MAKING THE MOST OF AVC: HOW TO SET UP ENCODER FOR OPTIMUM RESULTS

WHITE PAPER FROM ELECARD

August 2020

Contents

Company overview

9700 companies

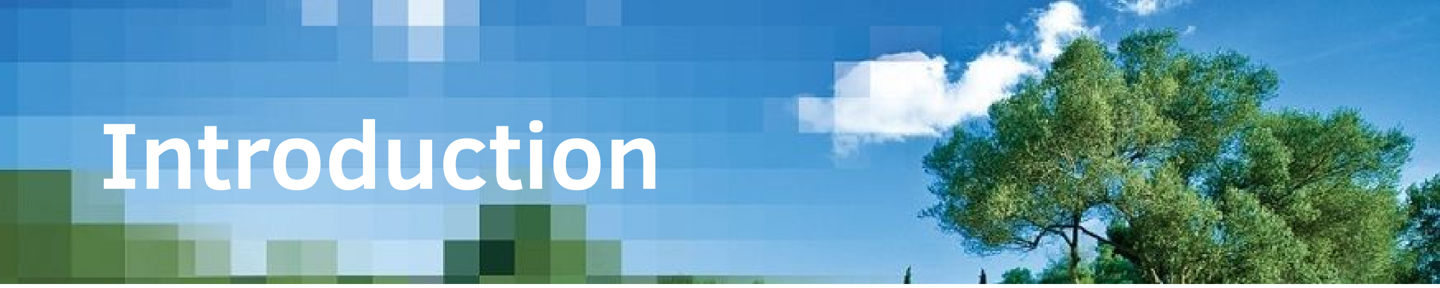
150 countries

20M users

Elecard company, founded in 1988, is a leading provider of software products for encoding, decoding, processing, monitoring and analysis of video and audio data in various formats.

Elecard is a vendor of professional software products and software development kits (SDKs); products for in-depth high-quality analysis and monitoring of the media content; solutions for IPTV and OTT projects, digital TV broadcasting and video streaming; transcoding servers. Elecard is based in the United States, Russia, and China with headquarters located in Tomsk, Russia.

Elecard products are highly appreciated and widely used by the leaders of IT industry such as Intel, Cisco, Netflix, Huawei, Blackmagic Design, etc. For more information, please visit www.elecard.com.



Introduction

Video compression is the key step in video processing. Compression allows broadcasters and premium TV providers to deliver their content to their audience.

Many video compression standards currently exist in TV broadcasting. Each standard has different properties, some of which are better suited to traditional live TV while others are more suited to video on demand (VoD).

Two basic standards can be identified in the history of video compression:

- MPEG-2, a legacy codec used for SD video and early digital broadcasting. It is still relevant due to its support in old set-top boxes.
- H.264, also referred to as Advanced Video Coding (AVC), that was created with the aim of transitioning to HD video. With growth of the Internet, H.264/AVC came to be used for over-the-top (OTT) delivery using the HTTP protocol.

Both standards were developed under supervision of the International Telecommunications Union (ITU). They are mainly geared toward the broadcasting market and have established themselves in the industry.

Today, more advanced coding standards are available on the market such as HEVC, AV1, and VP9. They improve compression efficiency by up to 50% compared to AVC but are sometimes difficult to integrate and use. HEVC was considered a successor to AVC, but limited uptake of 4K video and complex licensing mechanism have held back its implementation considerably. In addition, newer and more efficient standards such as VVC and EVC are almost ready to be released.

In future, there will be several dominant players in this market niche. For example, AV1 and VP9 may consolidate their position as the "codecs of choice" for streaming media distribution while AVC/H.264 and HEVC may remain as the primary TV broadcasting codecs. That being said, each broadcaster or premium TV provider has its unique infrastructure, which means any single codec is unlikely to ever emerge as a clear winner here.

Broadcasters naturally want to use codecs based on more advanced standards. However, large numbers of people still consume SD and HD content, making it necessary to invest in updating the old codecs so that they meet the demand for quality of content required by most of the audience. It has been a good while since AVC emerged, but it is still relevant and considered a "gold standard" in TV broadcasting and OTT services. AVC has enough parameters to cover all types of end-user devices.

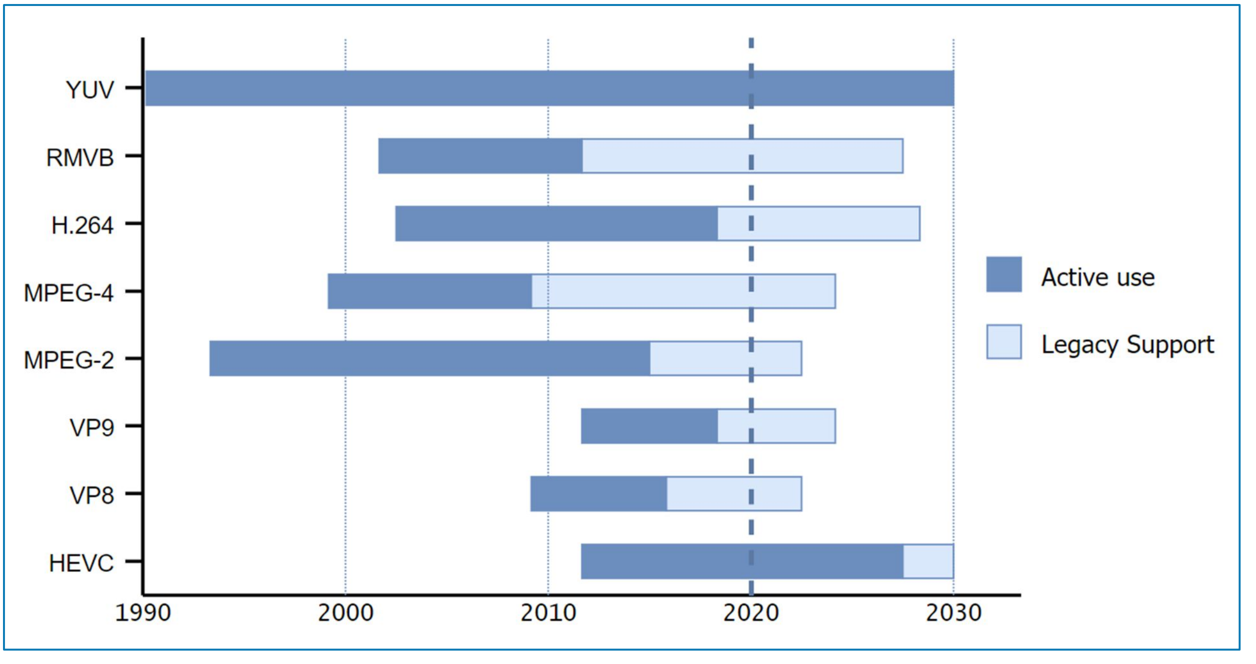


Fig. 1 Development of video compression standards. The current status of AVC/H.264 is shown according to ITU-T Recommendations for HDTV (709) and UHD (2020).

To achieve the best video quality, you need to set up the encoder carefully for your specific application. This is difficult to do, given that one popular AVC library has more than 200 parameters. Most encoders offer configuration presets, and as a last resort, you can click the Default button. But if you broadcast highly dynamic content, such as a football match, it is not recommended to rely on presets. You need to set up the encoder manually yourself or have professionals help you to do it.

This article will briefly describe key aspects of the H.264/AVC standard and consider the most important parameters for a correct encoder setup in TV broadcasting.

What is AVC

MPEG-4 AVC, also known as MPEG-4 Part 10 or H.264, was produced by the Joint Video Team (JVT), a joint project between the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. The first specification of this standard was published in 2003.

AVC is now one of the most popular video codecs used for streaming and recording. It has diverse applications, including:

- DVB/ATSC digital broadcasting
- Cable TV services
- Video conferencing
- Video surveillance
- Blu-Ray disc recording
- Other areas in the digital video field.

Advantages of AVC leading to market dominance

- High quality at low bitrates. This is the most crucial factor for all providers. With AVC, it is possible to carry two channels where MPEG-2 only allowed one. This lowers the cost of broadcasting considerably and allows acceptable broadcasting quality with limited bandwidth.
- Fault tolerance, making it possible to reproduce the picture despite errors in data transmission through various networks.
- "One-stop" licensing model.
- Low implementation cost and wide choice of supporting devices. This advantage is a consequence of the previous one: thanks to simple licensing, many device manufacturers implemented support for the standard. This improved compatibility between AVC-based solutions and appeared commercially attractive.

Making sense of profiles, levels, and bitrate

The standard provides capabilities for video compression in terms of profiles (sets of algorithmic parameters) and levels (classes of operation). The profile and level combined determine the quality, bitrate, resolution, and computational requirements for the encoding device. Compared to other standards, AVC has fewer options and parameters — the engineers made every effort to create a simple and straightforward solution.

Profile

A profile is a subset of the standard that is focused on a specific application and specifies the algorithms and tools for compression. AVC includes seven main profiles, each intended for a different application. Profile defines a set of algorithms that the encoder can use and limits complexity of decoder implementation.

Conformance to a certain profile and level is specified in the syntax element known as the Sequence Parameter Set (SPS). Decoders use this information to determine their capabilities in advance and allocate the required amount of memory even before receiving any frames.

The higher the profile, the better the quality is and the more demanding the computational requirements are (at the same bitrates).

Level

Levels within each profile are related to parameters of the picture being encoded. AVC has 11 levels that determine the required functionality, bandwidth, and memory. The level specifies the data rate, coding rate in macroblocks per second, maximum picture resolution, and frame rate.

The coding level is specified in the header of a video stream and determines the encoder performance requirements. A decoder must process all bitstreams encoded for this level and all lower levels. The higher the level, the higher the resolution and the number of frames per second.

Bitrate

The bitrate is a fundamental parameter of video compression. It expresses the overall compression ratio of the stream and therefore determines the required data channel bandwidth. The higher the bitrate, the more picture details can be preserved and the more realistic the resulting video appears. Choosing a bitrate always involves a trade-off between encoding quality and channel bandwidth.

There are two basic approaches to distributing the bits within a stream: constant bitrate (CBR) and variable bitrate (VBR).

Engineers can mean two different things by the terms constant or variable bitrate. In the context of broadcasting, it means the constant or variable number of bits transmitted per second. In the context of coding, it means the constant or variable rate at which the buffer is being filled or emptied.

CBR

CBR shows a stable bitrate level regardless of the scene and does not cause abrupt load changes. It is not optimal in terms of quality, because the stream does not change depending on dynamics and complexity of video.

- CBR is ideal for data transmitted over limited-bandwidth multimedia channels because it makes it possible to use full capabilities of such channels.
- This mode affects quality when encoding complex (highly dynamic) fragments that normally require more space.
- Static video fragments are encoded using the same number of allocated bits, and available channel bandwidth is wasted.

VBR

VBR allows increase in bitrates and reduction of compression ratio in complex scenes to achieve better picture quality. In this mode, the bitrate is adjusted depending on the rate at which the picture changes. For example, a static video fragment (sunset) would have much lower bitrate than a fragment that features fast action (car chase).

- VBR responds only to activity that occurs in the frame and enables considerable disk space savings when working with files.
- This mode makes it possible to achieve much better picture quality than CBR at smaller encoded file size.
- VBR is well-suited to Internet broadcasting (OTT).
- As activity in the frame intensifies, the bitrate increases correspondingly so that the system throughput can approach the limit very quickly. This can cause lots of artifacts and distortions in the resulting picture or even complete loss of the picture.
- The upcoming changes are hard to predict. The bitrate increase takes place with a time lag during recording.

Tips & Tricks

- Do not rely on automatic modes; specify the profile and level manually in encoder settings. If possible, make sure that your selected values meet the guidelines provided by the authors of the standard in the Profile-Level-Bitrate tables.
- Decoders track profile and level very closely and allocate specific amount of memory to decode the stream based on their values. If the decoder does not support a certain profile or level, it will most likely skip playback of that video sequence.
- Any video contains headers that define general characteristics of video, such as profile, level, resolution, frame rate, bitrate, etc. To verify operation of the encoder, you can check if these headers match the set parameter values.
- Sometimes during direct broadcasting over a low-bandwidth channel (CBR), bitrate deviations up to ~10% are allowed. Use video stream analysis software to check if the encoder is using correct bitrate.

- To see which profile and level are set for an elementary stream, look at parametric set of a sequence (seq_parameter_set).

name	value
stream type	AVC/H.264
profile	Main
level	5.2

name	value
[0] seq_parameter_set()	
profile_idc	77 (Main)
constraint_set0_flag	0 (false)
constraint_set1_flag	1 (true)
constraint_set2_flag	0 (false)
constraint_set3_flag	0 (false)
constraint_set4_flag	0 (false)
constraint_set5_flag	0 (false)
reserved_zero_2bits	0
level_idc	52

Fig. 2 Seq_parameter_set in a stream header.

Group of pictures and its structure

A group of pictures (GOP) is a set of sequential pictures that defines the order in which intra (I) and inter (P and B) frames appear.

Often a GOP is denoted using two numbers, such as M = 3, N = 12. M specifies the distance between two reference frames (I or P) and N the distance between two full pictures (I frames). As an example, for M = 3 and N = 12, the GOP will have the following structure: IBBPBBPBBPBBI.

I (IDR) frames

I frames are compressed independently of any other frames in the video sequence. The IDR-frame, also referred to as a key frame, is a subtype of I frame. It is the frame at which decoding of the entire stream begins. No frames located between two IDR frames can reference any frames outside this interval.

Sometimes, when the scene view changes, the current and previous frames differ so much that it is more beneficial to use an I frame instead of a P or B frame at the beginning of a new scene. Encoders are capable of responding to such changes—this capability is called scene change detection (SCD).

P и B frames

P and B frames are used to encode the changes in the current frame relative to the preceding frames. The most versatile structure of a P and B frame sequence is two to three B frames per P frame.

B frames are usually a fraction of the size of a P frame, and each B frame adds latency because of buffering and frame reordering. The greater the number of P and B frames used, the higher the compression ratio.

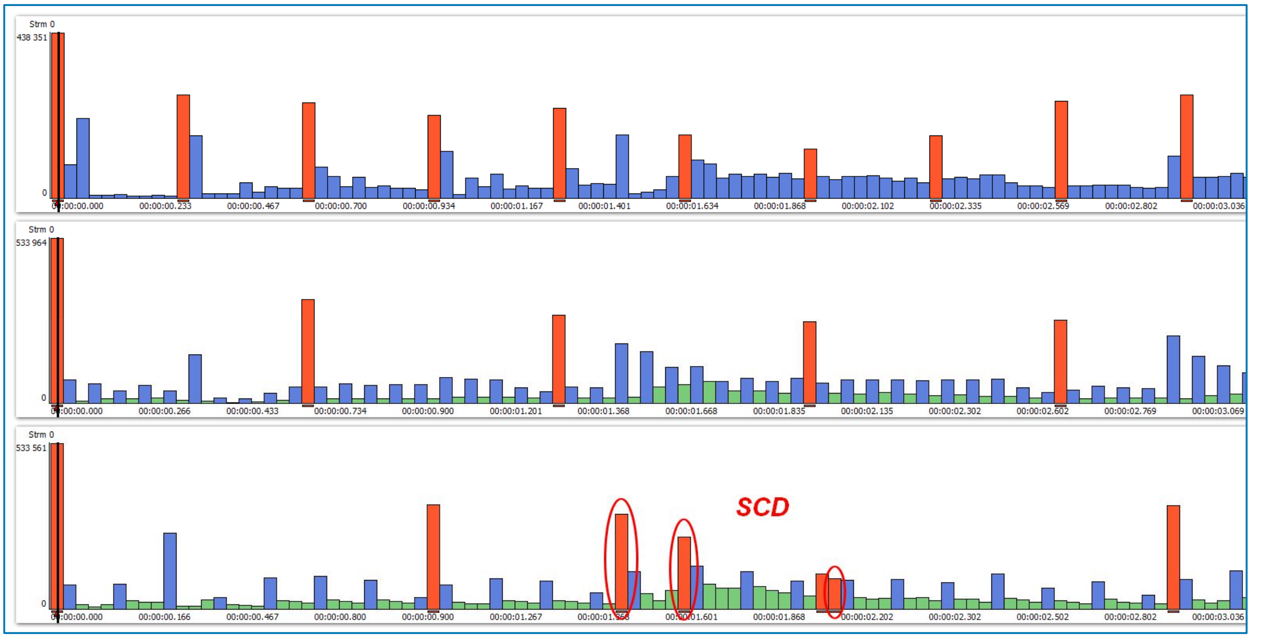


Fig. 3 The GOP structure of video encoded in different ways:
3.1: M=1 N=10, no B frames, no SCD;
3.2: M=2 N=20, B = 1, no SCD;
3.3: M=4 N=30, B = 3, pyramid, SCD.

Tips & Tricks

- Length. Long GOPs are used in files or in OTT broadcasting (for example, when the GOP length in seconds is equal to the chunk duration). For live broadcasting, a smaller GOP is recommended for several reasons, such as:
  - According to the DVB standard, PAT/PMT tables should appear at a rate of twice per second (equivalent to a period of 500 ms), and, as a rule, a PAT/PMT table is placed next to an I frame.
  - For DVB, the channel switching rate is also very important. The longer the GOP, the more time the switching will take.
- Structure. For better quality, use a hierarchical (pyramidal) GOP. This mode allows B frames to reference each other. Adaptive selection of the number of B frames is suited for encoding highly dynamic video sequences with complex motion. In moments of such complex motion, the number of P frames increases, and the GOP structure changes.

- Scene change detection. Most encoders detect a scene view change and insert a full I frame into the scene automatically. However, if the content features frequent scene view changes (e.g. news), inserting full frames can cause the GOP structure to change. This will add several extra seconds of latency to the stream. If a buffer overrun occurs in the receiving device, the viewers will see frozen pictures and pixelation (scene change detection can be seen in Fig. 3).
- Average (avg) encode ratio for the entire stream and I , P , and B frames: shows the compression ratio for the raw video. It can be used to verify the overall encoder performance, check whether the encoder has maintained the required proportions in terms of avg[EncRatio(I)] << avg[EncRatio(P)] << avg[EncRatio(B)], and compare the performance of two encoders using a common set of media files.

name	value	%
stream type	AVC/H.264	
profile	Main	
level	5.1	
chroma format	4:2:0	
resolution	1920 x 1080	
frame rate	29.97	
coding mode	CABAC	
declared bitrate	3 000 000	
duration	00:00:20:020	
mux duration	00:00:20:020	
epsnr	30.61	
frames		
count	600	
I	24 (4.00%)	
P	192 (32.00%)	
B	384 (64.00%)	
size (byte) / encode ratio (avg)	13036 / 239	
I	76 934 / 40	
P	19 904 / 156	
B	5 608 / 555	
bit allocation		
max	9 389 448 [589]	
avg	3 125 601	
min	671 640 [403]	
instant bitrate		
max / min	8 735 616 / 659 184	
qp		
qp max / min	51 / 15	
qp avg	36.25	

Fig. 4 Video sequence information (including encode ratio).

Macroblocks: partitioning and prediction modes

As in other codecs, AVC splits the captured frame into parts called macroblocks. Before AVC, coding standards specified a fixed macroblock size of 16x16 pixels. AVC allows the user to modify this parameter, with the minimum block size being 4x4 pixels.

The alternation between small, mid-sized, and large blocks in frames provides adaptive adjustment of the encoding and determines to a large extent the processor load during real time encoding. By varying the block size, the optimum picture quality is achieved.

For efficient compression, the encoder must make correct decisions when choosing block prediction and partitioning options.

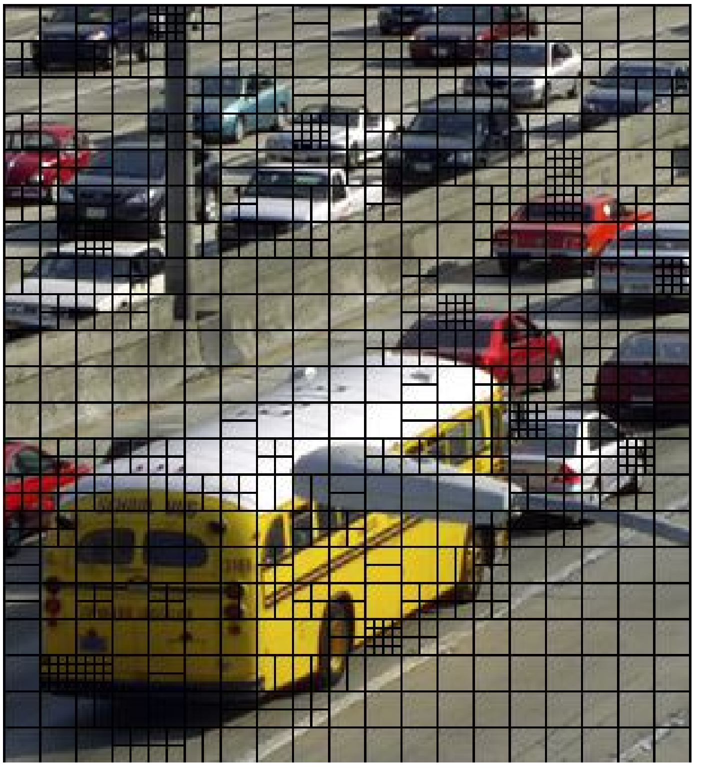


Fig. 5 Frame with minimum block size 4x4.

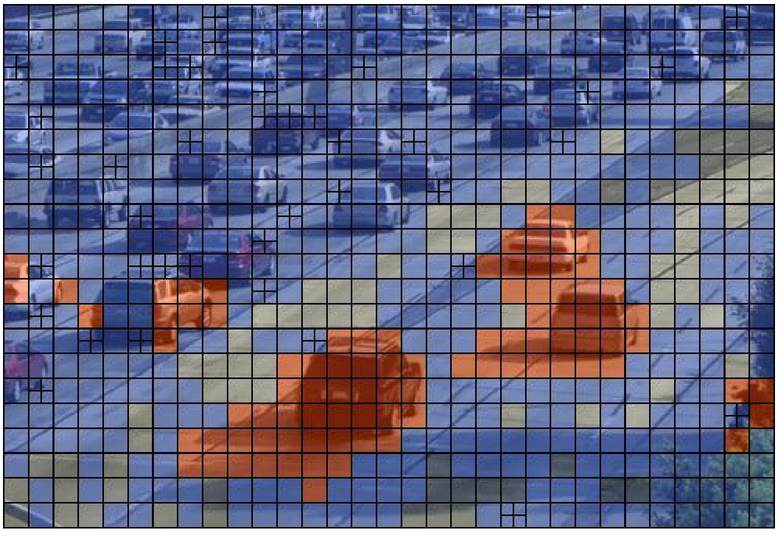
Tips & Tricks

- In broadcasting solutions, the macroblock size should be minimal but not so small as to cause dropped frames and added latency due to complex processing in the encoder.
- The macroblock size can be increased if necessary by pre-filtering the picture (e.g. blurring).

Eliminating spatial redundancy

Spatial (intra) prediction is an intra-frame prediction of macroblock values based on linear combinations of pixels in the adjacent macroblocks (located to the left and above). H.264/AVC allows up to 10 spatial block prediction options.

The encoder needs to select the most suitable spatial block prediction mode.



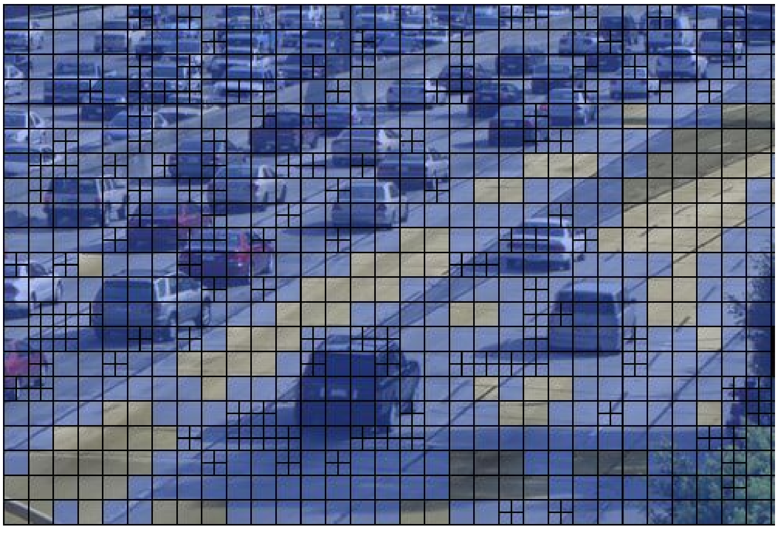


Fig. 6 Example of Intra (above) и Inter (below) prediction of one and the same frame.

Intra-blocks in P and B frames are only necessary when the prediction from the preceding frames differs significantly from information in the current block, such as when new objects appear in the scene or an object’s motion involves deformation. Besides that, the regions in the frame that represent static background should be encoded in most cases in the "skip" mode where transformation coefficients are skipped.

Availability of spatial prediction modes depends on the profile. For example, spatial prediction using 8×8 blocks is only available starting from a high profile.

In figure 6, intra-prediction blocks are shown in red. The blue regions are the temporal block predictions from previously encoded frames. Clear blocks show temporal predictions where residual information was not encoded. To reconstruct the block, it is sufficient to copy the relevant area from the previous frame. In the frame above, we can observe that this prediction is not the most efficient because the cars are moving steadily, and their motion can most likely be successfully predicted from the previously encoded frames, as in the frame below.

Tips & Tricks

- To reduce computational complexity of compression, the encoder is prevented from using all available prediction modes, which negatively affects the compression efficiency. The higher the profile, the wider the choice of spatial prediction modes that the encoder can use.
- The encoder is malfunctioning if it uses prediction modes that are not available for the current profile.

Stream ▾ prediction ▾ sizes ▾

type	4x4	8x8	16x16
intra			
PLANE	0.000	0.000	2.283
VERTICAL	15.835	0.000	5.502
HORIZONTAL	8.503	0.000	3.464
DC	57.065	0.000	1.510
DIAGONAL DOWN LEFT	1.909	0.000	0.000
DIAGONAL DOWN RIGHT	0.842	0.000	0.000
VERTICAL RIGHT MODE	0.609	0.000	0.000
HORIZONTAL DOWN	0.942	0.000	0.000
VERTICAL LEFT	0.645	0.000	0.000
HORIZONTAL UP	0.893	0.000	0.000

Stream ▾ prediction ▾ sizes ▾

type	4x4	8x8	16x16
intra			
PLANE	0.000	0.000	0.506
VERTICAL	24.846	0.000	0.899
HORIZONTAL	17.133	0.000	0.686
DC	20.206	0.000	0.278
DIAGONAL DOWN LEFT	7.988	0.000	0.000
DIAGONAL DOWN RIGHT	5.824	0.000	0.000
VERTICAL RIGHT MODE	5.141	0.000	0.000
HORIZONTAL DOWN	5.324	0.000	0.000
VERTICAL LEFT	5.110	0.000	0.000
HORIZONTAL UP	6.060	0.000	0.000

Stream ▾ prediction ▾ sizes ▾

type	4x4	8x8	16x16
intra			
PLANE	0.000	0.000	0.521
VERTICAL	17.900	2.042	1.156
HORIZONTAL	14.812	1.738	0.823
DC	12.451	1.633	0.271
DIAGONAL DOWN LEFT	7.814	1.535	0.000
DIAGONAL DOWN RIGHT	6.307	1.226	0.000
VERTICAL RIGHT MODE	5.785	1.383	0.000
HORIZONTAL DOWN	5.705	1.313	0.000
VERTICAL LEFT	5.608	1.476	0.000
HORIZONTAL UP	6.754	1.745	0.000

Fig. 7 Statistics for using various Intra prediction modes for the Baseline, Main, and High profiles. Compression efficiency can be improved by using various spatial prediction modes.

Eliminating temporal redundancy

Efficient elimination of temporal redundancy is the key to successful video compression. Much of the progress in video compression is due to improved motion prediction and reduced temporal differences, and all emerging technologies and future codec standards rely on developments in these areas.

H.264/AVC allows temporal prediction using block sizes of 16×16, 16×8, 8×16, 8×8, 4×8, 8×4, and 4×4.

An algorithm for motion estimation and compensation has the following form:

1. Partition the current frame into blocks.
2. For each block, analyze some adjacent area for reference frames.
3. Find the most similar block in the reference frame based on the chosen metric.
4. Obtain motion vectors that describe temporal block movement.
5. Apply the motion vectors to the blocks from the reference frames to obtain a compensated frame that is subtracted from the current one.
6. Compress the resulting residual using a discrete cosine transform (DCT).

Motion estimation algorithms have a high computational complexity and a large number of possible configurations in the encoder.

Block matching metric

There are several methods for estimating the similarity of images:

- Sum of squared differences (SSD). Ensures good quality but is computationally intensive and very sensitive to brightness changes.
- Sum of absolute differences (SAD). Executes in reasonable time and provides acceptable quality. This is the most commonly used method and is fairly fast.
- Sum of absolute transformed differences (SATD). More accurate and noise-tolerant than SAD but takes more time to complete.

Block matching algorithm

- Full Search. Iterates the coordinates of the desired block over some area around the block being processed. This algorithm is guaranteed to give the best results but is very computationally intensive.
- Template search. This algorithm is fast but does not give the best results.
- Spiral search. It is assumed that the closer the block is to the current one, the higher the probability of it being the desired block.

Motion estimation range

Efficiency of P and B frame compression is mainly determined by the motion estimation range in which the encoder searches for the most similar region of the frame.

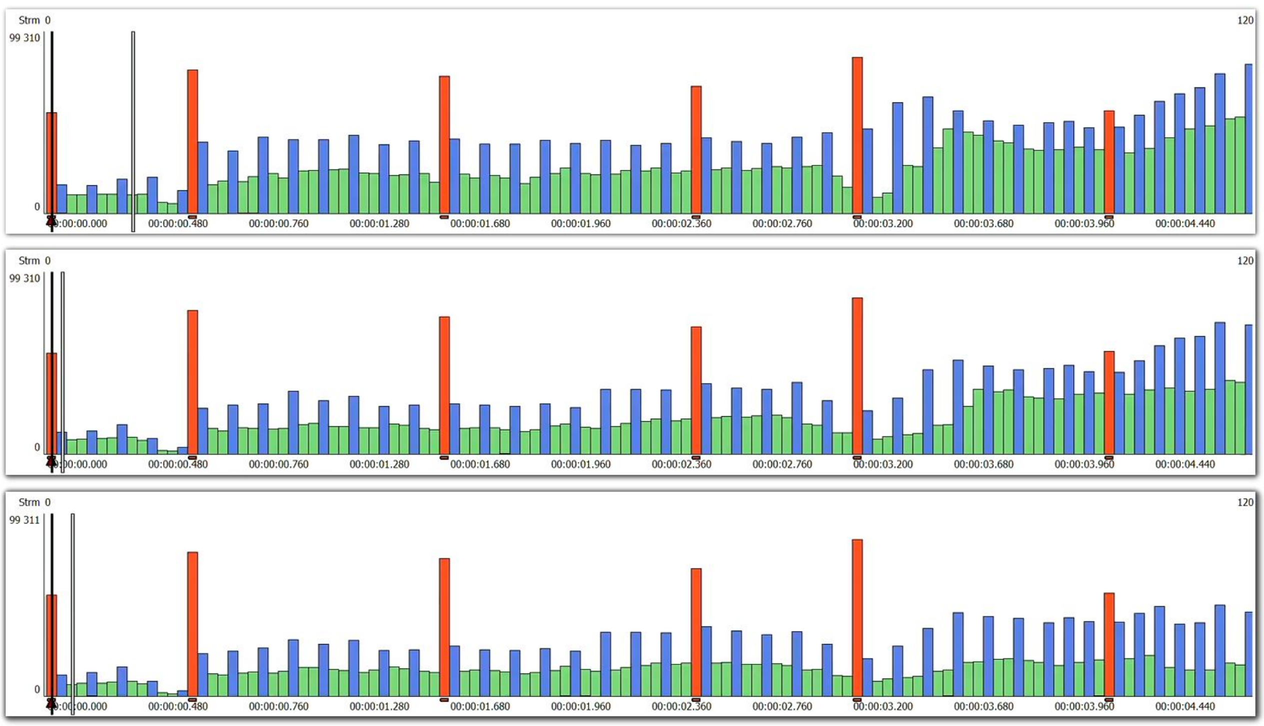


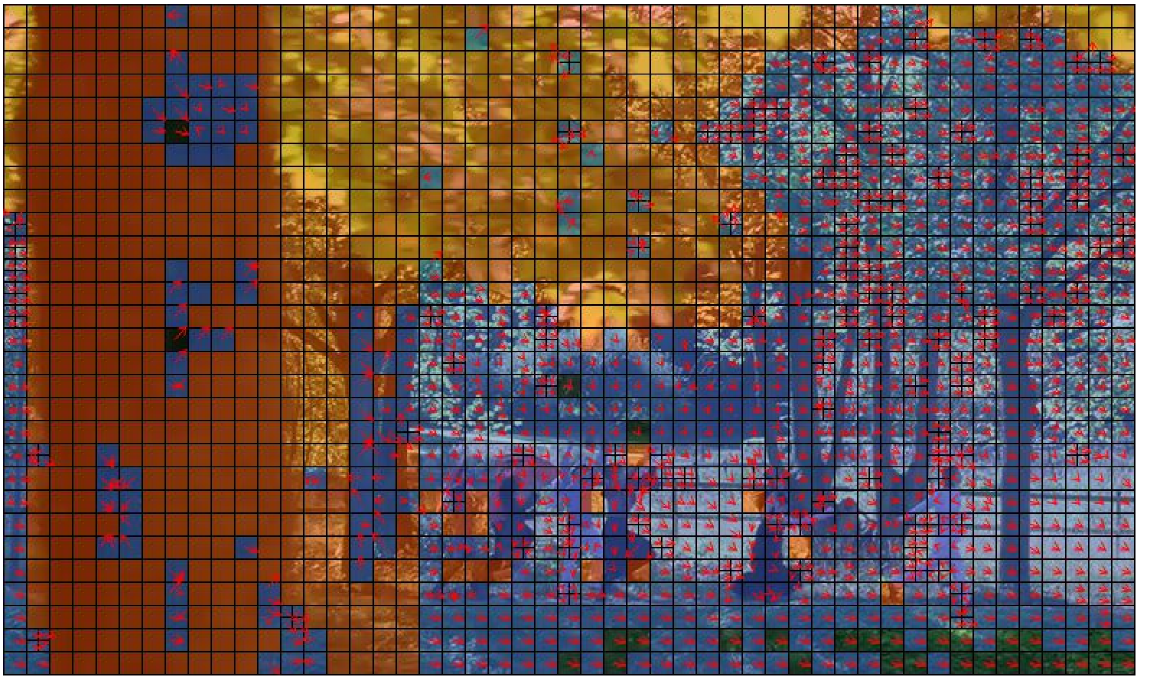
Fig. 8 Impact of the ME_range parameter in the encoder settings on the average P and B frame sizes.

8.1: ME_Range=0; Avg size P=37.5kb; Avg size B=16.4kb.

8.2: ME_Range=8; Avg size P=26.1kb; Avg size B=10.5kb.

8.3: ME_Range=64; Avg size P=22.4kb; Avg size B=8.5kb.

In the figure 9, it is clear that with a smaller motion estimation range, the encoder has to use intra-coding more often, which directly affects quality.



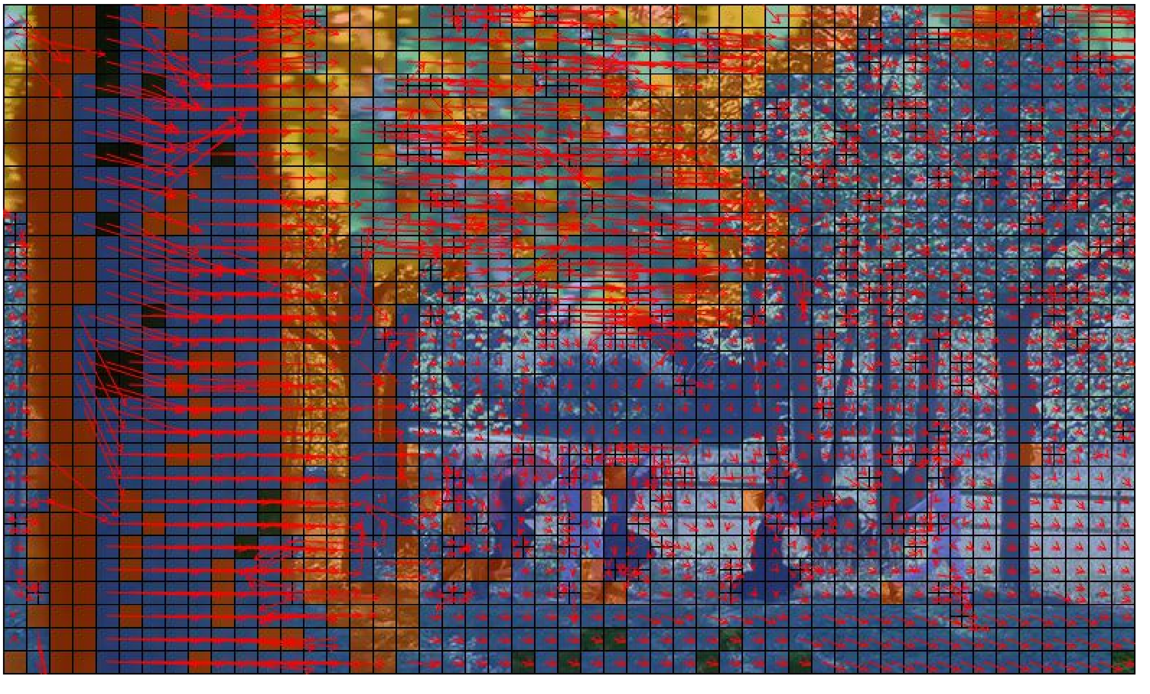


Fig. 9 Overlaid motion vectors in a P frame with different motion estimation range:

9.1 — me_range=8;

9.2 — me_range=64.

The blue, red, and green frame regions contain significant differences. Without an efficient temporal prediction, they will have to be written to the stream, thereby increasing bitrate of the encoded video.

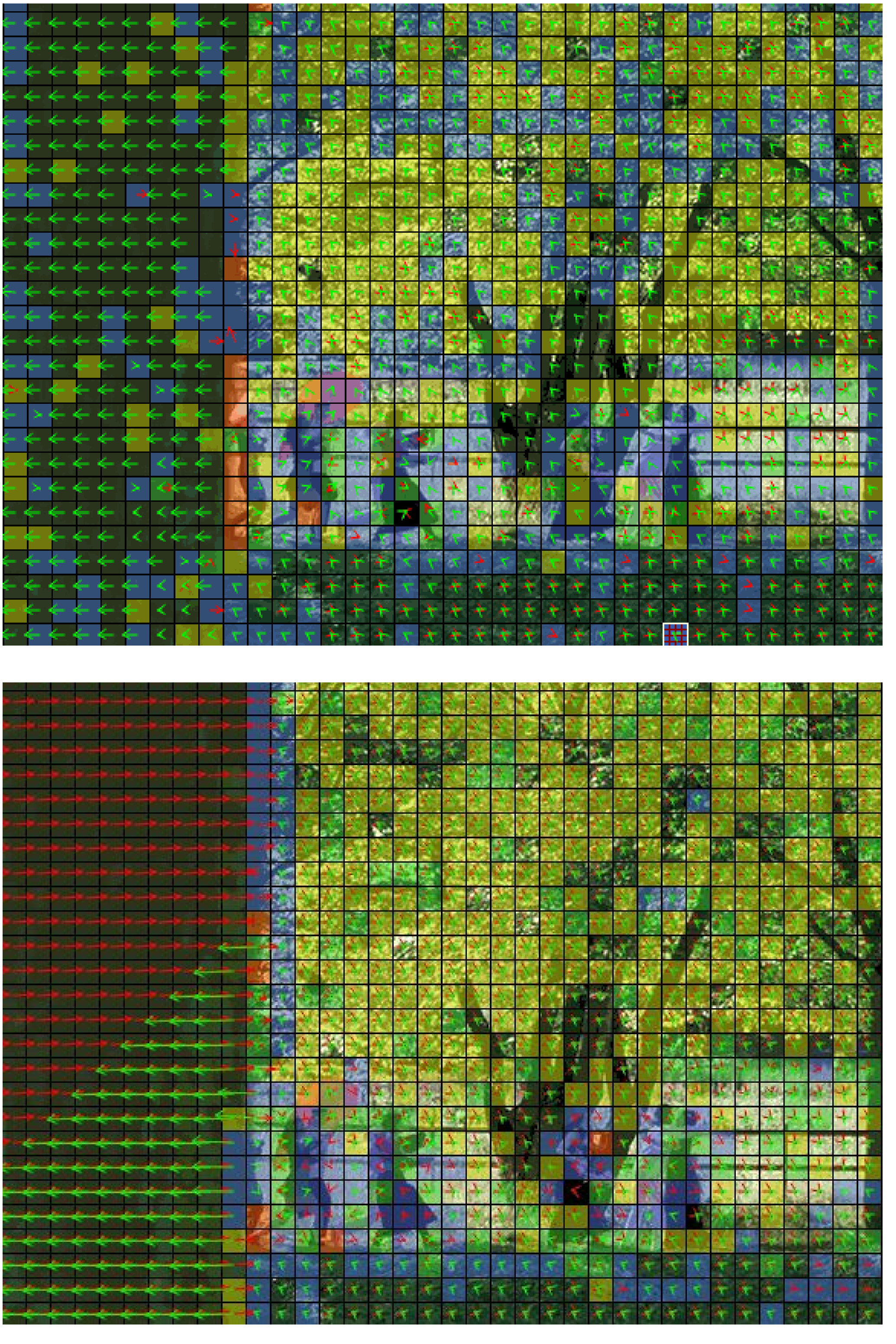


Fig. 10 Motion compensation with different motion estimation range for the same scene.

To analyze motion compensation that has been performed, you can visually compare the predicted and encoded images and look at their difference. This difference is a significant part of the video stream. It allows the original frame image to be restored during decoding, therefore the smaller it is, the better the compensation.

Pictures show the difference in temperature between the predicted and encoded images for a P frame and B frame. The brighter, the more differences there are.

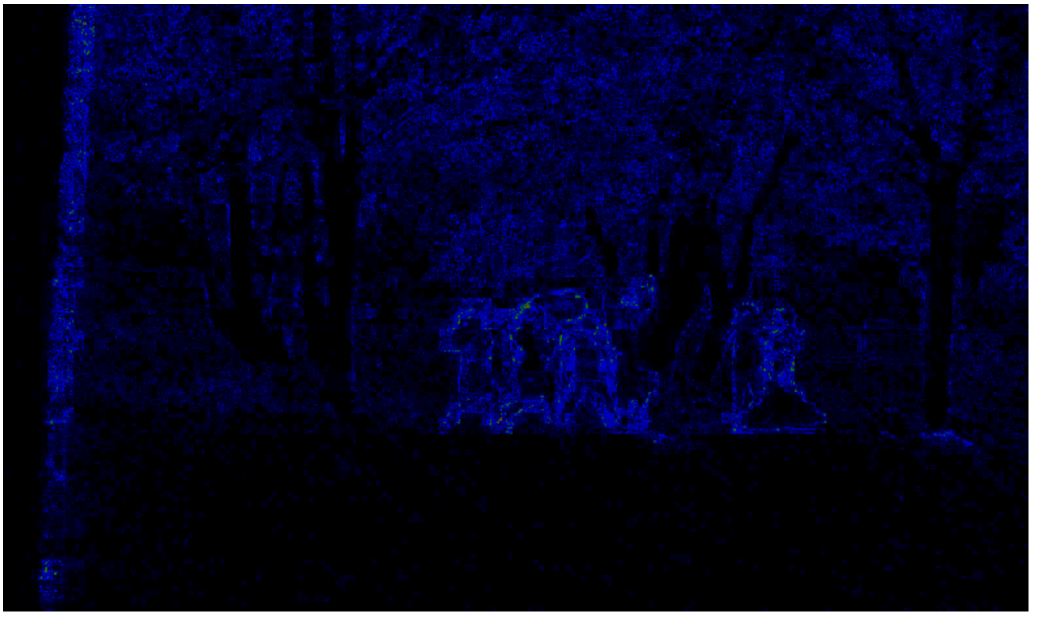


Fig. 11.1 The difference in temperature for a P frame.

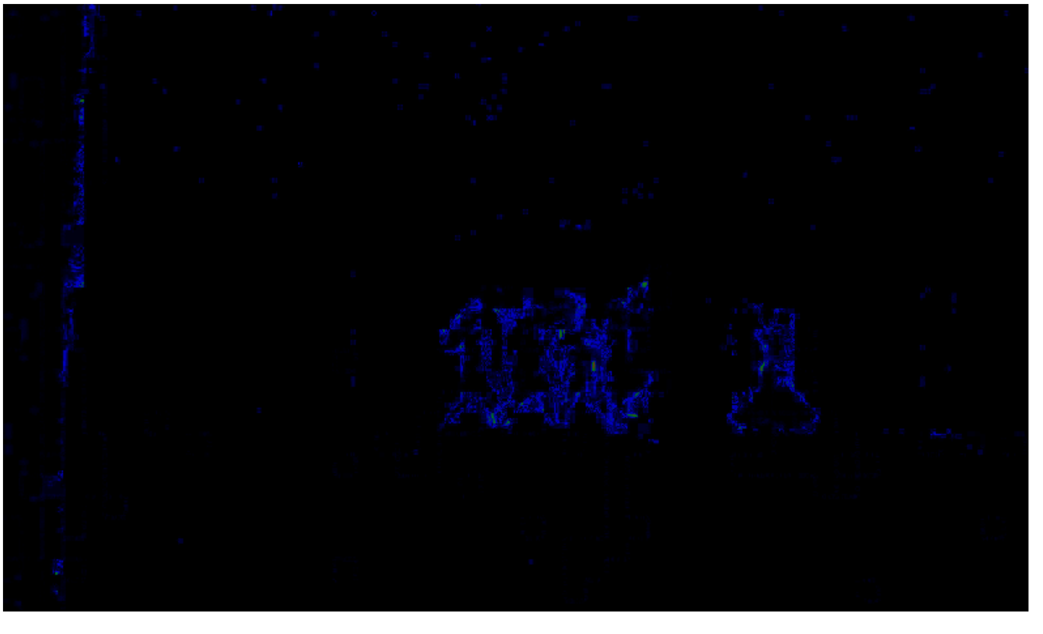
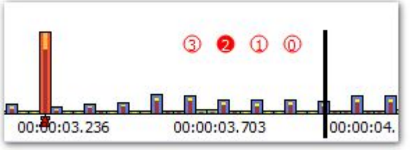


Fig. 11.2 The difference in temperature for a B frame.

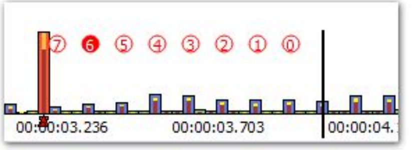
Number of reference frames

With intense motion, the more reference frames, the better the motion compensation algorithm will work, and the more efficient the compression will be. If, however, the motion is smooth enough, it will be more efficient, for example, to widen the motion estimation range than to increase the number of reference frames.

inter	4x4	8x8	16x16
RefIdx0	0.000	0.394	96.934
RefIdx1	0.000	0.121	2.551
RefIdx2	0.000	0.000	0.000
RefIdx3	0.000	0.000	0.000
RefIdx4	0.000	0.000	0.000
RefIdx5	0.000	0.000	0.000
RefIdx6	0.000	0.000	0.000
RefIdx7	0.000	0.000	0.000
RefIdx8	0.000	0.000	0.000
RefIdx9	0.000	0.000	0.000
RefIdx10	0.000	0.000	0.000
RefIdx11	0.000	0.000	0.000
RefIdx12	0.000	0.000	0.000
RefIdx13	0.000	0.000	0.000
RefIdx14	0.000	0.000	0.000
RefIdx15	0.000	0.000	0.000



inter	4x4	8x8	16x16
RefIdx0	0.000	0.399	93.035
RefIdx1	0.000	0.094	3.770
RefIdx2	0.000	0.057	1.677
RefIdx3	0.000	0.056	0.913
RefIdx4	0.000	0.000	0.000
RefIdx5	0.000	0.000	0.000
RefIdx6	0.000	0.000	0.000
RefIdx7	0.000	0.000	0.000
RefIdx8	0.000	0.000	0.000
RefIdx9	0.000	0.000	0.000
RefIdx10	0.000	0.000	0.000
RefIdx11	0.000	0.000	0.000
RefIdx12	0.000	0.000	0.000
RefIdx13	0.000	0.000	0.000
RefIdx14	0.000	0.000	0.000
RefIdx15	0.000	0.000	0.000



inter	4x4	8x8	16x16
RefIdx0	0.000	0.357	91.888
RefIdx1	0.000	0.081	3.797
RefIdx2	0.000	0.044	1.364
RefIdx3	0.000	0.031	0.768
RefIdx4	0.000	0.026	0.514
RefIdx5	0.000	0.022	0.417
RefIdx6	0.000	0.021	0.374
RefIdx7	0.000	0.025	0.268
RefIdx8	0.000	0.000	0.000
RefIdx9	0.000	0.000	0.000
RefIdx10	0.000	0.000	0.000
RefIdx11	0.000	0.000	0.000
RefIdx12	0.000	0.000	0.000
RefIdx13	0.000	0.000	0.000
RefIdx14	0.000	0.000	0.000
RefIdx15	0.000	0.000	0.000

Fig. 12 Statistics for using various block sizes for temporal prediction.
The max_num_ref_frames values are 2, 4, and 8, respectively.

Precision of motion compensation estimation

Subpixel prediction is yet another important encoder parameter. This type of prediction involves interpolating values between integer pixel samples, which enables more accurate motion prediction and improves compression efficiency significantly. When only integer prediction is used, the vector values will always be multiples of 4. When only half-pixel prediction is used, the vector values will always be multiples of 2. With quarter-pixel prediction, the motion vector will have an odd value.

H.264/AVC allows temporal prediction with quarter-pixel precision. Subpixel prediction is very efficient at compressing videos with smooth object motion.

Block Info	
name	value
mb location	736x464
slice_id	0
mb_type	B_Direct_16x16
mb_address	2366
field / frame:	frame
transform 8x8	false
sizes	
total	28
prediction	0
transform	21
transform	
qp	32
qp_delta	0
cbp	24
dimension	16x16
sub_pmode	Direct
mvs	
mv[0]/mvd[0]	168, -282, 0 / 0, 0
mv[1]/mvd[1]	36, -112, 0 / 0, 0

Block Info	
name	value
mb location	736x464
slice_id	0
mb_type	B_Bi_16x16
mb_address	2366
field / frame:	frame
transform 8x8	false
sizes	
total	74
prediction	35
transform	24
transform	
qp	32
qp_delta	0
cbp	30
dimension	16x16
sub_pmode	BI
mvs	
mv[0]/mvd[0]	-179, -333, 0 / -179, -7
mv[1]/mvd[1]	42, -126, 0 / 12, -20

Fig. 13 Motion vector length values for half-pixel (left) and quarter-pixel (right) precision.

Tips & Tricks

- The motion estimation algorithm and area size are the key factors affecting efficiency of P and B frame compression. The larger the motion estimation range, the better the motion compensation and the overall video compression.
- For broadcasting, the SAD metric represents the best trade-off between quality and computational complexity.
- The best motion compensation results are achieved using quarter-pixel prediction.
- Many encoders, especially hardware-based ones, do not use quarter-pixel prediction so as to reduce the computational load. Therefore, inability to find a motion vector with a length that is a multiple of 4 means that subpixel prediction was not used.
- Choose the number of reference frames based on intensity of motion in the content. AVC allows up to 16 reference frames to be used, but in practice, it is rare to use more than 5.
- When configuring your equipment, pay attention to how the encoder uses available temporal block prediction tools.

Adaptive quantization

Quantization is one of the main methods of bitrate control. Reducing bitrate increases compression losses, which in addition to general reduction in quality, makes intensity transitions at block boundaries sharper.

Adaptive block quantization provides an additional improvement in compression efficiency, allowing higher losses in the frame regions that contain less important information.

To tell whether adaptive quantization was used, you need to look at the average quantizer value per frame. If this value is an integer, then all blocks of the frame are highly likely to have been coded with the same quantizer, and vice versa.

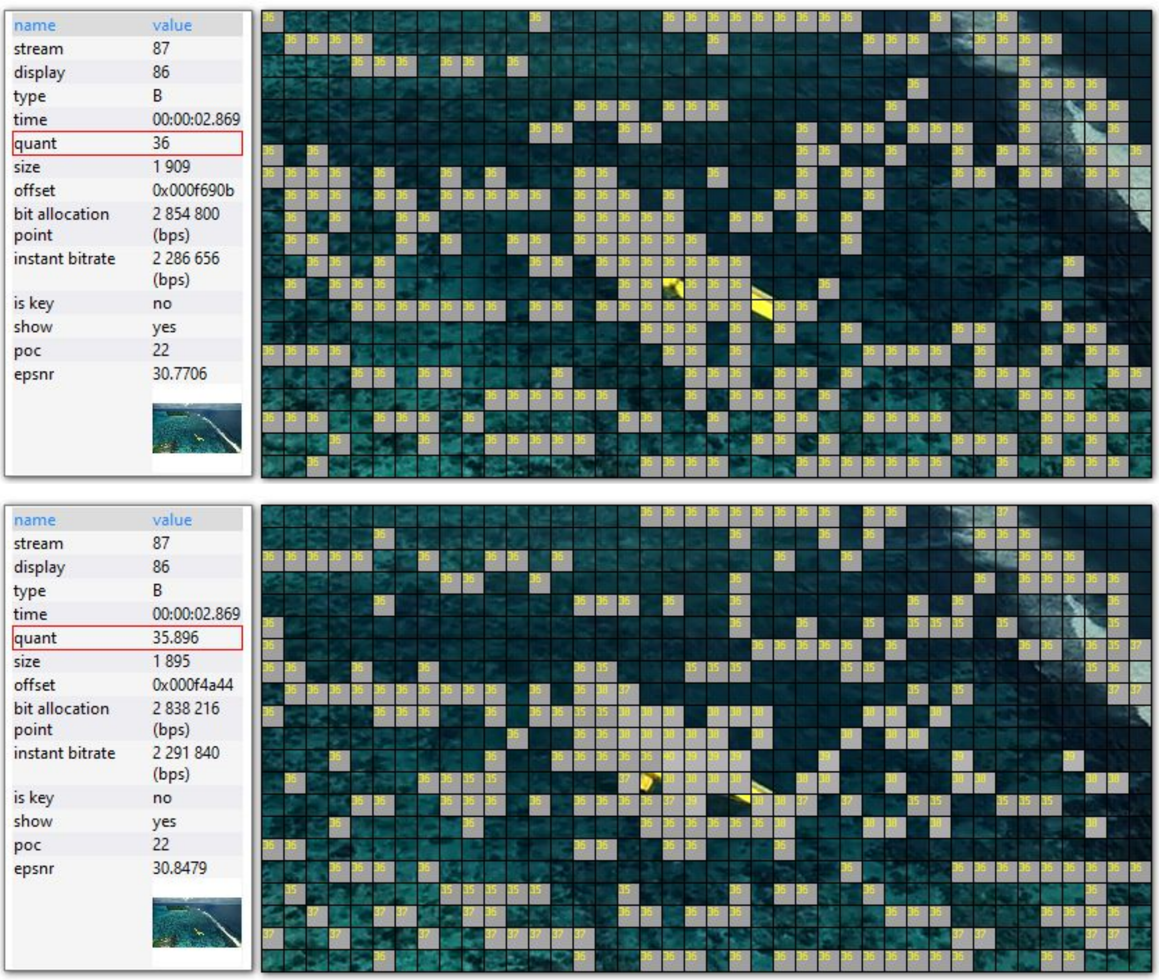


Fig. 14 Values of constant (above) and adaptive (below) quantizers for one and the same frame.

elecard VIDEO COMPRESSION GURU

Tips & Tricks

- An AVC encoder uses quantizer values in the range of 1 to 51. Values in the range of 22 to 30 indicate an acceptable picture quality. Many hardware-based encoders only compress video with the same quantization within a frame.
- Ideally, the average quantizer values per frame should increase depending on the frame type.
- avg(B quants) > avg(P quants) > avg(I quants)
- The higher the quantization level, the more noticeable the resulting artifacts at block boundaries.
- High average quantizer values suggest that the bitrate is too low for this content or that the encoder is performing inadequately.

Deblocking filtering

AVC allows inclusion of a deblocking filtering step in the compression and decoding processes. This helps reduce block-type artifacts that occur due to quantization. The filtering improves visual quality and efficiency of P and B frame compression.

Fig. 15 Decoded image before and after deblocking filtering.

Tips & Tricks

- Deblocking filtering is an optional step in the compression process and is specified in the encoder settings.
- If a deblocking filter was used during compression, it must also be used during decoding, which increases the number of computations somewhat.
- To tell whether deblocking filtering was used, check the SPS and PPS header fields:
- deblocking_filter_control_present_flag = 0 – Filtering is enabled by default;
- deblocking_filter_control_present_flag = 1 – Each header of a slice contains a disable_deblocking_filter_idc field;
- disable_deblocking_filter_idc = 1 – Deblocking filtering is not used in this slice.
- It is recommended to always enable deblocking filtering because its benefits outweigh the slight increase in the number of computations.

Entropy encoding

H.264/AVC has two options for entropy encoding of syntax elements in the stream:

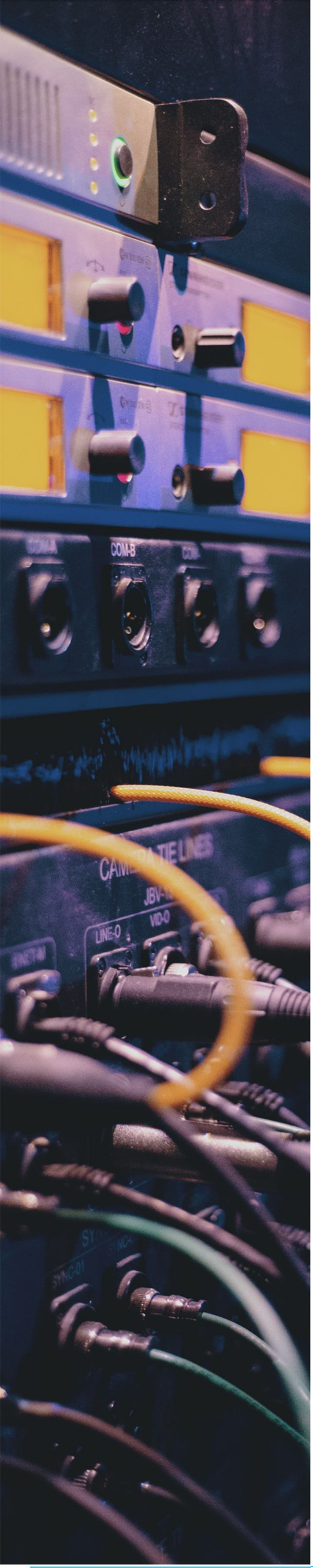
- CAVLC—context-adaptive encoding with variable code word length,
- CABAC—context-adaptive binary arithmetic coding.

CAVLC is mainly used in simplified compression systems and helps reduce the development cost of a software- or hardware-based encoder. In most video systems, CAVLC is not efficient.

CABAC provides more efficient compression but requires more computations during both compression and decoding. It is not supported in the Baseline and Constrained Baseline profiles.

Tips & Tricks

- CABAC makes it possible to achieve bitrate that is about 10% lower than with CAVLC at the same level of distortion.
- If CAVLC is used when the profile is higher than Baseline, the encoder is not configured optimally.



Conclusion

The advent of the AVC standard was a giant step in the field of video compression. AVC offered more accurate methods of both intra-prediction and motion compensation and provided very good quality at relatively low bitrates. The standard is widely used and therefore supported by virtually all video playback devices manufactured in the last 15 years.

Weak and unstable reception areas that cause interference and noise are becoming a thing of the past. Meanwhile, consumer screen sizes are steadily increasing, and broadcasters have to maintain a high standard of content quality. Setting up a transcoding solution for a specific TV company with diverse and unpredictable content is a constant headache for any engineering manager. There is only one answer: understand clearly the task at hand and configure the equipment properly.

Contact details

HEAD OFFICE	USA OFFICE	CHINA OFFICE
Address: 3 Razvitiya ave. Tomsk, Russia, 634055 **Phone:** +7 (3822) 488-580 Business hours: Monday-Friday 10.00 AM through 7.00 PM (local time GMT +07.00)	**Address:** 142 N. Milpitas blvd, #411 Milpitas, CA, 95035 **Phone:** +1 (323) 644-4063 Business hours: Monday-Friday 10.00 AM through 5.00 PM (local time PST) Corporate and OEM customers	**Address:** Vvyee halls, booth C06, 3rd floor, 96 Dong Hua Nan Lu street, YueXiu District, Guangzhou, China. **Phone:** +86-20-83717605

Email: sales@elecard.com
Website: www.elecard.com