


MP3编码分析

原创xiahouzuoxin于 2012-08-11 14:21:02 发布阅读量1.9w收藏 47点赞数 7

分类专栏:Image/Audio/ML...

Image/Audio/ML... 专栏收录该内容

4 订阅43 篇文章

目录

一、MP3文件格式解析....2

1、MP3文件及MPEG概述...2

二、MP3编码原理....4

1、MP3编码流程...4

2、子带 滤波器 排——编码流程图中编号为1.....5

3、改良后的DCT（MDCT）——编码流程图中编号为2.....7

4、声音心理学模型——编码流程图中编号为3.....8

5、位元分配、量化和Huffman编码——4.....12

三、SHINE程序分析....13

1、文件数据结构...13

2、编码前化工作...14

3、MP3编码...14

4、后处理...17

注：下面的资料参考网上论文整理而来

一、MP3文件格式解析

1、MP3文件及MPEG概述

MP3文件是由帧(frame)构成的，帧是MP3 文件最小的组成单位。MP3 的全称应为MPEG1 Layer-3 音频文件。

MPEG(MovingPicture Experts Group)，MPGE音频层指MPGE文件中的声音部分，根据编码质量和复杂程度分为3层，即Layer-1、Layer2、Layer3，对应MP1、MP2、M文件。

2、MP3文件结构

MP3文件分为TAG_V2(ID3V2)，Frame, TAG_V1(ID3V1)共3部分。


ID3V2	包含了作者，作曲，专辑等信息，长度不固定，扩展了 ID3V1 的信息量。
Frame	一系列的帧，个数由文件大小和帧长决定
•	每个 FRAME 的长度可能不固定，也可能固定，由位率 bitrate 决定
•	每个 FRAME 又分为帧头和数据实体两部分
•	帧头记录了 mp3 的位率，采样率，版本等信息，每个帧之间相互独立
Frame	
ID3V1	包含了作者，作曲，专辑等信息，长度为 128BYTE。

(1) Frame格式

FRAMEHEADER	CRC (free)	MAIN_DATA
4 BYTE	0 OR 2 BYTE	长度由帧头计算得出

帧头为4个字节，其结构如下

```
typedef FrameHeader{
```

xiahouzuoxin 关注

7 7

```
unsigned intsync:11;    //同步信息

unsigned intversion:2;    //版本

unsigned intlayer:2;    //层

unsigned intprotection:1;    // CRC校验

unsigned intbitrate:4;    //位率

unsigned intfrequency:2;    //采样频率

unsigned intpadding:1;    //帧长调节

unsigned intprivate:1;    //保留字

unsigned intmode:2;    //声道模式
unsigned int mode extension:2; //扩充模式

unsigned intcopyright:1;    // 版权

unsigned intoriginal:1;    //原版标志

unsigned intemphasis:2;    //强调模式
```

```
}HEADER, *LPHEADER;
```

无论帧多长，每帧播放时间为26ms。MAIN_DATA长度为

```
Length(MAIN_DATA)=((version==MPEG1)?144:72)* bitrate / frequency + padding;
```

(2) ID3V1格式

ID3V1存放在MP3文件结尾，共128Bytes，各项信息都顺序存放，不足部分使用'\0'补足，可使用UltraEdit打开查看。

```
typedef struct tagID3V1
{

char Header[3]; /*标签头必须是"TAG"否则认为没有标签*/
char Title[30]; /*标题*/
char Artist[30]; /*作者*/
char Album[30]; /*专集*/
char Year[4]; /*出品年代*/
char Comment[28]; /*备注*/
char reserve; /*保留*/
char track;; /*音轨*/
char Genre; /*类型*/

}ID3V1,*pID3V1;
```

(3) ID3V2格式

ID3V2存放在MP3文件的首部，由1个标签头和若干标签帧组成。

标签头为10个字节，


```
char Header[3];    /*必须为"ID3"否则认为标签不存在*/
char Ver;    /*版本号ID3V2.3 就记录3*/
char Revision;    /*副版本号此版本记录为0*/
char Flag;    /*存放标志的字节，这个版本只定义了三位，稍后详细解说*/
char Size[4];    /*标签大小，包括标签头的10 个字节和所有的标签帧的大小*/
```

每个标签帧都有一个10个字节的帧头和至少一个字节的固定长度的内容组成为，帧头的定义如下：

```
char FrameID[4]; /*用四个字符标识一个帧，说明其内容，稍后有常用的标识对照表*/
char Size[4];    /*帧内容的大小，不包括帧头，不得小于1*/
char Flags[2];    /*存放标志，只定义了6 位，稍后详细解说*/
```

MP3编码原理

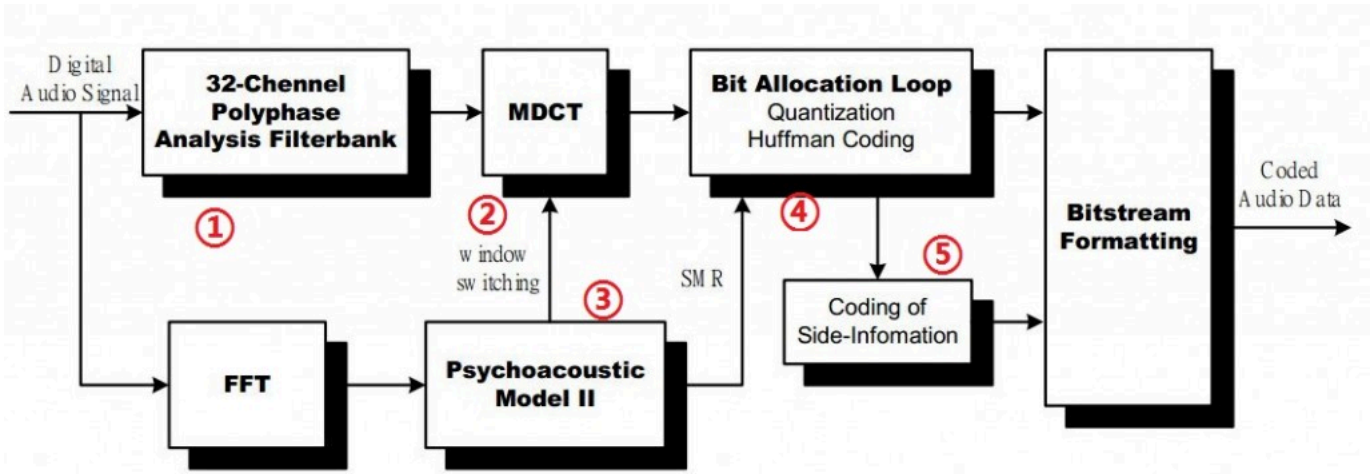
1、MP3编码流程

xiahouzuoxin

关注

7





MP3编码流程图

信号描述

- (1) MP3编码输入信号：PCM（Pulse Code modulation）声音信号，有些.wav格式的音频文件为PCM信号。
- (2) MP3编码输出信号：MP3格式码流

偏移地址	字节数	类型	内容
00H~03H	4	字符	资源交换文件标志（RIFF）
04H~07H	4	长整数	从下个地址开始到文件尾的总字节数
08H~0BH	4	字符	WAV文件标志（WAVE）
0CH~0FH	4	字符	波形格式标志（FMT）
10H~13H	4	整数	过滤字节（一般为00000010H）
14H~15H	2	整数	格式种类（值为1时，表示数据为线性PCM编码）
16H~17H	2	整数	通道数，单声道为1，双声音为2
18H~1BH	4	长整数	采样频率
1CH~1FH	4	长整数	波形数据数据传输速率（每秒平均字节数）
20H~21H	2	整数	数据的调整数（按字节计算）
22H~23H	2	整数	样本数据位数

wav文件头格式

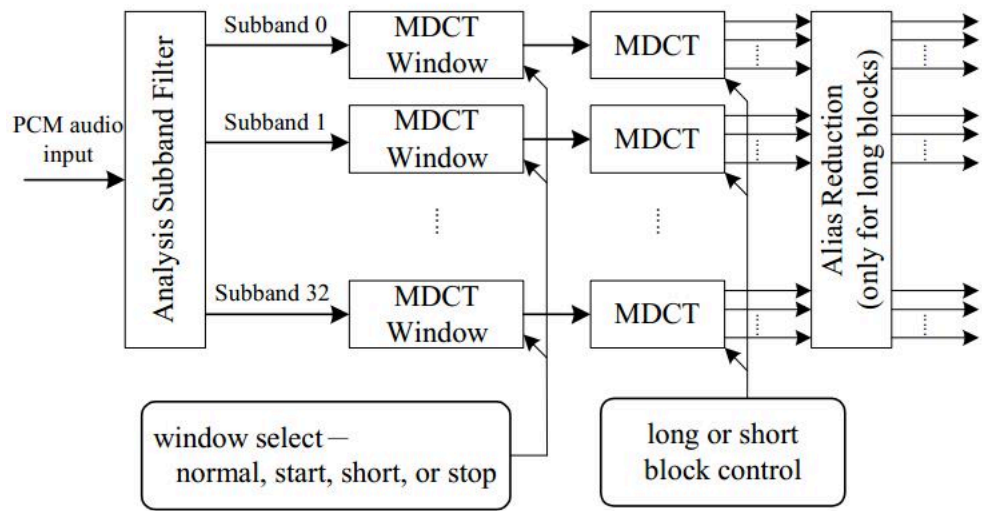
偏移地址	字节数	类型	内容
24H~27H	4	字符	数据标志符（data）
28H~2BH	4	长整型	采样数据总数
2CH...	...		采样数据

wav数据块

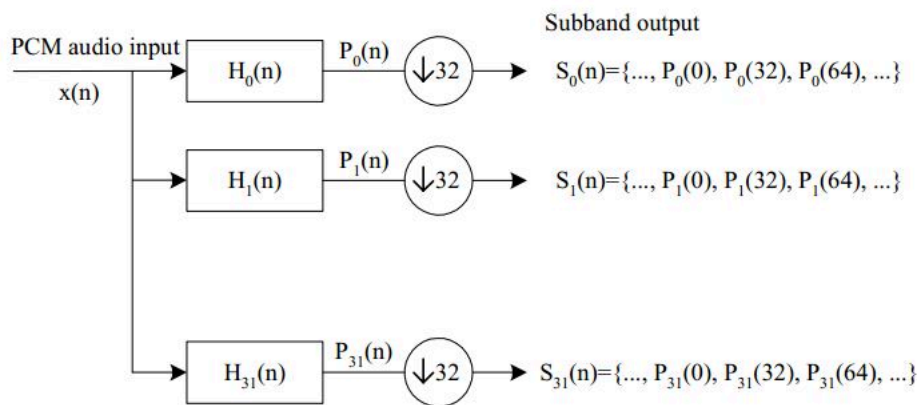
WAV格式文件所占容量= (取样频率X 量化位数X 声道)X 时间/ 8 (字节= 8bit)。

14H~15H的2个字节值为1时表示数据位PCM编码格式，可以作为MP3编码器的输入。

2、子带滤波器排——编码流程图中编号为1



子带滤波器及MDCT处理

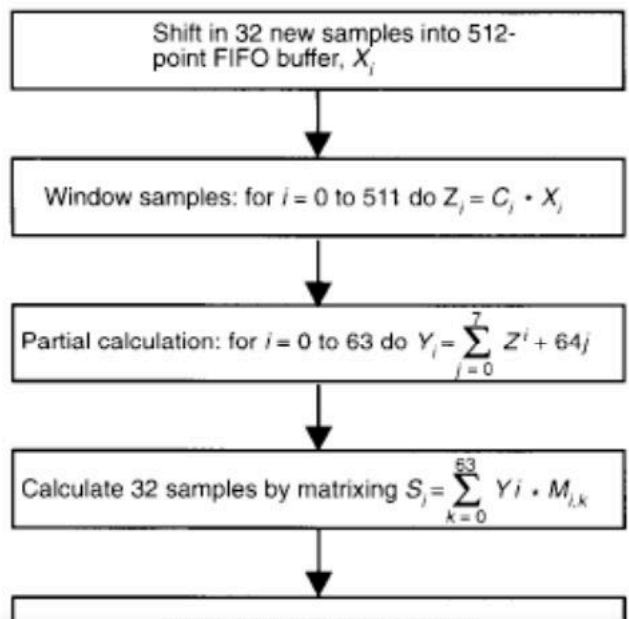


子带滤波过程

为多重相位滤波器，将PCM信号输入后，滤波器系统看做线性系统，则有

再将32点进行下采样，得子带滤波器输出结果为

ISO-M标准给出了如下图所示的实现方法，我们的程序将按照该实现方法编写。



xiahouzuoxin

关注

7

ISO-M给出的滤波器组的实现

将上图整个流程综合用表达式表示为

其中

为分析矩阵的系数。是窗函数的系数，共512个点，其值在ISO11172-3标准的 ANNEX_C.DOC文档中给出了，为子带序列号，范围为0~31，为第i个子带的样点，且t是取t倍。

3、改良后的DCT（MDCT）——编码流程图中编号为2

DCT变换的目的：进一步提高频谱解析度，将每一个子频带细分为18个次频带。

在正式DCT运算前，需要对子带信号进行加窗处理，有如下4中窗框，长窗框（Normal Window）、长短窗框（Start Window）、短窗框（Short Window）与短长窗框（Stop Window）。长窗框具有高的频谱解析度，短窗框的时间解析度比较高。

然后进行DCT变换，变换表达式 如下，

为DCT变换前加窗处理后的结果，如果加短窗框，则DCT运算中的，若加长框，则DCT运算中的。

在提高频域解析度的同时，加长框后会有假象（混叠现象）产生，因为不同子带之间存在混叠的信号，加长框会对混叠的信号当做该子带内正常信号处理，一种避免的办法是减小信号的强度。

噢，现在我们是知道了，所谓改良的DCT只不过是，（1）DCT变换前加窗处理（2）DCT变换（3）长框假象的处理 这3个过程。

到这里，还有一个问题，那就是窗框的选择问题，该如何选择窗框？窄的窗框具有好的时间分辨率，宽的窗框具有好的频率分辨率。我们回到编码流程图，

请注意图中的标号3，3表示声音心理学模型，窗框的宽窄选择与声音心里学模型相关，下面来分析该模型。

4、声音心理学模型——编码流程图中编号为3

研究声音心理学模型用途有：

- （1）研究模型的PE值决定做MDCT变换时使用长窗框还是短窗框
- （2）研究模型的SMR值决定量化编码时的比特数分配

现在不明白以上2条用途没关系，我们先来分析几个重要的概念。

- （1）SPL(Sound Pressure Level)，表示声音强度的名词，SPL是评价听觉刺激强度的标准，也就是说，我们对外界声音的感觉强度完全由它决定，其单位为dB。
- （2）静音门槛曲线

横轴为f(HZ)，纵轴为SPL(dB)，若声音强度（SPL）低于该曲线的值表示人听不到声音，如下图所示。从图中可以得出几条结论：

第一，人的听觉频率范围大约在10Hz~20KHz之间

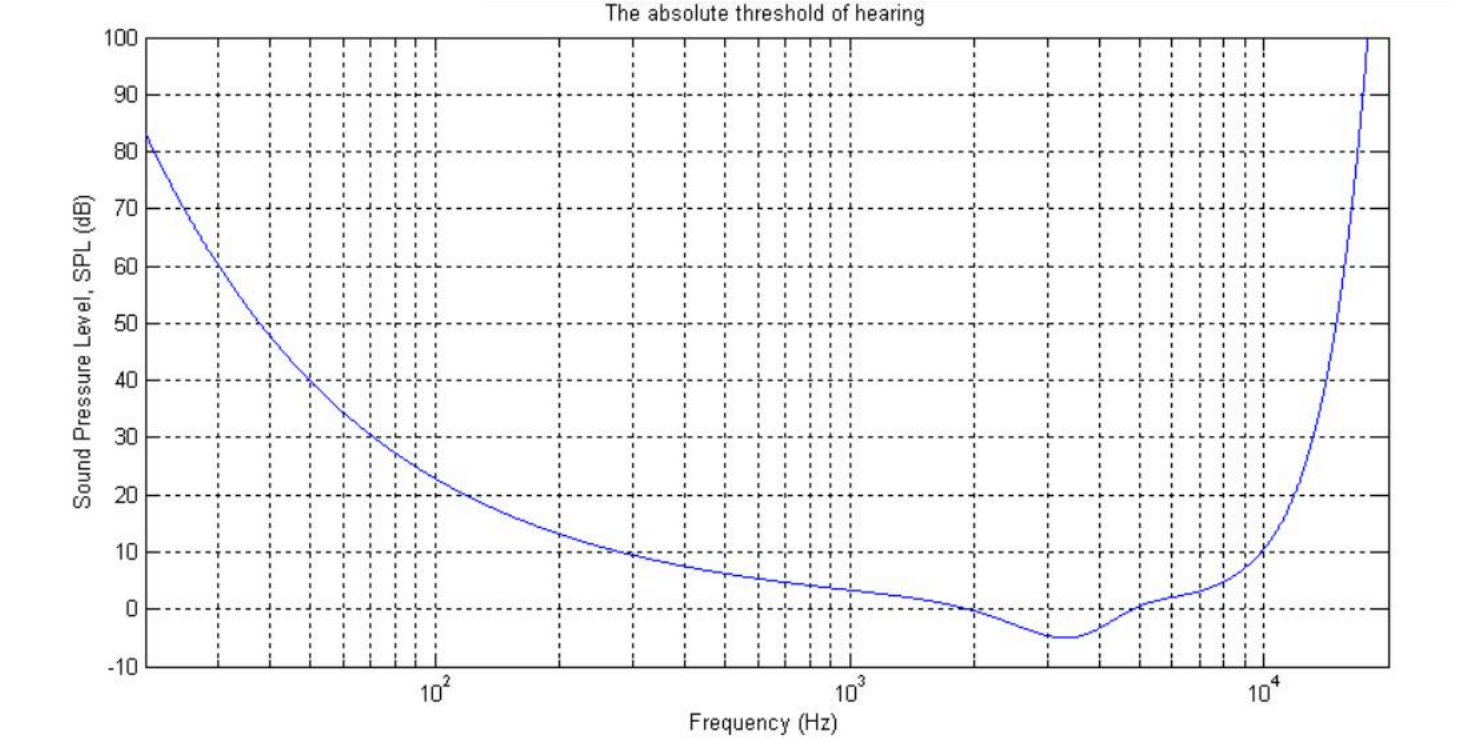
第二，大约在3KHz到4KHz时SPL有最小值，也就是所人在该频率范围内的听觉最敏锐



xiahouzuoxin

关注

7



(3) 临界频带（Critical Bands）

因为人耳对不同频率的敏感程度不同，MPEG1/Audio将22KHz范围内可感知的频率范围划分为23~26个临界频带，如下图。

Band No.	Center Freq. (Hz)	Bandwidth (Hz)	Band No.	Center Freq. (Hz)	Bandwidth (Hz)
1	50	– 100	14	2150	2000 – 2320
2	150	100 – 200	15	2500	2320 – 2700
3	250	200 – 300	16	2900	2700 – 3150
4	350	300 – 400	17	3400	3150 – 3700
5	450	400 – 510	18	4000	3700 – 4400
6	570	510 – 630	19	4800	4400 – 5300
7	700	630 – 770	20	5800	5300 – 6400
8	840	770 – 920	21	7000	6400 – 7700
9	1000	920 – 1080	22	8500	7700 – 9500
10	1175	1080 – 1270	23	10500	9500 – 12000
11	1370	1270 – 1480	24	13500	12000 – 15500
12	1600	1480 – 1720	25	19500	15500 –
13	1850	1720 – 2000			

表 2-1 理想的临界频带

从表中能得出几条结论：

第一，当中心频率值在500Hz以内时，不同临界频带的带宽()几乎相同，约100Hz

当中心频率值大于500Hz后，随着f值得上升，临界频带的带宽剧增

第二，从表中也可以看出，人耳对低频的解析度要比高频更好

(4) 频域上的遮蔽效应

SPL较大的信号容易掩盖频率相近的SPL较小的信号，叫声音的遮蔽效应。就比如在机场很难听到打电话的声音。

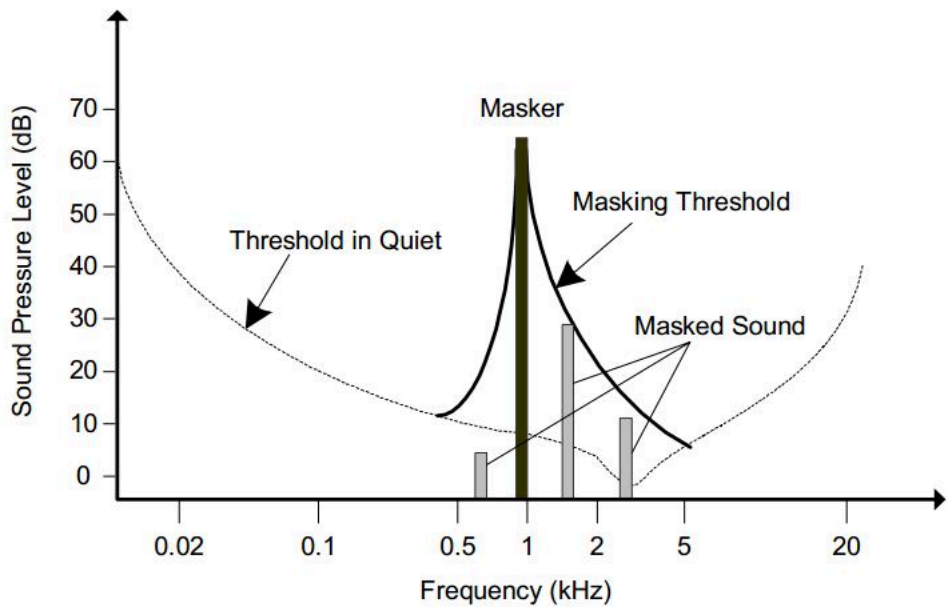


圖 2-9 遮噪門檻曲線與靜音門檻曲線

如上图所示，Masking Threshold将大约在0.7kHz，1.6kHz和2.3kHz的信号遮蔽了，当然0.7kHz信号的SPL在静音门檻曲线之下，不被遮蔽也是听不到的。

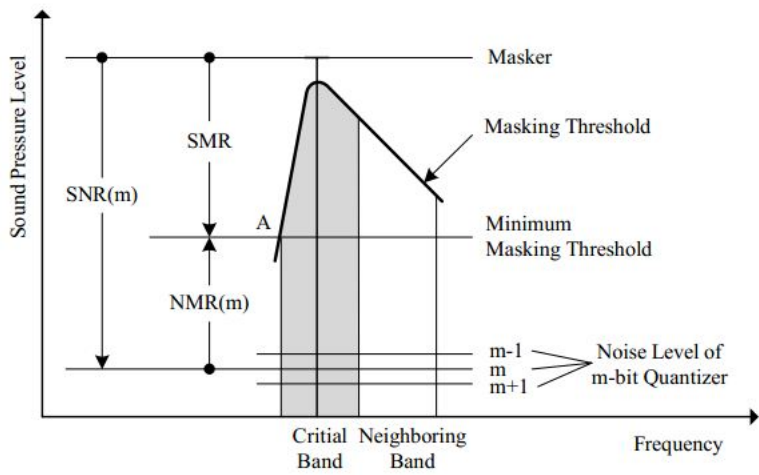
在这里，涉及3个重要的量——SMR、SNR和MNR。

SMR(signal-to-maskratio)：指在一个临界频带内，从masker到遮噪门檻值的距离。

SNR(signal-to-noiseratio)：指信号经过m位元量化后的信噪比，等于量化前信号方差和量化噪声的方差之比，。

MNR(mask-to-noise)：用来测量人耳可以感知的失真参数，

如下图所示，展示了三者之间的关系，其中的灰色区域Critical Band指临界频带，Masking Threshold就是遮噪门檻曲线，图中的SMR指在临界频带内最大的SMR值。



遮噪门檻曲线和SMR、SNR、NMR

值得注意的是，（1）我们上面讨论的SMR、SNR和MNR三者都是基于临界频带的，但遮蔽效应不仅对临界频带有影响，对临近的临界频带也有影响，称为遮噪延展性（2）的是一个临界频带内的一条遮噪曲线，实际情况存在多条遮噪曲线，结果是这些曲线的叠加。

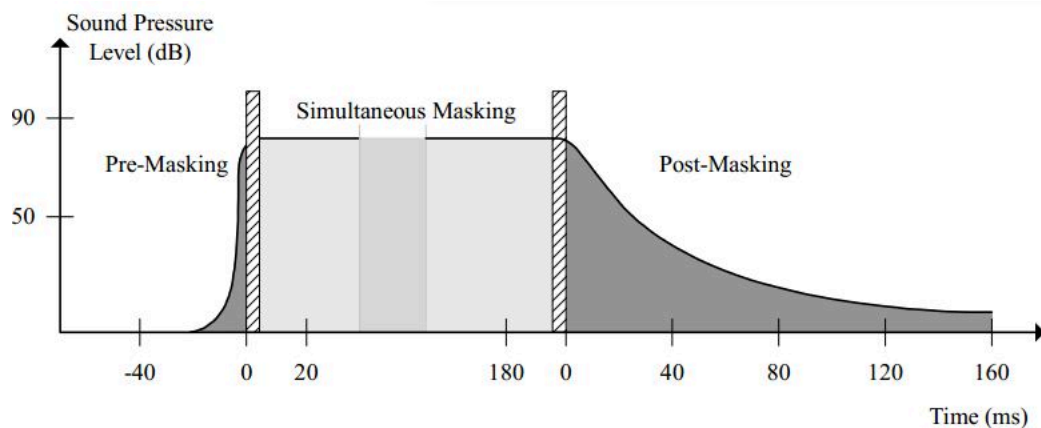
（5）时域上的遮噪曲线



xiahouzuoxin

关注

7



从上图可以看出，在一段很短的时间内（200ms左右），若出现了两个声音，不管出现的先后顺序，SPL大的声音(masker)会遮蔽SPL小的声音(maskee)。

若maskee出现在前，则遮噪曲线如上图的Pre-Masking；若maskee出现在后，则遮噪曲线如上图中的Post-Masking。由图中很容易看出，Post-Masking要比Pre-Masking长很多。Pre-Masking能遮蔽前回音，这是选择MDCT窗口的一个依据。

(6) 感知熵Perceptual Entropy (PE)

最重要的一点，PE能显示特定信号在理论上的压缩极限。PE的单位是bits/sample，代表每个取样在维持CD音质的情况下，能够压缩到的最低位元数。

重新回到本小节一开始就提到的声音心里学模型的用途，重新列一下：

(1) 研究模型的PE值决定做MDCT变换时使用长窗框还是短窗框

(2) 研究模型的SMR值决定量化编码时的比特数分配

对于第一条，MP3中定义，当 $PE > 1800$ 时，使用短窗框的MDCT来处理该grannul（MP3每个数据帧包含2个grannul，每个grannul包含 18×32 个subband采样）的号。因为当 $PE > 1800$ 表示这段音讯变化比较大，可能产生回音，不适合使用长框。

对于第二条，下面的位元分配将给出解释。

5、位元分配、量化和Huffman编码——4

(1) 位元分配

位元分配目的是使每个频带的MNR达到最大，使音质最佳。过程为：寻找最小的MNR频带，分配位元给该频带以提高MNR，接着重新计算各频带的MNR。重复上述过程，配结束。

[a]可编码位元数计算方法，1152指每个编码框的取样个数，

比如，以单声道为例，比特率为128kbps，采样频率为44.1kHz，则每个编码框可编码的位元数为3344。但考虑到挡头的32位，附属资料的136位和可选择的16位CRC，所以的位元数为 $3344 - 32 - 136 - 16 = 3160$ ，实际编码最小单位为grannul，所以每个grannul可用位元数为 $3160 / 2 = 1580$ 。

[b]MNR计算方法，在前面心理学模型中已经提到，

其中SMR由声音心理学模型提供，SNR信噪比则是由量化确定的。

(2) 非均匀量化

上式为MP3量化的公式，其中为MDCT输出并调整后的值，为量化后的整数，0.75是为了使量化器提供一致的SNR值，表示四舍五入。

下图为量化器的输入输出曲线，量化器的输入为浮点值频率，输出为整形值的频率。

由图知，量化器将输入的浮点值量化后变为整型值，且量化过程为非线性非均匀的。

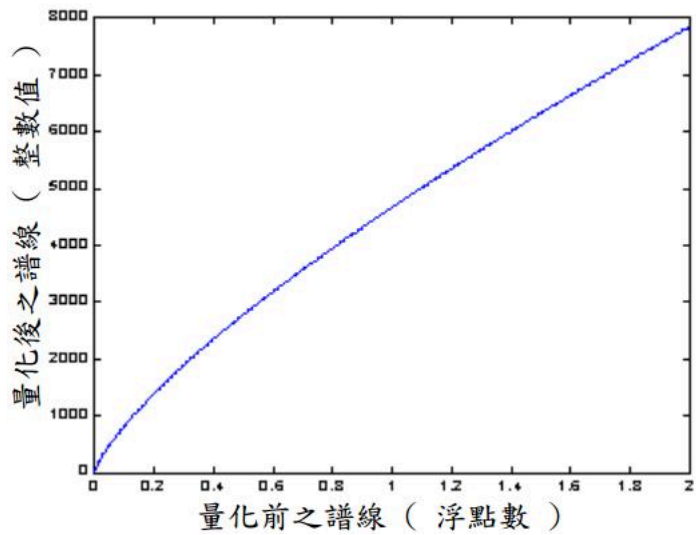


xiahouzuoxin

关注

7





频谱量化器的输入输出

三、SHINE程序分析

SHINE是一个C语言编写的MP编码程序，总共由11个源文件构成。将源文件添加到VC新建的控制台应用程序中即可运行，但运行时得使用命令行方式。

C Source (8)

bitstream.c	2003/6/23 19:15	C Source	15 KB
coder.c	2004/3/18 20:14	C Source	5 KB
huffman.c	2001/6/14 10:35	C Source	14 KB
layer3.c	2004/3/18 20:14	C Source	5 KB
loop.c	2003/6/23 19:14	C Source	16 KB
main.c	2004/3/18 20:59	C Source	6 KB
utils.c	2012/8/8 19:39	C Source	1 KB
wave.c	2012/8/8 20:16	C Source	5 KB

C/C++ Header (3)

table1.h	2003/6/20 16:16	C/C++ Header	17 KB
table2.h	2003/6/20 16:16	C/C++ Header	41 KB
types.h	2012/8/8 20:20	C/C++ Header	5 KB

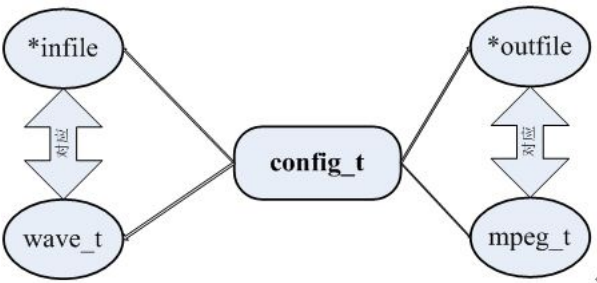
1、文件数据结构

在types.h中定义了一个config_t的结构体类型，并用它初始化了一个全局变量config，该变量作用相当于面向对象语言中的“对象”，用于在整个编码流程中对编码数据和参

管理。

定义了一个用于存储PCM脉冲格式文件信息的结构体类型wave_t，并且用wave_t在config_t中定义了wave变量，该变量保存了MP3编码的源的信息，作为MP3编码器的输

定义了一个用于存储MP3编码后信息的结构体类型mpeg_t，同样用mpeg_t在config_t中定义了mpeg变量，该变量存储的信息作为编码后的MP3参数信息输出。



```
typedef struct {  
    time_t start_time; /*记录编码起始时间*/  
    char* infile; /*编码输入文件*/  
    wave_t wave; /*PCM文件头信息*/  
    char* outfile; /*编码输出码流文件*/  
    mpeg_t mpeg; /*MP3文件头信息*/  
}
```

xiahouzuoxin

关注

7

```
} config_t;
```

以上的结构体主要用于保存“头”信息，编码输出后的字节流实体信息保存在bs结构体（bitstream.h文件中定义）中，bs结构体定义为

```
static struct
```

```
{
```

```
FILE *f; /* bitstream output file handle */
```

```
unsigned int i; /* file buffer index */
```

```
unsigned char *b; /* buffer pointer */
```

```
} bs;
```

结构体中的文件指针与指向同一个输出文件，b指向编码后的码流，编码结束后写入文件中。

2、编码前化工作

包括初始化config.mpeg结构体变量的默认值，打开wave类型文件（SHINE程序中此时只读取了文件头信息，没读取实体信息），根据读取的信息对mpeg输出信息进行配置

3、MP3编码

MP3编码主要由3步组成，分析子频带滤波器组，MDCT变换到频域，位元分配与量化。每次的操作对象为输入PCM的一帧。

MP3编码核心调用

```
/* polyphase filtering */
```

```
for(gr=0; gr<config.mpeg.granules; gr++)
```

```
for(ch=0; ch<config.mpeg.channels; ch++)
```

```
for(i=0; i<18; i++)
```

```
L3_window_filter_subband(&buffer[ch], &l3_sb_sample[ch][gr+1][i][0], ch);
```

```
/* apply mdct to the polyphase output */
```

```
L3_mdct_sub(l3_sb_sample, mdct_freq);
```

```
/* bit and noise allocation */
```

```
L3_iteration_loop(mdct_freq, &side_info, l3_enc, mean_bits);
```

```
/* write the frame to the bitstream */
```

```
L3_format_bitstream(l3_enc, &side_info);
```

(1) 子频带滤波器

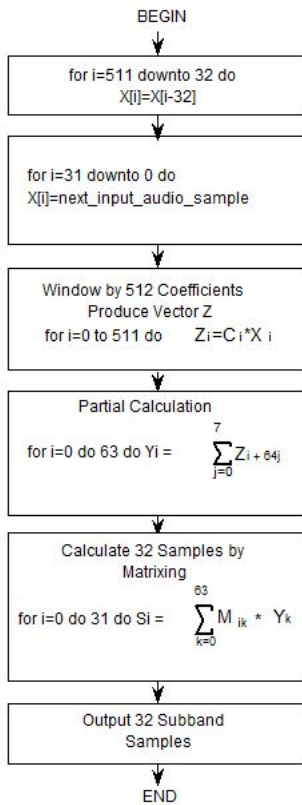
下图是ISO11172-3标准给出的**Analysis subband filter flow chart**。其步骤为



xiahouzuoxin

关注

7



【a】输入32个音频samples

【b】建立一个数组x[n], for n=0~511用于保存输入的采样值。将x[n]看做一个最多能容纳512个元素的队列，x[511]为队首，x[0]为队尾，每次接收新的samples前将队首32出，将samples放入队尾。

【c】加窗滤波器处理，窗函数系数为C[i], i=0~512，通过实现窗函数滤波器

【d】计算64个Yi值，表达式如流程图所示

【e】计算32个子带滤波器采样值Si，这里使用到矩阵M[i][k]，

$$M[i][k] = \cos[(2i+1)(k-16)\pi/64],$$

for i = 0 to 31, and k = 0 to 63.

实际计算时可以将非线性的运算用查Table的方法以减小运算的复杂度。

当然，在SHINE程序中，作者对M[i][k]×Y[k]的运算做了简化，主要从两方面：

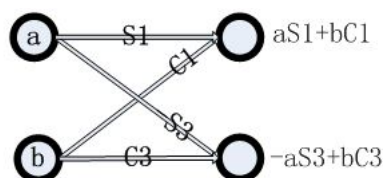
第一，cos函数在k=16和k=48处的对称性；第二，从各滤波器的相关性考虑，即cos(2i+1)的对称性考虑。

(2) MDCT实现

DCT（离散余弦变换）的原始表达式为

$$F(u) = a_0 c(u) \sum_{x=0}^{N-1} f(x) \cos \frac{(2x+1)u\pi}{2N}, \text{ for } u = 0, 1, \dots, N-1$$

DCT可以通过蝶形运算提高运算效率，具体内容可参考数字信号处理教材的内容。



FFT运算的蝶形图

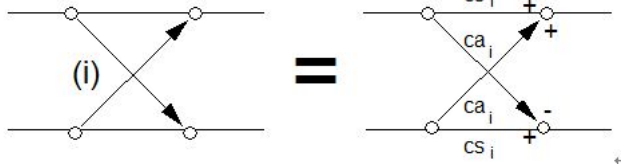
MDCT的表达式为



xiahouzuoxin

关注

7



MDCT快速算法蝶形图

计算一样可以通过蝶形图运算来提高效率，蝶形运算中最重要的是系数值，使用短窗框的MDCT运算点数为12，长窗框则为36。

ISO 11172-3的ANNEX_AB.DOC文档中Table 3-B.9给出了蝶形运算的系数如下

Table 3-B.9 Layer III coefficients for aliasing reduction:

(i)	ci
0	-0.6
1	-0.535
2	-0.33
3	-0.185
4	-0.095
5	-0.041
6	-0.0142
7	-0.0037

蝶形系数csi和cai通过下面2个式子计算

$$cs_i = \frac{1}{\sqrt{1 + c_i^2}} \quad ca_i = \frac{c_i}{\sqrt{1 + c_i^2}}$$

带假象处理的MDCT/IMDCT编解码图

32个频带每个频带的蝶形运算都需要8次，SHINE中蝶形运算的程序如下

```
for(band=31; band--;)
for(k=8;k--;)
{
/* must left justify result of multiplication here because the centre
* two values in each block are not touched.
*/
bu = muls(mdct_enc[band][17-k],cs[k])+ muls(mdct_enc[band+1][k],ca[k]);
bd =muls(mdct_enc[band+1][k],cs[k]) - muls(mdct_enc[band][17-k],ca[k]);
mdct_enc[band][17-k] = bu;
mdct_enc[band+1][k] = bd;
}
```



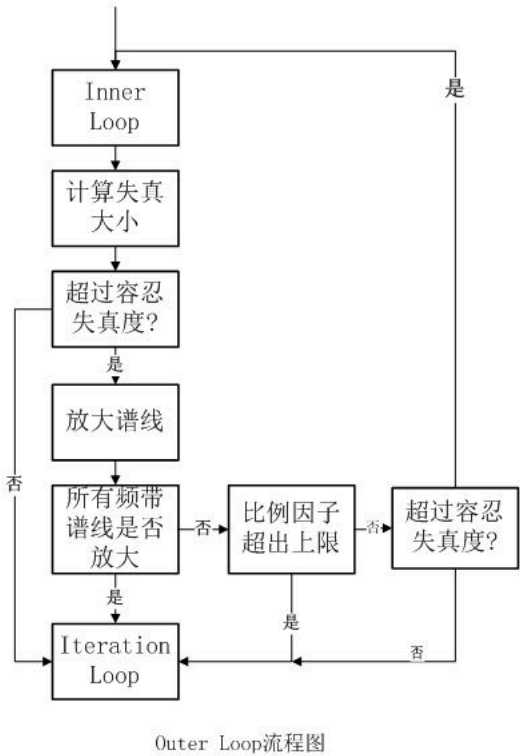
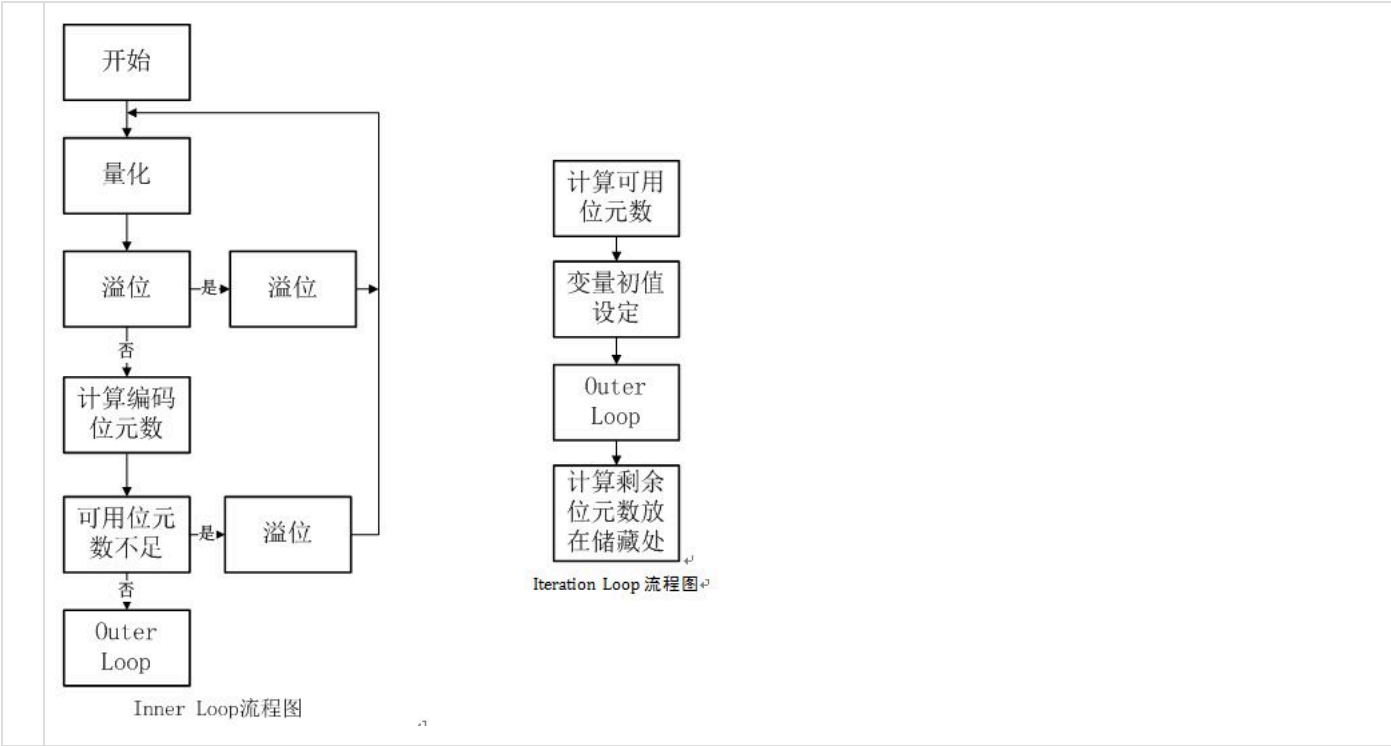
xiahouzuoxin 关注

7

(3) Huffman编码与位元分配

在经过声音心理模型和分析滤波器排之后提供信息之后，就可以对Audio进行位元分配和量化编码了。SHINE程序在该部分做了很多工作，位元分配和量化编码通过3个回圈[a]Iteration Loop [b]Outer Loop [c]Inner Loop。且[a]包含[b]，[b]包含[c]。

[c]主要完成量化工作，[b]计算量化后失真大小以决定是否需要重新量化，[a]计算剩余位元数并放在储存处。



位元分配与量化是整个编码过程中计算量最大的部分，是整个编码过程的核心。

【1】外部回圈分析

根据下面的公式计算失真度大小



di



xiahouzuoxin

关注

$$high(band) (|xr(i)| - ix(i))^{4/3} * \sqrt[4]{2^{stepsize}}$$

7

对于量化误差大于人耳所能最大可容忍失真的情况，则需将最大的可容忍失真度放大并且将每个未量化前的频带xr[i]放大。

【2】 内部回圈分析

首先，根据下面公式进行量化

$$ix(i) = n \operatorname{int}((\frac{|xr(i)|}{\sqrt[4]{2}^{stepsize}})^{0.75} - 0.0946)$$

若出现溢出或者位元数不够分配的情况，则按+1逐渐增大stepsize，直到前述情况不存在。

接着，计算编码所需位元数，通过位元数选择Huffman码表。

4、后处理

将结果写入比特流中，关闭PCM文件和码流存储文件。计算整个编码过程算法运行时间end_time -=config.start_time。

MP3文件格式与编码原理解码流程详解

水龙吟的备

1 文件格式 MP3文件格式四部分，按顺序排列如下： ID3V2 包含了作者，作曲，专辑等信息，长度不固定，扩展了ID3V1的信息量 Frame 音频帧序列 APEV2 包含了作者，

音频格式之MP3：(2)MP3编解码原理详解

littlezls的计

本文主要介绍mp3文件的存储格式，以及mp3编解码原理

11 条评论



huashuicaigou 热评 非常好

常见音频编码格式解析_一帧声音包含什么代码

1.MP3编码格式 1.1.MP3概述 MPEG-1 or MPEG-2 AudioLayerIII是一种音频压缩技术,其全称是动态影像专家压缩标准音频层面3(Moving Picture Experts Group Audio Layer

Mp3解码算法流程_mp3算法

不过MP3对音频信号采用的是有损压缩方式,为了降低声音失真度,MP3采取了“心理声学模型”,即编码时先对音频文件进行频谱分析,然后再根据心理声学模型把谱线分成若干

编码与解码 最新发布

scj1022的计

如果每个国家都定义一套自己的编码标准，结果相互之间谁也不懂谁的编码，就无法进行很好的沟通交流。所以 ISO（国际标准化组织）定义一套编码方案来解决所有国家

MP3文件格式解析 热门推荐

李世平的计

MP3文件格式解析Peter Lee 2008-06-05 目录一、概述... 二、整个MP3文件结构... 三、MP3帧格式... 1. 帧头格式... 2. MAIN_DATA.. 四、ID3标准... 1. ID3V1. 2. ID3V2. 五、

各种音频编码方式详解_音频编码方式有哪些

MP3(MPEG-1 audio layer 3) 类型:Audio 制定者:MPEG 所需频宽:128~112kbps(压缩10~12倍) 特性:编码复杂,用于互联网上的高质量声音的传输,如MP3音乐压缩10倍,2声

MP3文件格式解析_mp3 格式解析

MP3 文件是由帧(frame)构成的,帧是MP3 文件最小的组成单位。MP3 的全称应为MPEG1Layer-3 音频文件,MPEG(Moving Picture Experts Group)在汉语中译为活动图像专

MP3编码原理概述[转]

技术点亮人生，成功通向

音频压缩由编码和解码两个部分组成。把波形文件里的数字音频数据转换为高度压缩的形式(称为比特流)即为编码；要解码则把比特流重建为波形文件。 音频压缩可以

MP3编解码（MP3 encoder）

典型的Mp3编解码器源码，是学习音频编解码的良好样例

MP3解码算法原理解析_mp3编解码

频率反转:对逆向离散余弦变换的输出值中的奇数号子带(0到31号子带中的1,3,5,...,31)中的奇数号样本值(每个子带中的0到17号样本值的1,3,5,...,17号样本值)进行反相处理,用

文件名及mp3标签乱码问题_mp3 标题 怎么改编码格式

二。mp3标签 只需要把mp3标签里面用gbk、gb18030、big5等编码存储的中文内容修改为Unicode编码,那么基本上所有Linux下的播放器都能正常识别mp3标签了。关于r

LAME3.92经典MP3编码器

LAME（Low Altitude Military Encoder）是一款开源的MP3编码器，因其卓越的音质和高效率性能而备受赞誉。在音乐制作、音频编辑和数字音频处理领域，LAME3.92版本：

MP3CodeEncode.rar_mp3编码VC

MP3编码是音频压缩技术的一种，它通过有损压缩的方式减少音频数据的大小，以便于存储和传输。在VC++环境下进行MP3编码与解码的编程，涉及到的知识点广泛而深

对mp3 乱码问题的分析和解决(ZZ)_mp3 名称都是17位

1、了解 mp3 标签类型和使用的编码 首先说 mp3 标签类型和编码,大家应该知道目前主要存在这几种标准,ID3v1, ID3v2 2.3, ID3v2 2.4, APEv2,ID3v1 只支持 ISO-8859-1 编码

文件编码格式_mp3怎么验证格式编码

从文件编码的方式来看,文件可分为ASCII码文件和二进制码文件两种。 ASCII文件也称为文本文件,这种文件在磁盘中存放时每个字符对应一个字节,用于存放对应的ASCII码。

强大的mp3格式分析工具

在多媒体领域，音频格式的处理是一项重要任务，i



xiahouzuoxin

关注

7

