ACELP 算法在 PC 上实时播放的实现

李国汉，竺小松，刘 越
(解放军电子工程学院，合肥 230000)

摘要：对 TETRA 系统中所采用的 ACELP(Algebraic CELP)语音压缩算法进行了分析，该算法是基于 CELP 的改进算法，在语音编码中具有较强的代表性。在 Windows XP 环境下，利用 VS2005 实现将声卡采集到的数据进行压缩和解压缩，延时 10s 后通过声卡播放出来。实现过程用到许多 Windows 核心程序开发知识，并且给出了关键部分的源代码。对语音编码研究人员和程序员都有一定的借鉴意义。

关键词：ACELP 算法；语音质量检验；WaveX 函数
中图分类号：TB556 **文献标识码**：A **文章编号**：1000-3630(2010)-01-0060-03
DOI 编码：10.3969/j.issn1000-3630.2010.01.014

Real-time broadcasting of PC with ACELP algorithm

LI Guo-han, ZHU Xiao-song, LIU Yue
(*PLA Electronic Engineering Institute*, *Hefei* 230000, *China*)

Abstract: The ACELP (Algebraic CELP) algorithm used in the TETRA system is studied, which is a typical speech coding algorithm based on the improved CELP algorithm. In order to test the practical effect of this algorithm, the VS2005 is used to carry out the compression and decompression of the data acquired by sound card and play it back after 10s in the Windows XP environment. The whole process uses lots of the core Windows development knowledge and gives out the key part of the source code. This could be useful for speech coding researchers and programmers.
Key words: ACELP; voice quality testing; WaveX function

1 引 言

随着通信和互联网技术的发展，语音编码技术得到了快速的发展和广泛的应用。本文所研究的是 TETRA(Terrestrial Trunked Radio)系统中所采用的 ACELP 算法。TETRA 是欧洲电信标准协会 ETSI 制定的数字集群移动通信标准[1]。近几年发展迅速，已被许多国家所接受，并逐渐向世界标准迈进。

目前该算法已有许多在 DSP 上实现的方案[2,3]，但由于编译环境的限制，在 DSP 上进行软件的修改和优化较为复杂，因此本文直接利用 Windows 底层的 WaveX 函数，将采集到的语音利用该算法进行压缩和解压缩，再通过声卡输出设备播放出来，这样不仅可以直接实现算法的语音质量检验，也为后期的算法改进和软件优化提供了便利。

2 ACELP 算法分析

ACELP 是一种改进的代数码本激励线性预测算法，采用的是代数码本结构及聚焦搜索技术，提高了码本的灵活性和随机性，极大地降低了算法的复杂度。在线谱对(LSP)的转换和自适应生成方面也采用了有别于传统码本激励线形预测算法的新技术。ACELP 算法采用的输入语音采样率为 8kHz、每个样点用 16bits 进行量化，压缩后的输出为 137bits/s。压缩比约为 28。压缩部分主要分为三块：短时预测，主要提取声道滤波器的 LPC 系数并进行插值和量化；长时线性预测，分两部分：开环基音分析和闭环基音分析。前者用于获取基音周期的大致范围，后者获取基音周的准确值；码本搜索和增益计算主要用于计算合适的码本和增益，使重建语音的质量具有更高的清晰度和自然度。图 1 是该算法的原理图(压缩部分)。

ACELP 算法每帧长 30ms(240 个样点)，每帧语音分为 4 个子帧，对每个子帧求一次基音周期、自适应码本和增益值。G.929 中每帧为 10ms(80 个采样点)，每帧分为 2 个子帧，ACELP 和 G.723.1 中采用的算法较为相似，但压缩比比后者更大。

解码过程就是将编码比特流从各个参数中提取出来，重构输出信号和各个滤波器，最后得到重构语音。解码同样是通过分帧来完成的。首先将 LSP 矢量转换为 LP 滤波器 $A(z)$的系数，再通过基音

收稿日期：2008-10-22；**修回日期**：2008-12-30
作者简介：李国汉(1984-)，福建龙岩人，硕士研究生，研究方向为语音编码和扩频通信。
通讯作者：刘越，E-mail: ly.eei@hotmail.com

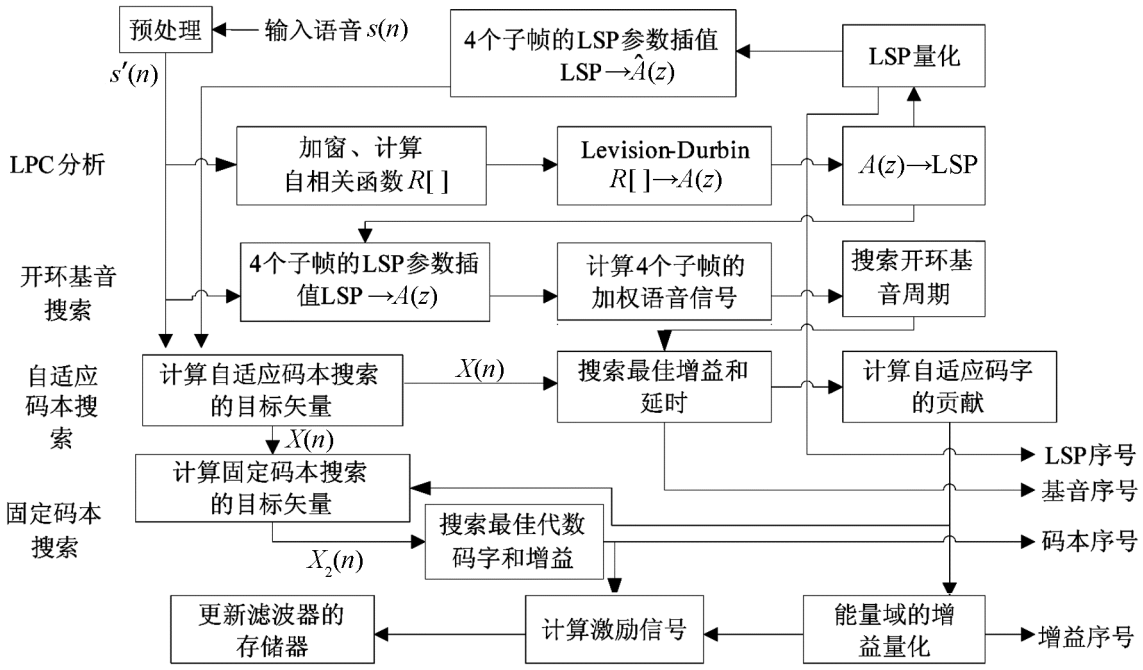

图 1　ACELP 压缩原理图

Fig.1　Schematic of ACELP compression

延时查找基音的整数和分数分量，将激励进行解码，对解码后的激励信号通过后基音滤波器送入合成滤波器，得到合成信号。合成信号经过共振峰后滤波器和增益放大单元控制得到输出的语音信号。

3　ACELP 算法在 PC 上的实时播放方案

3.1　函数简介和参数设定

采用 Windows 底层的 API 函数实现对声卡的控制。在 Windows 中提供了许多对声卡进行控制的函数，如 PlaySound、sndPlaySound、MCI 函数等，但上述函数仅能实现对文件的操作，如果想要实现语音的同步播放就必须实现对声卡缓冲区的操作。WaveX 函数可以直接设定声卡的采样率、声道数、语音量化的位数，并且可以具体到对播放缓冲区和录制缓冲区进行操作。为便于后面的介绍，现将要用到的函数和定义的变量进行介绍(见表 1 所示)。

表 1　程序中的函数和变量

Table 1 variables and function in the program

函数和变量	定义
waveInOpen	打开声卡录入设备
waveInPrepareHeader	初始化声卡，指定声卡采样的录入缓冲区等
waveInAddBuffer	将指定的缓冲区放入声卡录入设备
waveInStart	声卡开始录音
waveOutOpen	打开声卡输出设备
waveInPrepareHeader	初始化声卡，指定声卡播放的波特率和播放缓冲区等
waveInAddBuffer	将指定的缓冲区放入声卡播放设备
waveOutWrite	声卡开始放音
PWAVEHDR	语音缓冲区的头结构，可以定义缓冲区的长度等
WAVEFORMATEX	定义波形数据的格式，如波特率、声道数等
timeSetEvent	打开定时器，定时触发一个回调函数
Coder_Tetra	取两帧语音进行压缩
Decod_Tetra	对压缩语音进行解码

3.2　实现方案和关键代码

采用了多线程和定时器相结合的方案进行语音的采集、处理和播放。程序分为三个部分：主线程完成声卡设备的初始化，打开声卡录入设备，设置定时器，进行语音压缩的预处理，创建辅助线程；辅助线程延时 10s 后打开声卡播放设备；定时器中的回调函数每隔 30ms 将声卡录入设备中的两帧进行压缩和解压缩，然后放入播放缓冲区中。程序实现过程中遇到的问题及其解决办法：

(1) 语音采集、处理和语音播放之间必须满足一定时序关系，否则将不能够同步播放，或者播放的并非处理后的语音。 解决办法是播放线程开始 80ms 后(此时声卡缓冲区中至少已有两帧数据)，定时器中的回调函数从声卡录入缓冲区中取两帧数据进行压缩和解压缩，将处理后的数据放入声卡的播放缓冲区中，定时器每 30ms 触发回调函数运行一次，这样录入缓冲区的数据会不断地经过处理后放入播放缓冲区，10s 后打开声卡播放设备。

(2) Windows 操作系统是非实时的，提供的许多定时器是基于软件的，精度往往比较低，例如 Sleep 函数在定时精度小于 100ms 时，就会出现较大的偏差。若定时器偏快或偏慢，都会出现飘移，使得所取数据或是提前声卡录入设备，或者滞后于声卡播放设备。本文采用的是多媒体定时器 timeSetEvent，该定时器是获得硬件支持的定时器，在采用之前进行过验证。每 30ms 用该定时器触发回调函数，将当前时钟的运行时间(单位为 ms)输出到一个记事本中，经过连续 10min 的测试，记事本中的数据几乎和计算结果完全一致，偏差在 1ms 内，这是由于回调函数的处理是基于消息的，并非实时的，因此，定时器的计数是非常准确的。

(3) 通常进行声卡录音都是将数据放入 wave 文件中，但 wave 文件的头结构不是太容易懂，而且将数据放入文件，再从文件中取数据会增加不必要的开销，本文仅需要将数据进行处理后播放出来，并不需要保存数据。因此，本文直接将声卡录入设备中的数据拷贝到一个 960byte(两个语音帧)的数组中，处理后再放入声卡播放缓冲区中。

(4) WaveX 函数对声卡的控制是基于消息处理机制的，在声卡打开、关闭、缓冲区满等情况发生时，都会向操作系统发消息，必须及时捕获消息，并且进行相应的处理。在下面的程序中会介绍到消息映射和处理。

在实现声卡控制和语音编码的处理过程中，定义了许多变量和函数，为了便于介绍，仅以声卡播放设备的打开、消息处理为例进行介绍。在语音编码中省略了预处理和后处理等过程，仅以 Coder_Tetra 和 Decod_Tetra 代表语音的压缩和解压缩过程。下面是部分程序代码：

```
//打开声卡录入设备。hWaveIn 为声卡录入设备的句柄，waveform 为定义的波形数据的格式。
//psavebuffer1 为语音录入缓冲区，pWaveHdr1 为录入缓冲区的头结构
if (waveInOpen(&hWaveIn, WAVE_MAPPER, &waveform, (DWORD)this→m_hWnd, NULL, CALLBACK_WINDOW))
{    free(pSaveBuffer1);
     MessageBeep(MB_ICONEXCLAMATION);
     AfxMessageBox("Audio can not be open!");
     return;
}
waveInPrepareHeader(hWaveIn,pWaveHdr1,
     sizeof(WAVEHDR));
waveInAddBuffer(hWaveIn,pWaveHdr1,sizeof(WAVEHDR));
waveInStart(hWaveIn);
//开辟辅助线程，打开定时器
//ThreadFuc1 用于延时 10s，打开声卡播放设备
CreateThread(NULL,0,&ThreadFuc1,(LPVOID)0,0,NULL);
if((Retimer=timeSetEvent(30,1,(LPTIMECALLBACK)
     PrintTime,(DWORD)1,1))= =NULL)
{    printf("timer cannot be open!");
     return;
}
//回调函数，pSaveBuffer 为 960byte(两个语音帧)的数组，pSaveBuffer2 为播放缓冲区
void CALLBACK TimerProc(HWND hWnd,UINT message,
     UINT nTimerid,DWORD dwTime)
{    for(static int j=0;;j<200000; j++)
     {  Coder_Tetra(ana);
        Decod_Tetra(parm, synth);
        CopyMemory(pSaveBuffer2+j*480,synth,480);
        CopyMemory(pSaveBuffer,pSaveBuffer1+j*480,960);
        j++;
        if(j==20000)
           j=0;
           retun;
     }
}
//以声卡中的缓冲区满为例，介绍消息的声明和处理
ON_MESSAGE(MM_WIM_DATA,OnMM_WIM_DATA); //将消息映射到函数 OnMM_WIM_DATA
waveInAddBuffer(hWaveIn,(PWAVEHDR)lParam,
   sizeof(WAVEHDR));
```

4 验证和总结

以上程序在 Windows XP 环境下，利用 vs2005 编译通过，实现了预期的目的。经过主观试听验证，合成语音较好地保留了讲话人的特征，无背景噪声时，合成语音的清晰度和自然度都比较好。对 TETRA 系统中采用的 ACELP 算法进行了分析，利用 Windows 中底层的 WaveX 函数实现了语音的采集、压缩和解压缩，并在延时 10s 后将处理过的语音通过声卡播放出来，检验了该算法语音编码的质量，为以后算法的改进、程序代码的优化和程序向 DSP 中的移植打下了很好的基础，对语音编码的研究人员具有一定的实用价值。

参 考 文 献

[1] ETSI EN 300 395-2 V1.31 (2005-01) Terrestrial Trunked Radio (TETRA) Speech codec for full-rate traffic channel Part 2: TETRA codec.

[2] MSDN Library for Visual Studio 2005.

[3] 周敬利, 赵冕, 郭红星. G.729 语音编码器在 DSP 上的实时实现[J]. 微处理器, 2007.8.
ZHOU Jingli, ZHAO Mian, GUO Hongxing. The real-time realization of G.729 voice coder in the DSP[J]. Microprocessors, 2007.8.