x264码率控制参数详解  
  
  -q, --qp <integer>          Force constant QP (0-51, 0=lossless)  
默认：无  
说明：三种可选的码率控制方法(bitrate, CQP，CRF)之一。设置x264使用固定QP模式。设定的QP将被作为P帧的量化参数，I帧和B帧的量化参数由–ipratio and –pbratio参数进一步算出。CQP模式使用固定的QP，这意味着最终的文件大小是不可知的。设置为0将产出无损输出。相同视觉质量时，CQP模式编码输出的文件比CRF模式大。CQP模式将关闭自适应量化器，因为它是固定QP的。  
这个选项和 –bitrate和–crf是互斥的，三者只能选一个，选择的顺序是 bitrate > QP > CRF。一般而言crf都能代替QP模式，不过QP因为完全不需要预测所以它会运行地更快些。  
参见：–bitrate, –crf, –ipratio, –pbratio                        
  
  
  -B, --bitrate <integer>     Set bitrate (kbit/s)  
默认：无  
说明：三种可选的码率控制方法(bitrate, CQP，CRF)之一。设置x264使用固定目标比特率模式。固定目标比特率意味着最终文件的大小是可知的，但是目标的质量是不可知的。x264会试图让最终文件的整体码率与给定的码率相等。参数的量纲为Kbits/sec，x264程序里面1K是1000,而不是1024。  
通常这个选项和–pass选项配合进行2趟编码。在网络流媒体应用中选择这个方法。  
这个选项和 –qp和–crf是互斥的，三者只能选一个，选择的顺序是 bitrate > QP > CRF。  
参见：–qp, –crf, –ratetol, –pass, –stats  
  
  
      --crf <float>           Quality-based VBR (0-51) [23.0]  
默认：23.0  
说明：三种可选的码率控制方法(bitrate, CQP，CRF)之一。固定ratefactor。QP是固定量化参数，bitrate是固定文件大小，crf则是固定“质量”。CRF=n可以提供跟CQP=n一样的视觉的质量，但是文件更小。RF是Rate Factor。  
crf是通过降低那些“不重要”帧的质量做到这一切的。“不重要”意思是过于耗费码率又难以用肉眼察觉的帧，比如纹理复杂或者高速运行的场景。省下来的码率会用在其它更有效的帧上面。可以把CRF看作是一个动态的CQP，它根据一定的准则分配不同的QP到每一帧上。  
crf编码比2趟编码快，因为它相当于省略了第1趟编码。crf的最终码率也是不可预测的。一般压片字幕组都使用这种码率控制方式。  
这个选项和 –qp和–crf是互斥的，三者只能选一个，选择的顺序是 bitrate > QP > CRF。  
参见：–qp, –bitrate  
一些关于crf的备忘：a.相对于bitrate方式，cpu占用与内存占用均会下降；b.锐化滤镜会让crf的码率上升；c.vbv对crf依然有效；d. crf18就接近无损，字幕组惯用20-22。  
下面是对照表：  
crf bitrate-min  bitrate-max  
 20    420          1200  
 21    370          1000  
 22    330           850  
 23    300           700  
 24    250           600  
 25    220           500  
 26    200           432  
  
  
      --rc-lookahead <integer> Number of frames for frametype lookahead [40]  
默认：40  
说明：为mb-tree码率控制机制（Macroblock Tree Ratecontrol）和vbv-lookahead设置可用的帧的数量。最大可设置为250。  
对于mb-tree而言，调大这个值会得到更准确地结果，但也会更慢。mb-tree能使用的最大值是MIN(rc-lookahead, keyint)。  
对于vbv-lookahead而言，调大这个值会得更稳定和精确的码率控制。vbv-lookahead能使用的最大值是如下公式算出来的：MIN(rc-lookahead, MAX(keyint, MAX(vbv-maxrate, bitrate) / vbv-bufsize \* fps))  
参见：–no-mbtree, –vbv-bufsize, –vbv-maxrate  
rc-lookahead会增加编码启动群延时（编码器会储满rc-lookahead帧数据后，才开始第一帧编码），在实时视频通信系统中设置它为0。  
  
  
      --vbv-bufsize <integer> Set size of the VBV buffer (kbit) [0]  
默认：0  
说明：设置VBV可用的最大缓冲区，单位是kbits。VBV（Video Buffering Verifier，视频缓存检验器），它作用是处理各帧编码后大小不一和恒定输出码率的矛盾。可以将VBV想象成为一个水池，水池的入口连接着encoder的输出，出口为恒定码率的网络输出。为了使得输出恒定，encoder必须保证水池既不上溢也不下溢。下溢会导致无数据输出，上溢会导致数据丢失。所以encoder在编码一帧时会参考当前vbv的充盈情况，并由此计算出，当前帧应当编码出多少比特，从而保证既不上溢（增加QP）也不下溢（减少QP）。为了防止上溢，VBV会要求降低编码比特，encoder会采用更大一些的QP，这样必然降低了视频序列质量，因此只在真正需要的时候才设定它。一般在硬件解码端的时候，一定要做这个设置。例如蓝光定义最大的数据缓冲为30M比特，能够解码的最大视频码率为40Mbps。这是对硬件解码器资源选配的一个合理限制，否则解码器不知道预留多大的数据缓冲空间才是安全的。这类似264标准的level设定。  
http://mewiki.project357.com/wiki/X264\_Encoding\_Suggestions#VBV\_Encoding有三个例子是介绍vbv-bufsize和vbv-maxrate这两个参数的设定。  
  
  
      --vbv-maxrate <integer> Max local bitrate (kbit/s) [0]  
默认：0  
说明：设置VBV可用的最大码率，单位是kbps。使用它等于限制了I帧最大的码率输出（一般都是I帧达到maxrate的限制，除非bitrate设置过低不合理）。而I帧质量的降低，会拉低整个视频序列的视频质量。因此只在真正有最大码率限制的情况下才去设定它。  
参见：–vbv-bufsize, –vbv-init，  
在VBR和ABR情况下，可以设置vbv-maxrate和vbv-bufsize（很多情况下设置为vbv-maxrate = vbv-bufsize = bitrate，注意vbv-bufsize的量纲是不同，即最大缓存1s的数据）。  
在RC\_ABR码率控制方法下，如果vbv-maxrate == bitrate这时候其实进行的是CBR码率控制方法，encoder力争控制每一帧输出都稳定在bitrate上。  
如果设置vbv-maxrate则vbv-bufsize必须设置，否则会提示"VBV maxrate specified, but no bufsize, ignored"， vbv-maxrate会被改回为0。  
如果设置vbv-bufsize而vbv-maxrate没有设置，则会提示"VBV maxrate unspecified, assuming CBR"，vbv-maxrate会被设为bitrate，此时进行CBR编码。  
一般设置为 vbv-maxrate = vbv-bufsize = a\*bitrate。  
a=0，不启用VBV机制，编码性能最好，适用于硬盘文件编码；但它输出的码率波动性大，有可能某些帧的比特数过高，不适用于有实际带宽限制的流媒体传输。  
0<a<1，这样的设置没有什么意义，得不到任何的好处。  
a=1，等于CBR，CBR是一种复杂和平滑场景都不大讨好的码率控制方法，一般不采用这一方法。  
a>1，对每帧数据有限制，但又可以暂时超过平均码率，适用于流媒体传输。  
对于某些特殊场景编码，例如电脑屏幕编码，它的特点是I帧纹理细节丰富，编码数据极大，P帧变化很小或者根本没变化，p帧数据很小。  
这种情况，如果不设置vbv参数（保持缺省值0），I帧数据压不下来，码流会周期性的高低变动；有可能造成网络拥塞，数据包丢失，解码端花屏。  
如果设置a=1，I帧得到限制，可以压下来，但编码质量下降太多，主观质量差。  
因此需要根据网络设置成a>1，使得I帧可以暂时有限度的大于平均码率，而P帧编码时候还能把平均码率降下来。  
  
  
      --vbv-init <float>      Initial VBV buffer occupancy [0.9]  
默认：0.9  
说明：设置播放之前必须先载入多少码流到VBV缓冲中。如果值小于1，那么大小就为 vbv-init \* vbv-bufsize。如果大于1，则是以kbits为单位的值。  
参见：–vbv-maxsize,  
ABR时不要设置这个参数，可以完全忽略这个参数。  
  
  
      --crf-max <float>       With CRF+VBV, limit RF to this value May cause VBV underflows!  
默认：无  
说明：这个选项仅用于CRF和VBV同时启用的时候。它限制x264使用大于给定值的ratefactor（太大的CR，质量下降太多），哪怕会违反VBV。一般用于流服务器。  
在bitrate模式下无视它。  
  
  
      --qpmax <integer>       Set max QP [69]  
默认：QP\_MAX，范围[0, QP\_MAX]  
说明：设置x264可以使用的最大量化值。在H.264标准里面8比特编码的最大量化值为51，对应的输出视频质量非常差。一般而言不推荐调节这个值。  
参见：–qpmin, –pbratio, –crf-max  
在bitrate模式下，qpmin <= qp <= qpmax，保持他们的缺省值就可以。  
  
  
      --qpmin <integer>       Set min QP [0]  
默认：0，范围[0, qpmax]  
说明：设置x264可以使用的最小量化器。量化参数越小，输出越接近输入。使用某些值时，x264的输出可以和输入看起来完全一样（虽然其实并不是精确相同的），这样的QP就足够了，没有必要再选用更低的QP。  
如果开启了自适应量化器（默认开启），则不鼓励提高qpmin的值，那样可能会降低帧的平坦部分的质量。  
参见：–qpmax, –ipratio  
  
  
      --qpstep <integer>      Set max QP step [4]  
默认：4，范围[2, QP\_MAX]  
说明：设置2个码率控制单元间QP最大的可变值。码率控制单元可以是一帧，也可以是一行，在ABR+VBV的方法中，控制单元是一行。  
QP与视频质量有大致正比的关系，QP变化过大视频质量变化也会过大，周期性的视频质量变化，会被人类视觉感知，造成不良的视觉感受。  
  
  
      --ratetol <float>       Tolerance of ABR ratecontrol and VBV [1.0]  
默认：1.0， 范围[0.01 - inf], x264里面定义 inf = 1e9  
说明：这个参数有2个可能的含义：  
1、在1次bitrate编码时，这个参数控制x264可以偏离给定平均目标码率的百分比。1.0意味着允许bitrate\*1%比特的偏差，设置为inf是完全关闭码率溢出侦测。有些影片（比如说动作打斗片）在最后的片段里十分复杂。1趟编码并不知道哪里是最复杂的片断，往往到最后比特都已经用完了。把rateol设置为inf就能解决这个问题，它允许编码器用类似 –crf的方式工作，当然文件大小会溢出。  
2、当开启了VBV时（只要使用了任何–vbv-开头的选项就会开启），这个值定义偏离平均目标码率的程度，较大的值意味着允许更大的码率波动。  
较小的值码率控制的精度会更准确一些。建议保持缺省1.0。  
  
  
      --ipratio <float>       QP factor between I and P [1.40]  
默认：1.40，范围[0.01, )  
说明：设置平均I帧QP相比P帧QP的减值。更高的值意味着更高的I帧质量。ipratio与I帧QP成反比例关系。等于1的时候，I帧QP与P帧相同。  
参见：–pbratio  
rc->ip\_offset = 6.0 \* log2f( h->param.rc.f\_ip\_factor );  
rc->qp\_constant[SLICE\_TYPE\_I] = x264\_clip3( h->param.rc.i\_qp\_constant - rc->ip\_offset + 0.5, 0, QP\_MAX );  
  
  
      --pbratio <float>       QP factor between P and B [1.30]  
默认：1.30，范围[0.01, )  
说明：设置平均B帧QP相比P帧QP的增值。更高的值意味着更低的B帧质量。pbratio与B帧QP成正比例关系。等于1的时候，B帧QP与P帧相同。  
开启mbtree（默认开启）选项时不可用，mbtree会自适应地计算B帧量化器。  
参见：–ipratio  
rc->pb\_offset = 6.0 \* log2f( h->param.rc.f\_pb\_factor );  
rc->qp\_constant[SLICE\_TYPE\_B] = x264\_clip3( h->param.rc.i\_qp\_constant + rc->pb\_offset + 0.5, 0, QP\_MAX );  
  
  
      --chroma-qp-offset <integer>  QP difference between chroma and luma [0]  
默认：0，范围[-12, 12]  
说明：设置色度宏块的QP偏移值，chromaQP = LumaQP + chroma-qp-offset。正值则色度选用较大的QP，色度质量被降低；负值色度质量提升。在开启 psy-rd或 psy-trellis时，x264增加整体的quantizers来提升亮度质量，这间接的伤害了色度质量，因此人为的降低这个值对色度质量做一些补偿。（Psy RDO increases overall quantizers to improve the quality of luma--this indirectly hurts chroma quality so we lower the chroma QP offset to compensate）。注意：H.264标准中定义QP<30亮度和色度使用同样的量化器。30<=QP<=51，色度使用比亮度好的量化器。  
  
  
      --aq-mode <integer>     AQ method [1]  
                                  - 0: Disabled  
                                  - 1: Variance AQ (complexity mask)  
                                  - 2: Auto-variance AQ (experimental)  
默认：1  
说明：自适应量化器模式。在CRF和bitrate方法下使用，可以在一帧中不同宏块间重新分配量化值，可以提高细节质量，对于纹理边缘的蚊子噪声等有抑制效果。它有以下选项：  
0 —— 关闭自适应量化  
1 —— 允许自适应量化在所有视频帧内部分配比特。  
2 —— 根据前一帧强度决策的自变量化（实验性的）。  
参见： –aq-strength  
当aq-mode大于0并且subpel\_refine<10. 为了减少qp\_delta比特数，如果当前宏块QP与前一个宏块QP差值不大于1，那么当前宏块采用前一个宏块QP。  
在一些非官方x264编码器中，实现了aq-mod=3的自适应量化算法，该算法在细节上更加出众。  
www.x264.info/2009-08/aq-mode-3/  
自适应量化对视频主观质量的影响是非常大，甚至决定了一个编码器的优劣，“编码器评测结果一次又一次的证明，没有自适应量化的编码器完全没法与具备的编码器相提并论。”开启自适应量化对编码速度的影响并不大；对性能的提升在PSNR上看并不明显，在主观质量上还是有一些细节改善，因此建议保持默认值。  
  
  
      --aq-strength <float>   Reduces blocking and blurring in flat and textured areas. [1.0]  
默认：1.0，范围[0.0, 3.0]  
说明：自适应量化强度。减少平坦区域块效应和纹理区域模糊效应的强度。强度越高，高频信息消减越多，应对平坦区域的块效益就越好，但是对于纹理区域的模糊就越大。一般来说画面动态较高就选低点的aq-strength（0.6-0.8），以免在平坦区域浪费太多码率，而损伤了动态区域；在静态画面较多的场景就选高点aq-strength（>=1.0），防止暗部因比特不够而产生色带。  
一般情况下如果aq-strength=0，强制aq-mode为Disabled。  
在MB-tree的码率控制下，如果aq-mode为Disabled，强制aq-mode为Variance AQ，aq-strength为0。  
参见：–aq-mode  
常用场景预设值：颗粒（Grain） 0.5; 动画 0.6; 缺省 1.0; 静止图片 1.2; touhou 1.3 (东方，处理弹幕场景，弹幕是指视频播放的时候有很多评论文本从屏幕上滑过)  
  
  
  -p, --pass <integer>        Enable multipass ratecontrol  
                                  - 1: First pass, creates stats file  
                                  - 2: Last pass, does not overwrite stats file  
                                  - 3: Nth pass, overwrites stats file  
默认：无  
说明：这对2次编码是一个重要的选项。这个选项控制了x264如何处理 –stats指定的文件。它有三个选项。  
1 —— 建立一个新的stats文件，用于第一趟编码。  
2 —— 读取stats文件，用于最后一趟编码。  
3 —— 读取stats文件，而且更新它。  
stats文件包含了每个输入帧的信息，x264可以读取这些信息来改进输出。由于有前一次编码的结果做参照，编码器会在所有视频序列帧之间更好地分配比特（因此已经知道所有序列中那部分运动剧烈，那部分运动平缓）。从而提高整体视频质量。  
参见：–stats, –bitrate, –slow-firstpass X264\_statsfile  
2次编码的第一遍编码可以采用ABR也可以采用CRF，这两种方式一般情况下是相同的，除非是第一遍编码的码率变化太大，影响第二遍编码时qp-bitrate-cost的预测。因为ABR和CRF使用的码率控制方法其实是一样的，除了ABR在码率快要暴的时候增加了一个QP限制机制。  
  
  
      --stats <string>        Filename for 2 pass stats ["x264\_2pass.log"]  
默认：’x264\_2pass.log’  
说明：设置x264读取和写入的stats文件名  
  
  
      --no-mbtree             Disable mb-tree ratecontrol.  
默认：无  
说明：关闭基于宏块树的码率控制（macroblock tree ratecontrol）。基于宏块树的码率控制通过持续监控宏块在帧间的运动和相对权重来提升视频质量。它会单独生成一个很大的stats文件。  
建议值：保持默认  
参见：–rc-lookahead  
MBtree的基本原理是分配更多的比特给时间轴上平滑的区域。这是基于人眼视觉系统的一个特性：视频变化率越大，人眼的敏感度就越低。也就是人眼可以容忍快速变化场景的一些缺陷，但对于平滑场景的少量误差，人眼也能敏感的察觉到。这里的平滑是时间轴上场景的变化频率，并不是指空间上的平坦区域。而衡量一个区域是否平滑的标准，在MBtree里面是用宏块被参考数来表达。当一个宏块被它后续很多帧中的宏块参考，那么这个宏块在时间轴上被认为是保持平滑的，应该足够重视，多分配些比特。而且如果一个宏块被后续很多帧的宏块参考，那么提升这一个宏块的质量，有助于提升后续很多宏块的质量，同时因为前向参考块的细节度足够，后续这些宏块都可以省出来一些比特，这充分发挥了码率控制比特分配的精髓。  
为了计算各帧宏块之间的参考性，需要做运动估计，为了减少此处计算量，x264采用了在半分辨率（lowres）上做运动估计。  
由于MBtree会统计各帧宏块之间的参考关系，因此在编码之前需要存储满很多帧（MIN(rc-lookahead, keyint)）,这会引起很大的编码麇延时，不利于实时交互的媒体应用，因此在这种场景下，选择关闭MBtree。  
MBtree打开的时候pbratio=1。  
  
  
      --qcomp <float>         QP curve compression [0.60]  
默认：0.60，范围[0.0, 1.0]  
说明：量化器曲线压缩参数。0.0意味着恒定比特率，1.0意味着恒定QP。如果开启了mbtree，这个选项会影响mbtree的强度（更高的值意味着更弱的mbtree， qcomp=1.0， mbtree=0）。  
建议：保持默认  
参见：–cplxblur, –qblur  
  
  
      --cplxblur <float>      Reduce fluctuations in QP (before curve compression) [20.0]  
默认：20，范围>=0  
说明：把给定值作为高斯模糊的半径应用到量化曲线上。这意味着分配给每个帧的QP会被它附近帧的QP平均掉，这样会达到限制QP波动的效果。  
参见：–qcomp, –qblur  
仅用于二次编码的第二遍编码。  
  
  
      --qblur <float>         Reduce fluctuations in QP (after curve compression) [0.5]  
默认：0.5，范围>=0  
说明：在量化曲线压缩后，把给定值作为高斯模糊的半径应用到量化曲线上。这不是一个很重要的选项。  
参见：–qcomp, –cplxblur  
仅用于二次编码的第二遍编码。  
  
  
      --zones <zone0>/<zone1>/...  Tweak the bitrate of regions of the video  
                              Each zone is of the form  
                                  <start frame>,<end frame>,<option>  
                                  where <option> is either  
                                      q=<integer> (force QP)  
                                  or  b=<float> (bitrate multiplier)  
默认：无  
说明：调节视频的特殊片断。可以用它修改大多数x264选项。  
一个单独的zone使用<start frame>,<end frame>,<options>的形式表达。多个zone用’/'分隔。  
<options>  
  
  
      --qpfile <string>       Force frametypes and QPs for some or all frames  
                              Format of each line: framenumber frametype QP  
                              QP is optional (none lets x264 choose). Frametypes: I,i,K,P,B,b.  
                                  K=<I or i> depending on open-gop setting  
                              QPs are restricted by qpmin/qpmax.  
说明：手工设定序列中某些帧的帧类型及QP。格式是’帧序号 帧类型 量化器’。帧类型包括：I（IDR frame, Keyframe）,i (non-IDR I frame, maybe keyframe), K(I or i Keyframe), P（P frame）,B（Referenced B-frame）,b（Non-referenced B-frame）  
例子：  
0 I 18 < IDR (key) I-frame  
1 P 20 < P-frame  
2 B 22 < Referenced B-frame  
3 i 21 < Non-IDR (non-key) I-frame  
4 b 18 < Non-referenced B-frame  
5 K 16 < Keyframe\*  
注意：  
不用指定每一个帧。  
只想设置帧类型时，可以把量化器设置为-1来表明允许x264自行选择优化的QP，  
如果手工指定大量的帧类型和量化器之间间歇穿插着让x264自行决策的部分，则会降低x264的表现。  
‘Keyframe’ 表示可以seek到的关键帧。如果–open-gop设置为none则I,i都是Keyframe，如果open-gop打开（本GOP内的帧B会参考到前面一个GOP内的帧）这样的I帧就不一定是keyframe.  
  
参考  
X264 Settings mewiki.project357.com/wiki/X264\_Settings  
x264 settings x264-settings.wikispaces.com/x264+settings  
 [AVC(H.264)] 抛砖引玉话MBTree www.nmm-hd.org/bbs/thread-906-1-1.html  
【 翻译 】x264参数介绍（一、帧类型和码率控制） blog.yikuyiku.com/?p=2029  
【教程】x264常用参数科普+详解【教你压出高清画质】 tieba.baidu.com/p/3521990720