

# 语音采集压缩在 Windows Mobile 移动终端的研究与实现

俞柏锋, 江 涛, 葛宝忠

(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

**【摘 要】**文中研究了语音采集压缩技术在 Windows Mobile 中的应用, 使用 Windows API 封装的录音函数采集语音, 采用 G. 726 语音压缩编码技术压缩, 最后, 基于微软最新的 mobile 6.0 平台在智能手机中实现。实验结果表明, 压缩后的语音数据量是先前的 1/8, 而且通过解压后的语音质量和原来的基本一致, 从而为语音在移动智能终端上的结合提供了一种较好的方法。

**【关键词】**语音采集; Windows Mobile; G. 726; 智能手机

**【中图分类号】** TN911.7

**【文献标识码】** A

**【文章编号】** 1002-0802 (2008) 03-0115-03

## Research and Realization of Audio Collection Compression in Windows Mobile Terminal

YU Bai-feng, JIANG Tao, GE Bao-zhong

(National Digital Switching System Engineering & Technological R&D Center, Zhengzhou Henan 450002, China)

**【Abstract】** This paper studies the application of the audio compression technology in Windows Mobile. Windows API functions for recording is used to collect audio signals, G. 726 audio compression coding is adopted to compress, and the latest Microsoft mobile 6.0 platform is realized in the smart phone. It provides a fairly good method for the audio application on the mobile terminals.

**【Key words】** audio collect; windows mobile; G. 726; smart phone

### 0 引言

语音通信是现代通信中最简便、实用的通信方式之一, 人们无时无刻都在利用各种媒体传递语音数据信息量, 因而在很多场合都需要用到语音数据, 对语音数据的采集目前都是基于 Windows API 中封装的录音函数进行的, 但由于经过这一级 PCM 编码调制采集下来的语音数据一般都比较小, 大概有 1M 左右, 这个数据量对于带宽有限的网络传输是无法接受的, 所以对原有的数据必须进行压缩。

Windows Mobile 系统是微软为智能移动终端设备使用的操作系统, 发展至今已有 11 年历史。Windows Mobile 的核心是 Windows CE。基于 Windows Mobile 操作系统的智能终端设备<sup>[1]</sup>分 Smart Phone 和 Pocket PC 两大类, 其中 Smart Phone 为智能手机系统, Pocket PC 则为个人掌上电脑, 现在在很多 Pocket PC 也加入电话功能成为 Pocket PC Phone。

微软为上述智能移动终端设备提供了基于 Pocket PC2003、Smart Phone2003、Mobile 5.0 的开发平台, Mobile 6.0 是最新的开发平台, 安装完 Visual Studio2005 开发工具后, 再安装 Mobile 6.0 的 SDK 就可以开发基于 Windows Mobile 的应用程序。

ITU-T G. 726 语音压缩编码—自适应差分脉冲编码调制 (ADPCM) 是波形编码中非常有效的一种数字编码方式。在电信领域, 因为它利用了语音信号样点间的相关性, 并针对语音信号的非平稳特点, 使用自适应预测和自适应量化, 在保证可接受恢复语音质量的同时, 能有效降低比特流数。

### 1 语音采集

文中的语音采集是通过麦克风将声音录到相关设备中, 录音是把语音通过录音设备进行 PCM 编码调制成二进制数据

收稿日期: 2007-09-12。

作者简介: 俞柏锋 (1982—), 男, 硕士研究生, 主要从事移动通信、嵌入式开发应用方面的研究工作; 江 涛 (1974—), 男, 讲师, 主要从事嵌入式系统开发应用工作; 葛宝忠 (1965—), 男, 硕士研究生导师, 教授, 主要从事移动通信系统研究工作。

存放至内存，在录音停止后，应该可以通过回放来检查录音的效果，录音可以调用 Windows API 来实现，Windows 中自带的录音机就是通过这种方法实现的。

### 1.1 录音流程

一个简单的录音流程为：打开录音设备 waveInOpen→准备 wave 数据头 waveInPrepareHeader→准备数据块 waveInAddBuffer→开始录音 waveInStart→停止录音 waveInReset→关闭录音设备 waveInClose。

相应的回放录音数据的流程为：打开回放设备 waveOutOpen→准备 wave 数据头→waveOutPrepareHeader→写 wave 数据 waveOutWrite→停止放音 waveOutReset→关闭回放设备 waveOutClose。

### 1.2 录音格式设置

录音的格式使用一个结构来设置，此结构如下：

```
typedef struct {
    WORD wFormatTag;
    WORD nChannels;
    DWORD nChannels;
    DWORD nSamplesPerSec;
    WORD nBlockAlign;
    WORD wBitsPerSample;
    WORD cbSize;
} WAVEFORMATEX;
```

定义 WAVEFORMATEX waveform, 设定音频格式为 WAVE\_FORMAT\_PCM, 即令 waveform.wFormatTag=WAVE\_FORMAT\_PCM, 通道数为 1, wave.nChannels=1, 采样频率 wave.nSamplePerSec=8000, 每个采样点需要的总字节数 wave.nBlockAlign=2, 每个采样点需要比特数 wave.wBitsPerSample=16000, 该结构除上面变量外还可以增加的字节数。

### 1.3 录音处理的消息响应函数

在录音程序的设计中，每当录音函数被调用时，相应的消息会产生，对不同的消息进行处理可以完成录音的控制。MM\_WIM\_OPEN: 当录音设备被打开时发出；MM\_WIM\_DATA: 当缓存已满或停止录音时发出；MM\_WIM\_CLOSE: 当录音设备关闭时发出；MM\_WOM\_OPEN: 当录音设备被关闭时发出；MM\_WOM\_DONE: 停止放音时发出；MM\_WOM\_CLOSE: 录音设备关闭时发出。本程序的关键是要对各种消息进行处理完成模块功能，主要对消息 MM\_WIM\_DATA 进行分析处理。

## 2 语音压缩

G. 726—ADPCM 语音压缩编解码器<sup>[2-3]</sup>完成对 64 kb/s A 率或  $\mu$  率脉冲编码调制 (PCM) 信号和 40 kb/s, 32 kb/s, 24 kb/s 或 16 kb/s ADPCM 信号之间的转换。图 1 给出了 ADPCM 编解码原理的简化框图。

以编码函数<sup>[4]</sup>为例引用如下：

[g726lib.lib]

```
Extern void g726_Encode(unsigned char * speech, char
*bitstream)
{
    g726_state state_ptr;
    short temp[480];
    int i;
    g726_init_state(&state_ptr);
    memcpy(temp, speech, 960);
    for(i=0;i<120;i++)
    { *(bitstream+i)=(((char)g726_16_encoder(temp[i*4],
AUDIO_ENCODING_LINEAR,&state_ptr)))
    }
}
```

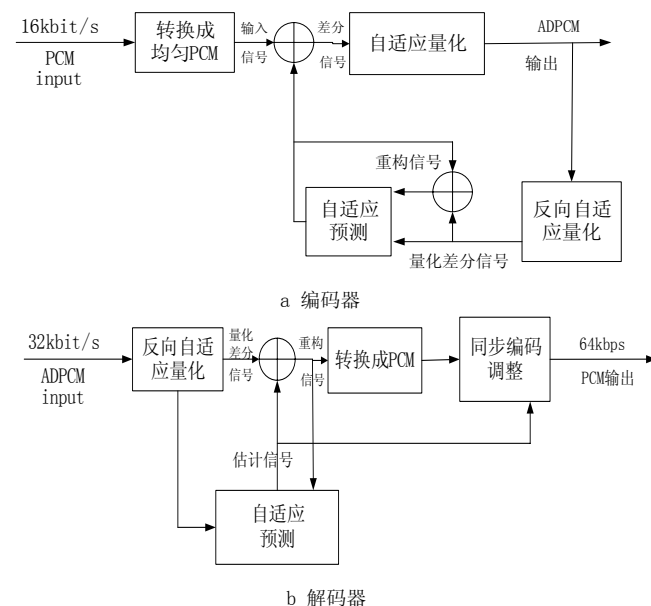


图 1 ADPCM 编解码器

## 3 Windows mobile 上的实现

开发工具为 Visual Studio2005, 安装了 mobile 6.0SDK 工具包，先选择模拟器上仿真实现，最后通过 Microsoft ActiveSync 工具连接智能手机(本例中使用多普达的 P800W), 最终结果表明本程序能很好的在智能终端上实现录音—压缩—输出或解压播放的功能，主要处理代码是针对录音消息 MM\_WIM\_DATA 进行处理，完整的处理流程如图 2 所示。

MM\_WIM\_DATA 消息中的关键代码如下。

```
LRESULT CMyRecordDlg::OnMM_WIM_DATA(UINT wParam,
LONG lParam)
{
    DWORD dwBytesWritten, dwBytesWritten1, wBytes
Written2;
    char pin[1920], pout[240], dpout[1920];
    //pout[240] 编码输出缓存
    char pin1[960], pout1[120];
```

```

//pin1[960],pout1[120]临时存放使用
    int iOut=120;
    int iIn=1920;
    LPWAVEHDR pHdr = (LPWAVEHDR) lParam;
    memcpy(pin, pHdr->lpData, pHdr->dwBytesRecorded);
    if(m_fh) //输出原始文件 1. dat
        WriteFile(m_fh, pin, pHdr->dwBytesRecorded,
&dwBytesWritten, NULL);
    memcpy(pin1, pin, 960); 编码到缓存 pout[240]
    g726_Encode((unsigned char *)pin1, pout1); //编码
前字节
    memcpy(pout, pout1, 120);
    memcpy(pin1, pin+960, 960);
    g726_Encode((unsigned char *)pin1, pout1); //编码
后字节
    memcpy(pout+120, pout1, 120); //编码结果存放
于 pout[240]中
    if(m_fh1)
        WriteFile(m_fh1, pout, 240, &dwBytesWritten1, ULL);
// 输出压缩文件 2. dat
    memcpy(pout1, pout, 120);
    g726_Decode(pout1, (unsigned char *)pin1); //译码
前字节
    memcpy(dpout, pin1, 960);
    memcpy(pout1, pout+120, 120);
    g726_Decode(pout1, (unsigned char *)pin1); //译码
后字节
    memcpy(dpout+960, pin1, 960);
//译码结果存放于 dpout[1920]之中
    if(m_fh2)
        WriteFile(m_fh2, dpout, iIn, &dwBytesWritten2, ULL);
    .....

    CopyMemory (pSaveBuffer + dwDataLength, dpout, //
将每次录音压缩且解压后的数据保存到
    ((PWAVEHDR) lParam)->dwBytesRecorded); // pSaveBuffer
所指向的内存中, 以便回放
    dwDataLength += ((PWAVEHDR) lParam)->dwBytes
Recorded ;
    .....

    return a;
}

```

需要注意的是在 Windows Mobile 中处理的录音播放控制函数中有些函数的返回类型和数据类型与在其他平台下的情况有所不同, 比如在 MM\_WIM\_DATA 消息响应函数中, 必须有一个 LRESULT 类型的返回值, 其他不同的数据格式可以借助 Visual Studio 2005 下的帮助文档适当地转换和处理。录音

中的缓存至少准备两个, 以保证连续录音, 每段缓存可以适当大些, 由于本文中还要对每段录音的数据进行压缩, 而压缩是通过分段进行的, 输入 960 个字节数据, 压缩后输出 120 个字节数据, 因此输入的缓存字节必须是 960 的倍数, 考虑到复杂性和消息响应连续录音的时间中若处理压缩的时间过长, 则会导致语音压缩时消息响应无法返回, 不能实现连续的录音, 经过数次实验比较, 本文采用了每段 1920 个字节的缓存大小, 结果表明能很好的实现录音数据的压缩处理和播放。

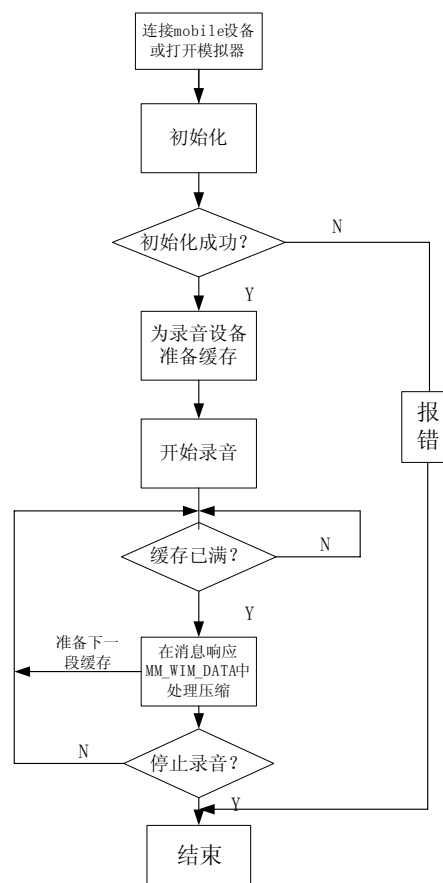


图 2 完整的处理流程示意

## 4 结语

文中就 Windows Mobile 设备采集语音和压缩处理的过程和方法进行了分析, 并给出了具体的实现, 在移动智能终端的使用越来越广泛的情况下, 研究基于 Mobile 的开发具有很实际的意义, 同时, 掌握对音频数据的处理, 可以在很多领域得到应用, 基于音频的移动开发具有很广阔的开发前景。

## 参考文献

- 1 傅曦, 齐宇, 徐骏. Windows Mobile 手机应用开发[M]. 北京: 人民邮电出版社, 2005: 5-27.
- 2 杨行峻等. 语音信号数字处理[M]. 北京: 电子工业出版社, 1995: 90-96.
- 3 王炳锡. 语音编码[M]. 西安: 西安电子科技大学出版社, 2002: 45-66.
- 4 陈健. 语音信号压缩编码[J]. 电子技术, 1998, (02): 39-42.