|  |  |
| --- | --- |
| |  | | --- | | Frame-type options:  --keyint <integer> (x264) **-g <integer>** (FFmpeg) [ No Change on MVs] Keyframe interval, also known as GOP length. This determines the maximum distance between I-frames. Very high GOP lengths will result in slightly more efficient compression, but will make seeking in the video somewhat more difficult. Recommended default: 250  --min-keyint <integer> (x264)  **-keyint\_min <integer>** (FFmpeg)  [ No Change on MVs] Minimum GOP length, the minimum distance between I-frames. Recommended default: 25  --scenecut <integer> (x264)  **-sc\_threshold <integer>** (FFmpeg) [ No Change on MVs] Adjusts the sensitivity of x264's scenecut detection. Rarely needs to be adjusted. Recommended default: 40  --pre-scenecut (x264) **none** (FFmpeg) Slightly faster (but less precise) scenecut detection. Normal scenecut detection decides whether a frame is a scenecut after the frame is encoded, and if so then re-encodes the frame as an I-frame. This is not compatible with threading, however, and so --pre-scenecut is automatically activated when multiple encoding threads are used.  --bframes <integer> (x264) **-bf <integer>** (FFmpeg) B-frames are a core element of H.264 and are more efficient in H.264 than any previous standard. Some specific targets, such as HD-DVD and Blu-Ray, have limitations on the number of consecutive B-frames. Most, however, do not; as a result, there is rarely any negative effect to setting this to the maximum (16) since x264 will, if B-adapt is used, automatically choose the best number of B-frames anyways. This parameter simply serves to limit the max number of B-frames. Note that Baseline Profile, such as that used by iPods, does not support B-frames. Recommended default: 16  --b-adapt <integer> (x264) **-b\_strategy <integer>** (FFmpeg) x264, by default, adaptively decides through a low-resolution lookahead the best number of B-frames to use. It is possible to disable this adaptivity; this is not recommended. Recommended default: 1  0: Very fast, but not recommended. Does not work with pre-scenecut (scenecut must be off to force off b-adapt).  1: Fast, default mode in x264. A good balance between speed and quality.  2: A much slower but more accurate B-frame decision mode that correctly detects fades and generally gives considerably better quality. Its speed gets considerably slower at high bframes values, so its recommended to keep bframes relatively low (perhaps around 3) when using this option. It also may slow down the first pass of x264 when in threaded mode.  --b-bias 0 (x264) **-bframebias 0** (FFmpeg) Make x264 more likely to choose higher numbers of B-frames during the adaptive lookahead. Not generally recommended. Recommended default: 0  --b-pyramid (x264) **-flags2 +bpyramid** (FFmpeg) Allows B-frames to be kept as references. The name is technically misleading, as x264 does not actually use pyramid coding; it simply adds B-references to the normal reference list. B-references get a quantizer halfway between that of a B-frame and P-frame. This setting is generally beneficial, but it increases the DPB (decoding picture buffer) size required for playback, so when encoding for hardware, disabling it may help compatibility.  --no-cabac (x264) **-coder 0** (FFmpeg) [No change on MVs] CABAC is the default entropy encoder used by x264. Though somewhat slower on both the decoding and encoding end, it offers 10-15% improved compression on live-action sources and considerably higher improvements on animated sources, especially at low bitrates. It is also required for the use of trellis quantization. Disabling CABAC may somewhat improve decoding performance, especially at high bitrates. CABAC is not allowed in Baseline Profile. Recommended default: -coder 1 (CABAC enabled)  --ref <integer> (x264)  **-refs <integer>** (FFmpeg) One of H.264's most useful features is the abillity to reference frames other than the one immediately prior to the current frame. This parameter lets one specify how many references can be used, through a maximum of 16. Increasing the number of refs increases the DPB (Decoded Picture Buffer) requirement, which means hardware playback devices will often have strict limits to the number of refs they can handle. In live-action sources, more reference have limited use beyond 4-8, but in cartoon sources up to the maximum value of 16 is often useful. More reference frames require more processing power because every frame is searched by the motion search (except when an early skip decision is made). The slowdown is especially apparent with slower motion estimation methods. Recommended default: -refs 6  --no-deblock (x264)  **-flags -loop**(FFmpeg) Disable loop filter. Recommended default: -flags +loop (Enabled)  --deblock <alpha:beta> (x264)  **-deblockalpha <integer>** (FFmpeg) **-deblockbeta <integer>** (FFmpeg) One of H.264's main features is the in-loop deblocker, which avoids the problem of blocking artifacts disrupting motion estimation. This requires a small amount of decoding CPU, but considerably increases quality in nearly all cases. Its strength may be raised or lowered in order to avoid more artifacts or keep more detail, respectively. Deblock has two parameters: alpha (strength) and beta (threshold). Recommended defaults:-deblockalpha 0 -deblockbeta 0 (Must have '-flags +loop')  --interlaced (x264) **none**(FFmpeg) Enables interlaced encoding. x264's interlaced encoding is not as efficient as its progressive encoding; consider deinterlacing for maximum effectiveness.  Ratecontrol:  --qp <integer> (x264) **-cqp <integer>** (FFmpeg) **[very important to motion values]** Constant quantizer mode. Not exactly constant completely--B-frames and I-frames have different quantizers from P-frames. Generally should not be used, since CRF gives better quality at the same bitrate.  --bitrate <integer> (x264) **-b <integer>** (FFmpeg) Enables target bitrate mode. Attempts to reach a specific bitrate. Should be used in 2-pass mode whenever possible; 1-pass bitrate mode is generally the worst ratecontrol mode x264 has.  --crf <float> (x264) **-crf <float>** (FFmpeg) Constant quality mode (also known as constant ratefactor). Bitrate corresponds approximately to that of constant quantizer, but gives better quality overall at little speed cost. The best one-pass option in x264.  --vbv-maxrate <integer> (x264) **-maxrate <integer>** (FFmpeg) Specifies the maximum bitrate at any point in the video. Requires the VBV buffersize to be set. This option is generally used when encoding for a piece of hardware with bitrate limitations.  --vbv-bufsize <integer> (x264) **-bufsize <integer>** (FFmpeg) Depends on the profile level of the video being encoded. Set only if you're encoding for a hardware device.  --vbv-init <float> (x264) **-rc\_init\_occupancy <float>** (FFmpeg) Initial VBV buffer occupancy. Note: Don't mess with this.  --qpmin <integer> (x264) **-qmin <integer>** (FFmpeg) Minimum quantizer. Doesn't need to be changed. Recommended default: -qmin 10  --qpmax <integer> (x264) **-qmax <integer>** (FFmpeg) Maximum quantizer. Doesn't need to be changed. Recommended default: -qmax 51  --qpstep <integer> (x264) **-qdiff <integer>** (FFmpeg) Set max QP step. Recommended default: -qdiff 4  --ratetol <float> (x264) **-bt <float>** (FFmpeg) Allowed variance of average bitrate  --ipratio <float> (x264) **-i\_qfactor <float>** (FFmpeg) Qscale difference between I-frames and P-frames. Note: -i\_qfactor is handled a little differently than --ipratio. Recommended: -i\_qfactor 0.71  --pbratio <float> (x264) **-b\_qfactor <float>** (FFmpeg) Qscale difference between P-frames and B-frames.  --chroma-qp-offset <integer> (x264) **-chromaoffset <integer>** (FFmpeg) QP difference between chroma and luma.  --aq-strength <float> (x264) **none** (FFmpeg) Adjusts the strength of adaptive quantization. Higher values take more bits away from complex areas and edges and move them towards simpler, flatter areas to maintain fine detail. Default: 1.0  --pass <1,2,3> (x264) **-pass <1,2,3>** (FFmpeg) Used with --bitrate. Pass 1 writes the stats file, pass 2 reads it, and 3 both reads and writes it. If you want to use three pass, this means you will have to use --pass 1 for the first pass, --pass 3 for the second, and --pass 2 or 3 for the third.  --stats <string> (x264) **none** (FFmpeg) Allows setting a specific filename for the firstpass stats file.  --rceq <string> (x264) **-rc\_eq <string>** (FFmpeg) Ratecontrol equation. Recommended default: -rc\_eq 'blurCplx^(1-qComp)'  --qcomp <float> (x264) **-qcomp <float>** (FFmpeg) QP curve compression: 0.0 => CBR, 1.0 => CQP. Recommended default: -qcomp 0.60  --cplxblur <float> (x264) **-complexityblur <float>**(FFmpeg) Reduce fluctuations in QP (before curve compression) [20.0]  --qblur <float> (x264) **-qblur <float>** (FFmpeg) Reduce fluctuations in QP (after curve compression) [0.5]  --zones <zone0>/<zone1> (x264) **none** (FFmpeg) Allows setting a specific quantizer for a specific region of video.  --qpfile (x264) **none** (FFmpeg) Allows one to read in a set of frametypes and quantizers from a file. Useful for testing various encoding options while ensuring the exact same quantizer distribution.  Analysis:  --partitions <string> (x264) **-partitions <string>**(FFmpeg)  p8x8 (x264) /**+partp8x8 (FFmpeg)**  p4x4 (x264) /**+partp4x4 (FFmpeg)**  b8x8 (x264) /**+partb8x8 (FFmpeg)**  i8x8 (x264) /**+parti8x8 (FFmpeg)**  i4x4 (x264) /**+parti4x4 (FFmpeg)**  One of H.264's most useful features is the ability to choose among many combinations of inter and intra partitions. P-macroblocks can be subdivided into 16x8, 8x16, 8x8, 4x8, 8x4, and 4x4 partitions. B-macroblocks can be divided into 16x8, 8x16, and 8x8 partitions. I-macroblocks can be divided into 4x4 or 8x8 partitions. Analyzing more partition options improves quality at the cost of speed. The default is to analyze all partitions except p4x4 (p8x8, i8x8, i4x4, b8x8), since p4x4 is not particularly useful except at high bitrates and lower resolutions. Note that i8x8 requires 8x8dct, and is therefore a High Profile-only partition. p8x8 is the most costly, speed-wise, of the partitions, but also gives the most benefit. Generally, whenever possible, all partition types except p4x4 should be used.  --direct <integer> (x264) **-directpred <integer>** (FFmpeg) B-frames in H.264 can choose between spatial and temporal prediction mode. Auto allows x264 to pick the best of these; the heuristic used is whichever mode allows more skip macroblocks. Auto should generally be used.  --weightb (x264) **-flags2 +wpred** (FFmpeg) This allows B-frames to use weighted prediction options other than the default. There is no real speed cost for this, so it should always be enabled.  --me <dia,hex,umh,esa> (x264) **-me\_method <epzs,hex,umh,full>** (FFmpeg)  dia (x264) / **epzs (FFmpeg)** is the simplest search, consisting of starting at the best predictor, checking the motion vectors at one pixel upwards, left, down, and to the right, picking the best, and repeating the process until it no longer finds any better motion vector.  hex (x264) / **hex (FFmpeg)** consists of a similar strategy, except it uses a range-2 search of 6 surrounding points, thus the name. It is considerably more efficient than DIA and hardly any slower, and therefore makes a good choice for general-use encoding.  umh (x264) / **umh (FFmpeg)** is considerably slower than HEX, but searches a complex multi-hexagon pattern in order to avoid missing harder-to-find motion vectors. Unlike HEX and DIA, the merange parameter directly controls UMH's search radius, allowing one to increase or decrease the size of the wide search.  esa (x264) / **full (FFmpeg)** is a highly optimized intelligent search of the entire motion search space within merange of the best predictor. It is mathematically equivalent to the bruteforce method of searching every single motion vector in that area, though faster. However, it is still considerably slower than UMH, with not too much benefit, so is not particularly useful for everyday encoding.  One of the most important settings for x264, both speed and quality-wise.  --merange <integer> (x264) **-me\_range <integer>** (FFmpeg) MErange controls the max range of the motion search. For HEX and DIA, this is clamped to between 4 and 16, with a default of 16. For UMH and ESA, it can be increased beyond the default 16 to allow for a wider-range motion search, which is useful on HD footage and for high-motion footage. Note that for UMH and ESA, increasing MErange will significantly slow down encoding.  --mvrange (x264) **none** (FFmpeg) Limits the maximum motion vector range. Since x264 by default limits this to 511.75 for standards compliance, this should not be changed.  --subme 6 (x264) **-subq 6** (FFmpeg)  1: Fastest, but extremely low quality. Should be avoided except on first pass encoding.  2-5: Progressively better and slower, 5 serves as a good medium for higher speed encoding.  6-7: 6 is the default. Activates rate-distortion optimization for partition decision. This can considerably improve efficiency, though it has a notable speed cost. 6 activates it in I/P frames, and subme7 activates it in B frames.  8-9: Activates rate-distortion refinement, which uses RDO to refine both motion vectors and intra prediction modes. Slower than subme 6, but again, more efficient.  An extremely important encoding parameter which determines what algorithms are used for both subpixel motion searching and partition decision.  --psy-rd <float>:<float> (x264) **none** (FFmpeg) First value represents the amount that x264 biases in favor of detail retention instead of max PSNR in mode decision. Requires subme >= 6. Second value is psy-trellis, an experimental algorithm that tries to improve sharpness and detail retention at the expense of more artifacting. Recommended starting values are 0.1-0.2. Requires trellis >= 1. Recommended default: 1.0:0.0  --mixed-refs (x264) **-flags2 +mixed\_refs** (FFmpeg) H.264 allows p8x8 blocks to select different references for each p8x8 block. This option allows this analysis to be done, and boosts quality with little speed impact. It should generally be used, though it obviously has no effect with only one reference frame.  --no-chroma-me (x264) **none** (FFmpeg) Chroma is used in the last steps of the subpixel refinement by default. For a slight speed increase, this can be disabled (at the cost of quality).  --8x8dct (x264) **-flags2 +dct8x8** (FFmpeg) Gives a notable quality boost by allowing x264 to choose between 8x8 and 4x4 frequency transform size. Required for i8x8 partitions. Speed cost for this option is near-zero both for encoding and decoding; the only reason to disable it is when one needs support on a device not compatible with High Profile.  --trellis <0,1,2> (x264) **-trellis <0,1,2>** (FFmpeg)  0: disabled  1: enabled only on the final encode of a MB  2: enabled on all mode decisions  The main decision made in quantization is which coefficients to round up and which to round down. Trellis chooses the optimal rounding choices for the maximum rate-distortion score, to maximize PSNR relative to bitrate. This generally increases quality relative to bitrate by about 5% for a somewhat small speed cost. It should generally be enabled. Note that trellis requires CABAC.  --no-fast-pskip (x264) **-flags2 -fastpskip** (FFmpeg) By default, x264 will skip macroblocks in P-frames that don't appear to have changed enough between two frames to justify encoding the difference. This considerably speeds up encoding. However, for a slight quality boost, P-skip can be disabled. In this case, the full analysis will be done on all P-blocks, and the only skips in the output stream will be the blocks whose motion vectors happen to match that of the skip vector and motion vectors happen to match that of the skip vector and which have no residual. The speed cost of enabling no-fast-pskip is relatively high, especially with many reference frames. There is a similar B-skip internal to x264, which is why B-frames generally encode much faster than P-frames, but it cannot be disabled on the commandline.  --no-dct-decimate(x264) **none**(FFmpeg) By default, x264 will decimate (remove all coefficients from) P-blocks that are extremely close to empty of coefficents. This can improve overall efficiency with little visual cost, but may work against an attempt to retain grain or similar. DCT decimation should be left on unless there's a good reason to disable it.  --nr(x264) **none**(FFmpeg) a fast, built-in noise reduction routine. Not as effective as external filters such as hqdn3d, but faster. Since x264 already naturally reduces noise through its quantization process, this parameter is not usually necessary.  --deadzone-inter (264) --deadzone-intra (x264) **none** (FFmpeg) **none** (FFmpeg) When trellis isn't activated, deadzone parameters determine how many DCT coefficients are rounded up or down. Rounding up results in higher quality and more detail retention, but costs more bits--so rounding is a balance between quality and bit cost. Lowering these settings will result in more coefficients being rounded up, and raising the settings will result in more coefficients being rounded down. Recommended: keep them at the defaults.  --cqm (264) --cqpfile (x264) **none** (FFmpeg) **none** (FFmpeg) Allows the use of a custom quantization matrix to weight frequencies differently in the quantization process. The presets quant matrices are "jvt" and "flat". --cqpfile reads a custom quant matrices from a JM-compatible file. Recommended only if you know what you're doing. | |