1. 视频帧的概念

视频播放过程可以简单理解为一帧一帧的画面按照时间顺序来显示。实际应用中，并不是每一帧都是完整的画面，因为若都是完整画面的话，那视频的体积会很大，这样网络传输和视频存储的成本就很高。所以我们会对视频流中的部分画面进行压缩处理，也就是编码。压缩处理方式不同，视频中的画面帧就分成不同类别：I帧、B帧、P帧。

I帧：关键帧，属于帧内压缩，压缩率是7（和JPG差不多）。保留了一帧完整的画面，解码时只需本帧数据就可完成；

P帧：采用帧间编码方式，是差别帧，压缩率是20。它表示的是这一帧和之前的I帧或者P帧的差别，解码时需要本帧定义差别和之前缓存的画面进行叠加，才能生成最终画面‘

B帧：采用帧间编码方式，是双向差别帧，压缩率是最高的，是50。记录的是本帧与前后帧的差别。解码B帧需要和前后帧数据叠加才能得到最终画面。

（视频帧存放的顺序都是按照显示顺序存放的）

网络电影很多时候会卡是什么原因？

网络电影很多都采用B帧，B帧因为记录前后帧的差别，可以节约更多的空间。如果B帧丢失掉，就会和之前的画面简单重复，就造成了画面卡。如果播放器不支持B帧播放，画面就越卡。

GOP是指两个I帧之间的距离；Reference（参考周期）指两个P帧之间的距离。

一个I帧所占的字节数大于一个P帧，一个P帧所占用的字节数大于一个B帧。

1. 在视频流中，先到的B帧无法立即解码，需要依赖后来的I帧和B帧先解码才能完成。这样播放时间和解码时间不一致，那此时帧要如何播放？PTS(Presentation Time Stamp)：显示时间戳，告诉播放器什么时候显示这帧数据；

DTS(Decoding Time Stamp)：解码时间戳，告诉播放器什么时候解码这帧数据；

若没有B帧时，PTS和DTS的顺序是一致的。有B帧，解码顺序和显示顺序不一致了，即视频输出是非线性的了。

音频播放时也有PTS、DTS的概念，但是音频中没有类似B帧的东西，所以音频帧的DTS、PTS顺序是一致的。

我们看视频时出现音画不同步，是什么原因？

是由于

想实现音视频同步，在播放时选定个参考时钟，读取帧的PTS，同时跟参考时钟来动态调节播放。因为人对声音的敏感度要强于视频，一般以音频时钟为参考时钟，把视频同步到音频上。

1. PTS、DTS的时间单位，time\_base时间基数

Time\_base是度量时间的，time\_base = {1,25}；表示一个刻度就是1/25秒，所以时间基数就是表示每个刻度是多少秒。

在ffmpeg中。av\_q2d(time\_base)=每个刻度是多少秒 ，那比如PTS=20个刻度，根据pts来计算一桢在整个视频中的时间位置：   
timestamp(秒) = pts \* av\_q2d(st->time\_base)；

pts：格子数   
av\_q2d(st->time\_base): 秒/格

duration表示当前帧的持续时间占多少格。或者理解是两帧的间隔时间是占多少格：

计算视频长度：   
time(秒) = st->duration \* av\_q2d(st->time\_base)

音频pts的计算：   
音频sample\_rate:samples per second，即采样率，表示每秒采集多少采样点。   
比如44100HZ，就是一秒采集44100个sample.   
即每个sample的时间是1/44100秒

一个音频帧的AVFrame有nb\_samples个sample，所以一个AVFrame耗时是nb\_samples\*（1/44100）秒   
即标准时间下duration\_s=nb\_samples\*（1/44100）秒，   
转换成AVStream时间基下   
duration=duration\_s / av\_q2d(st->time\_base)   
基于st->time\_base的num值一般等于采样率,所以duration=nb\_samples.   
pts=n\*duration=n\*nb\_samples

1. 采样率/样本率：单位HZ，每秒钟单个声道采样的次数。值越大，声音越连贯；

样本数(nb\_samples)：一帧中单个声道的音频样本数；

位宽（位深）：每种音频格式都有不同的量化精度(位宽)，位数越多，声音表现越精准。

FFMpeg中音频格式有以下几种，每种格式有其占用的字节数信息：

enum AVSampleFormat {

AV\_SAMPLE\_FMT\_NONE = -1,

AV\_SAMPLE\_FMT\_U8, ///< unsigned 8 bits

AV\_SAMPLE\_FMT\_S16, ///< signed 16 bits

AV\_SAMPLE\_FMT\_S32, ///< signed 32 bits

AV\_SAMPLE\_FMT\_FLT, ///< float

AV\_SAMPLE\_FMT\_DBL, ///< double

AV\_SAMPLE\_FMT\_U8P, ///< unsigned 8 bits, planar

AV\_SAMPLE\_FMT\_S16P, ///< signed 16 bits, planar

AV\_SAMPLE\_FMT\_S32P, ///< signed 32 bits, planar

AV\_SAMPLE\_FMT\_FLTP, ///< float, planar

AV\_SAMPLE\_FMT\_DBLP, ///< double, planar

AV\_SAMPLE\_FMT\_S64, ///< signed 64 bits

AV\_SAMPLE\_FMT\_S64P, ///< signed 64 bits, planar

AV\_SAMPLE\_FMT\_NB ///< Number of sample formats. DO NOT USE if linking dynamically

};

音频帧的数据量计算：

一帧音频的数据量 = channel数 \* nb\_samples样本数 \* 每个样本占用的字节数；

音频帧播放时间计算：

每帧播放时间 = 一帧的sample个数 \* （1000ms / 采样率）；

以采样率44100Hz来计算，每秒44100个sample，而正常一帧为1024个sample，可知每帧播放时间/1024=1000ms/44100，得到每帧播放时间=1024\*1000/44100=23.2ms

帧率（framerate）：单位fps，每秒钟刷新图片的帧数。值越大，画面越流畅

如果视频源来自摄像头，24FPS已经是肉眼极限，所以一般20帧的FPS就已经可以达到很好的用户体验了。

Time\_base = 1/framerate；