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命令行参数用法教程

[照上表下载ffmpeg和x265并记住路径, 此处置于D盘根目录下]



[打开Windows自带的CMD.exe, 点开始再输入c, m, d即可]



[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

[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

[导出与导入] x265.exe [参数] --output C:\文件夹\导出.mp4 C:\文件夹\导入.mp4

[导出与pipe导入] 见下方ffmpeg, VapourSynth, avs2yuv传递参数

[参数用例] D:\ffmpeg.exe -i F:\视频.mov -an -pix\_fmt yuv420p10 -f yuv4mpegpipe -strict unofficial - | D:\x265-10bit.exe -D 10 --input-csp i444 --allow-non-conformance --rect --ctu 64 --min-cu-size 8 --limit-tu 1 --tu-intra-depth 4 --tu-inter-depth 4 --max-tu-size 16 --me star --subme 6 --merange 48 --analyze-src-pics --max-merge 4 --early-skip --b-intra --no-open-gop --radl 3 --min-keyint 5 --keyint 240 --ref 3 --fades --bframes 14 --b-adapt 2 --crf 16.5 --rdoq-level 2 --psy-rdoq 4 --aq-mode 4 --qg-size 16 --rd 5 --limit-modes --limit-refs 1 --rskip 1 --rd-refine --splitrd-skip --no-sao --tskip --master-display G(8500,39850)B(6550,2300)R(35400,14600)WP(15635,16450)L(10000000,1) --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

### ass字幕渲染

* 单字体, 有算符∑∫∞, 无上下标用: avs texttosub()滤镜
* 多字体, 算符+艺术体𝓚𝔎+上下标9⁹₉: ffmpeg -filter\_complex "ass='F\:/字幕.ass'"滤镜

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

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

**ffmpeg 封装视音频, 后缀名指定封装格式**

* ffmpeg.exe -i ".\视频流入.mp4" -an -c:v copy -i ".\音频流入.aac" -c copy "封装出.mov"

**ffmpeg 替换音频流, itoffset±秒数以对齐**

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

**ffmpeg转固定/可变/删帧率**: -vsync cfr (1) / vfr (2) / drop

**ffmpeg内置缩放**: -sws\_flags bicubic bitexact gauss bicublin lanczos spline +full\_chroma\_int +full\_chroma\_inp +accurate\_rnd (例: -sws\_flags bitexact+full\_chroma\_int+full\_chroma\_inp+accurate\_rnd)

### 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(?,1)

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

bt2020: G(8500,39850)B(6550,2300)R(35400,14600)WP(15635,16450)L(?,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)>

-- cll <和master-display 的L最大值一样>

--colormatrix <照源, 例: gbr bt709 fcc bt470bg smpte170m YCgCo bt2020nc bt2020c smpte2085 ictcp>

--transfer <照源, 例: gbr bt709 fcc bt470bg smpte170m YCgCo bt2020nc bt2020c smpte2085 ictcp>

通用·简单

---自定义项目all off，方便急用但降低了压缩率

**分块-变换**

**动搜-补偿**

**溯块-帧控**

**量化**

**自适应量化**

**率失优-决策**

**去块-取偏**

**输入输出**

**目标色深**

**多处理器分配**

**去黑边加速**

**分场视频**

--min-cu-size 16 --limit-tu 1 --tu-intra-depth 2 --tu-inter-depth 2 --rdpenalty 1

--me umh --subme 6 --merange 48 --rskip 1 --weightb --mctf

--ref 3 --early-skip --no-open-gop --min-keyint 5 --fades --bframes 11 --b-adapt 2 --radl 2 --fast-intra --hist-scenecut

--crf 18 --crqpoffs -2

--aq-mode 4 --qg-size 16

--rd 5 --splitrd-skip --rdoq-level 1 --limit-modes --rect --tskip-fast

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

--hash crc --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)

--display-window <整数"←,↑,→,↓"像素>

--field

**(ffmpeg pipe) x265 CLI命令**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -an -f yuv4mpegpipe -strict unofficial - | x265.exe --min-cu-size 16 --limit-tu 1 --tu-intra-depth 2 --tu-inter-depth 2 --rdpenalty 1 --me umh --subme 4 --merange 48 --rskip 1 --weightb --mctf --ref 3 --early-skip --no-open-gop --max-merge 2 --min-keyint 5 --fades --bframes 11 --b-adapt 2 --radl 2 --fast-intra --hist-scenecut --crf 18 --crqpoffs -2 --aq-mode 4 --qg-size 16 --rd 5 --splitrd-skip --rdoq-level 1 --limit-modes --rect --tskip-fast --limit-sao --sao-non-deblock --deblock 0:-1 --hash crc --allow-non-conformance --y4m - --output ".\输出.mp4"

**libx265 CLI, 兼容libav**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libx265 -x265params "min-cu-size=16:limit-tu=1:tu-intra-depth=2:tu-inter-depth=2:rdpenalty=1:me=umh:subme=4:merange=48:rskip=1:weightb=1:mctf=1:ref=3:early-skip=1:max-merge=2:open-gop=0:min-keyint=5:fades=1:bframes=11:b-adapt=2:radl=2:fast-intra=1:hist-scenecut=1:crf=18:crqpoffs=-2:aq-mode=4:qg-size=16:rd=5:splitrd-skip=1:rdoq-level=1:limit-modes=1:rect=1:tskip-fast=1:limit-sao=1:sao-non-deblock=1:deblock=0:-1:hash=crc:allow-non-conformance=1" -c:a copy ".\输出.mp4"
* **目标色深-色彩空间:** -pix\_fmts yuv420p / yuv422p / yuv444p / yuv420p10 / yuv422p10 / yuv444p10…

**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 -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.mp4"

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 --analyze-src-pics --weightb --mctf

--ref 3 --max-merge <2快, 3中, 4慢> --early-skip --no-open-gop --min-keyint 5 --keyint <9×帧率> --fades --bframes 11 --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 <16~18超清 19 ~20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 5 --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 crc --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)

--display-window <整数"←,↑,→,↓"像素>

--field

--dither

**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**

**去黑边加速**

**输入分场视频**

**抖动降低色深**

**(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 - | 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 --analyze-src-pics --weightb --mctf --ref 3 --max-merge 〇 --early-skip --no-open-gop --min-keyint 5 --fades --bframes 11 --b-adapt 2 --radl 3 --pbratio 1.2 --hist-scenecut --fast-intra --b-intra --constrained-intra --crf 〇 --crqpoffs -3 --crqpoffs -1 --rdoq-level 〇 --aq-mode 4 --aq-strength 〇 --rd 5 --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 crc --allow-non-conformance --y4m - --output ".\输出.mp4"

**libx265 CLI, 兼容libav**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libx265 -x265params "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:analyze-src-pics=1:weightb=1:mctf=1:ref=3:max-merge=〇:early-skip=1:open-gop=0:min-keyint=5:fades=1:bframes=11:b-adapt=2:radl=3:pbratio=1.2: hist-scenecut=1: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=5: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=crc:allow-non-conformance=1" -c:a copy ".\输出.mp4"
* **目标色深-色彩空间:** -pix\_fmts yuv420p / yuv422p / yuv444p / yuv420p10 / yuv422p10 / yuv444p10…

高压·录像

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

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

--ref 3 --max-merge 4 --no-open-gop --min-keyint 3 --keyint <13×帧率> --fades --bframes 14 --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 <无噪点动画=0.8, 录像=1> --qg-size 8

--rd 5 --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> --rd-refine <实验性: --qp-adaptation-range 3>

--deblock 0:0

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

--hash crc --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)

--display-window <整数"←,↑,→,↓"像素>

**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**

**黑边跳过**

**(ffmpeg pipe) x265 CLI命令**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -an -f yuv4mpegpipe -strict unofficial - | x265.exe --tu-intra-depth 4 --tu-inter-depth 4 --limit-tu 1 --me star --subme 〇 --merange 48 --analyze-src-pics --weightb --mctf --ref 3 --max-merge 4 --no-open-gop --min-keyint 3 --keyint 〇 --fades --bframes 14 --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 〇 --qg-size 8 --rd 5 --limit-refs 0 --rskip 0 --rc-lookahead 〇 --rect --amp --psy-rd 〇 --rd-refine --qp-adaptation-range 3 --deblock 0:0 --limit-sao --sao-non-deblock --selective-sao 3 --hash crc --allow-non-conformance --y4m - --output ".\输出.mp4"

**libx265 CLI, 兼容libav**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libx265 -x265params "tu-intra-depth=4:tu-inter-depth=4:limit-tu=1:me=star:subme=〇:merange=48:weightb=1:mctf=1:ref=3:max-merge=4:open-gop=0:min-keyint=3:keyint=〇:fades=1:bframes=14:b-adapt=2:radl=3:hist-scenecut=1: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=5:limit-refs=0:rskip=0:rc-lookahead=〇:rect=1:amp=1:psy-rd=〇:rd-refine=1:qp-adaptation-range=3:deblock=0:0:limit-sao=1:sao-non-deblock=1:selective-sao=3:hash=crc:allow-non-conformance=1" -c:a copy ".\输出.mp4"
* **目标色深-色彩空间:** -pix\_fmts yuv420p / yuv422p / yuv444p / yuv420p10 / yuv422p10 / yuv444p10…

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

--splitrd-skip --rd-refine

--deblock 0:-1

--hash crc --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)

**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**

**(ffmpeg pipe) x265 CLI命令**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -an -f yuv4mpegpipe -strict unofficial - | x265.exe --ctu 32 --me star --subme 〇 --merange 48 --analyze-src-pics --max-merge 4 --early-skip --b-intra --hist-scenecut --no-open-gop --min-keyint 1 --keyint 〇 --ref 3 --fades --bframes 7 --b-adapt 2 --crf 17 --crqpoffs -3 --cbqpoffs -2 --rd 5 --limit-modes --limit-refs 1 --rskip 1 --rc-lookahead 〇 --splitrd-skip --deblock -1:-1 --hash crc --allow-non-conformance --tune grain --y4m - --output ".\输出.mp4"

**libx265 CLI, 兼容libav**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libx265 -x265params "ctu=32:me=star:subme=〇:merange=48:analyze-src-pics=1:max-merge=4:early-skip=1:hist-scenecut=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=5:limit-modes=1:limit-refs=1:rskip=1:rc-lookahead=〇:splitrd-skip=1:deblock=-1:-1:hash=crc:allow-non-conformance=1:tune=grain" -c:a copy ".\输出.mp4"
* **目标色深-色彩空间:** -pix\_fmts yuv420p / yuv422p / yuv444p / yuv420p10 / yuv422p10 / yuv444p10…

动漫·高压·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年代旧动漫缺乏光线变化, 可不写weightb> --max-merge 4 --early-skip --mctf

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

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

--hevc-aq --qg-size 8

--rd 5 --limit-modes --limit-refs 1 --rskip 1 --rc-lookahead <2.5×帧率> --rect --amp

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

--deblock 0:-1

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

--hash crc --allow-non-conformance --single-sei <外/内网NAS串流: --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)

--display-window <整数"←,↑,→,↓"像素>

--rc-grain

**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**

**黑边跳过**

**qp计算抗噪**

**(ffmpeg pipe) x265 CLI命令**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -an -f yuv4mpegpipe -strict unofficial - | 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 --mctf --ref 3 --no-open-gop --min-keyint 5 --keyint 〇 --fades --bframes 16 --b-adapt 2 --radl 3 --bframe-bias 20 --hist-scenecut --constrained-intra --b-intra --crf 19 --crqpoffs -4 --cbqpoffs -2 --ipratio 1.6 --pbratio 1.3 --cu-lossless --tskip --psy-rdoq 2.3 --rdoq-level 2 --hevc-aq --qg-size 8 --rd 5 --limit-modes --limit-refs 1 --rskip 1 --rc-lookahead 〇 --rect --amp --psy-rd 1.5 --rd-refine --splitrd-skip --rdpenalty 2 --qp-adaptation-range 4 --deblock -1:0 --limit-sao --sao-non-deblock --hash crc --allow-non-conformance --single-sei --y4m - --output ".\输出.mp4"

**libx265 CLI, 兼容libav**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libx265 -x265params "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:mctf=1:ref=3:open-gop=0:min-keyint=5:keyint=〇:fades=1:bframes=16:b-adapt=2:radl=3:bframe-bias=20:hist-scenecut=1:constrained-intra=1:b-intra=1:crf=19: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:qg-size=8:rd=5:limit-modes=1:limit-refs=1:rskip=1:rc-lookahead=〇:rect=1:amp=1:psy-rd=1.5:rd-refine=1:splitrd-skip=1:rdpenalty=2:qp-adaptation-range=4:deblock=-1:0:limit-sao=1:sao-non-deblock=1:hash=crc:allow-non-conformance=1:single-sei=1" -c:a copy ".\输出.mp4"
* **目标色深-色彩空间:** -pix\_fmts yuv420p / yuv422p / yuv444p / yuv420p10 / yuv422p10 / yuv444p10…

动漫·Ripper冷战·仅限HEDT工作站

要求暗场下放大四五倍的暂停画面和源差不多。因此相比字幕组参数的码率更高，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 --mctf

--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 --crqpoffs -t5 --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 0 --rc-lookahead <2.5×帧率> --rect --amp --no-cutree

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

--deblock -2:-2

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

--hash crc --allow-non-conformance --single-sei <外/内网NAS串流: --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)

**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**

**(ffmpeg pipe) x265 CLI命令**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -an -f yuv4mpegpipe -strict unofficial - | 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 --mctf --ref 3 --no-open-gop --min-keyint 1 --keyint 〇 --fades --bframes 16 --b-adapt 2 --radl 2 --hist-scenecut --b-intra --crf 16 --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 0 --rc-lookahead 〇 --rect --amp --no-cutree --psy-rd 1.5 --rd-refine --rdpenalty 2 --qp-adaptation-range 5 --deblock -2:-2 --limit-sao --sao-non-deblock --selective-sao 1 --hash crc --allow-non-conformance --single-sei --y4m - --output ".\输出.mp4"

**libx265 CLI, 兼容libav**

* ffmpeg.exe -loglevel 16 -hwaccel auto -y -hide\_banner -i ".\导入.mp4" -c:v libx265 -x265params "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:mctf=1:ref=3:open-gop=0:min-keyint=1:keyint=〇:fades=1:bframes=16:b-adapt=2:radl=2:hist-scenecut=1:b-intra=1:crf=16: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=0:rc-lookahead=〇:rect=1:amp=1:cutree=0:psy-rd=1.5:rd-refine=1:rdpenalty=2:qp-adaptation-range=5:deblock=-2:-2:limit-sao=1:sao-non-deblock=1:selective-sao=1:hash=crc:allow-non-conformance=1:single-sei=1" -c:a copy ".\输出.mp4"
* **目标色深-色彩空间:** -pix\_fmts yuv420p / yuv422p / yuv444p / yuv420p10 / yuv422p10 / yuv444p10…