Technology Tarena 达内科技

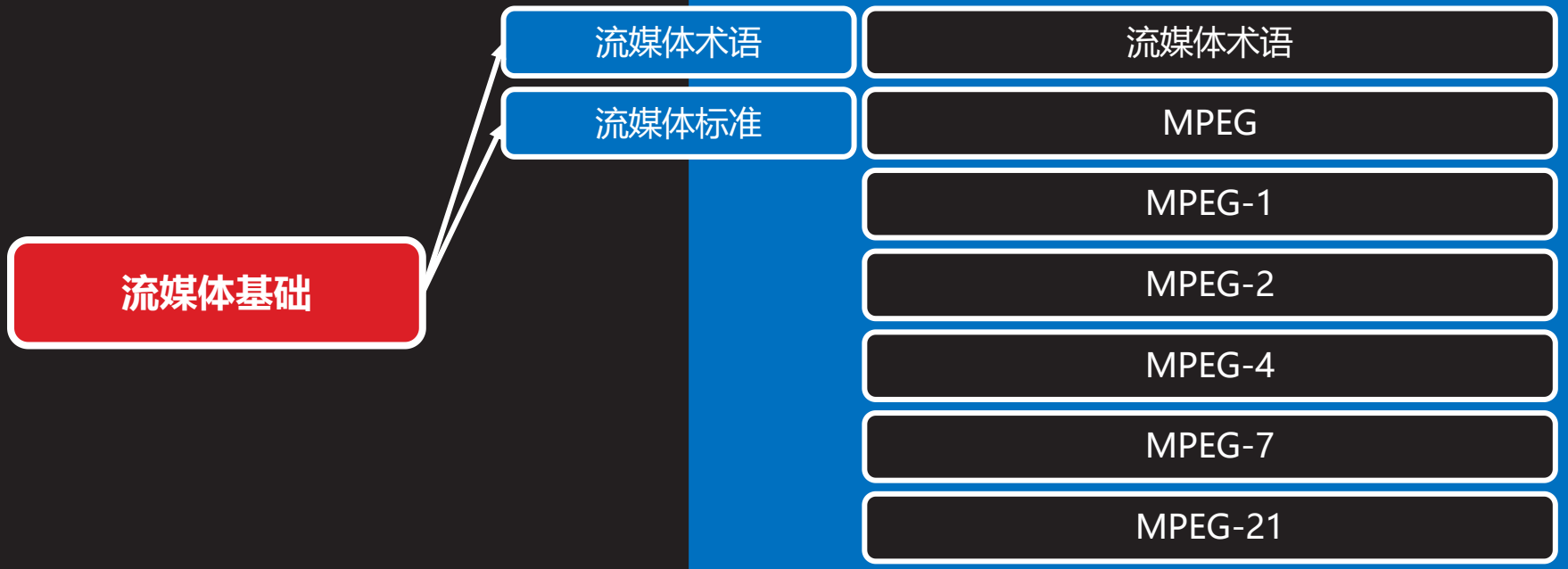
流媒体高级编程

STREAMING MEDIA DAY01

内容

上午	09:00 ~ 09:30	流媒体基础
	09:30 ~ 10:20	
	10:30 ~ 11:20	FFmpeg基础
	11:30 ~ 12:20	
下午	14:00 ~ 14:50	FFmpeg命令
	15:00 ~ 15:50	
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑

流媒体基础


流媒体基础
流媒体术语
流媒体标准
流媒体术语
MPEG
MPEG-1
MPEG-2
MPEG-4
MPEG-7
MPEG-21

Technology Tarena 达内科技

流媒体术语

流媒体术语

- 容器(Container)
  - 文件格式，如flv、mkv等，由文件头和流组成
- 流(Stream)
  - 多媒体数据，音频流、视频流、字幕流、数据流、附加流
- 打包(mux)
  - 将多个不同流按规则放入容器
- 解包(Demux)
  - 从容器中单独提取某个流

知识讲解

流媒体术语(续1)

知识讲解

- 帧(Frame)
  - 一幅静止图像，I帧、P帧、B帧
    - ✓ I (Intra)帧：帧内编码帧，可被直接解码为一张图片
    - ✓ P (Predictive)帧：预测编码帧，参考前一个I帧或B帧，还原一张图片
    - ✓ B (Bi-directional interpolated prediction)帧：双向内插预测编码帧，参考前一个I帧或P帧及后一个P帧，还原一张图片
- 编码(Code)
  - 压缩音视频数据
- 解码(Decode)
  - 解压音视频数据

合称编解码(Codec)

Technology Tarena 达内科技

流媒体标准

MPEG

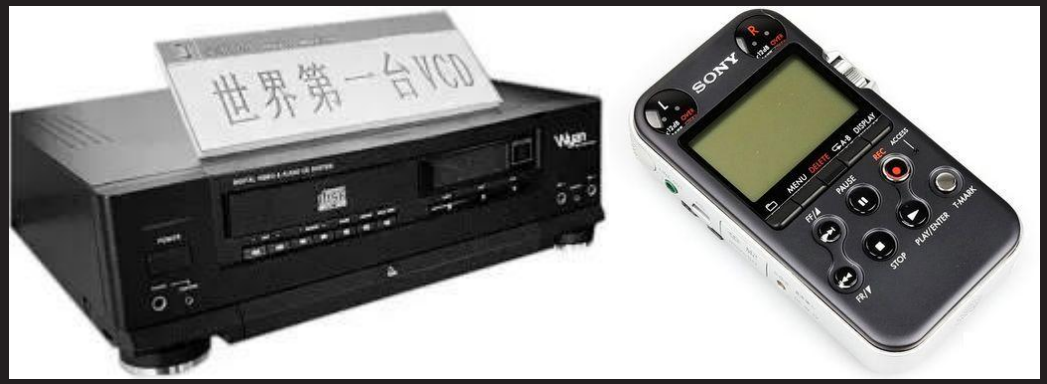
知识讲解

- MPEG (Moving Picture Experts Group，活动图像专家组)，ISO (International Standardization Organization，国际标准化组织)与IEC (International Electrotechnical Commission，国际电工委员会)于1988年成立的专门针对活动图像和声音信号制定压缩解压缩标准的国际化组织
- 截至目前MPEG共推出以下五个标准：
  - MPEG-1
  - MPEG-2
  - MPEG-4
  - MPEG-7
  - MPEG-21

MPEG-1

- 1992年，ISO/IEC11172，具体分为三层
  - 第1层：算法复杂度最低，数字录音盒带、VCD中的音频部分
  - 第2层：算法复杂度居中，数字音频广播、VCD中的视频部分
  - 第3层：算法复杂度最高，网络高质音频，如MP3

知识讲解



MPEG-2

- 1994年，ISO/IEC13818，按压缩比分为五个档次(Profile)，每个档次按图像清晰度分为四个级别(Level)，共20种组合，其中较为常见的11种组合分别应用于不同场合

级别	档次				
	Simple	Main	SNR Scalable	Spatially Scalable	High
Low		VCD	✓		
Main	✓	DVD	✓		✓
High-1440		HDTV		✓	✓
High		HDTV			✓

- Low : 352x240x30/352x288x25
  Main : 720x480x30/720x576x25
  High-1440 : 1440x1080x30 (4:3)
  High : 1920x1080x30 (16:9)

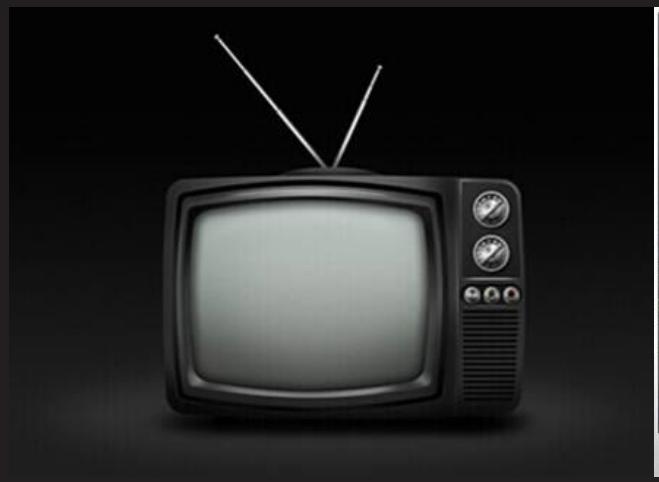
知识讲解

MPEG-2(续1)

Technology Tarena 达内科技

- 在电视行业里对视频清晰度的划分还有另外一套规范
  - 标　清：480x320/640x480
  - 高　清：1024x720/1920x1080i (i表示隔行扫描)
  - 全高清：1920x1080p (p表示逐行扫描)
  - 超高清：3840x2160 (4K)/7680x4320 (8K)

知识讲解



MPEG-4

- 1998年，ISO/IEC14496，利用帧重建技术进一步压缩数据，以求用最少的数据获得最佳的图像质量，图像质量下降不大，数据量却可比MPEG-2缩减数倍
- MPEG-4在嵌入式、互联网等存储、传输条件较为苛刻的领域大有用武之地

知识讲解

MPEG-7

Technology Tarena 达内科技

知识讲解

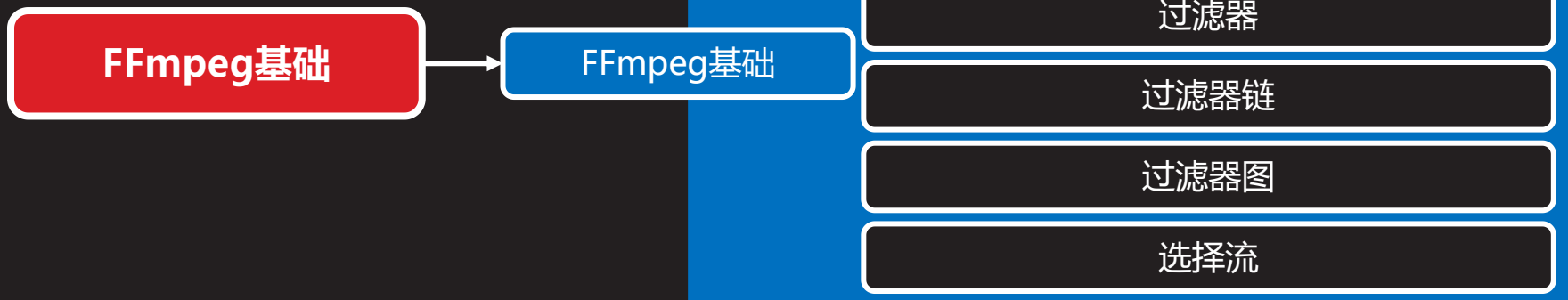
- 2002年，ISO/IEC15938，即MPEG-1+2+4，并非针对音视频数据的压缩编码方法，而是一种用来描述多媒体内容的标准
- 建立MPEG-7标准的出发点是依靠众多的参数对图像与声音实现分类，并可对存储它们的数据库进行查询
- 可用于影像编目、音乐词典、多媒体查询、频道选取、个性化电子新闻、媒体创作等领域

MPEG-21

- 正在制定中的多媒体框架标准，旨在解决如何将不同的技术和标准结合在一起，以及需要什么新的标准以实现不同标准的结合
- 通常所说的MP4即指MPEG-4，但MP3却并非MPEG-3
  - 在MPEG标准的制定历史中，确曾出现过MPEG-3，不过很快就被放弃了，现在所说的MP3实际指的是MPEG-1第3层，即针对网络高质音频的标准部分
- MPEG-5和MPEG-6压根就不存在
- DivX和MPEG-4有关联但并不完全等价
  - DivX实际上是将MP4标准的视频流和MP3标准的音频流打包到AVI格式的容器中，其数据量比MPEG-2小很多，但画质却和MPEG-2非常接近

知识讲解

FFmpeg基础



Technology Tarena 达内科技

FFmpeg基础

FFmpeg是什么?

- FFmpeg (Fast Forward mpeg，快进MPEG)，针对音频、视频流媒体的采集、转换、存储、传输和播放的开源软件库和应用程序
- 截至目前FFmpeg的用户包括Google、Facebook、Youtube、优酷、爱奇艺、土豆等等

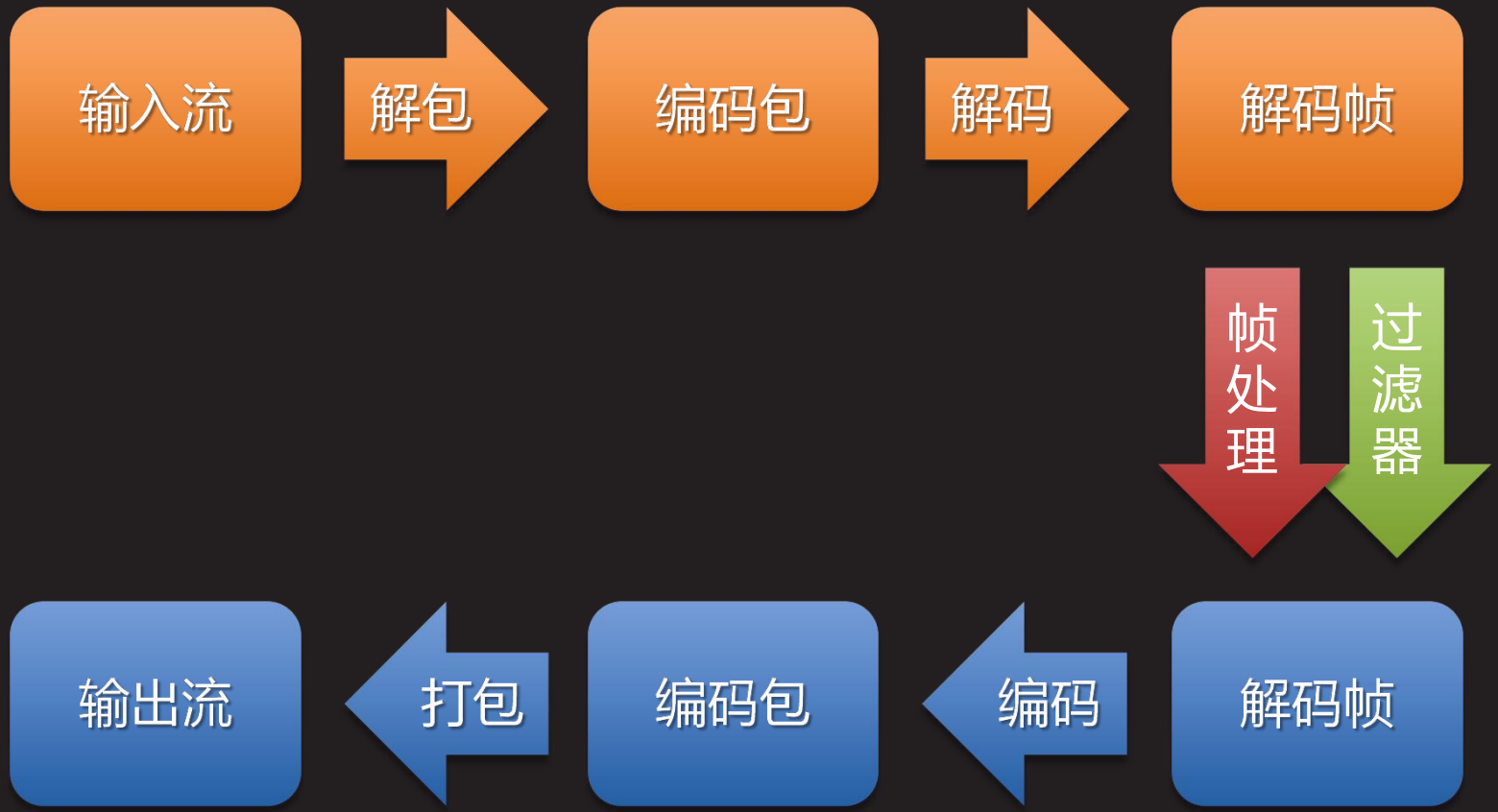
知识讲解

FFmpeg是什么？(续1)

- FFmpeg软件开发工具包(SDK)包括8个库、3个工具，以及头文件、文档和示例代码

```
– ffmpeg
  |
  +- bin/                          // 动态库及可执行程序
  |   +- avdevice-57.dll           // 音视频设备函数动态库
  |   +- ...
  |   +- ffmpeg.exe                // 流媒体处理工具
  |   +- ...
  +- include/                      // 头文件
  +- lib/                          // 导入库
  +- doc/                          // 文档
      +- examples/                 // 示例代码
```


FFmpeg处理流程



过滤器

- 音频过滤器：-af 过滤器名=参数
  - 例如：以原速度的50%播放音频文件
    ffplay audio.flac -af atempo=0.5
- 视频过滤器：-vf 过滤器名=参数
  - 例如：宽高取原尺寸的一半播放视频文件
    ffplay video.vob -vf scale=iw/2:-1
  - 例如：顺时针旋转90度播放视频文件
    ffplay video.vob -vf transpose=1

知识讲解

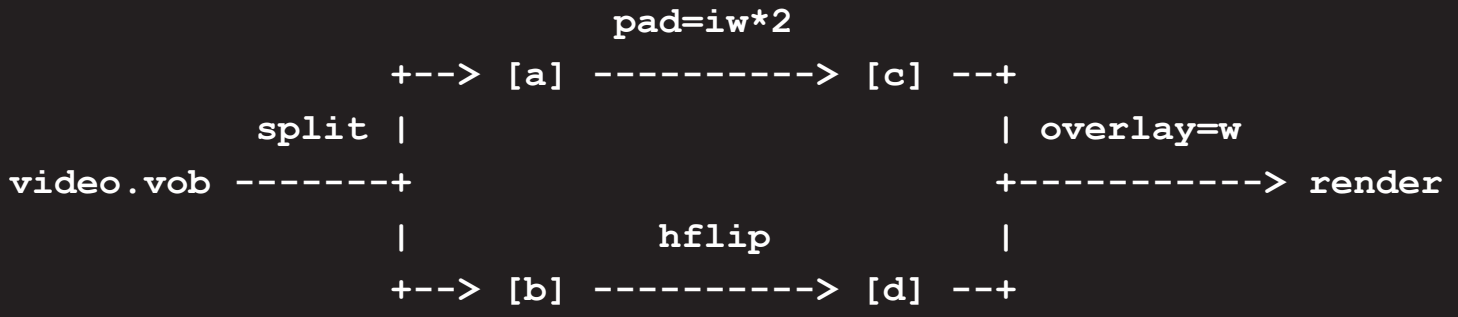
过滤器链

- 过滤器名1=参数1, 过滤器名2=参数2, ...
  - 例如：宽高取原尺寸的一半顺时针旋转90度播放视频文件
    ffplay video.vob -vf scale=iw/2:-1,transpose=1

知识讲解

过滤器图

- 过滤器链1; 过滤器链2; ...
  - 例如：同时播放视频文件原始图像和水平镜像
    ffplay video.vob -vf
    split[a][b];[a]pad=iw*2[c];[b]hflip[d];[c][d]overlay=w



知识讲解

选择流

- -map 文件索引[:流类型[:流索引]]
  - `-map 0` // 第0个容器中的所有流
    `-map 0:a` // 第0个容器中的所有音频(audio)流
    `-map 0:a:0` // 第0个容器中的第0个音频流
    `-map 0:v` // 第0个容器中的所有视频(video)流
    `-map 0:s` // 第0个容器中的所有字幕(subtitles)流
    `-map 0:d` // 第0个容器中的所有数据(data)流
    `-map 0:t` // 第0个容器中的所有附加(attachment)流
  - 例如：将audio.flac中的第0个音频流和video.vob中的第0个视频流合并到map1.mp4中
    ffmpeg -i audio.flac -i video.vob -map 0:a:0 -map 1:v:0 map1.mp4

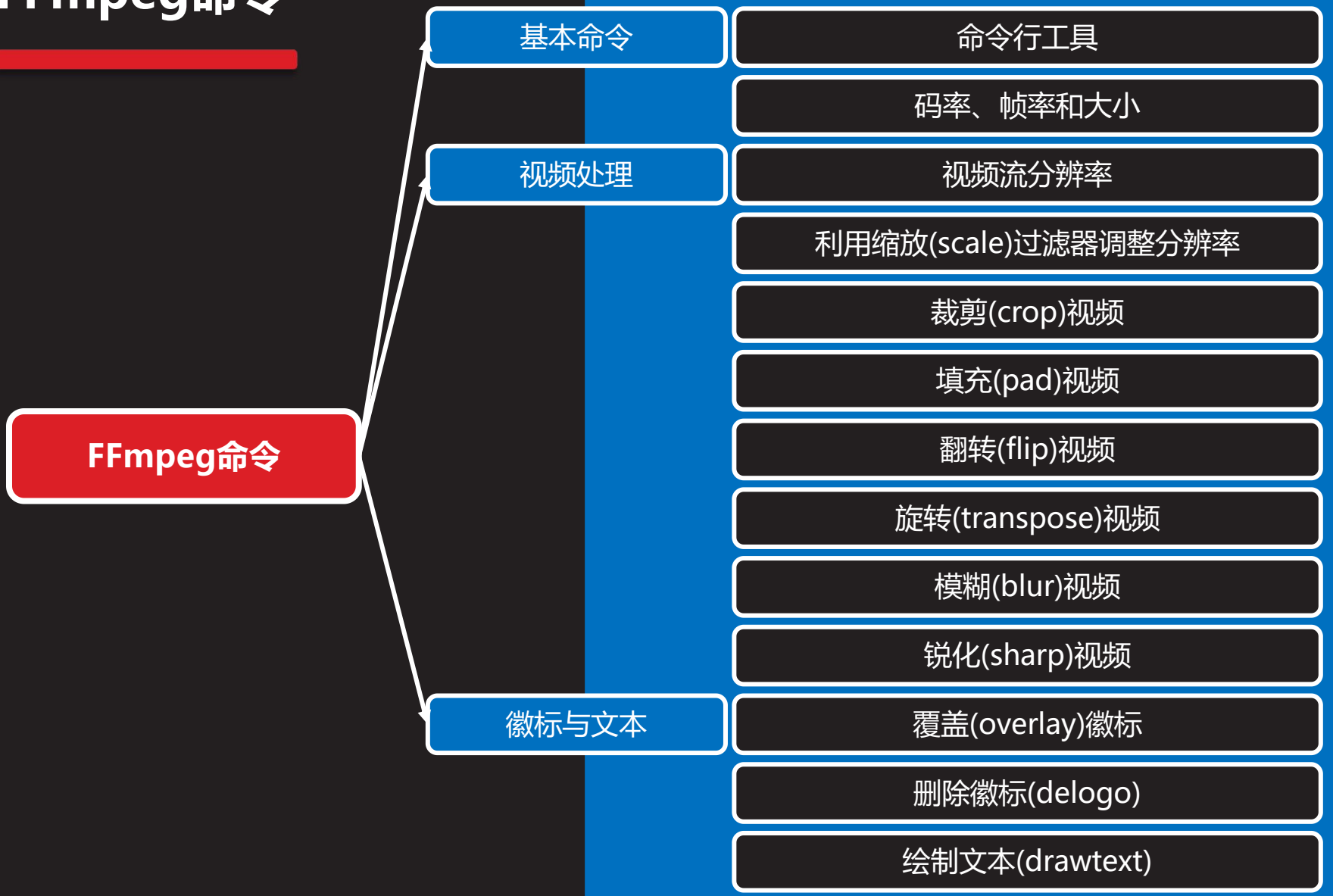
知识讲解

选择流(续1)

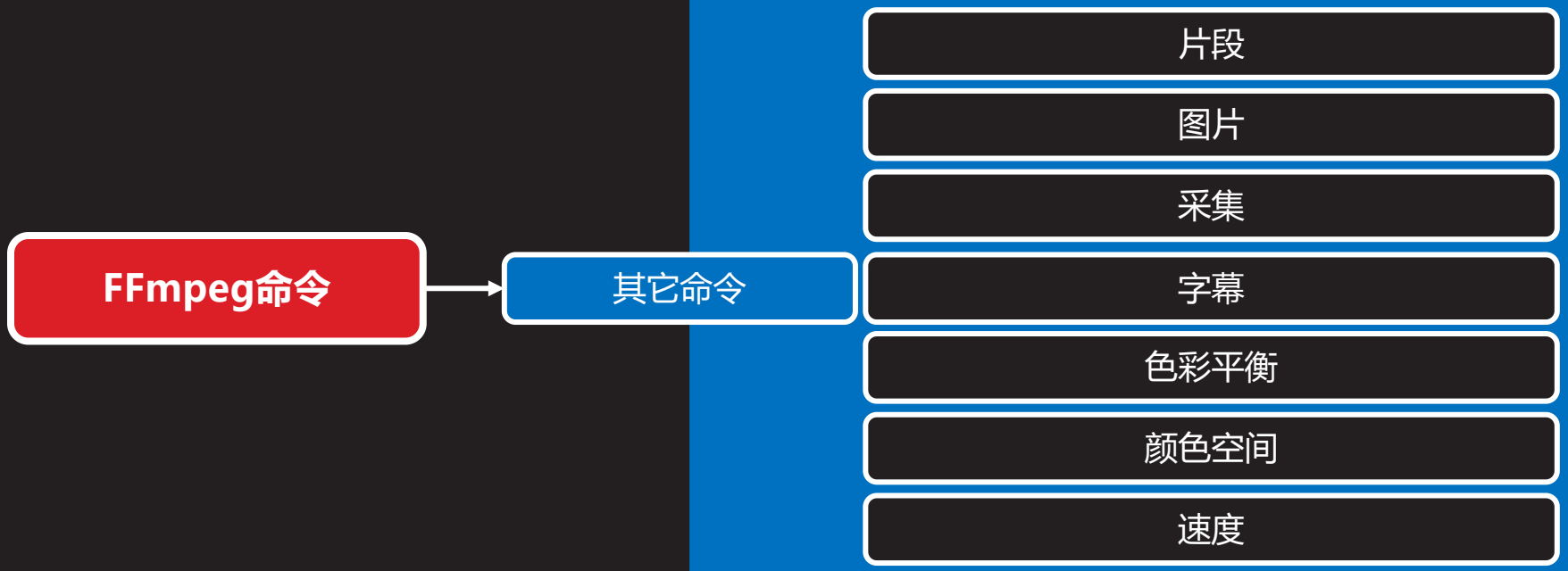
- -map 文件索引[:流类型[:流索引]]
  - 例如：将audio.flac和map1.mp4中除map1.mp4音频流以外的所有流合并到map2.mp4中
    ffmpeg -i audio.flac -i map1.mp4 -map 0 -map 1 -map -1:a map2.mp4
  - 也可以通过-an/vn/sn/dn/tn表示除所有音频/视频/字幕/数据/附加流之外
    例如：将audio.flac和map2.mp4中除音频流以外的所有流合并到map3.mp4中
    ffmpeg -i audio.flac -i map2.mp4 -an map3.mp4

知识讲解

FFmpeg命令


FFmpeg命令
基本命令
命令行工具
码率、帧率和大小
视频处理
视频流分辨率
利用缩放(scale)过滤器调整分辨率
裁剪(crop)视频
填充(pad)视频
翻转(flip)视频
旋转(transpose)视频
模糊(blur)视频
锐化(sharp)视频
徽标与文本
覆盖(overlay)徽标
删除徽标(delogo)
绘制文本(drawtext)

FFmpeg命令



Technology Tarena 达内科技

基本命令

命令行工具

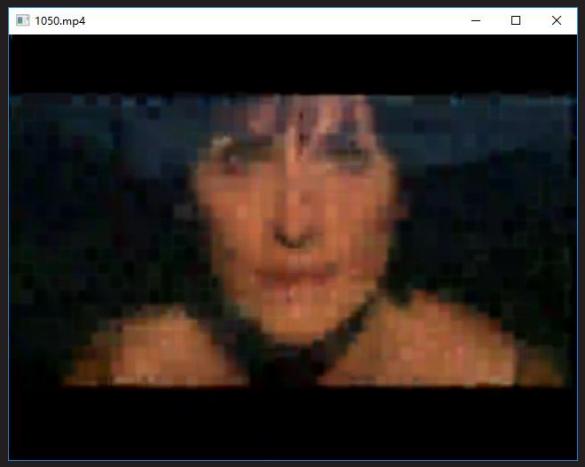
- FFmpeg包括三个命令行工具：ffmpeg、ffplay和ffprobe，分别用于流媒体的转换、播放和格式查看，它们各自带有一套命令行参数
  - ffmpeg [选项] [[输入流选项] -i 输入流] ... [[输出流选项] 输出流] ...
    - ✓ `ffmpeg -version` // 打印版本信息
    - ✓ `ffmpeg -L` // 打印许可证信息
    - ✓ `ffmpeg -h` // 打印帮助信息
    - ✓ `ffmpeg -h long` // 打印更多帮助信息
    - ✓ `ffmpeg -h full` // 打印全部帮助信息
    - ✓ `ffmpeg -h filter=atempo` // 打印atempo过滤器的帮助信息
    - ✓ `ffmpeg -formats` // 打印文件格式信息
    - ✓ `ffmpeg -encoders` // 打印编码器信息
    - ✓ `ffmpeg -decoders` // 打印解码器信息
    - ✓ `ffmpeg -filters` // 打印过滤器信息
    - ✓ `ffmpeg -bsfs` // 打印bitstream过滤器信息
    - ✓ `ffmpeg -layouts` // 打印声道布局信息
    - ✓ `ffmpeg -pix_fmts` // 打印像素格式信息
    - ✓ `ffmpeg -protocols` // 打印传输协议信息

知识讲解

码率、帧率和大小

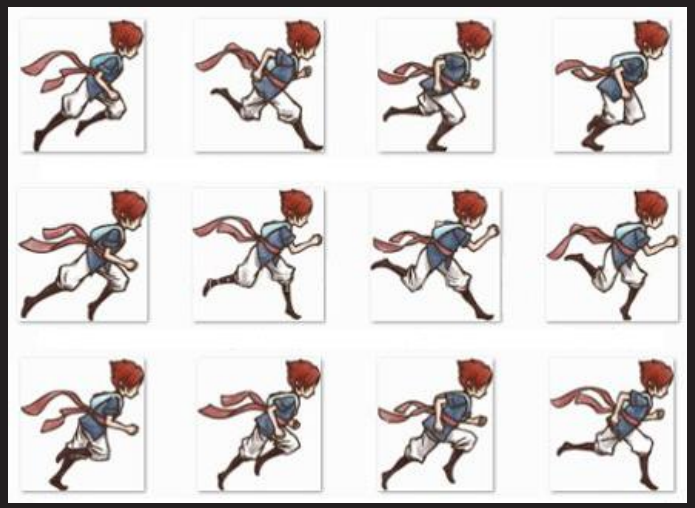
- 码率，亦称比特率，即媒体流中每一秒的二进制位数，记作bps (bit per second，比特/秒)，音频流的码率越高，音质越好，视频流的码率越高，画面越清晰
  - 例如：将video.vob中的音频和视频码率分别降至10kbps和50kbps，存到1050.mp4中
    ffmpeg -i video.vob -b:a 10k -b:v 50k 1050.mp4

知识讲解



码率、帧率和大小(续1)

- 帧率，亦称帧速率，即视频流中每一秒的画面帧数，记作fps (frame per second，帧/秒)，视频流的帧率越高，画面越连贯，人类肉眼对连续变化影像的识别，至少需要15fps的帧率
  - 例如：以每秒一帧即1fps的帧率播放视频文件
    ffplay video.vob -vf fps=1



知识讲解

码率、帧率和大小(续2)

Technology Tarena 达内科技

- 媒体流大小(字节)=(音频码率+视频码率)x时间长度/8
  - 例如：video.vob，音频码率192kbps，视频码率9611kbps，时间长度217秒，则其媒体流大小为：(192000+9611000)x217/8=265906375≈254M(字节)，再加上文件打包额外增加的数据，总字节数会比这更多，但其实际文件大小只有约65M，这是压缩编码带来的效果
  - 例如：从video.vob头部截取一段内容保存到1M.mp4中，使后者大小在1M字节左右
    ffmpeg -i video.vob -fs 1M 1M.mp4

知识讲解

Technology Tarena 达内科技

视频处理

视频流分辨率

- 视频流的分辨率用水平和垂直方向上的像素数表示，在相同几何尺寸和分辨率设置的显示设备上，视频流的分辨率越高，画面越大
  - 例如：将video.vob的视频流分辨率降至320x240，保存到qvga.mp4中
    ffmpeg -i video.vob -s 320x240 qvga.mp4，或者
    ffmpeg -i video.vob -s qvga qvga.mp4

知识讲解

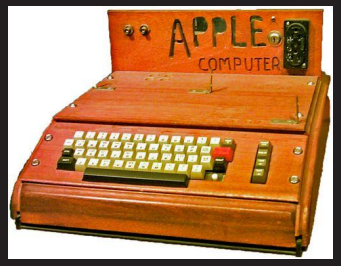
视频流分辨率(续1)

Technology Tarena 达内科技

知识讲解

- 预定义分辨率
  - 128 x 96, sqcif, 老式手机
    176 x 144, qcif, 手机
    320 x 200, cga, 老式CRT显示器
    320 x 240, qvga, 手机、网络摄像头
    352 x 288, cif, 手机
    640 x 350, ega, 老式CRT显示器
    640 x 480, vga, 显示器、网络摄像头
    800 x 600, svga, 显示器
    1024 x 768, xga, 显示器
    1280 x 1024, sxga, 显示器
    1366 x 768, wxga, 宽屏显示器
    1600 x 1200, uxga, 显示器
    1920 x 1080, hd1080, 宽屏显示器、高清电视
    ...





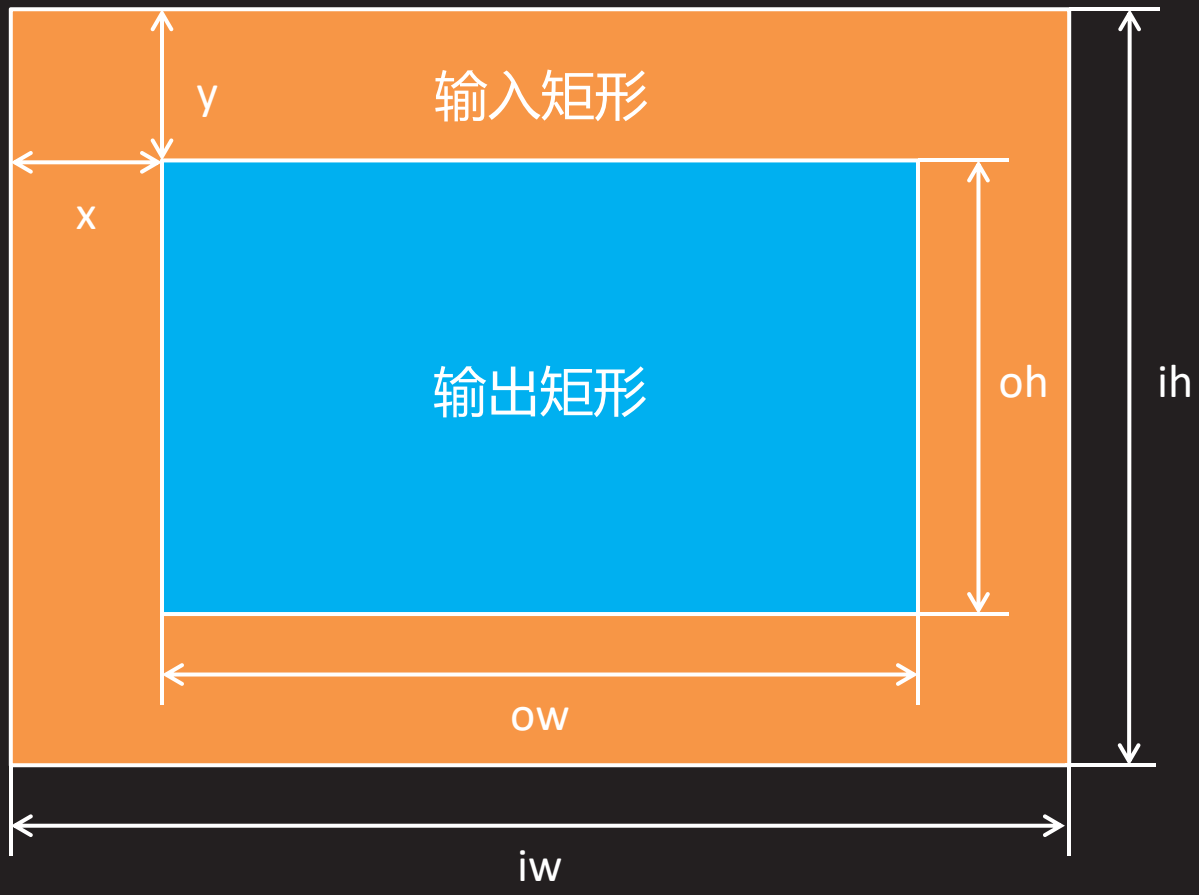

利用缩放(scale)过滤器调整分辨率

- ... -vf scale=<宽度>:<高度> ...
  - 例如：播放video.vob，分辨率320x240
    ffplay video.vob -vf scale=320:240
  - <宽度>和<高度>除使用具体数值外，还可使用以下标识：
    - ✓ `iw/in_w` ： 输入宽度
    - ✓ `ih/in_h` ： 输入高度
    - ✓ `ow/out_w` ： 输出宽度
    - ✓ `oh/out_h` ： 输出高度
    - ✓ 例如：播放video.vob，分辨率宽高各取一半
      ffplay video.vob -vf scale=iw/2:in_h*.5
  - "a"表示原始纵横比(aspect ratio)
    - ✓ 例如：播放video.vob，分辨率高度240，纵横比不变
      ffplay video.vob -vf scale=240*a:240
  - "-1"表示原始纵横比下的宽度或高度
    - ✓ 例如：播放video.vob，分辨率高度240，纵横比不变
      ffplay video.vob -vf scale=-1:240

知识讲解

裁剪(crop)视频

- 所谓裁剪视频即从输入视频的矩形区域中选取一个局部矩形作为输出

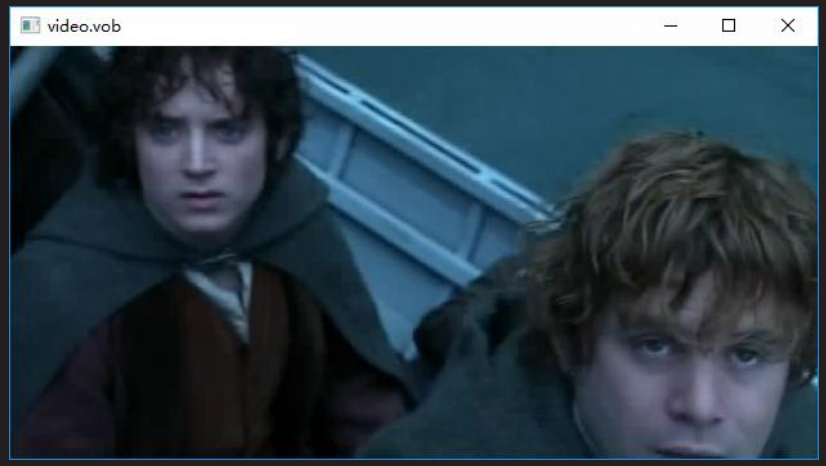


知识讲解

裁剪(crop)视频(续1)

- ... -vf crop=<ow>:<oh>:<x>:<y> ...
  - 例如：去除video.vob的视频黑边
    ffplay video.vob -vf crop=704:320:6:72
  - 例如：裁剪video.vob，输出矩形宽高为输入矩形宽高的一半，位置居中
    ffplay video.vob -vf crop=iw/2:ih/2:(iw-ow)/2:(ih-oh)/2

知识讲解

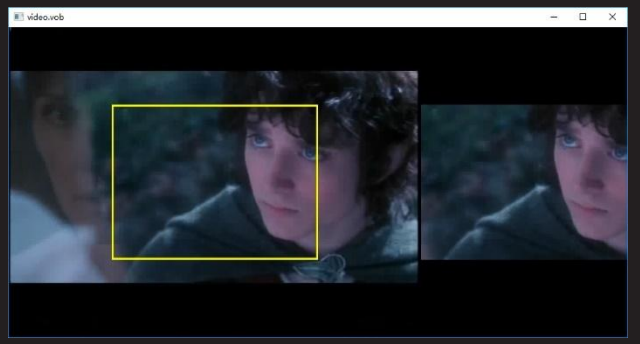


裁剪(crop)视频(续2)

- 若<x>和<y>取缺省值，则输出矩形相对于输入矩形水平垂直居中
  - 例如：裁剪video.vob，输出矩形宽高为输入矩形宽高的一半，位置居中

    ffplay video.vob -vf crop=iw/2:ih/2
  - 例如：裁剪video.vob，输出矩形宽高为输入矩形宽高的一半，位置居中，并与原始视频同屏显示

    ffplay video.vob -vf split[a][b];
    [a]drawbox=(iw-iw/2)/2:(ih-ih/2)/2:iw/2:ih/2:c=yellow[c];
    [c]pad=iw*3/2[d];
    [b]crop=iw/2:ih/2[e];
    [d][e]overlay=w*2:(H-h)/2



知识讲解

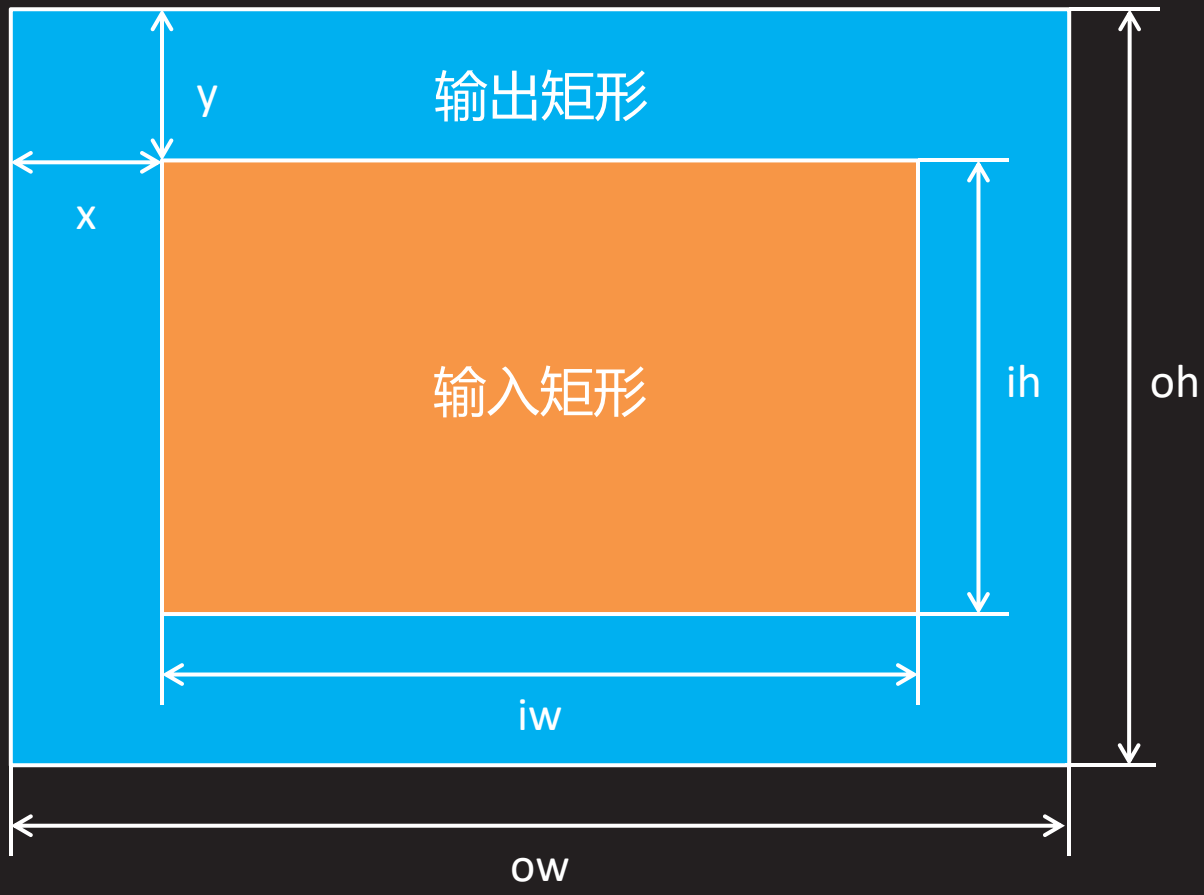
裁剪(crop)视频(续3)

- cropdetect过滤器可打印出针对视频黑边的裁剪矩形
  - 例如：打印video.vob视频黑边的裁剪矩形
    ffplay video.vob -vf cropdetect
    ... crop=704:320:6:72 ...
    ffmpeg -i video.vob -vf crop=704:320:6:72 noframe.mp4
  - 例如：通过裁剪宽度将noframe.mp4的视频变为16:9比例
    ffplay noframe.mp4 -vf crop=ih*16/9:ih
  - 例如：视频颤抖
    ffplay noframe.mp4 -vf crop=iw*3/4:ih*3/4:(iw-ow)/2*(1+sin(n/10)):(ih-oh)/2*(1+sin(n/7))

知识讲解

填充(pad)视频

- 所谓填充视频即将输入视频的矩形区域中放在一个更大的输出矩形中，多出的区域填以特定的颜色

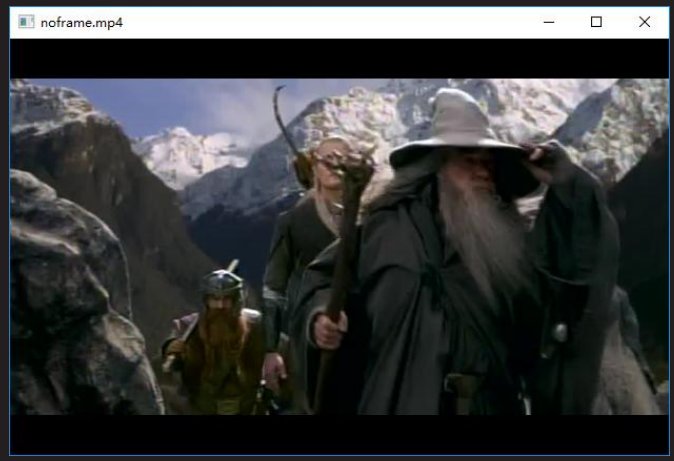


知识讲解

填充(pad)视频(续1)

- ... -vf pad=<ow>:<oh>:<x>:<y>:<color> ...
  - 例如：为noframe.mp4的视频增加40像素宽的粉色边框
    ffplay noframe.mp4 -vf pad=iw+80:ih+80:(ow-iw)/2:(oh-ih)/2:pink
  - 例如：通过填充高度将noframe.mp4的视频变为16:9比例
    ffplay noframe.mp4 -vf pad=iw:iw*9/16:0:(oh-ih)/2

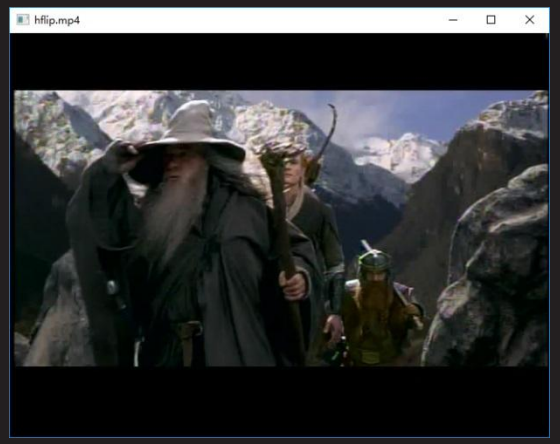
知识讲解



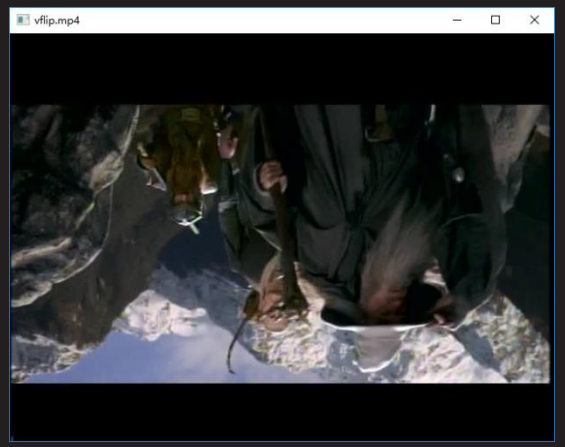
翻转(flip)视频

知识讲解

- 水平翻转：... -vf hflip ...
  - 例如：水平翻转video.vob的视频
    ffmpeg -i video.vob
    -vf hflip hflip.mp4



- 垂直翻转：... -vf vflip ...
  - 例如：垂直翻转video.vob的视频
    ffmpeg -i video.vob
    -vf vflip vflip.mp4



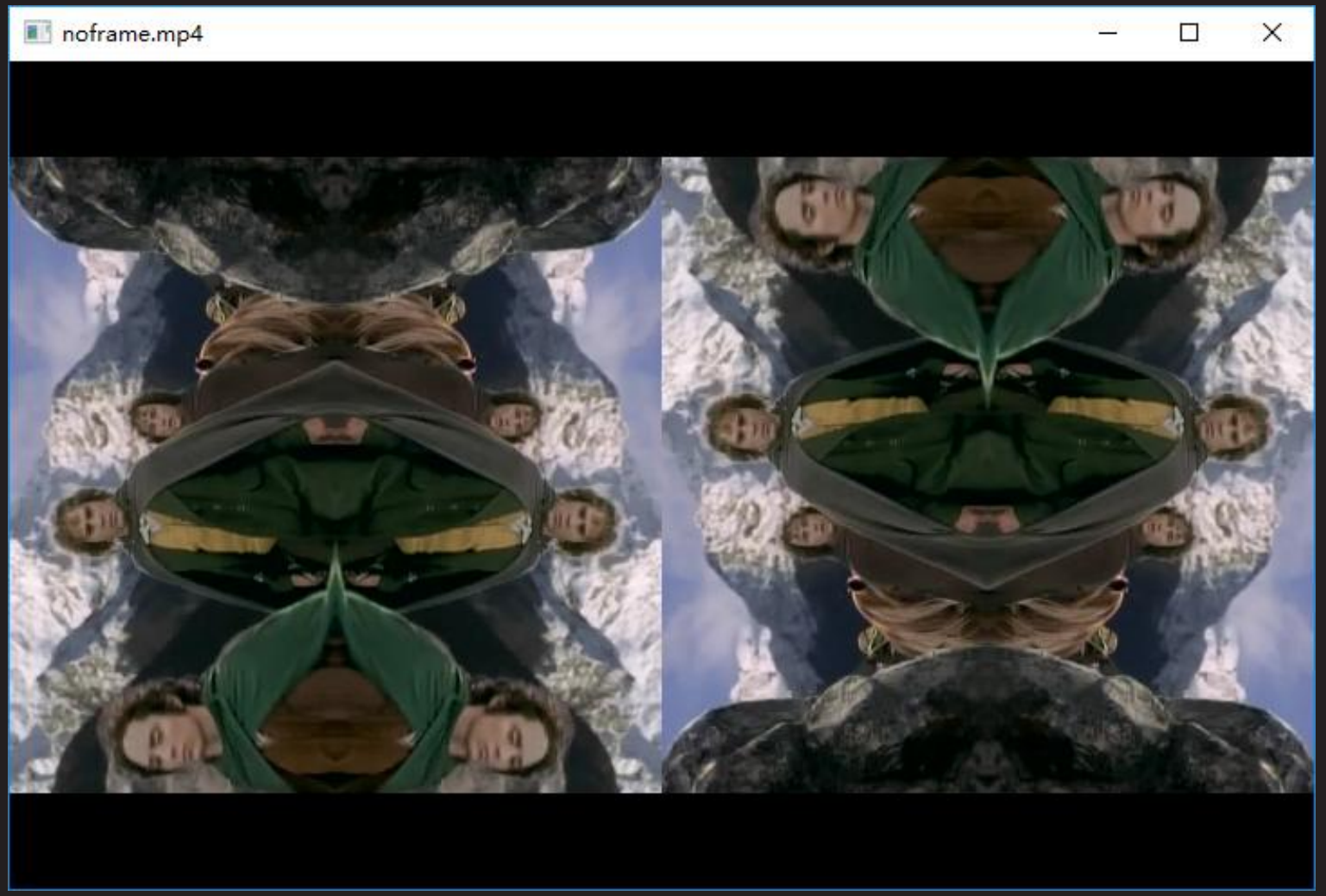
旋转(transpose)视频

Technology Tarena 达内科技

知识讲解

- ... -vf transpose=0/1/2/3 ...
  - 0：逆时针旋转90度后垂直翻转
    1：顺时针旋转90度
    2：逆时针旋转90度
    3：顺时针旋转90度后垂直翻转
  - 例如：同屏显示noframe.mp4视频的四种旋转
    ffplay noframe.mp4 -vf split=4[a][b][c][d];
    [a]transpose=0[e];[b]transpose=1[f];[c]transpose=2
    [g];[d]transpose=3[h];[e]pad=iw*4:ih[i];[i][f]overlay=
    w[j];[j][g]overlay=w*2[k];[k][h]overlay=w*3

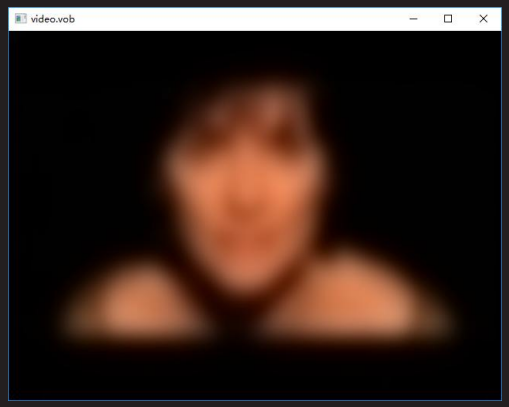
旋转(transpose)视频(续1)



知识讲解

模糊(blur)视频

- ... -vf boxblur=<lr>:<lp>:<cr>:<cp>:<ar>:<ap> ...
  - lr (luma radius) : 亮度模糊半径
    lp (luma power) : 亮度模糊次数
    cr (chroma radius) : 色度模糊半径
    cp (chroma power) : 色度模糊次数
    ar (alpha radius) : 透明度模糊半径
    ap (alpha power) : 透明度模糊次数
    模糊次数的缺省值均为2，取0表示不做模糊处理
  - 例如：对video.vob的视频做模糊处理
    ffplay video.vob -vf boxblur=9:5



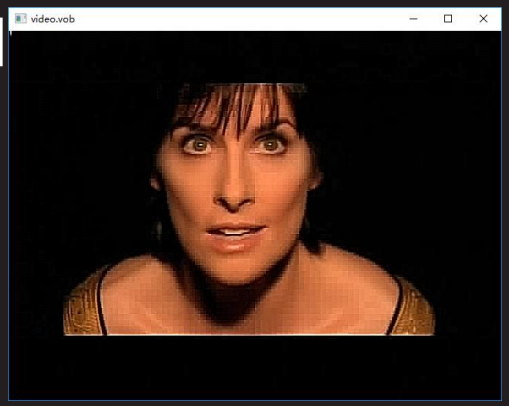
知识讲解

锐化(sharp)视频

Technology Tarena 达内科技

知识讲解

- ... -vf unsharp=l_msize_x:l_msize_y:l_amount:c_msize_x:c_msize_y:c_amount ...
  - `l_msize_x` ： 水平亮度矩阵，取值范围3 - 13，默认值5
    `l_msize_y` ： 垂直亮度矩阵，取值范围3 - 13，默认值5
    `l_amount` ： 亮度强度，取值范围-2.0 - 5.0，负数为模糊，默认值1.0
    `c_msize_x` ： 水平色彩矩阵，取值范围3 - 13，默认值5
    `c_msize_y` ： 垂直色彩矩阵，取值范围3 - 13，默认值5
    `c_amount` ： 色彩强度，取值范围-2.0 - 5.0，负数为模糊，默认值0.0
  - 例如：对video.vob的视频做锐化处理
    ffplay video.vob -vf unsharp=9:9:5

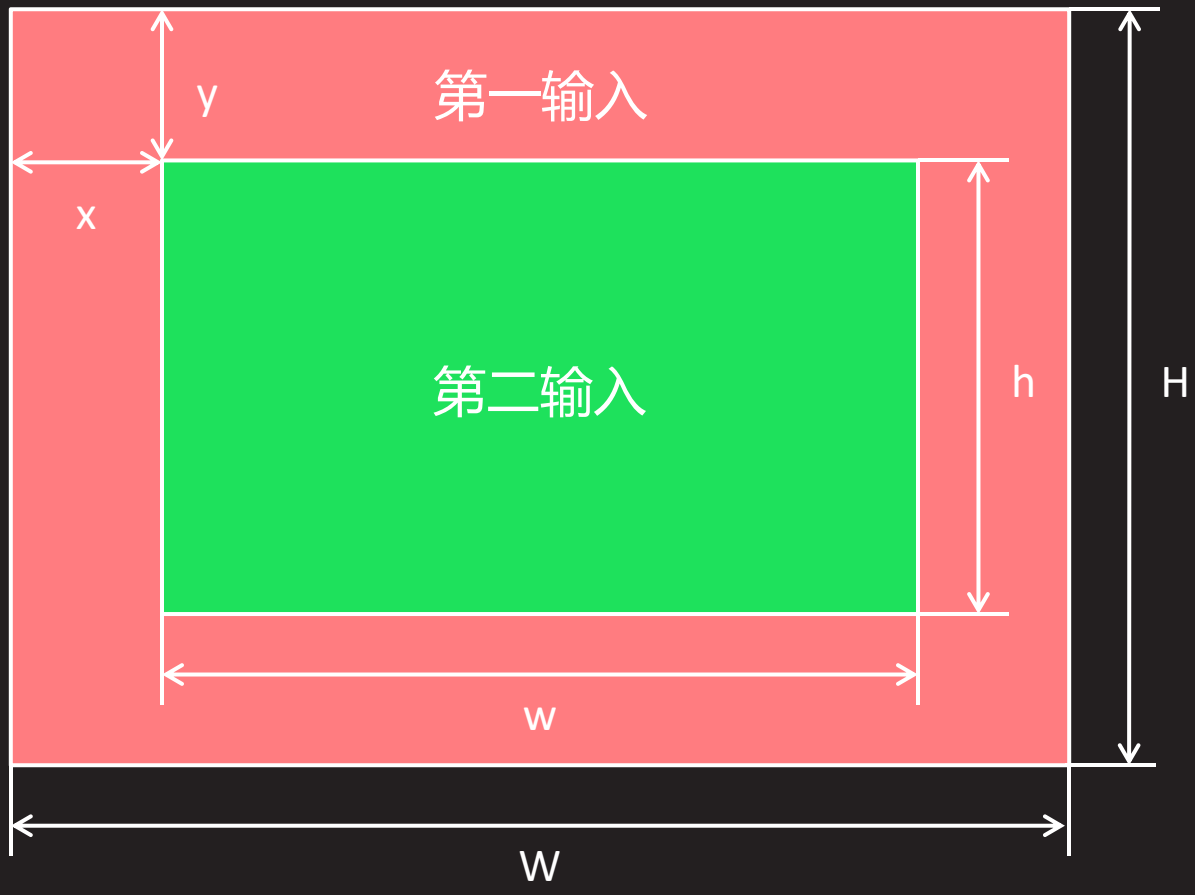


Technology Tarena 达内科技

徽标与文本

覆盖(overlay)徽标

- 所谓覆盖即将第二输入覆盖到第一输入的指定位置处，可以此实现画中画及添加徽标的功能

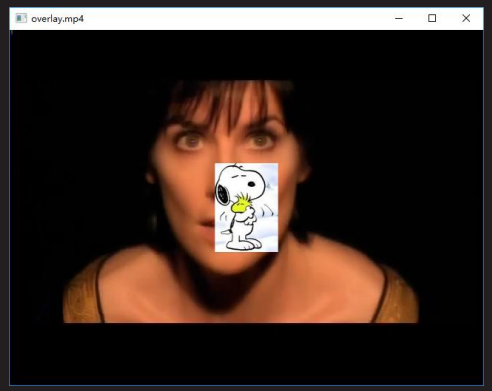


知识讲解

覆盖(overlay)徽标(续1)

Technology Tarena 达内科技

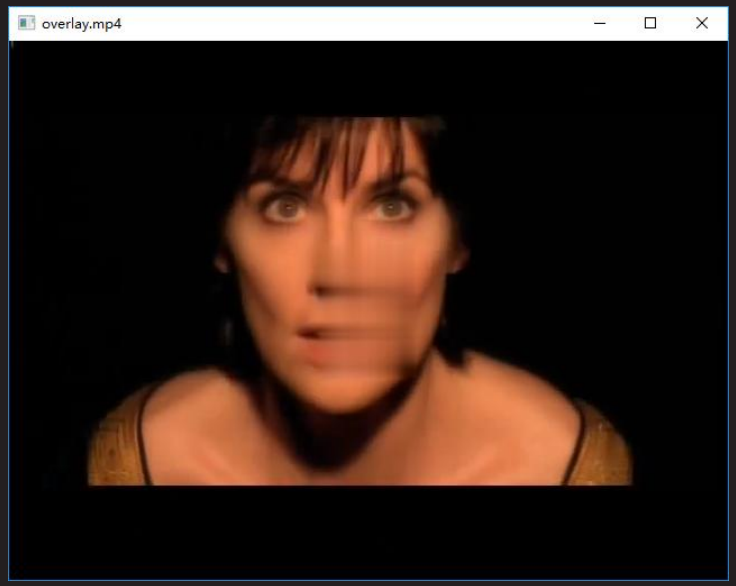
- ... -filter_complex overlay=<x>:<y> ...
  - 例如：将snoopy.jpg覆盖到video.vob视频的中心位置，保存到overlay.mp4中
    ffmpeg -i video.vob -i snoopy.jpg -filter_complex overlay=(W-w)/2:(H-h)/2 overlay.mp4
  - 例如：播放video.vob，将snoopy.jpg覆盖到视频中，垂直居中，水平滚动
    ffplay video.vob -vf movie=snoopy.jpg[a]; [in][a]overlay=mod(W+w-t*50\,W+w+1)-w:(H-h)/2
  - 例如：播放video.vob，将snoopy.jpg覆盖到视频的中心位置，每三秒钟显示一秒钟
    ffplay video.vob -vf movie=snoopy.jpg[a]; [in][a]overlay='if(lt(mod(t\,3)\,1),(W-w)/2,NAN)':(H-h)/2



知识讲解

删除徽标(delogo)

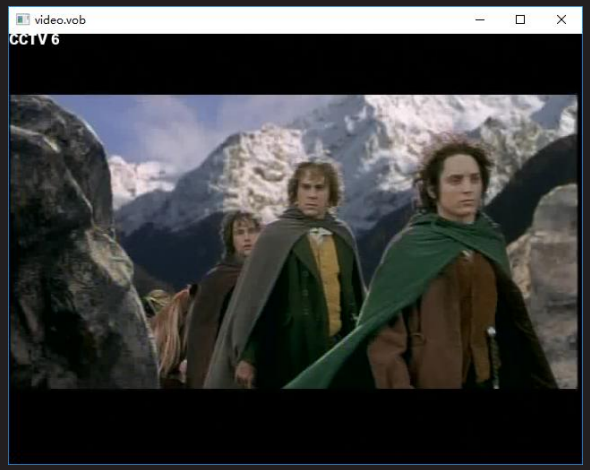
- ... -vf delogo=<左>:<顶>:<宽>:<高>:<边框厚度>:<是否显示绿框> ...
  - 例如：删除overlay.mp4视频中心位置的图片
    ffplay overlay.mp4 -vf delogo=312:180:96:120:0:0



知识讲解

绘制文本(drawtext)

- ... -vf drawtext="fontfile=<字体>:text='<文本>':fontcolor=<颜色>:fontsize=<字号>:x=<水平坐标>:y=<垂直坐标>:enable=<是否显示(1/0)>" ...
  - 例如：用Arial黑体显示CCTV 6白色于左上角
    ffplay video.vob -vf drawtext="fontfile=ariblk.ttf:text='CCTV 6':fontcolor=white"



知识讲解

绘制文本(drawtext)(续1)

- 例如：用Arial黑体显示CCTV 6白色20号字于中心位置

  ffplay video.vob -vf drawtext="fontfile=ariblk.ttf:text='CCTV 6':fontcolor=white:fontsize=20:x=(w-tw)/2:y=(h-th)/2"

  其中tw和th分别表示文本的宽度和高度

知识讲解

绘制文本(drawtext)(续2)

Technology Tarena 达内科技

知识讲解

- 例如：用Arial黑体显示CCTV 6白色20号字垂直居中水平自右向左循环滚动

  ffplay video.vob -vf drawtext="fontfile=ariblk.ttf:text='CCTV 6':fontcolor=white:fontsize=20:x=mod(w+tw-t*50\,w+tw+1)-tw:y=(h-th)/2"

  其中t表示时间，以秒为单位

绘制文本(drawtext)(续3)

Technology Tarena 达内科技

知识讲解

- 例如：用Arial黑体显示当前时间白色20号字于右上角

```
ffplay video.vob -vf
drawtext="fontfile=ariblk.ttf:text='%{localtime\:%H\\
\:%M\\\:%S}':fontcolor=white:fontsize=20:x=w-tw-
50:y=100"
```

- 例如：用Arial黑体显示CCTV 6白色20号字于右上角，每三秒钟显示一秒钟

```
ffplay video.vob -vf drawtext="fontfile=ariblk.ttf:
text='CCTV 6':fontcolor=white:
fontsize=20:x=w-tw-50:
y=100:enable=lt(mod(t\,3)\,1)"
```

Technology Tarena 达内科技

其它命令

片段

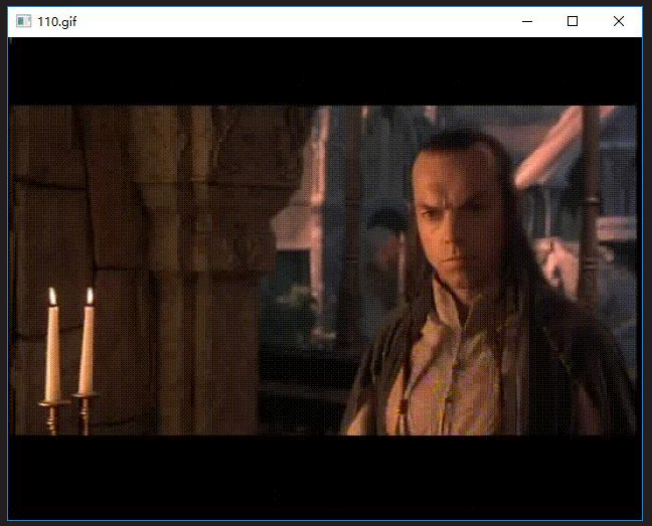
- ... -ss <起始时间> -t <片段时长> ...
  - 例如：播放video.vob从第1分钟开始10秒时长
    ffplay -ss 00:01:00 -t 10 video.vob

知识讲解

图片

- … -ss <起始时间> -t <片段时长> -pix_fmt <图片格式> …
  - 例如：将video.vob从第1分钟开始10秒时长的内容保存到110.gif中

    ffmpeg -ss 00:01:00 -t 10 -i video.vob -pix_fmt rgb24 110.gif



知识讲解

图片(续1)

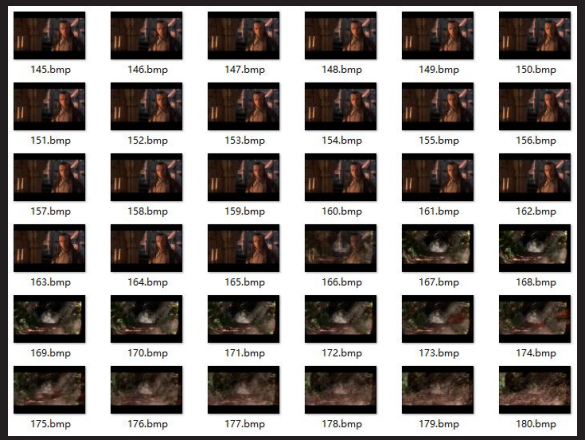
知识讲解

- 例如：将noframe.mp4从第1分钟开始10秒时长的内容每30帧(1秒)截一张图片保存到30.png中

```
ffmpeg -ss 00:01:00 -t 10
-i noframe.mp4 -pix_fmt rgb24
-vf "select=not(mod(n\,30)),
scale=-1:120,tile=2x5" 30.png
```

- 例如：将video.vob从第1分钟开始10秒时长的内容拆成单帧图片保存到110目录下

```
ffmpeg -ss 00:01:00 -t 10
-i video.vob -pix_fmt rgb24
./110/%03d.bmp
```



图片(续2)

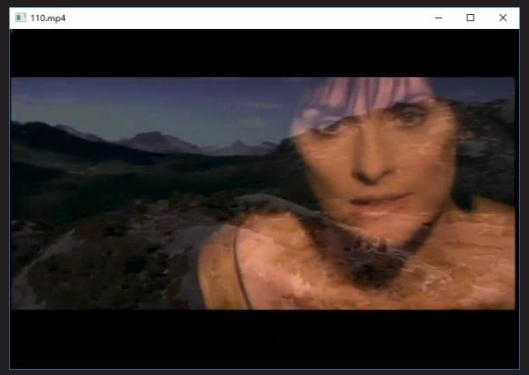
知识讲解

- 例如：将video.vob从第1分钟开始10秒时长的内容每隔2秒截一张图片保存到1102目录下

  ffmpeg -ss 00:01:00 -t 10 -i video.vob -pix_fmt rgb24 -vf fps=fps=1/2 ./1102/%d.bmp

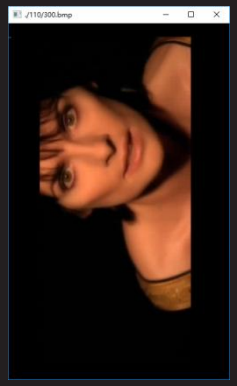
- 例如：将110目录下单帧图片以30fps的帧速率还原成视频文件110.mp4

  ffmpeg -i ./110/%03d.bmp -r 30 110.mp4

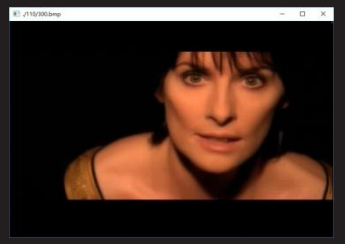


图片(续3)

- 通过视频过滤器裁剪、填充、翻转、旋转和覆盖图片
  - 例如：去除./110/300.bmp的黑边，裁剪宽度至16:9，增加40像素宽的粉色边框
    ffplay ./110/300.bmp -vf crop=704:320:6:72,crop=ih*16/9:ih,pad=iw+80:ih+80:(ow-iw)/2:(oh-ih)/2:pink



  - 例如：将./110/300.bmp水平翻转
    ffplay ./110/300.bmp -vf hflip



  - 例如：将./110/300.bmp逆时针旋转90度后垂直翻转
    ffplay ./110/300.bmp -vf transpose=0
  - 例如：将snoopy.jpg覆盖到./110/300.bmp的中心位置，保存到overlay.bmp中
    ffmpeg -i ./110/300.bmp -i snoopy.jpg -filter_complex overlay=(W-w)/2:(H-h)/2 overlay.bmp

知识讲解

采集

- 例如：列出采集设备

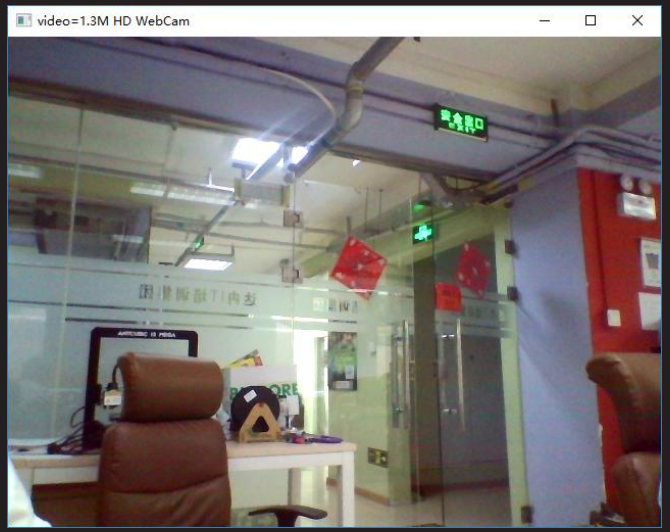
  ffmpeg -list_devices 1 -f dshow -i dummy

  "1.3M HD WebCam"

  "麦克风 (Realtek High Definition Audio)"

- 例如：回放摄像头拍摄到的影像

  ffplay -f dshow -i video="1.3M HD WebCam"



知识讲解

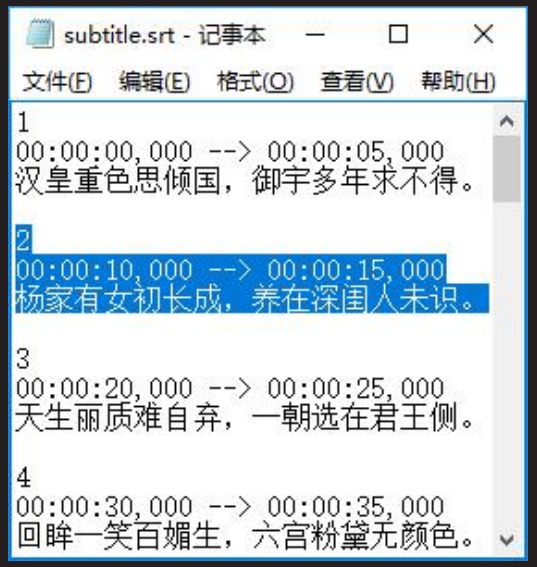
采集(续1)

- 例如：回放麦克风采集到的声音
  ffplay -f dshow -i audio="麦克风 (Realtek High Definition Audio)"
- 例如：录制音视频保存到record.mp4中
  ffmpeg -y -f dshow -i video="1.3M HD WebCam" -f dshow -i audio="麦克风 (Realtek High Definition Audio)" -s 320x240 -r 25 -b:v 800k -b:a 44.1k record.mp4

知识讲解

字幕

- … -vf subtitles=<字幕文件> …
  - 例如：为video.vob添加字幕
    set FONTCONFIG_PATH=. // 设置字体配置文件(fonts.conf)路径环境变量
    ffmpeg -i video.vob -vf subtitles=subtitle.srt subtitle.mp4

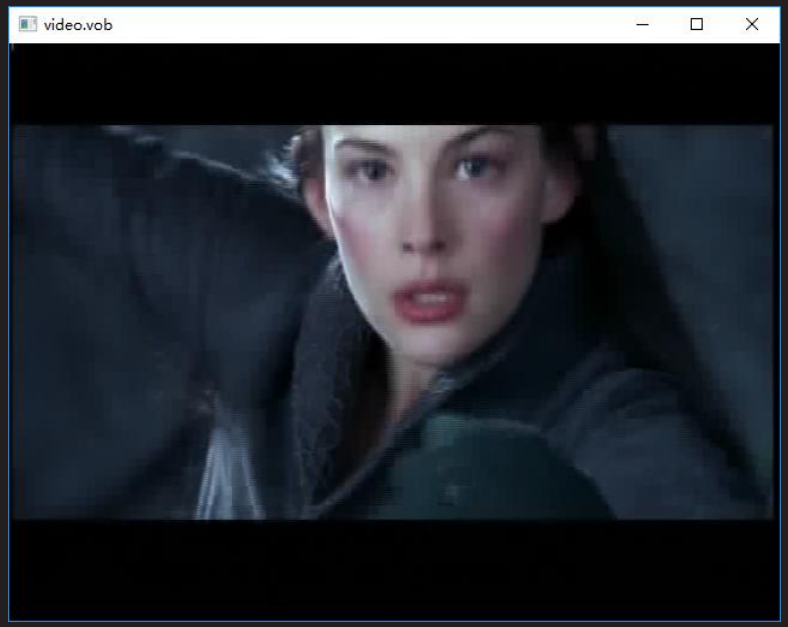


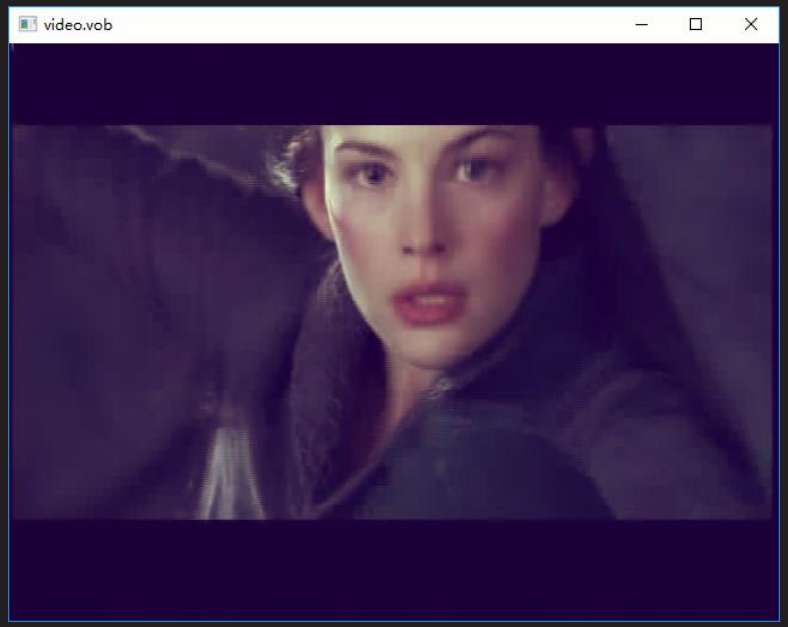
知识讲解

色彩平衡

- … -vf curves=<平衡曲线> …
  - 例如：色彩平衡
    ffplay video.vob -vf curves=vintage

知识讲解

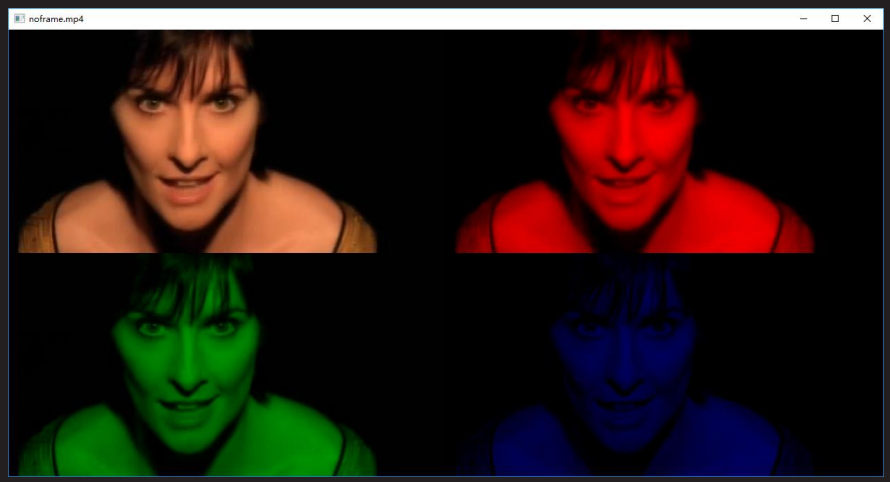




颜色空间

- ... -vf lutrgb=<RGB通道表> ...
  - `r` ： 红色
    `g` ： 绿色
    `b` ： 蓝色
  - 例如：同屏显示noframe.mp4视频及其红绿蓝通道

```
ffplay noframe.mp4 -vf split=4[a][b][c][d];
[a]pad=iw*2:ih*2[e];[b]lutrgb="g=0:b=0"[f];
[c]lutrgb="r=0:b=0"[g];
[d]lutrgb="r=0:g=0"[h];
[e][f]overlay=w[i];
[i][g]overlay=0:h[j];
[j][h]overlay=w:h
```



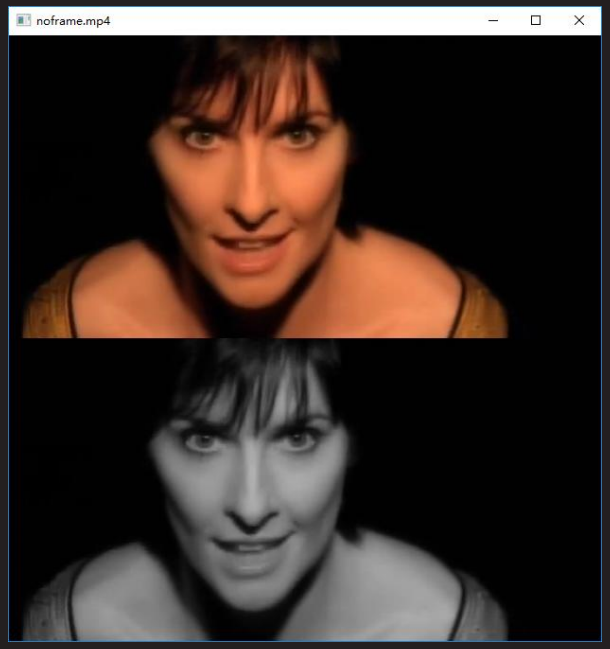
知识讲解

颜色空间(续1)

Technology Tarena 达内科技

- ... -vf lutyuv=<YUV通道表> ...
  - y ： 亮度
    u ： 色调
    v ： 饱和度
  - 例如：同屏显示noframe.mp4视频及其灰度变换

```
ffplay noframe.mp4
-vf split[a][b];
[a]pad=iw:ih*2[c];
[b]lutyuv="u=128:v=128"[d];
[c][d]overlay=0:h
```

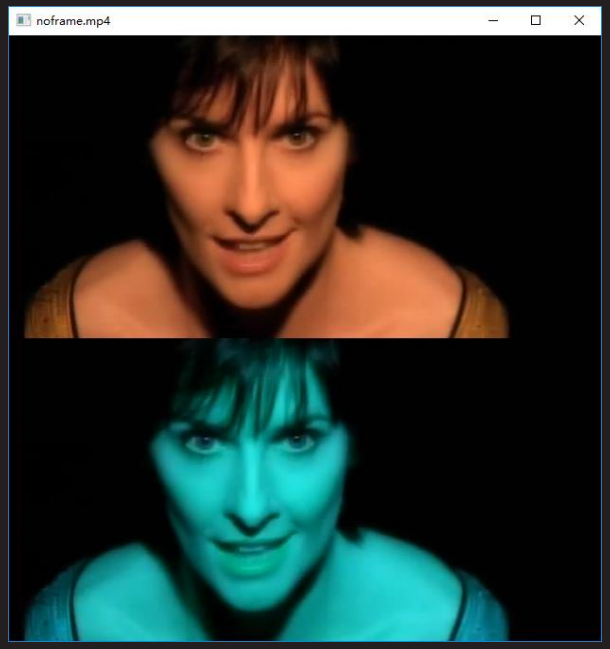


知识讲解

颜色空间(续2)

- ... -vf hue=<HSB通道表> ...
  - `h` ： 色相角(角度)
    `H` ： 色相角(弧度)
    `s` ： 饱和度
    `b` ： 亮度
  - 例如：同屏显示noframe.mp4视频及其色调变换

```
ffplay noframe.mp4
-vf split[a][b];
[a]pad=iw:ih*2[c];
[b]hue="H=2*PI*t:
s=sin(2*PI*t)+1"[d];
[c][d]overlay=0:h
```



知识讲解

速度

- ... -vf setpts=PTS/<视频倍速> ...
  - 例如：八倍速播放video.vob的视频
    ffplay video.vob -vf setpts=PTS/8
- ... -af atempo=<音频倍速> ...
  - 例如：两倍速播放video.vob的视频和音频
    ffplay video.vob -vf setpts=PTS/2 -af atempo=2

知识讲解

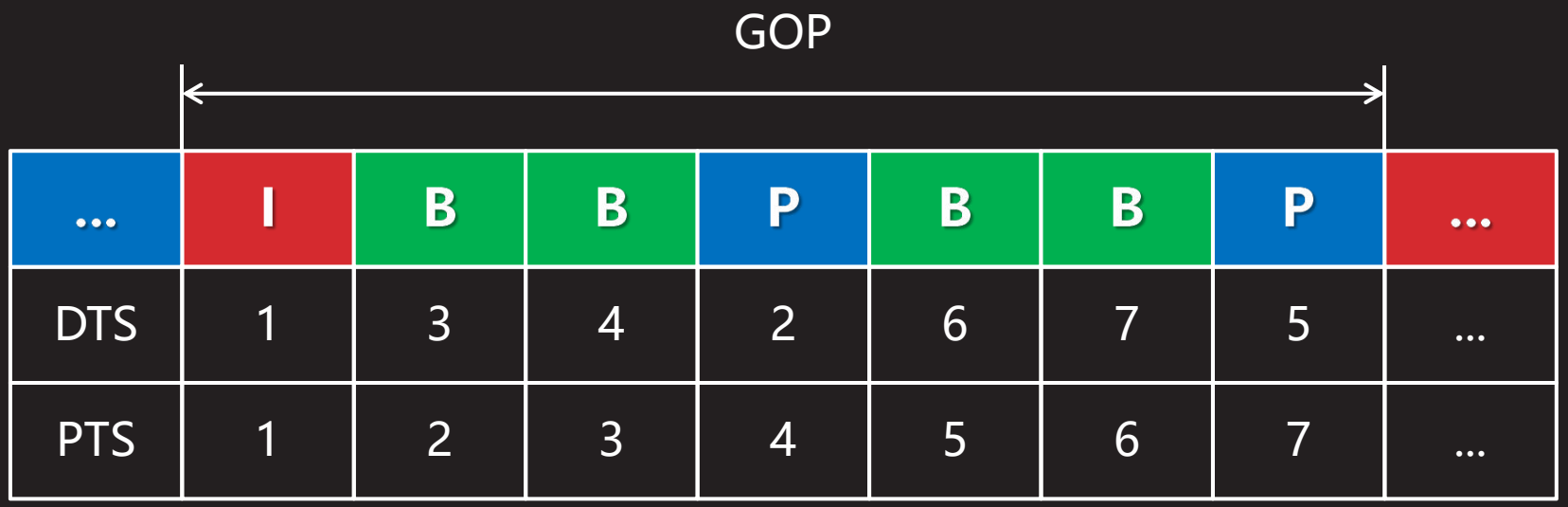
速度(续1)

- 帧、帧组和时间戳
  - I (Intra)帧：帧内编码帧，可被直接解码为一张图片
    P (Predictive)帧：预测编码帧，参考前一个I帧或B帧，还原一张图片
    B (Bi-directional interpolated prediction)帧：双向内插预测编码帧，参考前一个I帧或P帧及后一个P帧，还原一张图片
  - GOP (Group of Pictures)：帧组，从一个I帧到下一个I帧之前的P帧为一个GOP
    DTS (Decode Time Stamp)：解码时间戳，控制视频帧的解码顺序
    PTS (Presentation Time Stamp)：播放时间戳，控制视频帧的显示顺序

知识讲解

速度(续2)

- 帧、帧组和时间戳



知识讲解

Technology Tarena 达内科技

附录

FFmpeg与libav

Technology Tarena 达内科技

- libav是什么?
  - libav是一个自由软件，可以对多种音视频格式，进行录制、转码和过滤处理。它包含了libavcodec音视频编解码库、libavformat音视频格式库等多个功能组件
- 为什么会有libav?
  - 2011年1月19日，FFmpeg的现任维护者Michael Niedermayer在邮件列表上披露，FFmpeg发生了政变，一些开发者占领了官方网站，关闭了其它人的写入权限。随后政变者宣布FFmpeg已建立新政权，维护任务将由他们接手，并宣称只有维护团队才能拥有主源代码库的写入权限
  - 新内阁成员之一Diego Biurrun解释了他们的行动，称政变是迫不得已，其动机是为了维护统一，FFmpeg社区分裂的情况已经严重到令他们忍无可忍的地步，他们希望为FFmpeg项目建立一个健康友好的开发环境
  - 2011年3月13日，政变中失败的一方，同时也是FFmpeg项目的最初发起者，Fabrice Bellard等人退出FFmpeg，开启了新的项目libav，同时制定了一套关于项目继续发展和维护的规则，自此与FFmpeg分道扬镳

知识讲解

Technology Tarena 达内科技

总结和答疑