从x264教程综合版总结出的参数配置. 目的是把参数直接贴到软件里用. 要求x265 v3.5+69或v3.6

|  |  |
| --- | --- |
| [**LigH**](http://www.mediafire.com/?6lfp2jlygogwa) | .hevc GCC10 [单文件8-10-12bit] 附x86, Windows XP x86版 附libx265.dll |
| [**Rigaya**](https://drive.google.com/drive/u/0/folders/0BzA4dIFteM2dWEpvWGZXV3ZhdTA) | .hevc GCC 9.3 [8-10-12bit] 附x86版 |
| [**Patman**](http://www.mediafire.com/folder/arv5xmdqyiczc) | .hevc GCC 11+MSVC1925 [8-10-12bit] |
| [**ShortKatz**](https://forum.doom9.org/showthread.php?p=1937773#post1937773) | arm64~64e加x86版 [?] 需macOS运行编译命令文件 ? |
| **[DJATOM-aMod](https://github.com/DJATOM/x265-aMod/releases/)** | opt-Intel架构与zen1~2优化 [10bit], opt-znver3代表zen3优化 [10-12bit] GCC 10.2.1+GCC10.3 |
| **[MeteorRain-yuuki](https://down.7086.in/)** | lsmash.mkv/mp4或.hevc [能封装, 但传说lavf不如pipe可靠] GCC 9.3+ICC 1900+MSVC 1916 [8][10][12bit]+[8-10-12bit] |
| [**ffmpeg**](http://ffmpeg.org/download.html) 多系统兼容, 备用地址 ottverse.com/ffmpeg-builds | |
| [**mpv播放器**](https://mpv.io/installation/)比Potplayer好在没有音频滤镜, 不用手动关; 没有颜色偏差, 文件体积小 | |
| [**x265GuiEx**](https://drive.google.com/drive/folders/0BzA4dIFteM2dRkRzWXZMT0lkM2M) **(Rigaya)** 日本語, auto-setup安装, [教程点此](https://aviutl.info/x265guiex/#toc4) | |
| [**Voukoder**; **V-Connector**](https://www.voukoder.org/)免费Premiere/Vegas/AE插件, 直接用ffmpeg内置编码器, 不用帧服务器/导无损再压/找破解. 只要下两个压缩包, 放Plug-Ins\Common文件夹就行了 | |

### x265.exe命令行用法教程

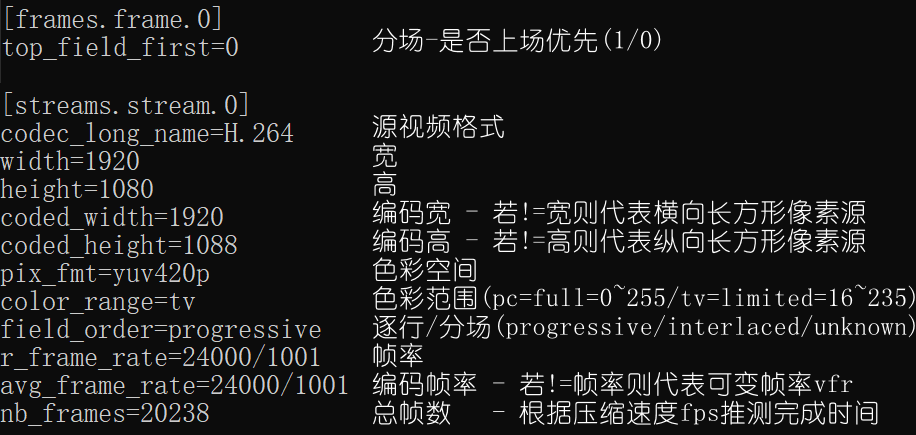
[照上表下载ffmpeg, ffprobe和x265并记住路径, 此处置于D盘根目录下]



[打开Windows的CMD/PowerShell或Linux/MacOS的Bash/Terminal, 分别输入ffmpeg, ffprobe, x265的路径并回车, 确认程序存在]

[查ffmpeg版本信息] C:\文件夹\ffmpeg.exe; [查x265版本信息] C:\文件夹\x265.exe -V 

[CMD路径自动填充] 路径名写一半, 然后按[Tab]直到文件名匹配为止

[用ffprobe获取视频编码格式名, 宽, 编码宽, 高, 编码高, 色彩空间格式, 色彩空间范围, 逐行/分行, 帧率, 平均帧率, 总帧数] ffprobe.exe -i ".\视频.mp4" -select\_streams v:0 -v error -hide\_banner -show\_streams -show\_frames -read\_intervals "%+#1" -show\_entries frame=top\_field\_first:stream=codec\_long\_name,width,coded\_width,height,coded\_height,pix\_fmt,color\_range,field\_order,r\_frame\_rate,avg\_frame\_rate,nb\_frames -of ini

[遇到分场源] 据top\_field\_first判断<上/下场优先>, 添加x265参数--interlaced<tff/bff>

[源视频为可变帧率] 手机省电用. 因兼容性问题应添加ffmpeg参数-vsync cfr转换为恒定帧率cfr

[长方形像素] 日本电视台缩宽, 旧版优酷缩高, 现今被抛弃的压缩手段. 能换源则尽可能换

[压制用时] 总帧数÷压缩速度fps=时间(秒)

[x265所需信息] ffmpeg输入-pix\_fmt<源视频的色彩空间>, 类似图中结果

[参数用例] D:\ffmpeg.exe -i .\视频.mov -an -pix\_fmt yuv420p10 -f yuv4mpegpipe -strict unofficial - | D:\x265-10bit.exe --preset slow --me umh --subme 5 --merange 48 --weightb --aq-mode 4 --bframes 5 --ref 3 --hash 2 --allow-non-conformance --qg-size 16 --rd 5 --limit-modes --limit-refs 1 --rskip 1 --splitrd-skip --no-sao --tskip --colorprim bt2020 --colormatrix bt2020nc --transfer smpte2084 --y4m - --output F:\导出.hevc 2>D:\桌面\ffmpeg或x265报错.txt

### ffmpeg, VS, avs2yuv pipe

ffmpeg -i [源] -an -f yuv4mpegpipe -strict unofficial - | x265 --y4m - --output

ffmpeg -i [源] -an -f rawvideo - | x265.exe --input-res [分辨率] --fps [] - --output

-i导入, -f格式, -an关音频编码, -strict unofficial允许自定格式, --y4m指"YUV for MPEG"未压缩格式以便pipe传输, "ffmpeg - | x265 -"之间的"-"是pipe格式

VSpipe.exe [脚本].vpy --y4m - | x265.exe - --y4m --output

VSpipe/avs2yuv [脚本].vpy - | x265.exe --input-res [分辨率] --fps [] - --output

avs2yuv.exe [脚本].avs -raw - | x265.exe --input-res [分辨率] --fps [] - --output

**ffmpeg .ass字幕渲染滤镜****:** -filter\_complex "ass='F\:/字幕.ass'"

中途停止压制, 并封装现有帧为视频**:** Ctrl+C, 部分人编译的x265.exe自带功能

**qaac 压制音频见**[**教程**](https://www.nazorip.site/archives/44/)**或**[**Github**](https://github.com/iAvoe/QAAC-Tutorial-Standalone/blob/master/%E6%95%99%E7%A8%8B.md)

**ffmpeg内置缩放:**

例: -sws\_flags bitexact+full\_chroma\_int+full\_chroma\_inp+accurate\_rnd

-sws\_flags bicubic/bitexact/gauss/bicublin/lanczos/spline/+full\_chroma\_int/+full\_chroma\_inp/+accurate\_rnd

**ffmpeg 封装视音频, 更改导出文件后缀名以指定封装格式(.mkv格式还支持封装字幕, 字体)**

* ffmpeg.exe -i ".\视频流入.hevc" -an -c:v copy -i ".\音频流入.aac" -c:a copy -i ".\传统字幕.srt" -c:s copy "封装出.mp4"
* ffmpeg -i ".\视频.hevc" -an -c:v copy -i ".\音轨1.aac" -c:a copy -i ".\音轨2.aac" -c:a copy -i ".\字幕1.ass" -c:s copy -i ".\字幕2.ass" -c:s copy -i ".\字体1.ttf" -c:t copy "封装出.mkv"

**不同封装格式的字幕格式支持:** [Wikipedia - Subtitle formats support](https://en.wikipedia.org/wiki/Comparison_of_video_container_formats)

**ffmpeg 替换封装中的音频流, itoffset±秒数以对齐:**

* ffmpeg.exe -i ".\封装入.mov" -i ".\新音频流入.aac" -c:v copy -map 0:v:0 -map 1:a:0 -c:a copy -itsoffset 0 ".\新封装出.mov"

**ffmpeg: small thread\_queue\_size 警告:**

-thread\_queue\_size<(源平均码率kbps+1000)/可调用CPU核心数>

**批处理: 完成后转换为普通命令窗(不退出):**

cmd /k

**+显示Windows版本:**

cmd -k

**ffmpeg 从NTSC 3:2 pulldown 60fps源恢复电影标准24fps:**

* ffmpeg.exe -i ".\60fps\_interlaced\_NTSC\_source.vob" -map 0:v:0 -vf "fieldmatch=order=auto:mode=pc\_n\_ub:field=auto:cthresh=8:combmatch=full:blockx=16:blocky=24:combpel=128,nnedi=weights=C:\[点此下载\nnedi3\_weights.bin](https://github.com/dubhater/vapoursynth-nnedi3/blob/master/src/nnedi3_weights.bin):field=af:nsize=s48x6:nns=n128:qual=slow:etype=mse:pscrn=new3" -fps 24

### x265

### HDR标识

**色域标识**

**色域转换**

--master-display <手动告知播放器拿什么色彩空间解码

DCI-P3: G(13250,34500)B(7500,3000)R(34000,16000)WP(15635,16450)L(maxCLL×10000,1)

bt709: G(15000,30000)B(7500,3000)R(32000,16500)WP(15635,16450)L(maxCLL×10000,1)

bt2020: G(8500,39850)B(6550,2300)R(35400,14600)WP(15635,16450)L(maxCLL×10000,1)

* 找到HDR元数据中的色彩范围，确认用以下哪个色彩空间后填上参数
* L的值没有标准，每个HDR视频元数据里可能都不一样

DCI-P3: G(x0.265, y0.690), B(x0.150, y0.060), R(x0.680, y0.320), WP(x0.3127, y0.329)

bt709: G(x0.30, y0.60), B(x0.150, y0.060), R(x0.640, y0.330), WP(x0.3127,y0.329)

bt2020: G(x0.170, y0.797), B(x0.131, y0.046), R(x0.708, y0.292), WP(x0.3127,y0.329)>

--max-cll <maxCLL,maxFALL>最大,平均光强度, MediaInfo查不出来就不用填

--colormatrix <照源, 例: gbr bt709 fcc bt470bg smpte170m YCgCo bt2020nc bt2020c smpte2084 ictcp>

--transfer <照源, 例: gbr bt709 fcc bt470bg smpte170m YCgCo bt2020nc bt2020c smpte2084 ictcp>

**杜比视界dolby vision/DV**

有DV-MEL (BL+RPU)和DV-FEL (BL+EL+RPU)两种带RPU的格式, x265支持共3种样式/profile的DV-MEL

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 样式 | 编码 | BL:EL分辨率 | x265支持 | 伽马 | 色彩空间 |
| 4 | 10bit hevc | 1:1/4 |  | SDR | YCbCr |
| 5 | 仅BL (DV-MEL) | √ |  | ICtCp |
| 7 | 4K=1:1/4; 1920x1080=1:1 |  | UHD蓝光 | YCbCr |
| 8.1 | 仅BL (DV-MEL) | √ | HDR10 |
| 8.2 | √ | SDR |
| 8.4 |  | HLG |
| 9 | 8bit avc | 仅BL (DV-MEL) |  | SDR | YCbCr |

--dolby-vision-profile

<选择5/8.1 (HDR10)/8.2 (SDR)>8.1需要写master-display和hdr10-opt

--dolby-vision-rpu

<路径>导入rpu二进制文件(.bin)用

# 速度参考

**处理器:** R7 5800X全8核负电压偏移超频4.5Ghz，FS140下烤机67°C稳，CbR23均15440 (PBO2负30偏移超频4.85Ghz，86°C仅提升2%算力故不用)

**内存:** 海力士MFR 2×2R×8GB/2x16GB，3000Mhz 15-17-17-35 1T 1.44V，F-U-MCLK等比同步

**源①:** 1920x1080 yuv420p8 24000/1001fps 312MB低清h.264录像源，高对比材质纹理，20238帧. **方:** 10bit crf 28以增加像素值偏移错误差距，用低清源来去掉画质差距，高对比细节动态增加计算量

**preset slow:** 16分27，平均~20.5fps，压缩后217MB，画质损失可见（高清源下更明显）

**通用·简单:** 24分48，平均~13.6fps，比slow慢1.5x，得159MB，损失可见

**高压·动漫:** 36分36，平均~9.21fps，比slow慢2.2x，得145MB，由于片源不匹配所以损失可见

**高压·录像:** 78分57，平均~4.27fps，比slow慢4.8x，得189MB，损失小但受源限制，要仔细看

**veryslow:** 133分16，平均~2.53fps，比slow慢8.1x，得221MB，损失小

**源②:** 3840x2160 yuv444p12 24fps 37GB PQ ProRes4444XQ, 6314帧. **法:** 检测极高成本动漫(时间复杂度高, 空间复杂度精确到1像素), 4:4:4上的压制速度, 以及稳定性

**通简-CRF16:** 75分21，1.4fps 1159MB，**CRF18:** 73分19，1.44fps 902MB，**CRF20:** 69分58，1.5fps 698MB，**CRF22:** 中途失败，剩一半但可后期拼接. 注: 以上开hist-scenecut的测试中途都失败了

**源③:** 1920x1080 yuv420p10 24000/1001fps 1.9GB h.264, 34095帧. **方:** 检测低成本动漫上, 4:2:0的压制速度. (空间-时间复杂度简单的源)

**高压·动漫:** 46分43，平均~12fps

**源④:** 1920x1080 yuv444p10 24000/1001fps 2.9GB h.264, 40920帧. **法:** 检测高成本动漫, 4:4:4上的压制速度. (空间-时间复杂度较为困难的源)

**高压·动漫:** 97分30，平均~6.99fps

# 通用·简单·低清

自定义项目all off，方便急用且速度仅比preset slow慢几fps

**预设-转场**

**动态搜索**

**自适应量化**

**帧控**

**输入输出**

**目标色深**

**多处理器分配**

**其它**

**目标色彩空间**

--preset slow --hist-scenecut

--me umh --subme 5 --merange 48 --weightb

--aq-mode 4

--bframes 5 --ref 3

--hash 2 --allow-non-conformance

-D 8/10/12 (单程序兼容多色深时建议手动指定, 一般默认8bit, 低勿转高, 高转低开--dither)

--pools ,,,, (举例-,+表示该电脑有两个CPU节点, 用第二个. 同时占用多个会造成严重的内存延迟)

**去黑边加速:** --display-window <整数"←,↑,→,↓"像素>, **≥16核cpu优化:** --pme, **分场视频:** --field, **抖动高质量降色深:** --dither, **开始; 结束帧:** --seek; --frames, **crf/abr缓解噪点影响:** --rc-grain

ffmpeg -pix\_fmt yuv420p / yuv422p / yuv444p / yuv420p10 / yuv422p10 / yuv444p10…

**(ffmpeg pipe) x265 CLI命令**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -an -f yuv4mpegpipe -strict unofficial -pix\_fmt<ffprobe pix\_fmt> - | x265.exe --preset slow --me umh --subme 5 --merange 48 --weightb --aq-mode 4 --bframes 5 --ref 3 --hash 2 --allow-non-conformance --y4m - --output ".\输出.hevc"

**libx265 CLI, 兼容libav**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libx265 -pix\_fmt<ffprobe pix\_fmt> -x265-params "preset=slow:me=umh:subme=5:merange=48:weightb=1:bframes=5:ref=3:hash=2:allow-non-conformance=1" -c:a copy ".\输出.hevc"

**libkvazaar CLI (实验性, 第三方, 暂缺crf)模式** **(libx265 ffmpeg CLI缺85%的命令, 无法使用)**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libkvazaar -pix\_fmt<ffprobe pix\_fmt> -kvazaar-params "limit-tu=1:tr-depth-intra=2:pu-depth-intra=4:pu-depth-inter=3:smp=1:amp=1:bipred=1:me=tz:subme=4:merange=48:me-early-termination=off:max-merge=2:ref=3:open-gop=0:period=360:gop=16:transform-skip=1:qp=16:fast-residual-cost=1:early-skip=1:max-merge=4:rd=3:mv-rdo=1:rdoq-skip=1:intra-rdo-et=1:sao=edge:hash=checksum" -c:a copy ".\输出.hevc"

通用·标准

含大量自定义项目，可以配出高压或高速参数

--tu-intra-depth 3 --tu-inter-depth 3 --limit-tu 1 --rdpenalty 1

--me umh --subme <24fps=3, 48fps=4, 60fps=5, 100fps=6> --merange 48 --weightb

--ref 3 --max-merge <2快, 3中, 4慢> --early-skip --no-open-gop --min-keyint 5 --fades --bframes 8 --b-adapt 2 --radl 3 <锐利线条: --pbratio 1.2>

--hist-scenecut <快: --fast-intra / 中: 不填 / 慢: --b-intra / 更慢: + --constrained-intra >

--crf <18~20超清, 19~22高清> --crqpoffs -3 --cbqpoffs -1

--rdoq-level <1快，2很慢>

<动漫源改--hevc-aq, 关aq-mode > --aq-mode 4 --aq-strength <多面=0.8, 多线=1>

--rd 3 --limit-modes --limit-refs 1 --rskip <3快, 2中, 1慢> --rc-lookahead <3×帧率> --tskip-fast --rect <很慢: --amp>

--psy-rd <录像=1.6，动画=0.6, ctu=64加0.6, =16减0.6> --splitrd-skip <实验性: --qp-adaptation-range 3>

--limit-sao --sao-non-deblock --deblock 0:-1

--hash 2 --allow-non-conformance <外/内网NAS串流: --idr-recovery-sei>

-D 8/10/12 (单程序兼容多色深时建议手动指定, 一般默认8bit, 低勿转高, 高转低开--dither)

--pools ,,,, (举例-,+表示该电脑有两个CPU节点, 用第二个. 同时占用多个会造成严重的内存延迟)

**去黑边加速:** --display-window <整数"←,↑,→,↓"像素>, **≥16核cpu优化:** --pme, **分场视频:** --field, **抖动高质量降色深:** --dither, **开始; 结束帧:** --seek; --frames, **crf/abr缓解噪点影响:** --rc-grain

ffmpeg -pix\_fmt yuv420p / yuv422p / yuv444p / yuv420p10 / yuv422p10 / yuv444p10…

**分块-变换**

**动搜-补偿**

**溯块-帧控**

**帧内编码**

**量化**

**率失优量化**

**自适应量化**

**模式决策**

**率失真优化**

**去块-取迁**

**输入输出**

**目标色深**

**多处理器分配**

**其它**

**目标色彩空间**

**(ffmpeg pipe) x265 CLI命令-共12个自定域, 1个危险自定义域**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -an -f yuv4mpegpipe -strict unofficial -pix\_fmt<ffprobe pix\_fmt> - | x265.exe --ctu 〇 --min-cu-size 16 --tu-intra-depth 3 --tu-inter-depth 3 --limit-tu 1 --rdpenalty 1 --me umh --subme 〇 --merange 48 --weightb --ref 3 --max-merge 〇 --early-skip --no-open-gop --min-keyint 5 --fades --bframes 8 --b-adapt 2 --radl 3 --pbratio 1.2 --fast-intra --b-intra --constrained-intra --crf 〇 --crqpoffs -3 --crqpoffs -1 --rdoq-level 〇 --aq-mode 4 --aq-strength 〇 --rd 3 --limit-modes --limit-refs 1 --rskip 〇 --rc-lookahead 〇 --tskip-fast --rect --amp --psy-rd 〇 --splitrd-skip --qp-adaptation-range 4 --limit-sao --sao-non-deblock --deblock 0:-1 --hash 2 --allow-non-conformance --y4m - --output ".\输出.hevc"

**libx265 CLI, 兼容libav**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libx265 -pix\_fmt<ffprobe pix\_fmt> -x265-params "ctu=〇:min-cu-size=16:tu-intra-depth=3:tu-inter-depth=3:limit-tu=1:rdpenalty=1:me=umh:subme=〇:merange=48:weightb=1:ref=3:max-merge=〇:early-skip=1:open-gop=0:min-keyint=5:fades=1:bframes=8:b-adapt=2:radl=3:pbratio=1.2:fast-intra=1:b-intra=1:constrained-intra=1:crf=〇:crqpoffs=-3:cbqpoffs=-1:rdoq-level=〇:aq-mode=4:aq-strength=〇:rd=3:limit-modes=1:limit-refs=1:rskip=〇:rc-lookahead=〇:tskip-fast=1:rect=1:amp=1:psy-rd=〇:splitrd-skip=1:qp-adaptation-range=4:limit-sao=1:sao-non-deblock=1:deblock=0:-1:hash=2:allow-non-conformance=1" -c:a copy ".\输出.mp4"

高压·录像·要求高画质源

要求高画质源，否则画质优势比不过通用-简单，慢很多

--tu-intra-depth 4 --tu-inter-depth 4 --limit-tu 1

--me star --subme <24fps=3, 48fps=4, 60fps=5, 100fps=6> --merange 48 --weightb

--ref 3 --max-merge 4 --no-open-gop --min-keyint 3 --keyint <9×帧率> --fades --bframes 8 --b-adapt 2 --radl 3

--hist-scenecut --constrained-intra --b-intra

--crf 21.8 --qpmin 8 --crqpoffs -3 --ipratio 1.2 --pbratio 1.5

--rdoq-level 2

--aq-mode 4 --aq-strength <1~1.3> --qg-size 8

--rd 3 --limit-refs 0 --rskip 0 --rc-lookahead <1.8×帧率> --rect --amp

--psy-rd <录像=1.6, 动画=0.6, ctu=64就加0.6, =16就减0.6> <实验性: --qp-adaptation-range 3>

--deblock 0:-1

--limit-sao --sao-non-deblock --selective-sao 3

--hash 2 --allow-non-conformance <外/内网NAS串流: --idr-recovery-sei>

-D 8/10/12 (单程序兼容多色深时建议手动指定, 一般默认8bit, 低勿转高, 高转低开--dither)

--pools ,,,, (举例-,+表示该电脑有两个CPU节点, 用第二个. 同时占用多个会造成严重的内存延迟)

**去黑边加速:** --display-window <整数"←,↑,→,↓"像素>, **≥16核cpu优化:** --pme, **分场视频:** --field, **抖动高质量降色深:** --dither, **开始; 结束帧:** --seek; --frames, **crf/abr缓解噪点影响:** --rc-grain

ffmpeg -pix\_fmt yuv420p / yuv422p / yuv444p / yuv420p10 / yuv422p10 / yuv444p10…

**分块-变换**

**动搜-补偿**

**溯块-帧控**

**帧内编码**

**量化**

**率失优量化**

**自适应量化**

**模式决策**

**率失真优化**

**去块**

**取样迁就偏移**

**输入输出**

**目标色深**

**多处理器分配**

**其它**

**目标色彩空间**

**(ffmpeg pipe) x265 CLI命令**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -an -f yuv4mpegpipe -strict unofficial -pix\_fmt<ffprobe pix\_fmt> - | x265.exe --tu-intra-depth 4 --tu-inter-depth 4 --limit-tu 1 --me star --subme 〇 --merange 48 --weightb --ref 3 --max-merge 4 --no-open-gop --min-keyint 3 --keyint 〇 --fades --bframes 8 --b-adapt 2 --radl 3 --constrained-intra --b-intra --crf 21.8 --qpmin 8 --crqpoffs -3 --ipratio 1.2 --pbratio 1.5 --rdoq-level 2 --aq-mode 4 --aq-strength 〇 --qg-size 8 --rd 3 --limit-refs 0 --rskip 0 --rc-lookahead 〇 --rect --amp --psy-rd 〇 --qp-adaptation-range 3 --deblock 0:-1 --limit-sao --sao-non-deblock --selective-sao 3 --hash 2 --allow-non-conformance --y4m - --output ".\输出.hevc"

**libx265 CLI, 兼容libav**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libx265 -pix\_fmt <ffprobe pix\_fmt> -x265-params "tu-intra-depth=4:tu-inter-depth=4:limit-tu=1:me=star:subme=〇:merange=48:weightb=1:ref=3:max-merge=4:open-gop=0:min-keyint=3:keyint=〇:fades=1:bframes=8:b-adapt=2:radl=3:constrained-intra=1:b-intra=1:crf=21.8:qpmin=8:crqpoffs=-3:ipratio=1.2:pbratio=1.5:rdoq-level=2:aq-mode=4:aq-strength=〇:qg-size=8:rd=3:limit-refs=0:rskip=0:rc-lookahead=〇:rect=1:amp=1:psy-rd=〇:qp-adaptation-range=3:deblock=0:-1:limit-sao=1:sao-non-deblock=1:selective-sao=3:hash=2:allow-non-conformance=1" -c:a copy ".\输出.mp4"

# 剪辑素材存档

--ctu 32

--me star --subme <24fps=3, 48fps=4, 60fps=5, 100fps=6> --merange 48 --analyze-src-pics

--max-merge 4 --early-skip --b-intra

--hist-scenecut --no-open-gop --min-keyint 1 --keyint <7×帧率>--ref 3 --fades --bframes 7 --b-adapt 2

--crf 17 --crqpoffs -3 --cbqpoffs -2

--rd 3 --limit-modes --limit-refs 1 --rskip 1 --rc-lookahead <4×帧率>

--splitrd-skip

--deblock -1:-1

--hash 2 --allow-non-conformance

--tune grain

-D 8/10/12 (单程序兼容多色深时建议手动指定, 一般默认8bit, 低勿转高, 高转低开--dither)

**去黑边加速:** --display-window <整数"←,↑,→,↓"像素>, **≥16核cpu优化:** --pme, **分场视频:** --field, **抖动高质量降色深:** --dither, **开始; 结束帧:** --seek; --frames, **crf/abr缓解噪点影响:** --rc-grain

ffmpeg -pix\_fmt yuv420p / yuv422p / yuv444p / yuv420p10 / yuv422p10 / yuv444p10…

**分块**

**动态搜索**

**帧内搜索**

**帧控制**

**量化**

**模式决策**

**率失真优化**

**环路滤波去块**

**输入输出**

**主控**

**目标色深**

**其它**

**目标色彩空间**

**(ffmpeg pipe) x265 CLI命令**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -an -f yuv4mpegpipe -strict unofficial -pix\_fmt<ffprobe pix\_fmt> - | x265.exe --ctu 32 --me star --subme 〇 --merange 48 --analyze-src-pics --max-merge 4 --early-skip --b-intra --no-open-gop --min-keyint 1 --keyint 〇 --ref 3 --fades --bframes 7 --b-adapt 2 --crf 17 --crqpoffs -3 --cbqpoffs -2 --rd 3 --limit-modes --limit-refs 1 --rskip 1 --rc-lookahead 〇 --splitrd-skip --deblock -1:-1 --hash 2 --allow-non-conformance --tune grain --y4m - --output ".\输出.hevc"

**libx265 CLI, 兼容libav**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libx265 -pix\_fmt<ffprobe pix\_fmt> -x265-params "ctu=32:me=star:subme=〇:merange=48:analyze-src-pics=1:max-merge=4:early-skip=1:open-gop=0:min-keyint=1:keyint=〇:ref=3:fades=1:bframes=7:b-adapt=2:radl=3:constrained-intra=1:b-intra=1:crf=17:crqpoffs =-3:cbqpoffs=-2:rd=3:limit-modes=1:limit-refs=1:rskip=1:rc-lookahead=〇:splitrd-skip=1:deblock=-1:-1:hash=2:allow-non-conformance=1:tune=grain" -c:a copy ".\输出.mp4"
* **目标色深-色彩空间:** -pix\_fmt yuv420p / yuv422p / yuv444p / yuv420p10 / yuv422p10 / yuv444p10…

# 高压·动漫·字幕组

遵守x265开发者定义的低到高成本动漫高压高画标准，秘诀是分块和TU开满+避免跳过，建议YUV4:2:0 8~10bit

--tu-intra-depth 4 --tu-inter-depth 4 --max-tu-size 16

--me umh --merange 48 --subme <24fps=3, 48fps=4, 60fps=5, 100fps=6> --weightb <80年代动漫缺乏光线变化, 可略weightb> --max-merge 4 --early-skip

--ref 3 --no-open-gop --min-keyint 5 --keyint <12×帧率> --fades --bframes 16 --b-adapt 2 --radl 3 --bframe-bias 20

--hist-scenecut --constrained-intra --b-intra

--crf 22 --crqpoffs -4 --cbqpoffs -2 --ipratio 1.6 --pbratio 1.3 --cu-lossless --tskip

--psy-rdoq 2.3 --rdoq-level 2

--hevc-aq --aq-strength 0.9 --qg-size 8

--rd 3 --limit-modes --limit-refs 1 --rskip 1 --rc-lookahead <2.5×帧率> --rect --amp

--psy-rd 1.5 --splitrd-skip --rdpenalty 2 <实验性: --qp-adaptation-range 4>

--deblock 0:-1

--limit-sao --sao-non-deblock

--hash 2 --allow-non-conformance <外/内网NAS串流: --single-sei --idr-recovery-sei>

--pools ,,,, (举例-,+表示该电脑有两个CPU节点, 用第二个. 同时占用多个会造成严重的内存延迟)

-D 8/10/12 (单程序兼容多色深时建议手动指定, 一般默认8bit, 低勿转高, 高转低开--dither)

**去黑边加速:** --display-window <整数"←,↑,→,↓"像素>, **≥16核cpu优化:** --pme, **分场视频:** --field, **抖动高质量降色深:** --dither, **开始; 结束帧:** --seek; --frames, **crf/abr缓解噪点影响:** --rc-grain

ffmpeg -pix\_fmt yuv420p / yuv422p / yuv444p / yuv420p10 / yuv422p10 / yuv444p10…

**分块-变换**

**动搜-补偿**

**溯块-帧控**

**帧内编码**

**量化**

**率失优量化**

**自适应量化**

**模式决策**

**率失真优化**

**去块**

**取样迁就偏移**

**输入输出**

**多处理器分配**

**目标色深**

**其它**

**目标色彩空间**

**(ffmpeg pipe) x265 CLI命令**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -an -f yuv4mpegpipe -strict unofficial -pix\_fmt<ffprobe pix\_fmt> - | x265.exe --tu-intra-depth 4 --tu-inter-depth 4 --max-tu-size 16 --me umh --subme 〇 --merange 48 --weightb --max-merge 4 --early-skip --ref 3 --no-open-gop --min-keyint 5 --keyint 〇 --fades --bframes 16 --b-adapt 2 --radl 3 --bframe-bias 20 --constrained-intra --b-intra --crf 22 --crqpoffs -4 --cbqpoffs -2 --ipratio 1.6 --pbratio 1.3 --cu-lossless --tskip --psy-rdoq 2.3 --rdoq-level 2 --hevc-aq --aq-strength 0.9 --qg-size 8 --rd 3 --limit-modes --limit-refs 1 --rskip 1 --rc-lookahead 〇 --rect --amp --psy-rd 1.5 --splitrd-skip --rdpenalty 2 --qp-adaptation-range 4 --deblock -1:0 --limit-sao --sao-non-deblock --hash 2 --allow-non-conformance --y4m - --output ".\输出.hevc"

**libx265 CLI, 兼容libav**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libx265 -pix\_fmt<ffprobe pix\_fmt> -x265-params "tu-intra-depth=4:tu-inter-depth=4:max-tu-size=16:me=umh:subme=〇:merange=48:weightb=1:max-merge=4:early-skip=1:ref=3:open-gop=0:min-keyint=5:keyint=〇:fades=1:bframes=16:b-adapt=2:radl=3:bframe-bias=20:constrained-intra=1:b-intra=1:crf=22:crqpoffs=-4:cbqpoffs=-2:ipratio=1.6:pbratio=1.3:cu-lossless=1:tskip=1:psy-rdoq=2.3:rdoq-level=2:hevc-aq=1:aq-strength=0.9:qg-size=8:rd=3:limit-modes=1:limit-refs=1:rskip=1:rc-lookahead=〇:rect=1:amp=1:psy-rd=1.5:splitrd-skip=1:rdpenalty=2:qp-adaptation-range=4:deblock=-1:0:limit-sao=1:sao-non-deblock=1:hash=2:allow-non-conformance=1" -c:a copy ".\输出.mp4"

# 动漫·Ripper冷战·仅限HEDT工作站与原画

比字幕组参数码率更高，压制速度更慢

--tu-intra-depth 4 --tu-inter-depth 4 --max-tu-size 4 --limit-tu 1

--me star --subme <24fps=3, 48fps=4, 60fps=5, 100fps=6> --merange 52 --analyze-src-pics --weightb --max-merge 4

--ref 3 --no-open-gop --min-keyint 1 --keyint <12×帧率> --fades --bframes 16 --b-adapt 2 --radl 2

--hist-scenecut --b-intra

--crf 16.5 --crqpoffs -5 --cbqpoffs -2 --ipratio 1.67 --pbratio 1.33

--cu-lossless

--psy-rdoq 2.5 --rdoq-level 2

--hevc-aq --aq-strength 1.4 --qg-size 8

--rd 5 --limit-refs 0 --rskip 2 --rskip-edge-threshold 3 --rc-lookahead <2.5×帧率> --rect --amp --no-cutree

--psy-rd 1.5 --rdpenalty 2 <实验性: --qp-adaptation-range 5>

--deblock -2:-2

--limit-sao --sao-non-deblock --selective-sao 1

--hash 2 --allow-non-conformance <外/内网NAS串流: --single-sei --idr-recovery-sei>

--pools ,,,, (举例-,+表示该电脑有两个CPU节点, 用第二个. 同时占用多个会造成严重的内存延迟)

-D 8/10/12 (单程序兼容多色深时建议手动指定, 一般默认8bit, 低勿转高, 高转低开--dither)

**去黑边加速:** --display-window <整数"←,↑,→,↓"像素>, **≥16核cpu优化:** --pme, **分场视频:** --field, **抖动高质量降色深:** --dither, **开始; 结束帧:** --seek; --frames, **crf/abr缓解噪点影响:** --rc-grain

**分块-变换**

**动搜-补偿**

**溯块-帧控**

**帧内编码**

**量化**

**无损量化**

**率失优量化**

**自适应量化**

**模式决策**

**率失真优化**

**去块**

**取样迁就偏移**

**输入输出**

**多处理器分配**

**目标色深**

**其它**

**(ffmpeg pipe) x265 CLI命令**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -an -f yuv4mpegpipe -strict unofficial -pix\_fmt<ffprobe pix\_fmt> - | x265.exe --tu-intra-depth 4 --tu-inter-depth 4 --max-tu-size 4 --limit-tu 1 --me star --subme 〇 --merange 52 --analyze-src-pics --weightb --max-merge 4 --ref 3 --no-open-gop --min-keyint 1 --keyint 〇 --fades --bframes 16 --b-adapt 2 --radl 2 --b-intra --crf 16.5 --crqpoffs -5 --cbqpoffs -2 --ipratio 1.67 --pbratio 1.33 --cu-lossless --psy-rdoq 2.5 --rdoq-level 2 --hevc-aq --aq-strength 1.4 --qg-size 8 --rd 5 --limit-refs 0 --rskip 2 --rskip-edge-threshold 3 --rc-lookahead 〇 --rect --amp --no-cutree --psy-rd 1.5 --rdpenalty 2 --qp-adaptation-range 5 --deblock -2:-2 --limit-sao --sao-non-deblock --selective-sao 1 --hash 2 --allow-non-conformance --y4m - --output ".\输出.hevc"

**libx265 CLI, 兼容libav**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libx265 -pix\_fmt<ffprobe pix\_fmt> -x265-params "tu-intra-depth=4:tu-inter-depth=4:max-tu-size=4:limit-tu=1:me=star:subme=〇:merange=52:analyze-src-pics=1:weightb=1:max-merge=4:mcstf=1:ref=3:open-gop=0:min-keyint=1:keyint=〇:fades=1:bframes=16:b-adapt=2:radl=2:b-intra=1:crf=16.5:crqpoffs=-5:cbqpoffs=-2:ipratio=1.6:pbratio=1.33:cu-lossless=1:psy-rdoq=2.5:rdoq-level=2:hevc-aq=1:aq-strength=1.4:qg-size=8:rd=5:limit-refs=0:rskip=2:rskip-edge-threshold=3:rc-lookahead=〇:rect=1:amp=1:cutree=0:psy-rd=1.5:rdpenalty=2:qp-adaptation-range=5:deblock=-2:-2:limit-sao=1:sao-non-deblock=1:selective-sao=1:hash=2:allow-non-conformance=1" -c:a copy ".\输出.mp4"