wiki: Encode / AV1

Login Preferences Help/Guide About Trac

Contents

Two-Pass

Constant Quality

Constrained Quality

Average Bitrate (ABR)

Keyframe placement

Lossless encoding

Presets and tunes

Keyframe placement

Film grain synthesis

Keyframe placement

Additional Resources

HDR and high bit depth

Controlling Speed / Quality

libaom

SVT-AV1

CRF

rav1e

AMD AMF AV1

Usage

Quality

Vbaq

Align

Enforce_hrd

WIKI

TIMELINE

VIEW TICKETS

Register Start Page Index

Forgot your password? SEARCH **TAGS**

Search

History

AV1 Video Encoding Guide

→ AV1 is an open source & royalty-free video codec developed by the → Alliance for Open Media (AOMedia), a non-profit industry consortium. Depending on the use case, AV1 can achieve about 30% higher compression efficiency than VP9, and about 50% higher efficiency than H.264.

There are currently three AV1 encoders supported by FFmpeg: libaom (invoked with libaom-av1 in FFmpeg), SVT-AV1 (libsvtav1), and rav1e (librav1e). This guide currently focuses on libaom and SVT-AV1. libaom

libaom (libaom-av1) is the reference encoder for the AV1 format. It was also used

for research during the development of AV1. <a>libaom is based on <a>libvpx and thus shares many of its characteristics in terms of features, performance, and usage. To install FFmpeg with support for [libaom-av1], look at the Compilation Guides and

compile FFmpeg with the [--enable-libaom] option.

libaom offers the following rate-control modes which determine the quality and file size obtained:

Constant quality

 Constrained quality • 2-pass average bitrate

• 1-pass average bitrate For a list of options, run [ffmpeg -h encoder=libaom-av1] or check FFmpeg's online documentation. For options that

currently no official online reference for them. **Note:** Users of libaom older than version 2.0.0 will need to add -strict experimental (or the alias -strict -2).

can be passed via -aom-params, checking the --help output of the aomenc application is recommended, as there is

Constant Quality

number of bits it deserves to achieve a certain (perceptual) quality level, rather than encoding each frame to meet a bit rate target. This results in better overall quality. If you do not need to achieve a fixed target file size, this should be your method of choice.

To trigger this mode, simply use the [-crf] switch along with the desired numerical value. ffmpeg -i input.mp4 -c:v libaom-av1 -crf 30 av1_test.mkv

this is not done, the **-crf** switch triggers the constrained quality mode with a default bitrate of 256kbps.

libaom-av1 has a constant quality (CQ) mode (like CRF in x264 and x265) which will ensure that every frame gets the

The CRF value can be from 0–63. Lower values mean better quality and greater file size. 0 means lossless. A CRF value of

23 yields a quality level corresponding to CRF 19 for x264 (⇒ source), which would be considered visually lossless. Note that in FFmpeg versions prior to 4.3, triggering the CRF mode also requires setting the bitrate to 0 with [-b:v 0]. If

Constrained Quality libaom-av1 also has a constrained quality (CQ) mode that will ensure that a constant (perceptual) quality is reached

while keeping the bitrate below a specified upper bound or within a certain bound. This method is useful for bulk encoding videos in a generally consistent fashion.

ffmpeg -i input.mp4 -c:v libaom-av1 -crf 30 -b:v 2000k output.mkv

The quality is determined by the [-crf], and the bitrate limit by the [-b:v] where the bitrate MUST be non-zero. You can also specify a minimum and maximum bitrate instead of a quality target:

Note: When muxing into MP4, you may want to add [-movflags +faststart] to the output parameters if the intended use for the resulting file is streaming.

Two-Pass

target bitrate. For two-pass, you need to run [ffmpeg] twice, with almost the same settings, except for:

ffmpeg -i input.mp4 -c:v libaom-av1 -minrate 500k -b:v 2000k -maxrate 2500k output.mp4

In order to create more efficient encodes when a particular target bitrate should be reached, you should choose two-

• In pass 1 and 2, use the [-pass 1] and [-pass 2] options, respectively. • In pass 1, output to a null file descriptor, not an actual file. (This will generate a logfile that FFmpeg needs for the

second pass.)

ffmpeg -i input.mp4 -c:v libaom-av1 -b:v 2M -pass 1 -an -f null /dev/null && \ ffmpeg -i input.mp4 -c:v libaom-av1 -b:v 2M -pass 2 -c:a libopus output.mkv

pass encoding. Two-pass encoding is also beneficial for encoding efficiency when constant quality is used without a

Note: Windows users should use NUL instead of /dev/null and ^ instead of \

• In pass 1, you can leave audio out by specifying [-an].

ffmpeg -i input.mp4 -c:v libaom-av1 -b:v 2M output.mkv

libaom-av1 also offers a simple "Average Bitrate" or "Target Bitrate" mode. In this mode, it will simply try to reach the specified bit rate on average, e.g. 2 MBit/s.

the other rate control methods described above.

Controlling Speed / Quality

threads than the number of encoded tiles.

presets higher than 8 cannot be used via FFmpeq).

Average Bitrate (ABR)

quality, and vice-versa. Valid values are from 0 to 8 inclusive. -row-mt 1 enables row-based multi-threading which maximizes CPU usage. To enable fast decoding performance, also add tiles (i.e. -tiles 4x1) or -tiles 2x2 for 4 tiles). Enabling row-mt is only faster when the CPU has more

-cpu-used sets how efficient the compression will be. Default is 1. Lower values mean slower encoding with better

Use this option only if file size and encoding time are more important factors than quality alone. Otherwise, use one of

Keyframe placement

-usage realtime activates the realtime mode, meant for live encoding use cases (livestreaming, videoconferencing,

etc). [-cpu-used] values between 7-10 are only available in the realtime mode (though due to ⇒a bug in FFmpeg,

By default, libaom's maximum keyframe interval is 9999 frames. This can lead to slow seeking, especially with content that has few or infrequent scene changes. The [-g] option can be used to set the maximum keyframe interval. Anything up to 10 seconds is considered reasonable for most content, so for 30 frames per second content one would use [-g 300], for 60 fps content [-g 600], etc.

To set a fixed keyframe interval, set both [-g] and [-keyint_min] to the same value. Note that currently [-keyint_min]

is ignored unless it's the same as [-g], so the minimum keyframe interval can't be set on its own.

HDR and high bit depth

incompatible hardware decoders.

color_primaries | For example, Youtube HDR uses

For intra-only output, use -g 0.

-colorspace bt2020nc -color_trc smpte2084 -color_primaries bt2020 AV1 includes 10-bit support in its Main profile. Thus content can be encoded in 10-bit without having to worry about

When encoding in HDR it's necessary to pass through color information; [-colorspace], [-color_trc] and [-

ffmpeg -help encoder=libaom-av1 for the supported pixel formats.

To utilize 10-bit in the Main profile, use -pix_fmt yuv420p10le. For 10-bit with 4:4:4 chroma subsampling (requires

the High profile), use -pix_fmt_yuv444p10le. 12-bit is also supported, but requires the Professional profile. See

Use -crf 0 for lossless encoding. Because of a bug present in FFmpeg versions prior to 4.4, the first frame will not be losslessly preserved (the issue was fixed on → March 21, 2021). As a workaround on pre-4.4 versions one may use -aomparams lossless=1 for lossless output.

SVT-AV1 (libsvtav1) is an encoder originally developed by Intel in collaboration with Netflix. In 2020, SVT-AV1 was adopted by AOMedia as the basis for the future development of AV1 as well as future codec efforts. The encoder

guide and is list of all parameters.

size. Lossless encoding is currently mot supported.

To check vmaf, ssim, psnr check [→] ffmetrics.

Presets and tunes

presets are 0 to 8.

supported since FFmpeg 5.1.

SVT-AV1

Lossless encoding

supports a wide range of speed-efficiency tradeoffs and scales fairly well across many CPU cores. To enable support, FFmpeg needs to be built with [--enable-libsvtav1]. For options available in your specific build of FFmpeg, see | ffmpeg -help encoder=libsvtav1|. See also ⇒ FFmpeg documentation, the upstream ⇒ encoder user

Many options are passed to the encoder with -svtav1-params. This was introduced in SVT-AV1 0.9.1 and has been

CRF is the default rate control method, but VBR and CBR are also available. **CRF**

Much like CRF in x264 and x265, this rate control method tries to ensure that every frame gets the number of bits it deserves to achieve a certain (perceptual) quality level.

ffmpeg -i input.mp4 -c:v libsvtav1 -crf 35 svtav1_test.mp4

For example:

Note that the -crf option is only supported in FFmpeg git builds since 2022-02-24. In versions prior to this, the CRF value is set with -qp.

The valid CRF value range is 0-63, with the default being 30. Lower values correspond to higher quality and greater file

CRF of 38 is usually enough for a VMAF of at least 95% and a good enough PSNR for source files of 4k-8k resolution.

for more information For source files of 1080p and 720p av1 otimization is not optimal, most of the time is better to go with the H265 for better vmaf/size ratio.

The trade-off between encoding speed and compression efficiency is managed with the [-preset] option. Since SVT-AV1 0.9.0, supported presets range from 0 to 13, with higher numbers providing a higher encoding speed. Note that preset 13 is only meant for debugging and running fast convex-hull encoding. In versions prior to 0.9.0, valid

Since SVT-AV1 0.9.1, the encoder also supports tuning for visual quality (sharpness). This is invoked with -svtav1-

In 0.9.1, the option accepts an integer from 1 to 3, with higher numbers resulting in easier-to-decode video. In 0.9.1,

decoder tuning is only supported for presets from 5 to 10, and the level of decoder tuning varies between presets.

Also supported since 0.9.1 is tuning the encoder to produce bitstreams that are faster (less CPU intensive) to decode, similar to the fastdecode tune in x264 and x265. Since SVT-AV1 1.0.0, this feature is invoked with -svtav1-params fast-decode=1.

As an example, this command encodes a video using preset 8 and a CRF of 35 while copying the audio:

ffmpeg -i input.mp4 -c:a copy -c:v libsvtav1 -preset 8 -crf 35 svtav1_test.mp4

params tune=0. The default value is 1, which tunes the encoder for PSNR.

By default, SVT-AV1's keyframe interval is 2-3 seconds, which is quite short for most use cases. Consider changing this up to 5 seconds (or higher) with the [-g] option (or keyint in svtav1-params); [-g 120] for 24 fps content, [-g 150] for 30 fps, etc.

Note that as of version 1.2.1, SVT-AV1 does not support inserting keyframes at scene changes. Instead, keyframes are

placed at set intervals. In SVT-AV1 0.9.1 and prior, the functionality was present but considered to be in a suboptimal

SVT-AV1 supports film grain synthesis, an AV1 feature for preserving the look of grainy video while spending very little bitrate to do so. The grain is removed from the image with denoising, its look is approximated and synthesized, and then added on top of the video at decode-time as a filter.

turning this off will lead to the original frames to be used instead.

video encoding scenarios and device requirements.

typical application scenarios, including:

Encoding profile and level

• Deblocking filter strength

• Rate control mode and strategy

• Bitrate and resolution constraints

• Motion estimation method and precision

• Adaptive quantization and rate distortion optimization

• GOP size and structure

• Multi-Pass encoding

Film grain synthesis

state and was disabled by default.

Keyframe placement

Higher numbers correspond to higher levels of denoising for the grain synthesis process and thus a higher amount of grain. The grain denoising process can remove detail as well, especially at the high values that are required to preserve the look of very grainy films. This can be mitigated with the film-grain-denoise=0 option, passed via svtav1-params.

While by default the denoised frames are passed on to be encoded as the final pictures (film-grain-denoise=1),

librav1e is the Xiph encoder for AV1. Compile with --enable-librav1e. See → FFmpeg doc and → upstream CLI

Rav1e claims to be the fastest software AV1 encoder, but that really depends on the setting.

The film grain synthesis feature is invoked with -svtav1-params film-grain=X, where X is an integer from 1 to 50.

AMD AMF AV1 The Advanced Media Framework (AMF) provides developers with optimal access to AMD GPU for multimedia processing. AMD AMF AV1 Encoder is a professional video encoder that provides powerful video encoding capabilities and a wide range of customization options. It is designed to meet the individual needs of different users. Users can

Usage The video encoder balances factors such as speed, quality, and latency. AMD has integrated several typical user scenario

rav1e

options.

• transcoding: Convert high-resolution or high-bitrate videos to low-resolution or low-bitrate videos for transmission or storage in bandwidth-limited network environments. • lowlatency: For video streaming live applications, lower latency and higher video quality are required.

parameter optimizations and presets cover the majority of the parameters, including but not limited to:

For each usage, AMF has optimized and preset the encoder's parameters based on the corresponding scenario. These

presets into the AMF encoder. Users can use these presets by setting the "usage" parameter. Usage parameter supports

- By using these presets, users can easily and efficiently select the appropriate encoding settings for their specific usage scenario without the need for in-depth knowledge of the encoder's parameters and their impact on video quality and performance. The usage scenario for transcoding
- ffmpeg -s 1920x1080 -pix_fmt yuv420p -i input.yuv -c:v h264_amf -usage transcoding output.mp4 The usage scenario for lowlatency ffmpeg -s 1920x1080 -pix_fmt yuv420p -i input.yuv -c:v av1_amf -usage lowlatency output.mp4
- applications that require a balance between the two, such as video conferencing and online gaming. • speed: This preset prioritizes speed over quality, making it suitable for applications that require real-time video encoding with low latency, such as online gaming and remote desktop applications. ffmpeg -i input.mp4 -c:v av1_amf -quality balanced output.mp4

• balanced: This preset balances the trade-off between quality and speed, making it suitable for a variety of

The Hypothetical Reference Decoder (HRD) helps to prevent buffer overflow and underflow, which can cause issues such as stuttering or freezing in the video playback. HRD may sacrifice a certain level of image quality. "enforce_hrd" parameter is not always necessary or appropriate for all types of scenario. It should be used selectively and with careful consideration of the specific characteristics of the video content being encoded.

Vbaq VBAQ is a technique used to improve the visual quality of the encoded video. It achieves this by adapting the

Align AV1 bitstream specification does not contain the cropping information for decoders to display the specific, pixel accurate resolution. It is expected that the proper cropping information should be presented in the container instead. AMF AV1 Encoder introduces parameter "align" to address the hardware alignment requirement such that the encoded

- resolution of 1920x1082. Two extra lines are padded at the bottom of the frame, filled with black pixels. • None: Videos with any resolution can be coded. However, for those videos whose resolution is not 64x16 aligned, their output resolution will be extrapolated to be 64x16 aligned and padded with black pixels. The exception is for resolution of 1080p, which will be padded to 1082p, as in the case of value "1080p".
- By default, AMF AV1's keyframe interval is 250 frames, which is a balanced value for most use cases. The "-g" option can be used to set the keyframe interval. For example, in broadcast television applications, it is typically desired to have a comfortable channel switching time for a good user experience. A 2-second keyframe is widely used as a common

Keyframe placement

Additional Resources

• ➡ SVT-AV1's FFmpeg guide • ➡SVT-AV1's user guide ➡SVT-AV1 issue tracker

• libaom issue tracker

- ➡rav1e issue tracker

Tags

av1_amf

encoder

libsvtav1

video

• libaom-av1

trac

Powered by **Trac 1.4.2**

By Edgewall Software.

av1

Visit the Trac open source project a

Last modified 3 weeks ago

http://trac.edgewall.org

adjust the encoder's parameter settings to meet different encoding requirements, such as resolution, bit rate, frame rate, encoding quality, and more. These parameter settings can be customized based on users' needs to meet different

Quality

Enforce_hrd

This parameter is used to select between video quality and speed. This parameter has a significant impact on encoding speed. It has three valid values: • quality: This preset is optimized for high-quality video output, suitable for applications such as video production, broadcasting, and live streaming.

ffmpeg -i input.mp4 -c:v av1_amf -quality quality output.mp4

ffmpeg -i input.mp4 -c:v av1_amf -quality speed output.mp4

ffmpeg -i input.mp4 -c:v av1_amf -enforce_hrd true output.mp4

quantization parameters for blocks based on the visual complexity of the content. It is particularly effective for encoding video with complex visual content, such as high-motion or high-detail scenes. ffmpeg -i input.mp4 -c:v av1_amf -vbaq true output.mp4

bitstream could be decoded and presented properly. Values for setting "align":

ffmpeg -i input.mp4 -c:v av1_amf -align 1080p output.mp4

• 64x16: Input videos whose resolution is aligned to 64x16 will be coded; input videos whose resolution is not aligned to 64x16 will not be coded; All other resolution videos will not be supported. • 1080p: Input videos whose resolution is aligned to 64x16, as well as 1920x1080 video, will be coded; All other resolution videos will not be supported. Note that for resolution of 1920x1080, the output video would have a

setting for this purpose. So, for content with a frame rate of 30 frames per second, one would use the command "-g 60". ffmpeg -i input.mp4 -c:v av1_amf -g 60 output.mp4

 SVT-AV1 parameters documentation • SVT-AV1: Common Questions and Topics of Interest

Note: See TracWiki for help on using the wiki. **Download in other formats:** Plain Text