

百味木屋

- - 点点滴滴，油盐酱醋

博客园 首页 新随笔 联系 订阅 管理

随笔 - 6 文章 - 25 评论 - 4 阅读 - 99537

公告

昵称： 刘文涛
园龄： 15年8个月
粉丝： 9
关注： 3
+加关注

< 2024年6月 >

日	一	二	三	四	五	六
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

搜索

找查看

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

Maven(1)

随笔分类

Java(4)
mysql(1)

随笔档案

2018年1月(1)
2016年8月(2)
2015年4月(2)
2008年9月(1)

文章分类

C(2)
java(10)
JNI(4)
Linux(4)
web(1)
文件解析(9)

阅读排行榜

FLAC 格式详解

<http://blog.csdn.net/ffgamelife/article/details/7893747>

1. FLAC 编码设置只对编码时间有较大影响而对解码影响不大；因为等级越高，编码器就会花越多的时间去寻找最佳的压缩算法，而解码器则根据给定的压缩算法直接解压。

2. FLAC简介

基本结构：

4byte 字符“flaC”：flac标志，用于识别flac数据流

STREAMINFO 文STREAMINFO 件信息描述块（metadata block）：包含必须的信息（采样率、声道数...）

可选的其他描述信息块：（解码是可以不用识别）

一个以上的音频帧（frame）

3. 编码的过程

Flac把未压缩的音频流划分为块（block），并独立压缩，压缩后的数据块形成数据帧（frame），把数据帧连接形成压缩后的flac数据流（stream）

分块（blocking） flac分块大小是可变的。分块大小应适当选择，太小影响压缩率（太多帧头信息），太大难以得到高效的压缩模型。一般44.1k线性采样，分块大小2~6k较合适（默认4096）

声道内解相关性? 立体声的左右声道数据之间有许多相关性，可以利用这种相关性压缩数据。Flac有四种方式表示声道数据。

独立模式：左右声道独立编码；

Mid-side模式：转换表达式：mid = (left + right) / 2, side = left - right.。

Left-side：Left不变，s

Right-side：

建模（modeling） 编码器尝试使用一个数学方法（近似）描述原始信号，这种描述信息一般来说比原始信息小得多，这些数学方法是编码器和解码器都已知的（flac现在有4个种类的预测方法，并可以加入更多方法）flac运行在各个块中使用不同的预测算法。大多时候不能完全精确的描述原始信息，此时还会剩下少量数据残渣（residual, residue, or error）。Flac有两种产生近似值的方法：1) 为信号找个合适的多项式。2) 简单线性预测（LPC）。前者更快但不精确。

数据残渣编码 对建模后剩余的数据进行编码，保证数据的无损。目前flac只使用一种编码方法。

帧（framing） 一个音频帧被一个帧头（frameheader）和帧脚（frame footer）包围。帧头以同步字开始，包含了解码这帧的最小信息如采样率，采样位数等，同时包含了这帧里的分块数或采样数及一个8位CRC校验码。帧头可以用来进行再同步。帧尾包含一个16位CRC校验码。如果解码器检测到CRC错误将产生一个silent block。

https://www.cnblogs.com/liuwt0911/articles/3730378.html

1/10

1. JSqlParser系列之二代码结构 (原) (23692)
2. MySQL5.6空间扩展 (原创) (8096)
3. Python系统运维常用库(6999)
4. JSqlParser系列之一源代码运行(6595)
5. 文本文件编码识别(2197)
评论排行榜
1. JSqlParser系列之二代码结构 (原) (2)
2. JSqlParser系列之一源代码运行(1)
3. 文本文件编码识别(1)
推荐排行榜
1. JSqlParser系列之二代码结构 (原) (3)
最新评论
1. Re:文本文件编码识别 6666 --规格严格-功夫到家
2. Re:JSqlParser系列之一源代码运行 6666666 --规格严格-功夫到家
3. Re:JSqlParser系列之二代码结构 (原) 666666 --规格严格-功夫到家
4. Re:JSqlParser系列之二代码结构 (原) 请问where部分表达式如何计算出具体的值。 --kris_zhang

4. 定义

(block、subblock 指没有被编码的原始数据; frame、subframe指编码后的数据)

Block: 一组或多组跨所有声道的采样点 (flac采样组数范围16~65535)。

Subblock: 一个或多个单声道的采样点, 一个Block包含几个声道就有几个subblock, 同个block内subblock的采样点数相同。

Blocksize: 一个block中任意subblock的采样点个数 (与声道数无关)。

Frame: 一个frame帧头加一个或多个Subframe

Subframe: 一个subframe帧头加某一声道上一个或多个编码后的采样点, 一个frame中的所有subframe包含的采样点数一样。

(每次一个subblock被编码成一个subframe, 多个subframe组成一个frame)

5. Flac格式 (format)

Flac中所有数值都是整形, 大端模式, 除非特别指出, 否则数值都是unsigned的。

一些概述:

Flac比特流以“flaC”开始, 接着是必须的metadata块 (STREAMINFO), 然后是其可选项的metadata块, 接着是音频帧。

Flac做多可以支持到128类metadata, 已经定义的包括下面几种:

A. STREAMINFO: 包含整个比特流的一些信息, 如采样率、声道数、采样总数等。他一定是第一个metadata而且必须有。之后可以接其他metadata, 这些metadata可以不用识别直接跳过。

B. APPLICATION: 包含第三方应用软件信息, 这个段里的32位识别码是flac维护组织提供的, 是唯一的。

C. PADDING: 没有意义的东西, 主要用来后期添加其他metadata。

D. SEEKTABLE: 保存快速定位点, 一个点由18bytes组成 (2k就可以精确到1%的定位), 表里可以有任意多个定位点。

E. VORBIS_COMMENT: 存储了一系列可读的“名/值”的键值对, 使用UTF-8编码。这是flac唯一官方支持的标签段。

F. CUESHEET: 存储用在cue sheet中的各种信息。可以用来划分音轨, 在备份CD时十分有用。

G. PICTURE: 保存相关图片, 同时还有url、分辨率等信息, 可以有不止一个picture block。

音频数据由一个或多个音频帧组成, 每一帧包含一个帧头: 同步字, 块大小, 采样率, 声道数...然后是8bitCRC校验码; 同时帧头还包含本帧第一个采样点的采样序号 (blocksize变长的文件) 或本帧的序号 (blocksize定长的文件), 他们用于精确定位。接着是编码后的subframes, 每个subframe代表一个声道。最后是一些有0填充的边界。每个subframe有他自己的帧头用于指出他是怎样被编码的。

当从一个文件中间开始解码时需要知道一个帧的起始点。可以通过一个14bit的同步字来判断。但是他可能会出现在subframe里面, 此时可以通过检测剩余数据 (没有无效数据) 和CRC8。

同样当从中间某帧开始解码时, 没有读取STREAMINFO, 为了得到采样率和声道数等信息, 需要再帧头里加入相关信息, 为了减少数据, 使用了查表的方法来定义常用的采样率。

格式:

长度	名称			说明
<32>	“flaC”			0x66 0x41 0x61 0x43 固定标签
<1>	METADATA BLOCK	HEADER		最后一个metadata为'1',其他为'0'
<7>			块信息类型	0 : STREAMINFO 1 : PADDING 2 : APPLICATION 3 : SEEKTABLE 4 : VORBIS_COMMENT 5 : CUESHEET 6 : PICTURE 7-126 : reserved 127 : 无效
<24>			后面跟的数据长度	单位是byte, 不包括此头的长度
		DATA	数据, 与头中的类型要相符	参加metadata data
	FRAME	HEADER	头	见FRAME HEADER
		SubFrame	子帧	包含一个通道的数据,见SUBFRAME
		?		填充0用于对齐
16		FOOTER	尾	CRC-16

Metadata block DATA:

Metadata类型	长度	说明
STREAMINFO	16	blocksize最小的中的采样数

		16	Blocksize中最大的采样数
		24	最小帧大小 (byte) , 0表示未知
		24	最大帧大小 (byte) , 0表示未知
		20	采样率 (Hz) 最大 655350Hz, 0无效
		3	声道数减一, flac支持1~8个声道
		5	采样位数减一, flac支持 4~32位采样位数
		36	一个声道的总采样数, 0表示未知
		128	未编码时的原始信号的MD5 信息
PADDING		N	'0' N必须是8的倍数
APPLICATION		32	应用程序ID
		N	应用程序数据, 必须是8的倍数
SEEKTAB LE	SEEKPOINT	64	目标帧中第一个采样点的序号
		64	相对于第一帧开始的偏移 (byte)
		16	目标帧中的采样数
	SEEKPOINT	同上	SEEKPOINT的个数等于 DTAT段长度/18

CUESHEET	64		音轨偏移量（单位是采样数）
	8		音轨数目
	12*8		音轨ISRC
	1		轨迹类型：0 音乐 1非音乐
	1		pre-emphasis标记
	6+13*8		保留 '0'
	8		应该的索引数目
	INDEXn（有多少个）	64	偏移（单位是采样点）
		8	索引号
		3*8	保留
PICTURE	32		图片类型（同ID3v2 APIC） 0 - Other 1 - 32x32 pixels 'file icon' (PNG only) 2 - Other file icon 3 - Cover (front) 4 - Cover (back) 5 - Leaflet page 6 - Media (e.g. label side of CD) 7 - Lead artist/lead performer/soloist 8 - Artist/performer 9 - Conductor 10 - Band/Orchestra 11 - Composer 12 - Lyricist/text writer 13 - Recording Location

		14 - During recording 15 - During performance 16 - Movie/video screen capture 17 - A bright coloured fish 18 - Illustration 19 - Band/artist logotype 20 - Publisher/Studio logotype
	32	MIME 类型说明字符长度 (byte)
	N*8	MIME类型说明字符
	32	描述符长度 (byte)
	N*8	描述符UTF-8
	32	图片宽度
	32	图片高度
	32	图片颜色深度
	32	索引图使用的颜色数目, 0非索引图
	32	图片数据长度
	N*8	图片二进制文件

Frame Header:

长度	说明	说明
14	同步字	'11111111111110'

1	保留	0 : 强制值 1 : 保留未来使用
1	分块策略	0: 固定块大小, 帧头包含帧的序号 1: 块大小可变, 帧头包含采样点序号
4	块内的采样数	0000 : reserved 0001 : 192 samples 0010-0101 : $576 * (2^{(n-2)})$ samples, i.e. 576/1152/2304/4608 0110 : get 8 bit (blocksize-1) from end of header 0111 : get 16 bit (blocksize-1) from end of header 1000-1111: $256 * (2^{(n-8)})$ 采样点
4	采样率	0000 : get from STREAMINFO metadata block 0001 : 88.2kHz 0010 : 176.4kHz 0011 : 192kHz 0100 : 8kHz 0101 : 16kHz 0110 : 22.05kHz 0111 : 24kHz 1000 : 32kHz 1001 : 44.1kHz 1010 : 48kHz 1011 : 96kHz 1100 : get 8 bit sample rate (in kHz) from end of header 1101 : get 16 bit sample rate (in Hz) from end of header 1110 : get 16 bit sample rate (in tens of Hz) from end of header 1111 : invalid, to prevent sync-fooling string of 1s
4	声道分配	0000-0111 :独立声道减一, 分配如下: 1 channel: mono 2 channels: left, right 3 channels: left, right, center 4 channels: left, right, back left, back right 5 channels: left, right, center, back/surround left, back/surround right

		<p>6 channels: left, right, center, LFE, back/surround left, back/surround right</p> <p>7 channels、8 channels: not defined</p> <p>1000 : left/side stereo: channel 0 is the left channel, channel 1 is the side(difference) channel</p> <p>1001 : right/side stereo: channel 0 is the side(difference) channel, channel 1 is the right channel</p> <p>1010 : mid/side stereo: channel 0 is the mid(average) channel, channel 1 is the side(difference) channel</p> <p>1011-1111 : reserved</p>
3	采样深度	<p>000 : get from STREAMINFO metadata block</p> <p>001 : 8 bits per sample</p> <p>010 : 12 bits per sample</p> <p>011 : reserved</p> <p>100 : 16 bits per sample</p> <p>101 : 20 bits per sample</p> <p>110 : 24 bits per sample</p> <p>111 : reserved</p>
1	保留	<p>0 : mandatory value 1 : reserved for future use</p>
?		<p>if(variable blocksize)</p> <p> <8-56>:"UTF-8" coded sample number (decoded number is 36 bits)</p> <p>else</p> <p> <8-48>:"UTF-8" coded frame number (decoded number is 31 bits)</p>
?		<p>if(blocksize bits == 011x)</p> <p> 8/16 bit (blocksize-1)</p>
?		<p>if(sample rate bits == 11xx)</p> <p> 8/16 bit sample rate</p>
8		<p>CRC-8 (polynomial = $x^8 + x^2 + x^1 + x^0$, initialized with 0) of everything before the crc, including the sync code</p>

SubFrame:

长度	域	说明
1	填充0	prevent sync-fooling string of 1s
6	Subframe类型	000000 : SUBFRAME_CONSTANT 000001 : SUBFRAME_VERBATIM 00001x : reserved 0001xx : reserved 001xxx : if(xxx <= 4) SUBFRAME_FIXED, xxx=order ; else reserved 01xxxx : reserved 1xxxxx : SUBFRAME_LPC, xxxxx=order-1
1+k	'Wasted bits-per-sample' flag	0 : no wasted bits-per-sample in source subblock, k=0 1 : k wasted bits-per-sample in source subblock, k-1 follows, unary coded; e.g. k=3 => 001 follows, k=7 => 0000001 follows.

分类: [文件解析](#)

好文要顶

关注我

收藏该文

微信分享



刘文涛

粉丝 - 9 关注 - 3

0

0

+加关注

升级成为会员

posted @ 2014-05-15 16:51 刘文涛 阅读(2585) 评论(0) 编辑 收藏 举报

会员力量，点亮园子希望

刷新页面 返回顶部

登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

- 【推荐】这场阿里云开发者社区有奖征文活动，期待您出文相助
- 【推荐】三生石上：ASP.NET Core中运行WebForms业务代码
- 【推荐】「指间灵动，快码加编」：通义灵码，再次降临博客园
- 【推荐】凡泰极客：跨越技术“鸿”沟，小程序一键生成鸿蒙App
- 【推荐】阿里云创新加速季，5亿补贴享不停，上云礼包抢先领



编辑推荐:

- 老生常谈！程序员为什么要阅读源代码？
- 如何安全地访问互联网
- 还在拼冗长的 Wherelf 吗？100行代码解放这个操作
- C#异步编程是怎么回事（番外）
- [WPF] Dispatcher 与消息循环

阅读排行:

- 「指间灵动，快码加编」：阿里云通义灵码，再次降临博客园
- 面试官：你讲下接口防重放如何处理？
- 如何安全地访问互联网
- 手机上玩 PC 游戏的开源项目「GitHub 热点速览」
- 【译】Visual Studio 17.10 发布了新版扩展管理器

Copyright © 2024 刘文涛

Powered by .NET 8.0 on Kubernetes