# H264/MPEG4 AVC

***Abstract***
*H264/MPEG4 AVC is a new standard (finalised in mid 2003) for video compression. H264/MPEG4 AVC outperforms previous H263, MPEG2 and MPEG4 standards by a factor of two or more with respect to compression ratio, albeit at the cost of increased complexity. With the increase in computational capabilities of today's microprocessors, DSPs and ordinary PCs, H264/MPEG4 AVC becomes a feasible codec solution for many applications.*

 

## History

MPEG4 AVC, also known as MPEG4 part 10 or H.264, is a new standard for video compression. The standard was developed as a Joint Video Team, JVT, by members from two standardisation organisations ITU-T and ISO/IEC. The JVT started its work in 1998 and delivered a finalised standard in the summer of 2003. ITU-T calls the standard H.264 and ISO/IEC calls it MPEG4 AVC, where AVC stands for "Advanced Video Codec".

## Licensing

The MPEG4 AVC is an open standard, allowing all companies to develop their own codec (encoder/decoder pair). The standard only defines the decoder and put no restraints on the encoder, apart from producing a standard compliant stream of compressed video, i.e. a stream decodable by a standard compliant decoder. Many of the companies participating in the JVT have patented some of the techniques involved in MPEG4 AVC. These companies has joined two separate "patent pools" organisations, MPEG LA (www.mpegla.com) and VIA licensing (www.vialicensing.com). MPEG LA and VIA licensing require modest fees for use of the participating company patents. For example, MPEG LA require no fees for the first 100.000 devices containing MPEG4 AVC codecs and VIA licensing require no fees for the first 50.000 devices containing MPEG4 AVC codecs (if the annual revenue is less than $US500,000 for these products)

## Codec overview

MPEG4 AVC  is a blockbased codec that divides a picture into "macroblocks" of 16x16 pixels. Each macroblock is predicted, either from other macroblocks in the same picture or from blocks located in other pictures. The macroblock prediction is compared to the same block in the original picture. The residual, i.e. difference, between the macroblock and original block is encoded with a discrete cosine transform followed by quantisation and an entropy coding algorithm.

## Codec enhancements

MPEG4 AVC is an evolution from previous blockbased codecs, refining many of the coding steps found in other standards such as H261, H263, MPEG1 and MPEG2. Some of the refinements are listed below.

**Partitioning of Macroblocks**

In MPEG4 AVC, the encoder may split each macroblock into a wide range of possible partitions. A macroblock may be split into at the most 16 smaller sub-partitions, where each sub-partition may be motion predicted from up to two previously encoded pictures.  Variable partitioning of macroblocks allows for finer resolution around areas in the picture where objects have different velocity, while for areas where there are no movement or all objects has the same velocity larger partitions are used.

**Higher resolution for motion vectors**

MPEG4 AVC has 1/8 pixel resolution for the chroma component motion vectors and ¼ pixel resolution for luma component motion vectors. This allows for a more accurate estimate of motion vectors and reducing the number of bits that are needed for coding the residual (difference) between the estimate and original picture, compared to previous compression standards.

**DCT and IDCT approximated**

The DCT (Discrete Cosine Transform) and IDCT (Inverse Discrete Cosine Transform) of luma and chroma macroblocks residuals used in MPEG2 have been approximated with integer transforms for both luma and chroma in MPEG4 AVC. The use of integer transforms eliminates the inverse-transform mismatch that was due to floating point arithmetic in MPEG2. The transforms require 16-bit arithmetic and may be implemented as both "shift and add" or "multiply and add" instructions and thus allowing fast implementations on both microprocessors and DSPs.

**New entropy coding**

Two methods, CAVLC (Context Adaptive Variable Length Coding) for baseline and extended profile and CABAC (Context Adaptive Binary Arithmetic Coder) for main profile, achieves a higher compression ratio than previous entropy coding methods.

**Deblocking filter**

The MPEG4 AVC encoder/decoder utilises a deblocking filter after the inverse transform of luma and croma macroblocks, smoothing over the blocking artefacts usually found in previous standards.

## Profiles

MPEG4 AVC supports three profiles, baseline, main and extended profile. The baseline profile is a low latency profile aimed for videoconferencing. This profile uses only I- and P-slices, described in the technical glossary below, and the less computational demanding CAVLC entropy coding method. The main profile is aimed for storage and the encoder may use I-, P- and B-slices and the CABAC entropy coding method, thus usually achieving higher compression ratio than the baseline profile. The extended profile is aimed for video streaming over error-prone transmission links. The extended profile is also designed for fast switching between video streams encoded with varying visual quality. For example, a client may be allowed to view a video stream of low quality but after payment could switch "on the fly" to a high quality video stream. The encoder may use all the available slice types (I-,P-,B-,SI- and SP-slices) in this profile, but use the CAVLC entropy coding method.

## Coding gains/costs

MPEG4 AVC delivers an increase in coding efficiency by a factor of 2-2,5 compared to MPEG2. The increase in coding efficiency comes at a price. The complexity of the decoder increases by a factor of 2 to 4 depending on the profile used compared to MPEG2. The complexity of the encoder increases by a factor of 4 to 5.

## Applications

MPEG4 AVC's increase in compression efficiency can be utilised in several new application areas. For example, video streams with TV (PAL) quality can be transmitted over xDSL connection at approximately 1 Mbit/s. Likewise; high quality videoconferencing over xDSL becomes feasible. It will also be possible to store a high definition movie on a conventional DVD, possibly delaying the switch to blue laser DVD.

It is also possible that the baseline or extended profile will be adopted for mobile communications, due to the high cost of bandwidth in 3G networks.



 

## Technical Glossary

**Luma/chroma**
A movie sequence is usually captured or displayed in the RGB colour space (CRT and LCD monitors displays in RGB and a video camera capture a scene by filtering out the Red, Green and Blue component). However, the human eye is more sensitive to brightness (luminance) than to colour (chrominance). An initial step in most compression methods is therefore to transform RGB data into the YCbCr colour space and down sample the colour components. Y is the luminance and Cb and Cr are the chrominance components. For MPEG4 AVC, the 4:2:0 sampling format is used, where Cb and Cr has half the horizontal and vertical resolution of Y.

**Macroblocks**
"macroblock" is the name for a grouping of 16x16 luminance samples and the 2 corresponding 8x8 chrominance samples in a picture.

**Macroblock partitioning in MPEG4 AVC**
A macroblock luma component may be split up in three ways, either as two 8x16, two 16x8 or four 8x8 partitions. When a 8x8 partition is used, it may also be split into sub-macroblock partitions in three ways, either as two 8x4, two 4x8 or four 4x4. For the croma components, the corresponding partitions are two 4x8, two 8x4 or four 4x4 partitions and the sub-partitions are two 4x2, two 2x4 or four 2x2.

**Slices**
A slice is a group of macroblocks encoded with the same type of scheme. There exist five types of slices in MPEG4 AVC

*I-slices*: Is a group of macroblocks encoded without reference to other slices. Note however that a macroblock in an I-slice typically is encoded by use data from already encoded neighbouring macroblocks and that only 16x16 and 4x4 macroblocks luma partitions (8x8 and 2x2 for chroma) are used for I-slices.

*P-slices*: Is a group of macroblocks with up to three types of macroblocks.

1. Intra coded.
   A macroblock is coded as macroblocks in I-slices
2. Inter coded.
   A macroblock is coded with references, i.e. motion vectors, to areas in already encoded/ decoded pictures and located before the current picture in time. Note that several pictures may be used as reference, but only one for each macroblock sub partition.
3. Skipped.
   A skipped macroblock flag implies that the motion vector for this macroblock should be calculated as the average of neighbouring macroblocks and thus allowing for higher compression.

*B-slices*: Is a group of macroblocks encoded/ decoded with references, i.e. motion vectors, to areas in already encoded/decoded pictures that may be located BOTH before and after the current picture in time. Each macroblock partition may refer to two pictures, where both pictures may be located earlier or later in time than the current picture. It is also possible that a macroblock refer to one picture earlier and one later in time than the current picture.

*SP- and SI-slices*: SP and SI slices are used to simplify the switch between different bitstreams of varying compression ratios.