LI Jin-yu, WANG Ren-hua

(Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China)

E-mail: jyli@mail.ustc.edu.cn

http://www.ustc.edu.cn

Abstract: A novel speech codec of high quality and low complexity—PS-ACELP (pre-search algebraic-code-excited linear-prediction) is presented in this paper, which is suitable for software based compressing and replaying speech on H/PC. The proposed codec is almost equal in speech quality with ITU-T G. 723. 1 5.3kbps codec, but one third of the complexity of the latter, only around 7 MIPS. How to achieve the low computing complexity is described in detail in this paper. Due to the advantages of high speech quality, low complexity and low bit rate, PS-ACELP codec is suitable for software based compressing and replaying of speech, voice mail and Internet phone etc.

Key words: H/PC; PS-ACELP (pre-search algebraic-code-excited linear-prediction); low complexity; pre-search; open-loop pitch estimation; adaptive codebook search; fixed codebook search

* Received June 29, 1999; accepted March 17, 2000

Supported by the National High Technology Development Program of China under Grant No. 863-306-03-02-2