

音频知识详解(一)



忠晓 关注

1 2018.03.07 16:35:28 字数 2,860 阅读 11,638

- 1
- Ps:做了几年的ios开发，但是对于现如今很火的小视频，直播app等知识点并不懂，很是惭愧😓！！
- 2
- 抱着对新知识的好奇与渴望，在此踏上音频之旅，借鉴了很多大佬的文章以及加上个人的理解，进行了
- 3
- 整理以方便学习，同时与大家进行分享，如果要是有啥不足的地方，希望大家给予指点，在此多谢！！

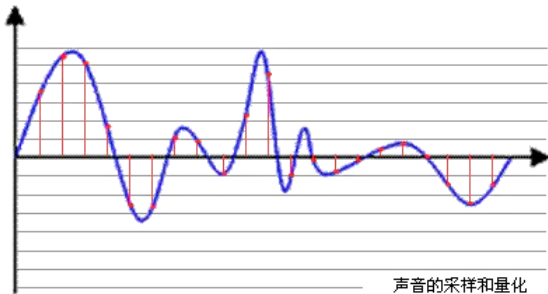
引言

现实生活中，我们听到的声音都是时间连续的，我们称为这种信号叫 **模拟信号**。模拟信号需要进行数字化以后才能在计算机中使用。

目前我们在计算机上进行音频播放都需要依赖于**音频文件**。**音频文件**的生成过程是将**声音信息采集、量化和编码**产生的**数字信号**的过程，人耳所能听到的声音，最低的频率是从20Hz起一直到最高频率20KHZ，因此音频文件格式的最大带宽是20KHZ。根据**奈奎斯特**的理论，只有**采样频率高于声音信号最高频率**的两倍时，才能把数字信号表示的声音还原成为原来的声音，所以音频文件的采样率一般在40~50KHZ，比如最常见的CD音质采样率44.1KHZ。

音频基本概念

采样：波是无限光滑的，采样的过程就是从波中抽取某些点的频率值，就是把**模拟信号数字化**。如下图所示：



蓝色代表模拟音频信号，红色代表采样得到的量化数值

采样频率：单位时间内对模拟信号的采样次数。采样频率越高，声音的还原就越真实越自然，当然数据量就越大。采样频率一般共分为22.05KHz、44.1KHz、48KHz三个等级。8KHz - 电话所用采样率, 对于人的说话已经足够，22.05KHz只能达到FM广播的声音品质(适用于语音和中等品质的音乐)，44.1KHz则是最常见的采样率标准，理论上的CD音质界限，48KHz则更加精确一些(对于高于48KHz的采样频率人耳已无法辨别出来了，所以在电脑上没有多少使用价值)。

- 1
- 小知识点：
- 2
- 5kHz的采样率仅能达到人们讲话的声音质量。
- 3
- 11kHz的采样率是播放小段声音的最低标准，是CD音质的四分之一。
- 4
- 22kHz采样率的声音可以达到CD音质的一半，目前大多数网站都选用这样的采样率。
- 5
- 44kHz的采样率是标准的CD音质，可以达到很好的听觉效果。

采样位数(也成量化级、样本尺寸、量化数据位数)：每个采样点能够表示的数据范围。采样位数通常有8bits或16bits两种，采样位数越大，所能记录声音的变化度就越细腻，相应的数据量

就越大。8位字长量化（低品质）和16位字长量化（高品质），16 bit 是最常见的采样精度。

1	"采样频率"和"采样位数"是数字化声音的两个最基本要素，相当于视频中的屏幕大小
2	（例如800*600）和颜色分辨率（例如24bit）。

量化：将采样后离散信号的幅度用二进制数表示出来的过程称为量化。（日常生活所说的量化，就是设定一个范围或者区间，然后看获取到的数据在这个条件内的收集出来）。

PCM：PCM（Pulse Code Modulation），即脉冲编码调制，对声音进行采样、量化过程，未经过任何编码和压缩处理。

1	PCM数据是最原始的音频数据完全无损，所以PCM数据虽然音质优秀但体积庞大，
2	为了解决这个问题先后诞生了一系列的音频格式，这些音频格式运用不同的方法对音频数据
3	进行压缩，其中有无损压缩（ALAC、APE、FLAC）和有损压缩（MP3、AAC、OGG、WMA）两种。

编码：采样和量化后的信号还不是数字信号，需要将它转化为数字编码脉冲，这一过程称为编码。模拟音频经采样、量化和编码后形成的二进制序列就是数字音频信号。

声道数：声道数是指支持能不同发声的音响的个数，它是衡量音响设备的重要指标之一。

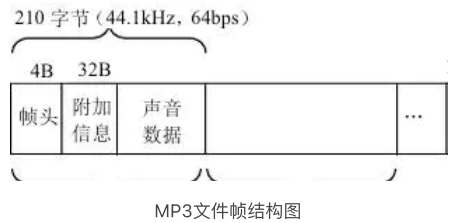
1	单声道的声道数为1个声道；
2	双声道的声道数为2个声道；
3	立体声道的声道数默认为2个声道；
4	立体声道（4声道）的声道数为4个声道。

码率：(也成位速、比特率)是指在一个数据流中每秒钟能通过的信息量，代表了压缩质量。比如MP3常用码率有128kbit/s、160kbit/s、320kbit/s等等，越高代表着声音音质越好。MP3中的数据有ID3和音频数据组成，ID3用于存储歌名、演唱者、专辑、音轨等我们可以常见的信息。

1	公式：
2	码率 = 采样率 * 采样位数 * 声道数
3	例如：
4	如果是CD音质，采样率44.1KHz，采样位数16bit，立体声(双声道)，
5	码率 = 44.1 * 1000 * 16 * 2 = 1411200bps = 176400Bps，那么录制一分钟的音乐，
6	大概176400 * 1 * 60 / 1024 / 1024 = 10.09MB。

音频帧：音频数据是流式的，本身没有明确的一帧帧的概念，在实际的应用中，为了音频算法处理/传输的方便，一般约定俗成取2.5ms~60ms为单位的数据量为一帧音频。这个时间被称之为“采样时间”，其长度没有特别的标准，它是根据编解码器和具体应用的需求来决定的。

1	举例：
2	
3	目前最为常用的音频格式是MP3，MP3是一种有损压缩的音频格式，设计这种格式的目的就是
4	为了大幅度的减小音频的数据量，它舍弃PCM音频数据中人类听觉不敏感的部分。
5	
6	MP3格式中的码率（BitRate）代表了MP3数据的压缩质量，现在常用的码率有
7	128kbit/s、160kbit/s、320kbit/s等等，这个值越高声音质量也就越高。
8	MP3编码方式常用的有两种"固定码率"(Constant bitrate, CBR)
9	和"可变码率"(Variable bitrate, VBR)。
10	
11	MP3格式中的数据通常由两部分组成，一部分为"ID3"用来存储歌名、演唱者、专辑、音轨数
12	等信息，另一部分为音频数据。音频数据部分以帧(frame)为单位存储，每个音频都有自己的帧头，
13	如图所示就是一个MP3文件帧结构图。MP3中的每一个帧都有自己的帧头，其中存储了采样率等
14	解码必须的信息，所以每一个帧都可以独立于文件存在和播放，这个特性加上高压缩比使得
15	MP3文件成为了音频流播放的主流格式。帧头之后存储着音频数据，这些音频数据是若干个
16	PCM数据帧经过压缩算法压缩得到的，对CBR的MP3数据来说每个帧中包含的PCM数据帧是固定的，
17	而VBR是可变的。
18	



声音具体处理流程

模拟信号 -> 输入设备（传递电压值）-> 声卡（经过采样跟量化（即设置声音大小等各种值））-> 磁盘（文件）-> 声卡 -> 输出设备 -> 模拟信号

我们声音在物理上用波形表示，那么我们将这些波形称作为**模拟信号**。而我们计算机磁盘只能存储（01010101）的格式。我们将模拟信号转换成能够被磁盘存储的格式（010101）称之为**数字信号**。这个转换的过程我们叫**模数转换**。

我们发出来的声音（模拟信号）是连续的，我们如果要一直的对模拟信号进行转化，产生的数字信号会很大。那么我们就需要采样，而采样精度就是每秒计算机对模拟信号进行采样的次数。最常见的采样精度就是上面提到的44.1khz/s,这个是经过大师们多年研究得出的数据，低于这个数据，效果就会很差，而高于这个数据，效果的差距不是很明显。

采样后就是变成了（01010101110100101...），那声音的音量是有大小的，那这串数据，怎样表示声音的大小呢？这就涉及到了**比特率**，它是指在一个数据流中每秒钟能通过的信息量。比特率就是将声音的大小划分为多少等级。举例下：8比特，在二进制中，表示有8位，表示的十进制的值就是0（00000000）~256（11111111），那每个数值就代表着一个声音大小。

经过采样、量化、编码后转化成数字信号，然后存储为文件。

文件是用来装数字信号的，文件包括了比特率、采样率、声道、编码方式、以及被编码过后的数字信号。

文件格式就是制造者自己规定的一种名称，在每个文件格式都会特定支持几种编码格式。打个比方就是**文件就是一个容器**，里面可以装不同的水，有的可以装一种，有的可以装好几种。

经过采样后的数字信号很大，有时候我们不需要这么大的，所以我们就要进行编码压缩，当然压缩技术都是有损的。在不大影响音频的效果的情况下，舍弃掉一些高频或者低频的数据。

编码格式可以理解为每种音频格式不同的编解码方式。

封装格式就是文件格式，编码就是编码格式。

- 1 | 比较通俗的理解：封装格式和编码的关系，就是和酒瓶与酒的关系差不多，而播放器就是开酒器。
- 2 | 为了可以喝更好的酒，最好先了解酒是什么酒，酒瓶是什么酒瓶，怎么使用开酒器开酒瓶，码率
- 3 | 或者可以比喻做酒的原料。对于同一个酒瓶和做同一种酒的情况下，如果原料太少，又要要求
- 4 | 用酒把酒瓶灌满，此时只好兑水了，酒的品质就会变差了。然而，如果原料太多，又会造成
- 5 | 原料浪费。所以要做好酒，我们就需要充足的原料。

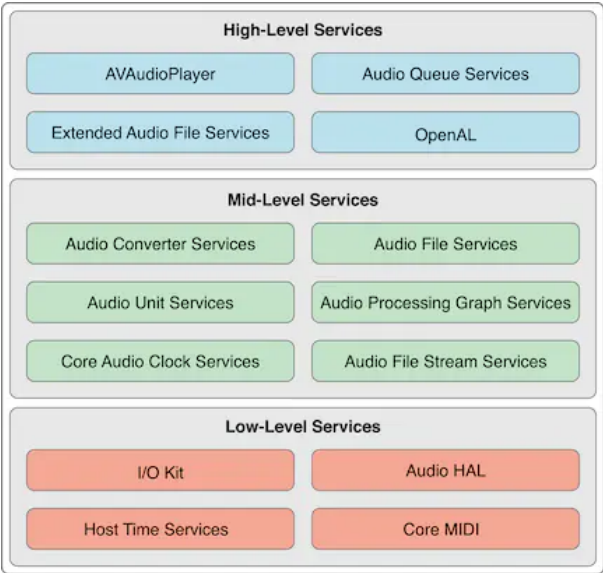
ios音频播放描述

了解了基础概念之后我们就可以列出一个经典的音频播放流程（以MP3为例）：

- 1、读取MP3文件
- 2、解析采样率、码率、时长等信息，分离MP3中的音频帧
- 3、对分离出来的音频帧解码得到PCM数据
- 4、对PCM数据进行音效处理（均衡器、混响器等，非必须）

- 5、把PCM数据解码成音频信号
- 6、把音频信号交给硬件播放
- 7、重复1-6步直到播放完成

在iOS系统中apple对上述的流程进行了封装并提供了不同层次的接口（图片引自[官方文档](#)）。



下面对其中的中高层接口进行功能说明：

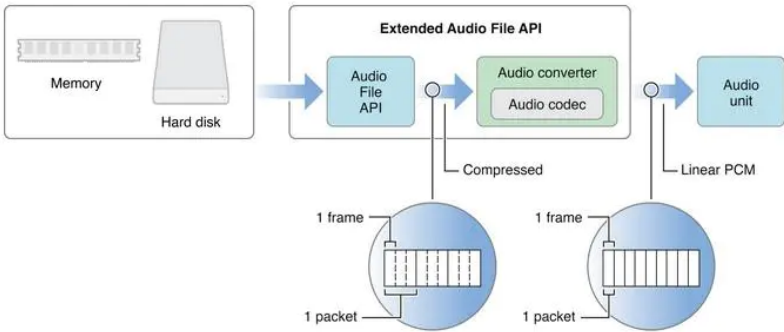
- Audio File Services：读写音频数据，可以完成播放流程中的第2步；
- Audio File Stream Services：对音频进行解码，可以完成播放流程中的第2步；
- Audio Converter services：音频数据转换，可以完成播放流程中的第3步；
- Audio Processing Graph Services：音效处理模块，可以完成播放流程中的第4步；
- Audio Unit Services：播放音频数据：可以完成播放流程中的第5步、第6步；
- Extended Audio File Services：Audio File Services和Audio Converter services的结合体；
- AVAudioPlayer/AVPlayer(AVFoundation)：高级接口，可以完成整个音频播放的过程（包括本地文件和网络流播放，第4步除外）；
- Audio Queue Services：高级接口，可以进行录音和播放，可以完成播放流程中的第3、5、6步；
- OpenAL：用于游戏音频播放，暂不讨论

可以看到apple提供的接口类型非常丰富，可以满足各种类别类需求：

- 如果你只是想实现音频的播放，没有其他需求AVFoundation会很好的满足你的需求。它的接口使用简单、不用关心其中的细节；
- 如果你的app需要对音频进行流播放并且同时存储，那么AudioFileStreamer加AudioQueue能够帮到你，你可以先把音频数据下载到本地，一边下载一边用NSFileHandler等接口读取本地音频文件并交给AudioFileStreamer或者AudioFile解析分离音频帧，分离出来的音频帧可以送给AudioQueue进行解码和播放。如果是本地文件直接读取文件解析即可。（这两个都是比较直接的做法，这类需求也可以用AVFoundation+本地server的方式实现，AVAudioPlayer会把请求发送给本地server，由本地server转发出去，获取数据后在本地server中存储并转送给AVAudioPlayer。另

一个比较trick的做法是先把音频下载到文件中，在下载一定量的数据后把文件路径给AVAudioPlayer播放，当然这种做法在音频seek后就回有问题了。）；

- 如果你正在开发一个专业的音乐播放软件，需要对音频施加音效（均衡器、混响器），那么除了数据的读取和解析以外还需要用到AudioConverter来把音频数据转换成PCM数据，再由AudioUnit+AUGraph来进行音效处理和播放（但目前多数带音效的app都是自己开发音效模块来坐PCM数据的处理，这部分功能自行开发在自定义性和扩展性上会比较强一些。PCM数据通过音效器处理完成后就可以使用AudioUnit播放了，当然AudioQueue也支持直接使对PCM数据进行播放。）。下图描述的就是使用AudioFile + AudioConverter + AudioUnit进行音频播放的流程（图片引自[官方文档](#)）。



参考文章

<https://msching.github.io/> 大神之作
<https://www.jianshu.com/p/5c5e95d89c4f> 写的相当不错
<https://www.jianshu.com/p/423726cc9090> 知识点很全
<https://www.jianshu.com/p/b3db09fb69dc> 总结的挺好
<https://www.jianshu.com/p/a75f2411225f> 有点专业，了解一部分
<http://blog.csdn.net/liusandian/article/details/52488078> 概念很清晰易懂

49人点赞 >

流媒体