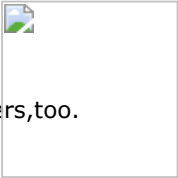


浮生若夢，為歡幾何

Always put yourself in the other's shoes.If you feel that it hurts you,it probably hurts others,too.



花生日记App扫码下载
邀请码D9CCBEO
自用省钱，分享赚钱



惠购物App扫码下载
海量天猫优惠券，省钱神器

昵称：[晏过留痕](#)
园龄：[4年6个月](#)
粉丝：[16](#)
关注：[1](#)
[+加关注](#)

<	2019年5月						>
日	一	二	三	四	五	六	
28	29	30	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	31	1	
2	3	4	5	6	7	8	

搜索

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

- [PopUpButton\(9\)](#)
- [Away3D\(6\)](#)
- [Accordion\(5\)](#)
- [ColorTransform\(5\)](#)
- [IFrame\(5\)](#)

[博客园](#) [首页](#) [新随笔](#) [联系](#) [订阅XML](#) [管理](#)

随笔-330 评论-9 文章-46

[FFmpeg] ffmpeg 常用命令

1. 视频转换

比如一个avi文件，想转为mp4，或者一个mp4想转为ts。
ffmpeg -i input.avi output.mp4
ffmpeg -i input.mp4 output.ts

2. 提取音频

ffmpeg -i test.mp4 -acodec copy -vn output.aac
上面的命令，默认mp4的audio codec是aac,如果不是，可以都转为最常见的aac。
ffmpeg -i test.mp4 -acodec aac -vn output.aac

3. 提取视频

ffmpeg -i input.mp4 -vcodec copy -an output.mp4

4. 视频剪切

下面的命令，可以从时间为00:00:15开始，截取5秒钟的视频。
ffmpeg -ss 00:00:15 -t 00:00:05 -i input.mp4 -vcodec copy -acodec copy output.mp4
-ss表示开始切割的时间，-t表示要切多少。上面就是从15秒开始，切5秒钟出来。

5. 码率控制

[Matrix\(4\)](#)
[LocalConnection\(4\)](#)
[NetConnection\(4\)](#)
[Socket\(4\)](#)
[transform.matrix3D.appendRotation\(4\)](#)
[更多](#)

随笔分类

[ActionScript 3.0\(189\)](#)
[AIR\(28\)](#)
[FFmpeg\(2\)](#)
[Flex\(53\)](#)
[HTML\(21\)](#)
[Java\(29\)](#)
[JavaScript\(33\)](#)
[Other\(2\)](#)
[Vuejs\(8\)](#)

随笔档案

[2019年4月 \(3\)](#)
[2019年2月 \(1\)](#)
[2019年1月 \(3\)](#)
[2018年12月 \(1\)](#)
[2018年9月 \(5\)](#)
[2018年8月 \(2\)](#)
[2018年7月 \(1\)](#)
[2018年6月 \(5\)](#)
[2018年5月 \(4\)](#)
[2018年4月 \(1\)](#)
[2018年3月 \(2\)](#)
[2018年1月 \(4\)](#)
[2017年12月 \(2\)](#)
[2017年11月 \(2\)](#)
[2017年9月 \(2\)](#)
[2017年8月 \(4\)](#)
[2017年7月 \(2\)](#)

码率控制对于在线视频比较重要。因为在线视频需要考虑其能提供的带宽。

那么，什么是码率？很简单：

$\text{bitrate} = \text{file size} / \text{duration}$

比如一个文件20.8M，时长1分钟，那么，码率就是：

$\text{biterate} = 20.8\text{M bit} / 60\text{s} = 20.8 * 1024 * 1024 * 8 \text{ bit} / 60\text{s} = 2831\text{Kbps}$

一般音频的码率只有固定几种，比如是128Kbps，

那么，video的就是

$\text{video biterate} = 2831\text{Kbps} - 128\text{Kbps} = 2703\text{Kbps}$ 。

那么ffmpeg如何控制码率。

ffmpg控制码率有3种选择，-minrate -b:v -maxrate

-b:v主要是控制平均码率。

比如一个视频源的码率太高了，有10Mbps，文件太大，想把文件弄小一点，但是又不破坏分辨率。

`ffmpeg -i input.mp4 -b:v 2000k output.mp4`

上面把码率从原码率转成2Mbps码率，这样其实也间接让文件变小了。目测接近一半。

不过，ffmpeg官方wiki比较建议，设置b:v时，同时加上 -bufsize

-bufsize 用于设置码率控制缓冲器的大小，设置的好处是，让整体的码率更趋近于希望的值，减少波动。（简单来说，比如1 2的平均值是1.5， 1.49 1.51 也是1.5，当然是第二种比较好）

`ffmpeg -i input.mp4 -b:v 2000k -bufsize 2000k output.mp4`

-minrate -maxrate就简单了，在线视频有时候，希望码率波动，不要超过一个阈值，可以设置maxrate。

`ffmpeg -i input.mp4 -b:v 2000k -bufsize 2000k -maxrate 2500k output.mp4`

6. 视频编码格式转换

比如一个视频的编码是MPEG4，想用H264编码，咋办？

`ffmpeg -i input.mp4 -vcodec h264 output.mp4`

相反也一样

`ffmpeg -i input.mp4 -vcodec mpeg4 output.mp4`

当然了，如果ffmpeg当时编译时，添加了外部的x265或者X264，那也可以用外部的编码器来编码。（不知道什么是X265，可以 Google一下，简单的说，就是她不包含在ffmpeg的源码里，是独立的一个开源代码，用于编码HEVC，ffmpeg编码时可以调用它。当然 了，ffmpeg自己也有编码器）

`ffmpeg -i input.mp4 -c:v libx265 output.mp4`

`ffmpeg -i input.mp4 -c:v libx264 output.mp4`

7. 只提取视频ES数据

`ffmpeg -i input.mp4 -vcodec copy -an -f m4v output.h264`

8. 过滤器的使用

8.1 将输入的1920x1080缩小到960x540输出：

[2017年6月 \(3\)](#)[2017年5月 \(8\)](#)[2017年4月 \(5\)](#)[2017年3月 \(2\)](#)[2017年2月 \(4\)](#)[2017年1月 \(4\)](#)[2016年12月 \(5\)](#)[2016年11月 \(3\)](#)[2016年10月 \(4\)](#)[2016年9月 \(5\)](#)[2016年8月 \(10\)](#)[2016年7月 \(4\)](#)[2016年6月 \(14\)](#)[2016年5月 \(6\)](#)[2016年4月 \(26\)](#)[2016年3月 \(8\)](#)[2016年2月 \(13\)](#)[2016年1月 \(10\)](#)[2015年12月 \(11\)](#)[2015年11月 \(13\)](#)[2015年10月 \(22\)](#)[2015年9月 \(18\)](#)[2015年8月 \(13\)](#)[2015年7月 \(14\)](#)[2015年6月 \(22\)](#)[2015年5月 \(8\)](#)[2015年4月 \(4\)](#)[2015年3月 \(20\)](#)[2015年2月 \(7\)](#)

文章分类

[开心一刻\(8\)](#)[美文美句\(14\)](#)[名人轶事\(5\)](#)[情感杂文\(17\)](#)

`ffmpeg -i input.mp4 -vf scale=960:540 output.mp4`

//ps: 如果540不写, 写成-1, 即scale=960:-1, 那也是可以的, ffmpeg会通知缩放滤镜在输出时保持原始的宽高比。

8.2 为视频添加logo

比如, 我有这么一个图片



想要贴到一个视频上, 那可以用如下命令:

```
./ffmpeg -i input.mp4 -i iQIYI_logo.png -filter_complex overlay output.mp4
```

结果如下所示:



要贴到其他地方? 看下面:

右上角:

```
./ffmpeg -i input.mp4 -i logo.png -filter_complex overlay=W-w output.mp4
```

左下角:

```
./ffmpeg -i input.mp4 -i logo.png -filter_complex overlay=0:H-h output.mp4
```

右下角:

```
./ffmpeg -i input.mp4 -i logo.png -filter_complex overlay=W-w:H-h output.mp4
```

8.3 去掉视频的logo

社会现象(2)

友情链接

<http://blog.project-nya.jp/1752>
<http://dead-knight.iteye.com/blog/>
<http://help.adobe.com/>
<http://shuxia.cc/tuya/create>
<http://www.flexpaper.org/>
<http://www.freeactionscrip.com/>
<http://www.html5tricks.com>
<http://www.litefeel.com>
<http://www.senocular.com>
<http://www.shouce.ren/>
http://www.w3cschool.cn/html5_reference.html
[淘宝天猫内部优惠券](#)

最新评论

1. Re:[ActionScript 3.0] AS3调用百度天气预报查询API

@凌落成迷很久以前申请的服务端类型不过后面百度系统升级有变动，下面这个是百度的公告公告：因新系统升级，自公告之日起，“服务端AK”不再支持浏览器端使用；在浏览器端使用，请选择“浏览器端AK”，例如Ja.....

--晏过留痕

2. Re:[FFmpeg] ffmpeg 常用命令 系统学习ffmpeg命令 网易云-音视频 FFMPEG命令从入门到提高

--成心以此

3. Re:[Vue] vue-cli3.0安装

@晏过留痕 已解决，在git.cmd中运行，vue -V成功了...

--无限yier

4. Re:[Vue] vue-cli3.0安装

@无限yier看提示，vue-cli3.0.1 已经安装成功了，百度搜索bash: vue:

语法：-vf delogo=x:y:w:h[:t[:show]]

x:y 离左上角的坐标

w:h logo的宽和高

t: 矩形边缘的厚度默认值4

show: 若设置为1有一个绿色的矩形，默认值0。

ffmpeg -i input.mp4 -vf delogo=0:0:220:90:100:1 output.mp4

结果如下所示：



9. 截取视频图像

ffmpeg -i input.mp4 -r 1 -q:v 2 -f image2 pic-%03d.jpeg

-r 表示每秒几帧

-q:v表示存储jpeg的图像质量，一般2是高质量。

如此，ffmpeg会把input.mp4，每隔一秒，存一张图片下来。假设有60s，那会有60张。

可以设置开始的时间，和你想要截取的时间。

ffmpeg -i input.mp4 -ss 00:00:20 -t 10 -r 1 -q:v 2 -f image2 pic-%03d.jpeg

-ss 表示开始时间

-t 表示共要多少时间。

如此，ffmpeg会从input.mp4的第20s时间开始，往下10s，即20~30s这10秒钟之间，每隔1s就抓一帧，总共会抓10帧。

10. 序列帧与视频的相互转换

command not found可以找到解决办法...

--晏过留痕

5. [Re:\[Vue\] vue-cli3.0安装](#)

在吗? 安装后出现ntyin@ntyin-PC

MINGW64 ~\$ npm i -g

@vue/cliD:\nodejs\nodejs\node_glo
ballvue -> D:\nodejs\nod.....

--无限yier

阅读排行榜

1. [\[FFmpeg\] ffmpeg 常用命令 \(41789\)](#)

2. [\[Flex\] IFrame系列 —— 在flex的web应用中嵌入html的方法\(3808\)](#)

3. [\[ActionScript 3.0\] AS3.0 让一个视频无缝循环播放的一个偏方\(3655\)](#)

4. [\[JavaScript\] iframe加载完成事件 \(2822\)](#)

5. [\[Flex\] IFrame系列 —— IFrame嵌入html后Alert弹出窗口被IFrame遮挡问题\(2780\)](#)

评论排行榜

1. [\[Vue\] vue-cli3.0安装\(3\)](#)

2. [\[ActionScript 3.0\] 如何控制加载swf动画的播放与暂停\(2\)](#)

3. [\[ActionScript 3.0\] AS3调用百度天气预报查询API\(2\)](#)

4. [\[ActionScript 3.0\] AS3调用百度地图API\(1\)](#)

5. [\[FFmpeg\] ffmpeg 常用命令\(1\)](#)

推荐排行榜

1. [\[FFmpeg\] ffmpeg 常用命令\(6\)](#)

2. [\[ActionScript3.0\] 运用JPEGEncoderOptions或者PNGEncoderOptions保存图片到本地\(1\)](#)

把darkdoor.[001-100].jpg序列帧和001.mp3音频文件利用mpeg4编码方式合成视频文件darkdoor.avi:

```
$ ffmpeg -i 001.mp3 -i darkdoor.%3d.jpg -s 1024x768 -author fy -vcodec mpeg4 darkdoor.avi
```

还可以把视频文件导出成jpg序列帧:

```
$ ffmpeg -i bc-cinematic-en.avi example.%d.jpg
```

其他用法

1. 输出YUV420原始数据

对于一下做底层编解码的人来说, 有时候常要提取视频的YUV原始数据, 如下:

```
ffmpeg -i input.mp4 output.yuv
```

那如果我只想要抽取某一帧YUV呢?

你先用上面的方法, 先抽出jpeg图片, 然后把jpeg转为YUV。

比如:

你先抽取10帧图片。

```
ffmpeg -i input.mp4 -ss 00:00:20 -t 10 -r 1 -q:v 2 -f image2 pic-%03d.jpeg
```

—

然后, 你就随便挑一张, 转为YUV:

```
ffmpeg -i pic-001.jpeg -s 1440x1440 -pix_fmt yuv420p xxx3.yuv
```

如果-s参数不写, 则输出大小与输入一样。

当然了, YUV还有yuv422p啥的, 你在-pix_fmt 换成yuv422p就行啦!

2. H264编码profile & level控制

背景知识

先科普一下profile&level。(这里讨论最常用的H264)

H.264有四种画质级别, 分别是baseline, extended, main, high:

- 1、Baseline Profile: 基本画质。支持I/P 帧, 只支持无交错 (Progressive) 和CAVLC;
- 2、Extended profile: 进阶画质。支持I/P/B/SP/SI 帧, 只支持无交错 (Progressive) 和CAVLC; (用的少)
- 3、Main profile: 主流画质。提供I/P/B 帧, 支持无交错 (Progressive) 和交错 (Interlaced), 也支持CAVLC 和CABAC 的支持;
- 4、High profile: 高级画质。在main Profile 的基础上增加了8x8内部预测、自定义量化、无损视频编码和更多的YUV 格式;

H.264 Baseline profile、Extended profile和Main profile都是针对8位样本数据、4:2:0格式(YUV)的视频序列。在相同配置情况下, High profile (HP) 可以比Main profile (MP) 降低10%的码率。

根据应用领域的不同, Baseline profile多应用于实时通信领域, Main profile多应用于流媒体领域, High profile则多应用于广电和存储领域。

下图清楚的给出不同的profile&level的性能区别。

profile



	Baseline	Extended	Main	High	High 10	High 4:2:2	High 4:4:4 Predictive
I and P Slices	Yes	Yes	Yes	Yes	Yes	Yes	Yes
B Slices	No	Yes	Yes	Yes	Yes	Yes	Yes
SI and SP Slices	No	Yes	No	No	No	No	No
Multiple Reference Frames	Yes	Yes	Yes	Yes	Yes	Yes	Yes
In-Loop Deblocking Filter	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CAVLC Entropy Coding	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CABAC Entropy Coding	No	No	Yes	Yes	Yes	Yes	Yes
Flexible Macroblock Ordering (FMO)	Yes	Yes	No	No	No	No	No
Arbitrary Slice Ordering (ASO)	Yes	Yes	No	No	No	No	No
Redundant Slices (RS)	Yes	Yes	No	No	No	No	No
Data Partitioning	No	Yes	No	No	No	No	No
Interlaced Coding (PicAFF, MBAFF)	No	Yes	Yes	Yes	Yes	Yes	Yes
4:2:0 Chroma Format	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Monochrome Video Format (4:0:0)	No	No	No	Yes	Yes	Yes	Yes
4:2:2 Chroma Format	No	No	No	No	No	Yes	Yes
4:4:4 Chroma Format	No	No	No	No	No	No	Yes
8 Bit Sample Depth	Yes	Yes	Yes	Yes	Yes	Yes	Yes
9 and 10 Bit Sample Depth	No	No	No	No	Yes	Yes	Yes
11 to 14 Bit Sample Depth	No	No	No	No	No	No	Yes
8x8 vs. 4x4 Transform Adaptivity	No	No	No	Yes	Yes	Yes	Yes
Quantization Scaling Matrices	No	No	No	Yes	Yes	Yes	Yes
Separate Cb and Cr QP control	No	No	No	Yes	Yes	Yes	Yes

Separate Color Plane Coding	No	No	No	No	No	No	Yes
Predictive Lossless Coding	No	No	No	No	No	No	Yes
	Baseline	Extended	Main	High	High 10	High 4:2:2	High 4:4:4 Predictive

level

Level number	Max macroblocks per second	Max frame size (macroblocks)	Max video bit rate (VCL) for Baseline, Extended and Main Profiles	Max video bit rate (VCL) for High Profile	Max video bit rate (VCL) for High 10 Profile	Max video bit rate (VCL) for High 4:2:2 and High 4:4:4 Predictive Profiles	Examples for high resolution @ frame rate (max stored frames) in Level
1	1485	99	64 kbit/s	80 kbit/s	192 kbit/s	256 kbit/s	128x96@30.9 (8) 176x144@15.0 (4)
1b	1485	99	128 kbit/s	160 kbit/s	384 kbit/s	512 kbit/s	128x96@30.9 (8) 176x144@15.0 (4)
1.1	3000	396	192 kbit/s	240 kbit/s	576 kbit/s	768 kbit/s	176x144@30.3 (9) 320x240@10.0 (3) 352x288@7.5 (2)
1.2	6000	396	384 kbit/s	480 kbit/s	1152 kbit/s	1536 kbit/s	320x240@20.0 (7) 352x288@15.2 (6)
1.3	11880	396	768 kbit/s	960 kbit/s	2304 kbit/s	3072 kbit/s	320x240@36.0 (7) 352x288@30.0 (6)
2	11880	396	2 Mbit/s	2.5 Mbit/s	6 Mbit/s	8 Mbit/s	320x240@36.0 (7) 352x288@30.0 (6)
2.1	19800	792	4 Mbit/s	5 Mbit/s	12 Mbit/s	16 Mbit/s	352x480@30.0 (7) 352x576@25.0 (6)
2.2	20250	1620	4 Mbit/s	5 Mbit/s	12 Mbit/s	16 Mbit/s	352x480@30.7(10) 352x576@25.6 (7) 720x480@15.0 (6) 720x576@12.5 (5)
3	40500	1620	10 Mbit/s	12.5 Mbit/s	30 Mbit/s	40 Mbit/s	352x480@61.4 (12) 352x576@51.1 (10) 720x480@30.0 (6) 720x576@25.0 (5)

3.1	108000	3600	14 Mbit/s	17.5 Mbit/s	42 Mbit/s	56 Mbit/s	720x480@80.0 (13) 720x576@66.7 (11) 1280x720@30.0 (5)
3.2	216000	5120	20 Mbit/s	25 Mbit/s	60 Mbit/s	80 Mbit/s	1280x720@60.0 (5) 1280x1024@42.2 (4)
4	245760	8192	20 Mbit/s	25 Mbit/s	60 Mbit/s	80 Mbit/s	1280x720@68.3 (9) 1920x1088@30.1 (4) 2048x1024@30.0 (4)
4.1	245760	8192	50 Mbit/s	50 Mbit/s	150 Mbit/s	200 Mbit/s	1280x720@68.3 (9) 1920x1088@30.1 (4) 2048x1024@30.0 (4)
4.2	522240	8704	50 Mbit/s	50 Mbit/s	150 Mbit/s	200 Mbit/s	1920x1088@64.0 (4) 2048x1088@60.0 (4)
5	589824	22080	135 Mbit/s	168.75 Mbit/s	405 Mbit/s	540 Mbit/s	1920x1088@72.3 (13) 2048x1024@72.0 (13) 2048x1088@67.8 (12) 2560x1920@30.7 (5) 3680x1536/26.7 (5)
5.1	983040	36864	240 Mbit/s	300 Mbit/s	720 Mbit/s	960 Mbit/s	1920x1088@120.5 (16) 4096x2048@30.0 (5) 4096x2304@26.7 (5)
Level number	Max macroblocks per second	Max frame size (macroblocks)	Max video bit rate (VCL) for Baseline, Extended and Main Profiles	Max video bit rate (VCL) for High Profile	Max video bit rate (VCL) for High 10 Profile	Max video bit rate (VCL) for High 4:2:2 and High 4:4:4 Predictive Profiles	Examples for high resolution @ frame rate (max stored frames) in Level

2.1 ffmpeg如何控制profile&level

举3个例子吧

```
ffmpeg -i input.mp4 -profile:v baseline -level 3.0 output.mp4
```

```
ffmpeg -i input.mp4 -profile:v main -level 4.2 output.mp4
```

```
ffmpeg -i input.mp4 -profile:v high -level 5.1 output.mp4
```

如果ffmpeg编译时加了external的libx264，那就这么写：

```
ffmpeg -i input.mp4 -c:v libx264 -x264-params "profile=high:level=3.0" output.mp4
```


从压缩比例来说, baseline < main < high, 对于带宽比较局限的在线视频, 可能会选择high, 但有些时候, 做个小视频, 希望所有的设备基本都能解码 (有些低端设备或早期的设备只能解码 baseline), 那就牺牲文件大小吧, 用baseline。自己取舍吧!

苹果的设备对不同profile的支持。

iOS Compatability (source)			
Profile	Level	Devices	Options
Baseline	3.0	All devices	<code>-profile:v baseline -level 3.0</code>
Baseline	3.1	iPhone 3G and later, iPod touch 2nd generation and later	<code>-profile:v baseline -level 3.1</code>
Main	3.1	iPad (all versions), Apple TV 2 and later, iPhone 4 and later	<code>-profile:v main -level 3.1</code>
Main	4.0	Apple TV 3 and later, iPad 2 and later, iPhone 4s and later	<code>-profile:v main -level 4.0</code>
High	4.0	Apple TV 3 and later, iPad 2 and later, iPhone 4s and later	<code>-profile:v high -level 4.0</code>
High	4.1	iPad 2 and later, iPhone 4s and later, iPhone 5c and later	<code>-profile:v high -level 4.1</code>
High	4.2	iPad Air and later, iPhone 5s and later	<code>-profile:v high -level 4.2</code>

2.2. 编码效率和视频质量的取舍(preset, crf)

除了上面提到的, 强行配置biterate, 或者强行配置profile/level, 还有2个参数可以控制编码效率。

一个是preset, 一个是crf。

preset也挺粗暴, 基本原则就是, 如果你觉得编码太快或太慢了, 想改改, 可以用profile。

preset有如下参数可用:

ultrafast, superfast, veryfast, faster, fast, medium, slow, slower, veryslow and placebo.
编码加快, 意味着信息丢失越严重, 输出图像质量越差。

CRF(Constant Rate Factor): 范围 0-51: 0是编码毫无丢失信息, 23 is 默认, 51 是最差的情况。相对合理的区间是18-28。
值越大, 压缩效率越高, 但也意味着信息丢失越严重, 输出图像质量越差。

举个例子吧。

```
ffmpeg -i input -c:v libx264 -profile:v main -preset:v fast -level 3.1 -x264opts crf=18
```

(参考自: <https://trac.ffmpeg.org/wiki/Encode/H.264>)

2.3. H265 (HEVC)编码tile&level控制

背景知识

和H264的profile&level一样, 为了应对不同应用的需求, HEVC制定了“层级”(tier) 和“等级”(level)。
tier只有main和high。

level有13级，如下所示：

Tiers and levels with maximum property values ^{[2][3]}																	
Level	Max luma sample rate (samples/s)	Max luma picture size (samples)	Max bit rate for Main and Main 10 profiles (kbit/s) ^[4]		Max bit rate for Main 12 profile (kbit/s)		Max bit rate for Main 4:4:4 12 profile (kbit/s)		Max bit rate for Main 4:4:4 16 Intra profile (kbit/s)		Max bit rate for High Throughput 4:4:4 16 Intra profile (kbit/s)		Example picture resolution @ highest frame rate ^[5] (MaxDpbSize ^[6]) Click here to hide/show additional examples	MinCR ^[7]	Max number of slice segments per picture ^[8]	Max number of tile ^[9]	
			Main tier	High tier	Main tier	High tier	Main tier	High tier	Main tier	High tier	Main tier	High tier				rows	columns
1	552,960	36,864	128	-	192	-	384	-	1,024	-	12,288	-	128×96@33.7 (6) 176×144@15.0 (6)	2	16	1	1
2	3,686,400	122,880	1,500	-	2,250	-	4,500	-	12,000	-	144,000	-	176×144@100.0 (16) 320×240@45.0 (6) 352×240@37.5 (6) 352×288@30.0 (6)	2	16	1	1
2.1	7,372,800	245,760	3,000	-	4,500	-	9,000	-	24,000	-	288,000	-	320×240@90.0 (12) 352×240@75.0 (12) 352×288@60.0 (12) 352×480@37.5 (6) 352×576@33.3 (6) 640×360@30.0 (6)	2	20	1	1
3	16,588,800	552,960	6,000	-	9,000	-	18,000	-	48,000	-	576,000	-	352×480@84.3 (12) 352×576@75.0 (12) 640×360@57.5 (12) 720×480@42.1 (6) 720×576@37.5 (6) 960×540@30.0 (6)	2	30	2	2
3.1	33,177,600	983,040	10,000	-	15,000	-	30,000	-	80,000	-	960,000	-	720×480@84.3 (12) 720×576@75.0 (12) 960×540@60.0 (6) 1280×720@33.7 (6)	2	40	3	3
4	66,846,720	2,228,224	12,000	30,000	18,000	45,000	36,000	90,000	96,000	240,000	1,152,000	2,880,000	1,280×720@68.0 (12) 1,920×1,080@32.0 (6) 2,048×1,080@30.0 (6)	4	75	5	5
4.1	133,693,440		20,000	50,000	30,000	75,000	60,000	150,000	160,000	400,000	1,920,000	4,800,000	1,280×720@136.0 (12) 1,920×1,080@64.0 (6) 2,048×1,080@60.0 (6)	4			
5	267,388,800	8,612,896	25,000	100,000	37,500	150,000	75,000	300,000	200,000	800,000	2,400,000	9,600,000	1,920×1,080@128.0 (16) 2,048×1,080@120.0 (16) 3,840×2,160@32.0 (6) 4,096×2,160@30.0 (6)	6	200	11	10
5.1	534,773,760		40,000	160,000	60,000	240,000	120,000	480,000	320,000	1,280,000	3,840,000	15,360,000	1,920×1,080@256.0 (16) 2,048×1,080@240.0 (16) 3,840×2,160@64.0 (6) 4,096×2,160@60.0 (6)	8			
5.2	1,069,547,520		60,000	240,000	90,000	360,000	180,000	720,000	480,000	1,920,000	5,760,000	23,040,000	1,920×1,080@300.0 (16) 2,048×1,080@300.0 (16) 3,840×2,160@128.0 (6) 4,096×2,160@120.0 (6)	8			
6	1,069,547,520	35,651,584	60,000	240,000	90,000	360,000	180,000	720,000	480,000	1,920,000	5,760,000	23,040,000	3,840×2,160@128.0 (16) 4,096×2,160@120.0 (16) 7,680×4,320@32.0 (6) 8,192×4,320@30.0 (6)	8	600	22	20
6.1	2,139,095,040		120,000	480,000	180,000	720,000	360,000	1,440,000	960,000	3,840,000	11,520,000	46,080,000	3,840×2,160@256.0 (16) 4,096×2,160@240.0 (16) 7,680×4,320@64.0 (6) 8,192×4,320@60.0 (6)	8			
6.2	4,278,190,080		240,000	800,000	360,000	1,200,000	720,000	2,400,000	1,920,000	6,400,000	23,040,000	76,800,000	3,840×2,160@300.0 (16) 4,096×2,160@300.0 (16) 7,680×4,320@128.0 (6) 8,192×4,320@120.0 (6)	8			

^[4] The maximum bit rate of the profile is based on the combination of bit depth, chroma sampling, and the type of profile.^[2] For bit depth the maximum bit rate increases by 1.5x for 12-bit profiles and 2x for 16-bit profiles.^[3] For chroma sampling the maximum bit rate increases by 1.5x for 4:2:2 profiles and 2x for 4:4:4 profiles.^[3] For the Intra profiles the maximum bit rate increases by 2x.^[2]

^[5] The maximum frame rate supported by HEVC is 300 frames per second (fps).^[2]

不多说，直接给出怎么用。（supposed你用libx265编码）
ffmpeg -i input.mp4 -c:v libx265 -x265-params "profile=high:level=3.0" output.mp4

----- Always put yourself in the other's shoes.If you feel that it hurts you,it probably hurts others,too.-----

分类: FFmpeg
标签: ffmpeg

好文要顶

关注我

收藏该文



晏过留痕
关注 - 1
粉丝 - 16

+加关注

« 上一篇: [FFmpeg] ffmpeg参数详解

» 下一篇: [HTML] IE=edge,chrome=1的META标签详解

posted on 2016-09-07 11:25 晏过留痕 阅读(41792) 评论(1) 编辑 收藏

评论:

#1楼 2019-02-27 21:04 | 成心以此

系统学习ffmpeg命令 网易云-音视频FFMPEG命令从入门到提高

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问](#)网站首页。

【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码

【活动】看雪2019安全开发者峰会, 共话安全领域焦点

相关博文:

- [ffmpeg命令学习](#)
- [FFmpeg常用命令 \(一\)](#)
- [ffmpeg常用命令](#)
- [H264编码profile & level控制](#)
- [FFmpeg介绍及参数详细说明](#)

最新新闻:

- [完全免费! GitHub发布软件包管理服务: NPM瑟瑟发抖](#)
 - [《复联 4》热潮过后, 电影院可能会付出更沉重的代价](#)
 - [百度沈抖: 智能小程序月活破2亿, 资源已全力倾斜](#)
 - [文艺科学: 显微“艺术照”, 黑白怀旧风](#)
 - [HTC宣布关闭线上手机旗舰店 但手机及其配件仍在售](#)
- » [更多新闻...](#)

历史上的今天:

2015-09-07 [\[ActionScript 3.0\] AS3 绘制正八面体 \(线条\)](#)

2015-09-07 [\[ActionScript 3.0\] AS3 绘制正二十面体 \(线条\)](#)

