FFmpeg 学习

一. FFmpeg 简介

1.1 FFmpeg 是什么

FF 代表“Fast Forward”。

FFmpeg 是一款音视频编解码工具，同时也是一组音视频编解码开发套件，为开发者提供了丰富的音视频处理的调用接口。

FFmpeg 提供了多种媒体格式的封装和解封装，包括多种音视频编码、多种协议的流媒体，多种色彩格式转换、多种采集率转换、多种码率转换等；FFmpeg 框架提供了多种丰富的插件模块，包含封装与解封装的插件、编码与解码的插件等。

1.2 FFmpeg 的组成

1、libavformat：用于各种音视频封装格式的生成和解析，包括获取解码所需信息以生成解码上下文结构和读取音视频帧等功能，包含 demuxers 和 muxer 库；
2、libavcodec：用于各种类型声音/图像编解码；
3、libavdevice：用于和多媒体设备交互的类库；
4、libavutil：包含一些公共的工具函数；
5、libswscale：用于视频场景比例缩放、色彩映射转换；
6、libpostproc：用于后期效果处理；
7、ffmpeg：是一个命令行工具，用来对视频文件转换格式，也支持对电视卡实时编码；
8、ffsever：是一个 HTTP 多媒体实时广播流服务器，支持时光平移；
9、ffplay：是一个简单的播放器，使用 ffmpeg 库解析和解码，通过 SDL 显示；

二、FFmpeg 编解码过程

2.1 编解码基础

什么是视频？其实就是一组（很多张）图片，时间间隔很小的连续展示出来，人们就觉得画面中的人物在动，这就是视频。那视频的实质就是 N 多张图片的集

合。那每张图片和帧又有什么关系呢？事实上，如果一部影片里面的图片，我们原封不动的全部存起来，空间会很大很大很大，但是如果通过一定的算法，把每一张图片压缩（编码_encode）一下，变成 **帧**。再把帧连起来变成**流**，再把不同的流放到某个**容器**里面，这就是我们平常看见的视频文件了，如文件“碟中谍4.H264.ACC.mkv”，他为什么要这样命名呢？ mkv 表达了它的容器是.mkv 的，且包含至少两个流，h264 的视频流，ACC 的音频流。这是一种典型的 牺牲时间来换取空间的做法。

术语：

容器(Container)——容器就是一种文件格式，比如 flv，mkv 等。包含下面 5 种流以及文件头信息。

流(Stream)——是一种视频数据信息的传输方式，5 种流：音频，视频，字幕，附件，数据。

帧(Frame)——帧代表一幅静止的图像，分为 I 帧，P 帧，B 帧。

编解码器(Codec)——是对视频进行压缩或者解压缩，CODEC =COde （编码）+DECode（解码）

复用/解复用(mux/demux)——把不同的流按照某种容器的规则放入容器，这种行为叫做复用（mux）；把不同的流从某种容器中解析出来，这种行为叫做解复用(demux)

附 1：I 帧,P 帧,B 帧 （http://blog.csdn.net/abcjennifer/article/details/6577934）

视频压缩中，每帧代表一幅静止的图像。而在实际压缩时，会采取各种算法减少数据的容量，其中 IPB 就是最常见的。

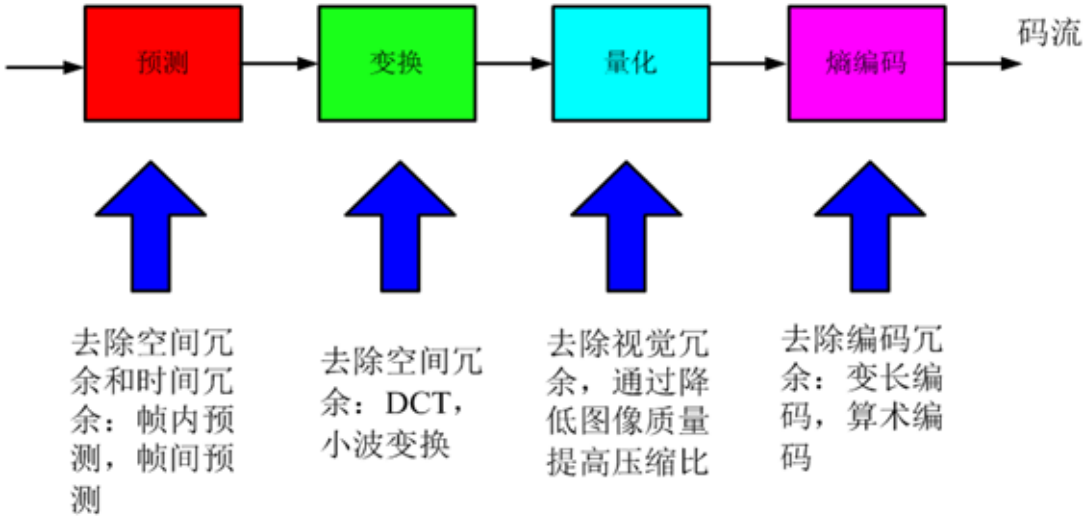
（1）I 帧表示关键帧，你可以理解为这一帧画面的完整保留；解码时只需要本帧数据就可以完成（因为包含完整画面）

（2）P 帧表示的是这一帧跟之前的一个关键帧（或 P 帧）的差别，解码时需要用之前缓存的画面叠加上本帧定义的差别，生成最终画面。（也就是差别帧，P 帧没有完整画面数据，只有与前一帧的画面差别的数据）

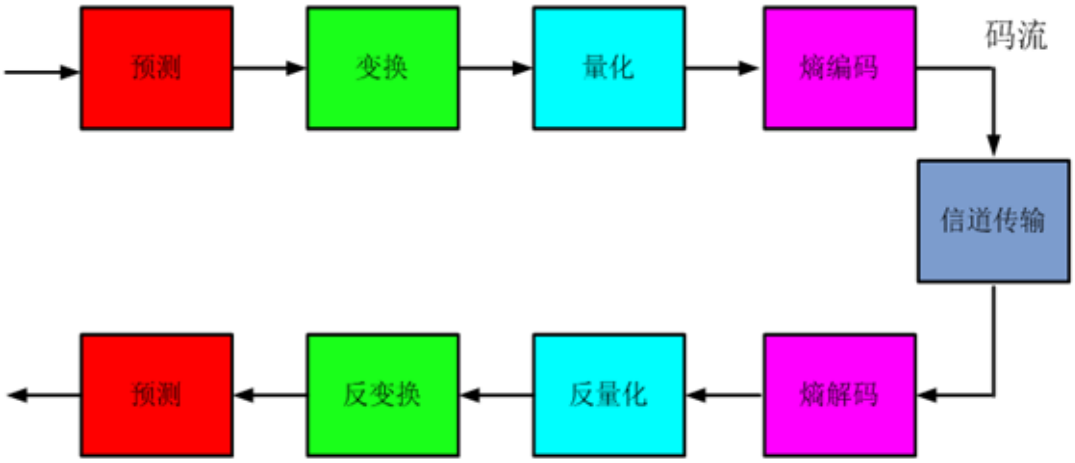
（3）B 帧是双向差别帧，也就是 B 帧记录的是本帧与前后帧的差别（具体比较复杂，有 4 种情况），换言之，要解码 B 帧，不仅要取得之前的缓存画面，还要解码之后的画面，通过前后画面的与本帧数据的叠加取得最终的画面。B 帧压缩率高，但是解码时 CPU 会比较累。

2.2 编解码关键技术

1、编码器中的关键技术



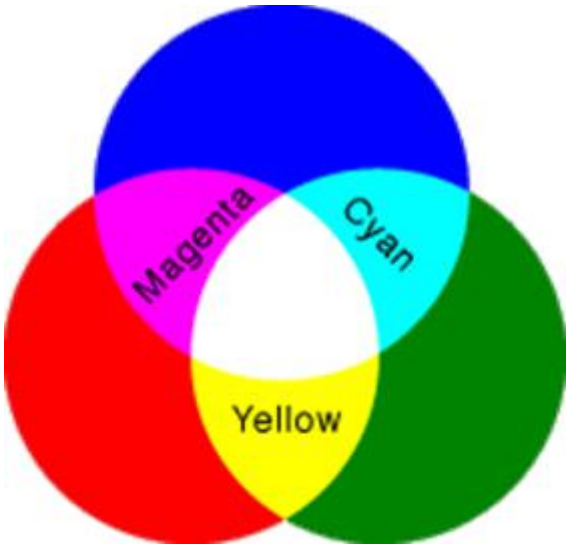
2、编解码器关键技术



3、视频编码色彩空间

（1）RGB 色彩空间

- 三原色：红（R），绿（G），蓝（B）。
- 任何颜色都可以通过按一定比例混合三原色产生。
- RGB 色度空间
- 由 RGB 三原色组成
- 广泛用于 BMP，TIFF，PPM 等
- 每个色度成分通常用 8bit 表示[0,255]



（2）YUV 色彩空间

- Y：亮度分量
- UV：两个色度分量
- YUV 更好的反映 HVS（人类视觉系统）特点

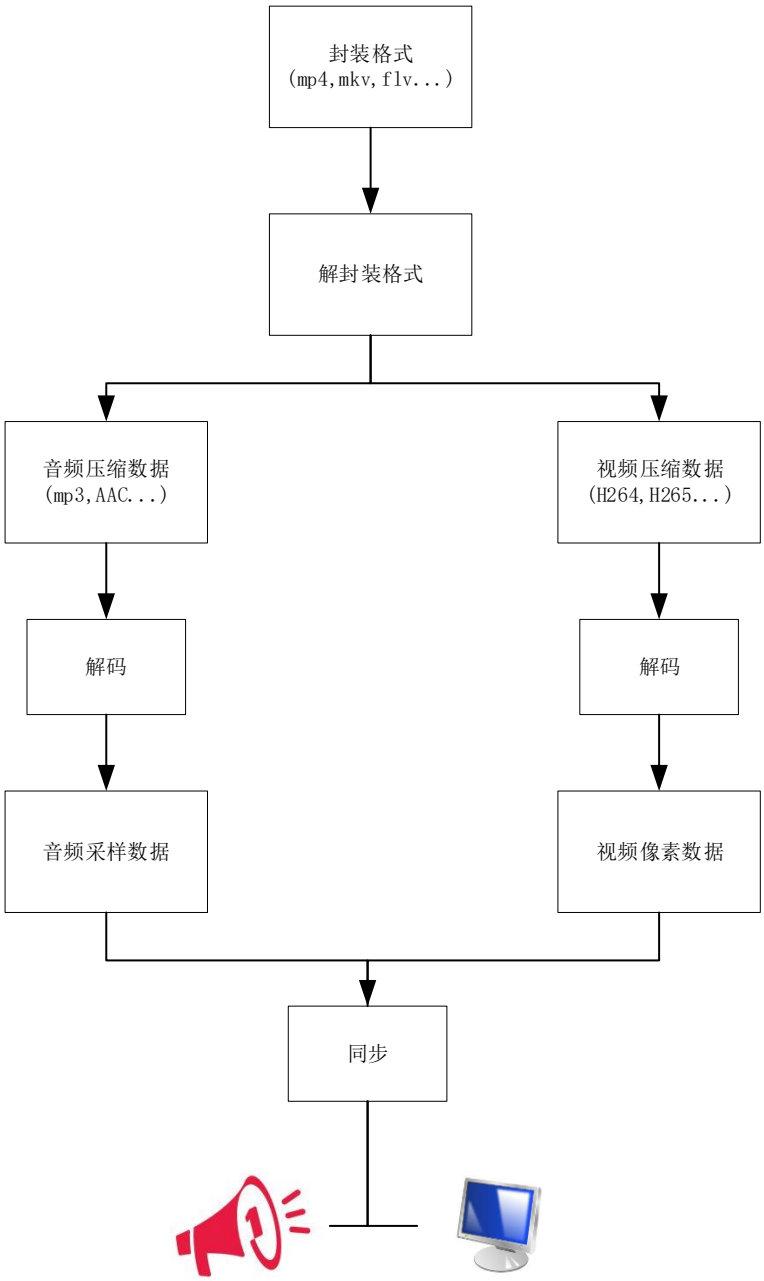
（3）RGB,YUV 空间的转换

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ -0.1687 & -0.3313 & 0.5000 \\ 0.5000 & -0.4187 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G+128 \\ B+128 \end{bmatrix}$$

主流的编解码标准的压缩对象都是 YUV 图像

2.3 视频播放器原理

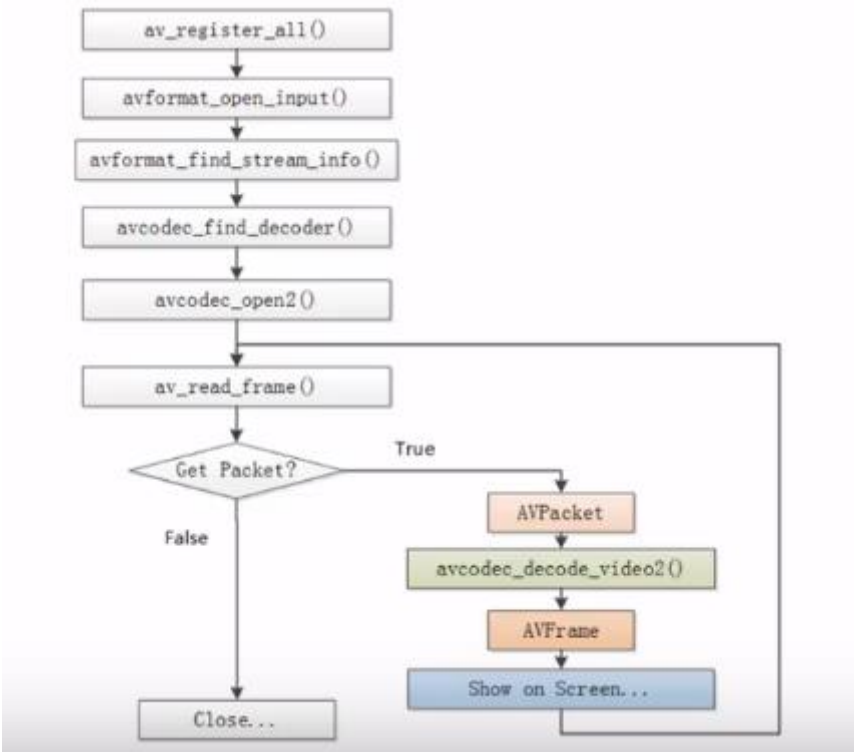
至此，我们可知视频播放器即是对封装在容器里的（mp4,mkv,…）的视频进行**解封装->解码->显示**的过程。



2.4 ffmpeg 视频解码过程

1. 注册所有容器格式和 CODEC:av_register_all()
2. 打开文件:av_open_input_file()
3. 从文件中提取流信息:av_find_stream_info()
4. 穷举所有的流，查找其中种类为 CODEC_TYPE_VIDEO
5. 查找对应的解码器:avcodec_find_decoder()
6. 打开编解码器:avcodec_open()
7. 为解码帧分配内存:avcodec_alloc_frame()

8. 不停地从码流中提取出帧数据:av_read_frame()
9. 判断帧的类型，对于视频帧调用:avcodec_decode_video()
10. 解码完后，释放解码器:avcodec_close()
11. 关闭输入文件:av_close_input_file()



2.5 ffmpeg 数据结构