H.264/MPEG-4 AVC

H.264/MPEG-4 Part 10 or **AVC** (Advanced Video Coding) is a standard for video compression, and is currently one of the most commonly used formats for the recording, compression, and distribution of high definition video. The final drafting work on the first version of the standard was completed in May 2003.

H.264/MPEG-4 AVC is a block-oriented motion-compensation-based codec standard developed by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC JTC1 Moving Picture Experts Group (MPEG). The project partnership effort is known as the Joint Video Team (JVT). The ITU-T **H.264** standard and the ISO/IEC **MPEG-4 AVC** standard (formally, ISO/IEC 14496-10 – MPEG-4 Part 10, Advanced Video Coding) are jointly maintained so that they have identical technical content.

H.264 is perhaps best known as being one of the codec standards for Blu-ray Discs; all Blu-ray Disc players must be able to decode H.264. It is also widely used by streaming internet sources, such as videos from Vimeo, YouTube, and the iTunes Store, web software such as the Adobe Flash Player and Microsoft Silverlight, and also various HDTV broadcasts over terrestrial (ATSC, ISDB-T, DVB-T or DVB-T2), cable (DVB-C) and satellite (DVB-S and DVB-S2).

Overview

The intent of the H.264/AVC project was to create a standard capable of providing good video quality at substantially lower bit rates than previous standards (i.e., half or less the bit rate of MPEG-2, H.263, or MPEG-4 Part 2), without increasing the complexity of design so much that it would be impractical or excessively expensive to implement. An additional goal was to provide enough flexibility to allow the standard to be applied to a wide variety of applications on a wide variety of networks and systems, including low and high bit rates, low and high resolution video, broadcast, DVD storage, RTP/IP packet networks, and ITU-T multimedia telephony systems.

The H.264 standard can be viewed as a "family of standards" composed of the profiles described below. A specific decoder decodes at least one, but not necessarily all profiles. The decoder specification describes which profiles can be decoded. The H.264 name follows the ITU-T naming convention, where the standard is a member of the H.26x line of VCEG video coding standards; the MPEG-4 AVC name relates to the naming convention in ISO/IEC MPEG, where the standard is part 10 of ISO/IEC 14496, which is the suite of standards known as MPEG-4. The standard was developed jointly in a partnership of VCEG and MPEG, after earlier development work in the ITU-T as a VCEG project called H.26L. It is thus common to refer to the standard with names such as H.264/AVC, AVC/H.264, H.264/MPEG-4 AVC, or MPEG-4/H.264 AVC, to emphasize the common heritage. Occasionally, it is also referred to as "the JVT codec", in reference to the Joint Video Team (JVT) organization that developed it. (Such partnership and multiple naming is not uncommon. For example, the video codec standard known as MPEG-2 also arose from the partnership between MPEG and the ITU-T, where MPEG-2 video is known to the ITU-T community as H.262.[1]) Some software programs (such as VLC media player) internally identify this standard as AVC1.

The standardization of the first version of H.264/AVC was completed in May 2003. In the first project to extend the original standard, the JVT then developed what was called the Fidelity Range Extensions (FRExt). These extensions enabled higher quality video coding by supporting increased sample bit depth precision and higher-resolution color information, including sampling structures known as Y'CbCr 4:2:2 (=YUV 4:2:2) and Y'CbCr 4:4:4. Several other features were also included in the Fidelity Range Extensions project, such as adaptive switching between 4×4 and 8×8 integer transforms, encoder-specified perceptual-based quantization weighting matrices, efficient inter-picture lossless coding, and support of additional color spaces. The design work on the Fidelity Range Extensions was completed in July 2004, and the drafting work on them was completed in September 2004.

Further recent extensions of the standard then included adding five other new profiles intended primarily for professional applications, adding extended-gamut color space support, defining additional aspect ratio indicators, defining two additional types of "supplemental enhancement information" (post-filter hint and tone mapping), and deprecating one of the prior FRExt profiles that industry feedback indicated should have been designed differently.

The next major feature added to the standard was Scalable Video Coding (SVC). Specified in Annex G of H.264/AVC, SVC allows the construction of bitstreams that contain sub-bitstreams that also conform to the standard, including one such bitstream known as the "base layer" that can be decoded by a H.264/AVC codec that does not support SVC. For temporal bitstream scalability (i.e., the presence of a sub-bitstream with a smaller temporal sampling rate than the main bitstream), complete access units are removed from the bitstream when deriving the sub-bitstream. In this case, high-level syntax and inter-prediction reference pictures in the bitstream are constructed accordingly. On the other hand, for spatial and quality bitstream scalability (i.e. the presence of a sub-bitstream with lower spatial resolution/quality than the main bitstream), the NAL (Network Abstraction Layer) is removed from the bitstream when deriving the sub-bitstream. In this case, inter-layer prediction (i.e., the prediction of the higher spatial resolution/quality signal from the data of the lower spatial resolution/quality signal) is typically used for efficient coding. The Scalable Video Coding extensions were completed in November 2007.

The next major feature added to the standard was Multiview Video Coding (MVC). Specified in Annex H of H.264/AVC, MVC enables the construction of bitstreams that represent more than one view of a video scene. An important example of this functionality is stereoscopic 3D video coding. Two profiles were developed in the MVC work: Multiview High Profile supports an arbitrary number of views, and Stereo High Profile is designed specifically for two-view stereoscopic video. The Multiview Video Coding extensions were completed in November 2009.

Standardization committee and history

In early 1998, the Video Coding Experts Group (VCEG – ITU-T SG16 Q.6) issued a call for proposals on a project called H.26L, with the target to double the coding efficiency (which means halving the bit rate necessary for a given level of fidelity) in comparison to any other existing video coding standards for a broad variety of applications. VCEG was chaired by Gary Sullivan (Microsoft, formerly PictureTel, USA). The first draft design for that new standard was adopted in August 1999. In 2000, Thomas Wiegand (Heinrich Hertz Institute, Germany) became VCEG co-chair. In December 2001, VCEG and the Moving Picture Experts Group (MPEG – ISO/IEC JTC 1/SC 29/WG 11) formed a Joint Video Team (JVT), with the charter to finalize the video coding standard.[2] Formal approval of the specification came in March 2003. The JVT was (is) chaired by Gary Sullivan, Thomas Wiegand, and Ajay Luthra (Motorola, USA). In June 2004, the Fidelity range extensions (FRExt) project was finalized. From January 2005 to November 2007, the JVT was working on an extension of H.264/AVC towards scalability by an Annex (G) called Scalable Video Coding (SVC). The JVT management team was extended by Jens-Rainer Ohm (Aachen University, Germany). From July 2006 to November 2009, the JVT worked on Multiview Video Coding (MVC), an extension of H.264/AVC towards free viewpoint television and 3D television. That work included the development of two new profiles of the standard: the Multiview High Profile and the Stereo High Profile.

Applications

The H.264 video format has a very broad application range that covers all forms of digital compressed video from low bit-rate Internet streaming applications to HDTV broadcast and Digital Cinema applications with nearly lossless coding. With the use of H.264, bit rate savings of 50% or more are reported. For example, H.264 has been reported to give the same Digital Satellite TV quality as current MPEG-2 implementations with less than half the bitrate, with current MPEG-2 implementations working at around 3.5 Mbit/s and H.264 at only 1.5 Mbit/s.[3] To ensure compatibility and problem-free adoption of H.264/AVC, many standards bodies have amended or added to their video-related standards so that users of these standards can employ H.264/AVC.

Both the Blu-ray Disc format and the now-discontinued HD DVD format include the H.264/AVC High Profile as one of 3 mandatory video compression formats.

The Digital Video Broadcast project (DVB) approved the use of H.264/AVC for broadcast television in late 2004.

The Advanced Television Systems Committee (ATSC) standards body in the United States approved the use of H.264/AVC for broadcast television in July 2008, although the standard is not yet used for fixed ATSC broadcasts within the United States.[4][5] It has also been approved for use with the more recent ATSC-M/H (Mobile/Handheld) standard, using the AVC and SVC portions of H.264.[6]

AVCHD is a high-definition recording format designed by Sony and Panasonic that uses H.264 (conforming to H.264 while adding additional application-specific features and constraints).

AVC-Intra is an intraframe-only compression format, developed by Panasonic.

XAVC is a recording format designed by Sony that uses level 5.2 of H.264/MPEG-4 AVC, which is the highest level supported by that video standard.[][] XAVC can support 4K resolution (4096 × 2160 and 3840 × 2160) at up to 60 frames per second (fps).[][] Sony has announced that cameras that support XAVC include two CineAlta cameras—the Sony PMW-F55 and Sony PMW-F5.[][] The Sony PMW-F55 can record XAVC with 4K resolution at 30 fps at 300 Mbit/s and 2K resolution at 30 fps at 100 Mbit/s.[] XAVC can record 4K resolution at 60 fps with 4:2:2 chroma subsampling at 600 Mbit/s.[][]

The CCTV (Closed Circuit TV) and Video Surveillance markets have included the technology in many products.

Canon and Nikon DSLRs use H.264 video wrapped in QuickTime MOV containers as the native recording format.

Patent licensing

In countries where patents on software algorithms are upheld, vendors and commercial users of products that use H.264/AVC are expected to pay patent licensing royalties for the patented technology that their products use.[7] This applies to the Baseline Profile as well.[8] A private organization known as MPEG LA, which is not affiliated in any way with the MPEG standardization organization, administers the licenses for patents applying to this standard, as well as the patent pools for MPEG-2 Part 1 Systems, MPEG-2 Part 2 Video, MPEG-4 Part 2 Video, and other technologies. The MPEG-LA patents in the US last at least until 2027.[9]

On August 26, 2010 MPEG LA announced that H.264 encoded internet video that is free to end users will never be charged royalties.[10] All other royalties remain in place, such as royalties for products that decode and encode H.264 video.[11] The license terms are updated in 5-year blocks.[12]

In 2005, Qualcomm, which was the assignee of U.S. Patent 5,452,104 [13] and U.S. Patent 5,576,767 [14], sued Broadcom in US District Court, alleging that Broadcom infringed the two patents by making products that were compliant with the H.264 video compression standard.[15] In 2007, the District Court found that the patents were unenforceable because Qualcomm had failed to disclose them to the JVT prior to the release of the H.264 standard in May 2003.[15] In December 2008, the US Court of Appeals for the Federal Circuit affirmed the District Court's order that the patents be unenforceable but remanded to the District Court with instructions to limit the scope of unenforceability to H.264 compliant products.[15]

Controversies

Controversies surrounding the H.264 video compression standard stem primarily from its use within the HTML5 Internet standard. HTML5 adds two new tags to the HTML standard—<video> and <audio>—for direct embedding of video and audio content into a web page. HTML5 is being developed by the HTML5 working group as an open standard to be adopted by all web browser developers. In 2009, the HTML5 working group was split between supporters of Ogg Theora, a free video format which is unencumbered by patents, and H.264, which contains patented technology. As late as July 2009, Google and Apple were said to support H.264, while Mozilla and Opera support Ogg Theora.[16] Microsoft, with the release of Internet Explorer 9, has added support for both HTML 5 and H.264. At the Gartner Symposium/ITXpo in November, 2010, Microsoft CEO Steve Ballmer answered the question "HTML 5 or Silverlight?" by saying "If you want to do something that is universal, there is no question the world is going HTML5."[17] In January 2011, Google announced that they were pulling support for H.264 from their Chrome browser and supporting both Theora and WebM/VP8 to use only open formats.[18] However, as of December 2012, Google has not followed through with this announcement and still supports H.264 in their Chrome browser through FFmpeg. No official statement was released on the matter.

On 18 March 2012, Mozilla announced support for H.264 in Firefox on mobile devices, due to prevalence of H.264-encoded video and the increased power-efficiency of using dedicated H.264 decoder hardware common on such devices.[19] On 20 February 2013, Mozilla implemented support in Firefox for decoding H.264 on Windows 7 and above. This feature relies on Windows' built in decoding libraries.[20]

Features

H.264/AVC/MPEG-4 Part 10 contains a number of new features that allow it to compress video much more effectively than older standards and to provide more flexibility for application to a wide variety of network environments. In particular, some such key features include:

- Multi-picture inter-picture prediction including the following features:
  - Using previously encoded pictures as references in a much more flexible way than in past standards, allowing up to 16 reference frames (or 32 reference fields, in the case of interlaced encoding) to be used in some cases. This is in contrast to prior standards, where the limit was typically one; or, in the case of conventional "B pictures" (B-frames), two. This particular feature usually allows modest improvements in bit rate and quality in most scenes.[*citation needed*] But in certain types of scenes, such as those with repetitive motion or back-and-forth scene cuts or uncovered background areas, it allows a significant reduction in bit rate while maintaining clarity.
  - Variable block-size motion compensation (VBSMC) with block sizes as large as 16×16 and as small as 4×4, enabling precise segmentation of moving regions. The supported luma prediction block sizes include 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, and 4×4, many of which can be used together in a single macroblock. Chroma prediction block sizes are correspondingly smaller according to the chroma subsampling in use.
  - The ability to use multiple motion vectors per macroblock (one or two per partition) with a maximum of 32 in the case of a B macroblock constructed of 16 4×4 partitions. The motion vectors for each 8×8 or larger partition region can point to different reference pictures.
  - The ability to use any macroblock type in B-frames, including I-macroblocks, resulting in much more efficient encoding when using B-frames. This feature was notably left out from MPEG-4 ASP.
  - Six-tap filtering for derivation of half-pel luma sample predictions, for sharper subpixel motion-compensation. Quarter-pixel motion is derived by linear interpolation of the halfpel values, to save processing power.
  - Quarter-pixel precision for motion compensation, enabling precise description of the displacements of moving areas. For chroma the resolution is typically halved both vertically and horizontally (see 4:2:0) therefore the motion compensation of chroma uses one-eighth chroma pixel grid units.
  - Weighted prediction, allowing an encoder to specify the use of a scaling and offset when performing motion compensation, and providing a significant benefit in performance in special cases—such as fade-to-black, fade-in, and cross-fade transitions. This includes implicit weighted prediction for B-frames, and explicit weighted prediction for P-frames.
- Spatial prediction from the edges of neighboring blocks for "intra" coding, rather than the "DC"-only prediction found in MPEG-2 Part 2 and the transform coefficient prediction found in H.263v2 and MPEG-4 Part 2. This includes luma prediction block sizes of 16×16, 8×8, and 4×4 (of which only one type can be used within each macroblock).
- Lossless macroblock coding features including:
  - A lossless "PCM macroblock" representation mode in which video data samples are represented directly,[21] allowing perfect representation of specific regions and allowing a strict limit to be placed on the quantity of

  coded data for each macroblock.
  - An enhanced lossless macroblock representation mode allowing perfect representation of specific regions while ordinarily using substantially fewer bits than the PCM mode.
- Flexible interlaced-scan video coding features, including:
  - Macroblock-adaptive frame-field (MBAFF) coding, using a macroblock pair structure for pictures coded as frames, allowing 16×16 macroblocks in field mode (compared with MPEG-2, where field mode processing in a picture that is coded as a frame results in the processing of 16×8 half-macroblocks).
  - Picture-adaptive frame-field coding (PAFF or PicAFF) allowing a freely selected mixture of pictures coded either as complete frames where both fields are combined together for encoding or as individual single fields.
- New transform design features, including:
  - An exact-match integer 4×4 spatial block transform, allowing precise placement of residual signals with little of the "ringing" often found with prior codec designs. This design is conceptually similar to that of the well-known discrete cosine transform (DCT), introduced in 1974 by N. Ahmed, T.Natarajan and K.R.Rao, which is Citation 1 in Discrete cosine transform. However, it is simplified and made to provide exactly specified decoding.
  - An exact-match integer 8×8 spatial block transform, allowing highly correlated regions to be compressed more efficiently than with the 4×4 transform. This design is conceptually similar to that of the well-known DCT, but simplified and made to provide exactly specified decoding.
  - Adaptive encoder selection between the 4×4 and 8×8 transform block sizes for the integer transform operation.
  - A secondary Hadamard transform performed on "DC" coefficients of the primary spatial transform applied to chroma DC coefficients (and also luma in one special case) to obtain even more compression in smooth regions.
- A quantization design including:
  - Logarithmic step size control for easier bit rate management by encoders and simplified inverse-quantization scaling
  - Frequency-customized quantization scaling matrices selected by the encoder for perceptual-based quantization optimization
- An in-loop deblocking filter that helps prevent the blocking artifacts common to other DCT-based image compression techniques, resulting in better visual appearance and compression efficiency
- An entropy coding design including:
  - Context-adaptive binary arithmetic coding (CABAC), an algorithm to losslessly compress syntax elements in the video stream knowing the probabilities of syntax elements in a given context. CABAC compresses data more efficiently than CAVLC but requires considerably more processing to decode.
  - Context-adaptive variable-length coding (CAVLC), which is a lower-complexity alternative to CABAC for the coding of quantized transform coefficient values. Although lower complexity than CABAC, CAVLC is more elaborate and more efficient than the methods typically used to code coefficients in other prior designs.
  - A common simple and highly structured variable length coding (VLC) technique for many of the syntax elements not coded by CABAC or CAVLC, referred to as Exponential-Golomb coding (or Exp-Golomb).
- Loss resilience features including:
  - A Network Abstraction Layer (NAL) definition allowing the same video syntax to be used in many network environments. One very fundamental design concept of H.264 is to generate self-contained packets, to remove the header duplication as in MPEG-4's Header Extension Code (HEC).[22] This was achieved by decoupling information relevant to more than one slice from the media stream. The combination of the higher-level parameters is called a parameter set.[22] The H.264 specification includes two types of parameter sets: Sequence Parameter Set (SPS) and Picture Parameter Set (PPS). An active sequence parameter set remains unchanged throughout a coded video sequence, and an active picture parameter set remains unchanged within

  a coded picture. The sequence and picture parameter set structures contain information such as picture size, optional coding modes employed, and macroblock to slice group map.[22]
  - Flexible macroblock ordering (FMO), also known as slice groups, and arbitrary slice ordering (ASO), which are techniques for restructuring the ordering of the representation of the fundamental regions (*macroblocks*) in pictures. Typically considered an error/loss robustness feature, FMO and ASO can also be used for other purposes.
  - Data partitioning (DP), a feature providing the ability to separate more important and less important syntax elements into different packets of data, enabling the application of unequal error protection (UEP) and other types of improvement of error/loss robustness.
  - Redundant slices (RS), an error/loss robustness feature that lets an encoder send an extra representation of a picture region (typically at lower fidelity) that can be used if the primary representation is corrupted or lost.
  - Frame numbering, a feature that allows the creation of "sub-sequences", enabling temporal scalability by optional inclusion of extra pictures between other pictures, and the detection and concealment of losses of entire pictures, which can occur due to network packet losses or channel errors.
- Switching slices, called SP and SI slices, allowing an encoder to direct a decoder to jump into an ongoing video stream for such purposes as video streaming bit rate switching and "trick mode" operation. When a decoder jumps into the middle of a video stream using the SP/SI feature, it can get an exact match to the decoded pictures at that location in the video stream despite using different pictures, or no pictures at all, as references prior to the switch.
- A simple automatic process for preventing the accidental emulation of start codes, which are special sequences of bits in the coded data that allow random access into the bitstream and recovery of byte alignment in systems that can lose byte synchronization.
- Supplemental enhancement information (SEI) and video usability information (VUI), which are extra information that can be inserted into the bitstream to enhance the use of the video for a wide variety of purposes.Wikipedia:Please clarify SEI FPA (Frame Packing Arrangement) message that contains the 3D arrangement:
  - 0: checkerboard - pixels are alternatively from L and R
  - 1: column alternation - L and R are interlaced by column
  - 2: row alternation - L and R are interlaced by row
  - 3: side by side - L is on the left, R on the right
  - 4: top bottom - L is on top, R on bottom
  - 5: frame alternation - one view per frame
- Auxiliary pictures, which can be used for such purposes as alpha compositing.
- Support of monochrome (4:0:0), 4:2:0, 4:2:2, and 4:4:4 chroma subsampling (depending on the selected profile).
- Support of sample bit depth precision ranging from 8 to 14 bits per sample (depending on the selected profile).
- The ability to encode individual color planes as distinct pictures with their own slice structures, macroblock modes, motion vectors, etc., allowing encoders to be designed with a simple parallelization structure (supported only in the three 4:4:4-capable profiles).
- Picture order count, a feature that serves to keep the ordering of the pictures and the values of samples in the decoded pictures isolated from timing information, allowing timing information to be carried and controlled/changed separately by a system without affecting decoded picture content.

These techniques, along with several others, help H.264 to perform significantly better than any prior standard under a wide variety of circumstances in a wide variety of application environments. H.264 can often perform radically better than MPEG-2 video—typically obtaining the same quality at half of the bit rate or less, especially on high bit rate and high resolution situations.[23]

Like other ISO/IEC MPEG video standards, H.264/AVC has a reference software implementation that can be freely downloaded.[24] Its main purpose is to give examples of H.264/AVC features, rather than being a useful application *per se*. Some reference hardware design work is also under way in the Moving Picture Experts Group. The above mentioned are complete features of H.264/AVC covering all profiles of H.264. A profile for a codec is a set of features of that codec identified to meet a certain set of specifications of intended applications. This means that many of the features listed are not supported in some profiles. Various profiles of H.264/AVC are discussed in next section.

Profiles

The standard defines 21 sets of capabilities, which are referred to as *profiles*, targeting specific classes of applications.

Profiles for non-scalable 2D video applications include the following:

Constrained Baseline Profile (CBP)

Primarily for low-cost applications, this profile is most typically used in videoconferencing and mobile applications. It corresponds to the subset of features that are in common between the Baseline, Main, and High Profiles.

Baseline Profile (BP)

Primarily for low-cost applications that require additional data loss robustness, this profile is used in some videoconferencing and mobile applications. This profile includes all features that are supported in the Constrained Baseline Profile, plus three additional features that can be used for loss robustness (or for other purposes such as low-delay multi-point video stream compositing). The importance of this profile has faded somewhat since the definition of the Constrained Baseline Profile in 2009. All Constrained Baseline Profile bitstreams are also considered to be Baseline Profile bitstreams, as these two profiles share the same profile identifier code value.

Main Profile (MP)

This profile is used for standard-definition digital TV broadcasts that use the MPEG-4 format as defined in the DVB standard.[25] It is not, however, used for high-definition television broadcasts, as the importance of this profile faded when the High Profile was developed in 2004 for that application.

Extended Profile (XP)

Intended as the streaming video profile, this profile has relatively high compression capability and some extra tricks for robustness to data losses and server stream switching.

High Profile (HiP)

The primary profile for broadcast and disc storage applications, particularly for high-definition television applications (for example, this is the profile adopted by the Blu-ray Disc storage format and the DVB HDTV broadcast service).

Progressive High Profile (PHiP)

Similar to the High profile, but without support of field coding features.

Constrained High Profile

Similar to the Progressive High profile, but without support of B (bi-predictive) slices.

High 10 Profile (Hi10P)

Going beyond typical mainstream consumer product capabilities, this profile builds on top of the High Profile, adding support for up to 10 bits per sample of decoded picture precision.

High 4:2:2 Profile (Hi422P)

Primarily targeting professional applications that use interlaced video, this profile builds on top of the High 10 Profile, adding support for the 4:2:2 chroma subsampling format while using up to 10 bits per sample of decoded picture precision.

High 4:4:4 Predictive Profile (Hi444PP)

This profile builds on top of the High 4:2:2 Profile, supporting up to 4:4:4 chroma sampling, up to 14 bits per sample, and additionally supporting efficient lossless region coding and the coding of each picture as three separate color planes.

For camcorders, editing, and professional applications, the standard contains four additional Intra-frame-only profiles, which are defined as simple subsets of other corresponding profiles. These are mostly for professional (e.g., camera and editing system) applications:

High 10 Intra Profile

The High 10 Profile constrained to all-Intra use.

High 4:2:2 Intra Profile

The High 4:2:2 Profile constrained to all-Intra use.

High 4:4:4 Intra Profile

The High 4:4:4 Profile constrained to all-Intra use.

CAVLC 4:4:4 Intra Profile

The High 4:4:4 Profile constrained to all-Intra use and to CAVLC entropy coding (i.e., not supporting CABAC).

As a result of the Scalable Video Coding (SVC) extension, the standard contains five additional *scalable profiles*, which are defined as a combination of a H.264/AVC profile for the base layer (identified by the second word in the scalable profile name) and tools that achieve the scalable extension:

Scalable Baseline Profile

Primarily targeting video conferencing, mobile, and surveillance applications, this profile builds on top of the Constrained Baseline profile to which the base layer (a subset of the bitstream) must conform. For the scalability tools, a subset of the available tools is enabled.

Scalable Constrained Baseline Profile

A subset of the Scalable Baseline Profile intended primarily for real-time communication applications.

Scalable High Profile

Primarily targeting broadcast and streaming applications, this profile builds on top of the H.264/AVC High Profile to which the base layer must conform.

Scalable Constrained High Profile

A subset of the Scalable High Profile intended primarily for real-time communication applications.

Scalable High Intra Profile

Primarily targeting production applications, this profile is the Scalable High Profile constrained to all-Intra use.

As a result of the Multiview Video Coding (MVC) extension, the standard contains two *multiview profiles*:

Stereo High Profile

This profile targets two-view stereoscopic 3D video and combines the tools of the High profile with the inter-view prediction capabilities of the MVC extension.

Multiview High Profile

This profile supports two or more views using both inter-picture (temporal) and MVC inter-view prediction, but does not support field pictures and macroblock-adaptive frame-field coding.

Multiview Depth High Profile

Feature support in particular profiles

Feature	CBP	BP	XP	MP	ProHiP	HiP	Hi10P	Hi422P	Hi444PP
Chroma formats	4:2:0	4:2:0	4:2:0	4:2:0	4:2:0	4:2:0	4:2:0	4:2:0/4:2:2	4:2:0/4:2:2/4:4:4
Sample depths (bits)	8	8	8	8	8	8	8 to 10	8 to 10	8 to 14
Flexible macroblock ordering (FMO)	No	Yes	Yes	No	No	No	No	No	No
Arbitrary slice ordering (ASO)	No	Yes	Yes	No	No	No	No	No	No
Redundant slices (RS)	No	Yes	Yes	No	No	No	No	No	No
Data Partitioning	No	No	Yes	No	No	No	No	No	No
SI and SP slices	No	No	Yes	No	No	No	No	No	No
Interlaced coding (PicAFF, MBAFF)	No	No	Yes	Yes	No	Yes	Yes	Yes	Yes
B slices	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CABAC entropy coding	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
8×8 vs. 4×4 transform adaptivity	No	No	No	No	Yes	Yes	Yes	Yes	Yes
Quantization scaling matrices	No	No	No	No	Yes	Yes	Yes	Yes	Yes
Separate C_b and C_r QP control	No	No	No	No	Yes	Yes	Yes	Yes	Yes
Monochrome (4:0:0)	No	No	No	No	Yes	Yes	Yes	Yes	Yes
Separate color plane coding	No	No	No	No	No	No	No	No	Yes
Predictive lossless coding	No	No	No	No	No	No	No	No	Yes

Levels

As the term is used in the standard, a "*level*" is a specified set of constraints that indicate a degree of required decoder performance for a profile. For example, a level of support within a profile specifies the maximum picture resolution, frame rate, and bit rate that a decoder may use. A decoder that conforms to a given level must be able to decode all bitstreams encoded for that level and all lower levels.

Levels with maximum property values

Level	Max decoding speed		Max frame size		Max video bit rate for video coding layer (VCL) kbit/s			Examples for high resolution @ highest frame rate (max stored frames) Toggle additional details
	Luma samples/s	Macroblocks/s	Luma samples	Macroblocks	Baseline, Extended and Main Profiles	High Profile	High 10 Profile	
1	380,160	1,485	25,344	99	64	80	192	128×96@30.9 (8) 176×144@15.0 (4)
1b	380,160	1,485	25,344	99	128	160	384	128×96@30.9 (8) 176×144@15.0 (4)
1.1	768,000	3,000	101,376	396	192	240	576	176×144@30.3 (9) 320×240@10.0 (3) 352×288@7.5 (2)
1.2	1,536,000	6,000	101,376	396	384	480	1,152	320×240@20.0 (7) 352×288@15.2 (6)

1.3	3,041,280	11,880	101,376	396	768	960	2,304	320×240@36.0 (7) 352×288@30.0 (6)
2	3,041,280	11,880	101,376	396	2,000	2,500	6,000	320×240@36.0 (7) 352×288@30.0 (6)
2.1	5,068,800	19,800	202,752	792	4,000	5,000	12,000	352×480@30.0 (7) 352×576@25.0 (6)
2.2	5,184,000	20,250	414,720	1,620	4,000	5,000	12,000	352×480@30.7(10) 352×576@25.6 (7) 720×480@15.0 (6) 720×576@12.5 (5)
3	10,368,000	40,500	414,720	1,620	10,000	12,500	30,000	352×480@61.4 (12) 352×576@51.1 (10) 720×480@30.0 (6) 720×576@25.0 (5)
3.1	27,648,000	108,000	921,600	3,600	14,000	17,500	42,000	720×480@80.0 (13) 720×576@66.7 (11) 1280×720@30.0 (5)
3.2	55,296,000	216,000	1,310,720	5,120	20,000	25,000	60,000	1,280×720@60.0 (5) 1,280×1,024@42.2 (4)
4	62,914,560	245,760	2,097,152	8,192	20,000	25,000	60,000	1,280×720@68.3 (9) 1,920×1,080@30.1 (4) 2,048×1,024@30.0 (4)
4.1	62,914,560	245,760	2,097,152	8,192	50,000	62,500	150,000	1,280×720@68.3 (9) 1,920×1,080@30.1 (4) 2,048×1,024@30.0 (4)
4.2	133,693,440	522,240	2,228,224	8,704	50,000	62,500	150,000	1,280×720@145.1 (9) 1,920×1,080@64.0 (4) 2,048×1,080@60.0 (4)
5	150,994,944	589,824	5,652,480	22,080	135,000	168,750	405,000	1,920×1,080@72.3 (13) 2,048×1,024@72.0 (13) 2,048×1,080@67.8 (12) 2,560×1,920@30.7 (5) 3,672×1,536@26.7 (5)
5.1	251,658,240	983,040	9,437,184	36,864	240,000	300,000	720,000	1,920×1,080@120.5 (16) 2,560×1,920@51.2 (9) 3,840×2,160@31.7 (5) 4,096×2,048@30.0 (5) 4,096×2,160@28.5 (5) 4,096×2,304@26.7 (5)
5.2	530,841,600	2,073,600	9,437,184	36,864	240,000	300,000	720,000	1,920×1,080@172.0 (16) 2,560×1,920@108.0 (9) 3,840×2,160@66.8 (5) 4,096×2,048@63.3 (5) 4,096×2,160@60.0 (5) 4,096×2,304@56.3 (5)

The maximum bit rate for High Profile is 1.25 times that of the Base/Extended/Main Profiles, 3 times for Hi10P, and 4 times for Hi422P/Hi444PP.

The number of luma samples is 16x16=256 times the number of macroblocks (and the number of luma samples per second is 256 times the number of macroblocks per second).

Decoded picture buffering

Previously encoded pictures are used by H.264/AVC encoders to provide predictions of the values of samples in other pictures. This allows the encoder to make efficient decisions on the best way to encode a given picture. At the decoder, such pictures are stored in a virtual *decoded picture buffer* (DPB). The maximum capacity of the DPB is in units of frames (or pairs of fields), as shown in parentheses in the right column of the table above, can be computed as follows:

Standard equation	`Min(Floor(MaxDpbMbs / (PicWidthInMbs * FrameHeightInMbs)), 16)`
Excel-compatible formula	`=MIN(FLOOR(MaxDpbMbs / (PicWidthInMbs * FrameHeightInMbs); 1); 16)`

Where MaxDpbMbs is a constant value provided in the table below as a function of level number, and PicWidthInMbs and FrameHeightInMbs are the picture width and frame height for the coded video data, expressed in units of macroblocks (rounded up to integer values and accounting for cropping and macroblock pairing when applicable). This formula is specified in sections A.3.1.h and A.3.2.f of the 2009 edition of the standard.

Level	1	1b	1.1	1.2	1.3	2	2.1	2.2	3	3.1	3.2	4	4.1	4.2	5	5.1	5.2
MaxDpbMbs	396	396	900	2,376	2,376	2,376	4,752	8,100	8,100	18,000	20,480	32,768	32,768	34,816	110,400	184,320	184,320

For example, for an HDTV picture that is 1920 samples wide (PicWidthInMbs = 120) and 1080 samples high (FrameHeightInMbs = 68), a Level 4 decoder has a maximum DPB storage capacity of Floor(32768/(120*68)) = 4 frames (or 8 fields) when encoded with minimal cropping parameter values. Thus, the value 4 is shown in parentheses in the table above in the right column of the row for Level 4 with the frame size 1920×1080.

It is important to note that the current picture being decoded is *not included* in the computation of DPB fullness (unless the encoder has indicated for it to be stored for use as a reference for decoding other pictures or for delayed output timing). Thus, a decoder needs to actually have sufficient memory to handle (at least) one frame *more* than the maximum capacity of the DPB as calculated above.

Versions

Versions of the H.264/AVC standard include the following completed revisions, corrigenda, and amendments (dates are final approval dates in ITU-T, while final "International Standard" approval dates in ISO/IEC are somewhat different and slightly later in most cases). Each version represents changes relative to the next lower version that is integrated into the text. **Bold faced versions** are versions that include relatively major technical enhancements.

- **Version 1:** (May 30, 2003) First approved version of H.264/AVC containing **Baseline, Main, and Extended** profiles.[]
- Version 2: (May 7, 2004) Corrigendum containing various minor corrections.[]
- **Version 3:** (March 1, 2005) Major addition to H.264/AVC containing the first amendment providing **Fidelity Range Extensions** (FRExt) containing **High**, **High 10**, **High 4:2:2**, and **High 4:4:4** profiles.[]
- Version 4: (September 13, 2005) Corrigendum containing various minor corrections and adding three **aspect ratio indicators**.[]
- Version 5: (June 13, 2006) Amendment consisting of removal of prior **High 4:4:4** profile (processed as a corrigendum in ISO/IEC).[]
- Version 6: (June 13, 2006) Amendment consisting of minor extensions like **extended-gamut color space support** (bundled with above-mentioned **aspect ratio indicators** in ISO/IEC).[]

- Version 7: (April 6, 2007) Amendment containing the addition of **High 4:4:4 Predictive** and four **Intra-only** profiles (**High 10 Intra**, **High 4:2:2 Intra**, **High 4:4:4 Intra**, and **CAVLC 4:4:4 Intra**).[]
- **Version 8:** (November 22, 2007) Major addition to H.264/AVC containing the amendment for **Scalable Video Coding** (SVC) containing **Scalable Baseline**, **Scalable High**, and **Scalable High Intra** profiles.[]
- Version 9: (January 13, 2009) Corrigendum containing minor corrections.[]
- Version 10: (March 16, 2009) Amendment containing definition of a new profile (the **Constrained Baseline** profile) with only the common subset of capabilities supported in various previously specified profiles.[]
- **Version 11:** (March 16, 2009) Major addition to H.264/AVC containing the amendment for **Multiview Video Coding** (MVC) extension, including the **Multiview High** profile.[]
- Version 12: (March 9, 2010) Amendment containing definition of a new MVC profile (the **Stereo High** profile) for two-view video coding with support of interlaced coding tools and specifying an additional SEI message (the **frame packing arrangement SEI message**).[]
- Version 13: (March 9, 2010) Corrigendum containing minor corrections.[]
- Version 14: (June 29, 2011) Amendment specifying a new level (**Level 5.2**) supporting higher processing rates in terms of maximum macroblocks per second, and a new profile (the **Progressive High profile**) supporting only the frame coding tools of the previously specified High profile.[]
- Version 15: (June 29, 2011) Corrigendum containing minor corrections.[]
- Version 16: (January 13, 2012) Amendment containing definition of three new profiles intended primarily for real-time communication applications: the **Constrained High**, **Scalable Constrained Baseline**, and **Scalable Constrained High** profiles.[]
- Version 17: (April 13, 2013) Amendment with additional SEI message indicators.[]
- Version 18: (April 13, 2013) Amendment to specify the coding of depth map data for 3D stereoscopic video, including a **Multiview Depth High** profile.[]
- Version 19: (April 13, 2013) Corrigendum to correct an error in the sub-bitstream extraction process for multiview video.[]
- Version 20: (April 13, 2013) Amendment to specify additional color space identifiers (including support of ITU-R Recommendation BT.2020 for UHDTV) and an additional model type in the tone mapping information SEI message.[]

Software encoder feature comparison

AVC software implementations

Feature	QT	Nero	LEAD	x264	MainConcept	Elecard	TSE	VSofts	ProCoder	Avivo	Elemental	IPP
B slices	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
SI and SP slices	No	No	No	No	No	No	No	No	No	No	No	No
Multiple reference frames	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Flexible Macroblock Ordering (FMO)	No	No	No	No	No	No	No	Yes	No	No	No	No
Arbitrary slice ordering (ASO)	No	No	No	No	No	No	No	No	No	No	No	No

Redundant slices (RS)	No	No	No	No	No	No	No	No	No	No	No	No
Data partitioning	No	No	No	No	No	No	No	No	No	No	No	No
Interlaced coding (PicAFF, MBAFF)	No	MBAFF	MBAFF	MBAFF	Yes	Yes	No	MBAFF	Yes	MBAFF	Yes	No
CABAC entropy coding	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
8×8 vs. 4×4 transform adaptivity	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Quantization scaling matrices	No	No	No	Yes	Yes	No	No	Yes	No	No	No	No
Separate C_b and C_r QP control	No	No	No	Yes	Yes	Yes	No	Yes	No	No	No	No
Monochrome (4:0:0)	No	No	No	No	No	Yes	No	Yes	No	No	No	No
Extended chroma formats	No	No	No	4:2:2[26]/4:4:4[27]	4:2:2	4:2:2	4:2:2	4:2:2/4:4:4	No	No	4:2:0, 4:2:2	No
Largest sample depth (bit)	8	8	8	10[28]	10	8	8	10	8	8	10	12
Separate color plane coding	No	No	No	No	No	No	No	No	No	No	No	No
Predictive lossless coding	No	No	No	Yes[29]	No	No	No	No	No	No	No	No
Film grain modeling	No	No	No	No	No	No	No	Yes	No	No	No	No

Hardware-based encoding and decoding

Because H.264 encoding and decoding requires significant computing power in specific types of arithmetic operations, software implementations that run on general-purpose CPUs are typically less power efficient. However, the latest quad-core general-purpose x86 CPUs have sufficient computation power to perform real-time SD and HD encoding. Compression efficiency depends on video algorithmic implementations, not on whether hardware or software implementation is used. Therefore, the difference between hardware and software based implementation is more on power-efficiency, flexibility and cost. To improve the power efficiency and reduce hardware form-factor, special-purpose hardware may be employed, either for the complete encoding or decoding process, or for acceleration assistance within a CPU-controlled environment.

CPU based solutions are known to be much more flexible, particularly when encoding must be done concurrently in multiple formats, multiple bit rates and resolutions (multi-screen video), and possibly with additional features on container format support, advanced integrated advertising features, etc. CPU based software solution generally makes it much easier to load balance multiple concurrent encoding sessions within the same CPU.

The 2nd generation Intel "Sandy Bridge" Core i3/i5/i7 processors introduced at the January 2011 CES (Consumer Electronics Show) offer an on-chip hardware full HD H.264 encoder, known as Intel Quick Sync Video.[30][31]

A hardware H.264 encoder can be an ASIC or an FPGA. An FPGA is a general programmable chip. To use an FPGA as a hardware encoder, an H.264 encoder design is required to customize the chip for the application. A full HD H.264 encoder could run on a single low cost FPGA chip by 2009 (High profile, level 4.1, 1080p, 30fps).

ASIC encoders with H.264 encoder functionality are available from many different semiconductor companies, but the core design used in the ASIC is typically licensed from one of a few companies such as Chips&Media, On2 (formerly Hantro, acquired by Google), Imagination Technologies, NGCodec. Some companies have both FPGA and ASIC product offerings.[32]

Texas Instruments manufactures a line of ARM + DSP cores that perform DSP H264 BP encoding 1080p at 30fps.[33] This permits flexibility with respect to codecs (which are implemented as highly optimized DSP code) while being more efficient than software on a generic CPU.

References

[2] Joint Video Team (http://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jvt.aspx), ITU-T web site.

[9] http://www.osnews.com/story/24954/US_Patent_Expiration_for_MP3_MPEG-2_H_264 has a MPEG-LA patent US 7826532 that was filed in Sep. 5, 2003 and has a 1546 day term extension. http://patft1.uspto.gov/netacgi/nph-Parser?patentnumber=7826532 http://www.google.com/patents/about?id=2onYAAAAEBAJ

[13] http://www.google.com/patents/US5452104

[14] http://www.google.com/patents/US5576767

[15] See Qualcomm Inc. v. Broadcom Corp. (http://caselaw.lp.findlaw.com/data2/circs/fed/071545p.pdf), No. 2007-1545, 2008-1162 (Fed. Cir. Dec. 1, 2008). For articles in the popular press, see signonsandiego.com, "Qualcomm loses its patent-rights case" (http://www.signonsandiego.com/news/business/20070127-9999-1b27verdict.html) and "Qualcomm's patent case goes to jury" (http://www.signonsandiego.com/news/business/20070126-9999-1b26qualcomm.html); and bloomberg.com "Broadcom Wins First Trial in Qualcomm Patent Dispute" (http://www.bloomberg.com/apps/news?pid=20601204&sid=aLX_DFMCEYWU&refer=technology)

[22] RFC 3984, p.3

[26] "x264 4:2:2 encoding support" (http://git.videolan.org/?p=x264.git;a=commit;h=bb9216dc319a39eb6f2a5508a98e36d6492ffa7e), Retrieved 2011-09-22.

[27] "x264 4:4:4 encoding support" (http://git.videolan.org/?p=x264.git;a=commit;h=c07df2f77dacfec444785bf60229f804dc43c10b), Retrieved 2011-06-22.

[28] "x264 support for 9 and 10-bit encoding" (http://git.videolan.org/?p=x264.git;a=commit;h=d058f37d9af8fc425fa0626695a190eb3aa032af), Retrieved 2011-06-22.

[29] "x264 replace High 4:4:4 profile lossless with High 4:4:4 Predictive" (http://git.videolan.org/?p=x264.git;a=commit;h=b35a044b95c7eab2c91f55a3ac4100ca26a29d92), Retrieved 2011-06-22.

Further reading

- Wiegand, Thomas; Sullivan, Gary J.; Bjøntegaard, Gisle; Luthra, Ajay (July 2003). "Overview of the H.264/AVC Video Coding Standard" (http://ip.hhi.de/imagecom_G1/assets/pdfs/csvt_overview_0305.pdf) (PDF). *IEEE Transactions on Circuits and Systems for Video Technology* **13** (7). Retrieved 31 January 2011.
- Topiwala, Pankaj; Sullivan, Gary J.; Luthra, Ajay (August 2004). "Overview and Introduction to the Fidelity Range Extensions" (http://www.fastvdo.com/spie04/spie04-h264OverviewPaper.pdf) (PDF). *SPIE Applications of Digital Image Processing XXVII*. Retrieved 31 January 2011.
- Ostermann, J.; Bormans, J.; List, P.; Marpe, D.; Narroschke, M.; Pereira, F.; Stockhammer, T.; Wedi, T. (First Quarter 2004). "Video coding with H.264/AVC: Tools, Performance, and Complexity" (ftp://ftp.tnt.uni-hannover.de/pub/papers/2004/CASM_2004_MN.pdf) (PDF). *IEEE Circuits and Systems Magazine* **4** (1). Retrieved 31 January 2011.

- Sullivan, Gary J.; Wiegand, Thomas (January 2005). "Video Compression—From Concepts to the H.264/AVC Standard" (http://ip.hhi.de/imagecom_G1/assets/pdfs/pieee_sullivan_wiegand_2005.pdf) (PDF). *Proceedings of the IEEE* **93** (1). Retrieved 31 January 2011.
- Richardson, Iain E. G. (2011 January). "Learn about video compression and H.264" (http://www.vcodex.com/h264.html). *VCODEX*. Vcodex Limited. Retrieved 31 January 2011.

External links

- ITU-T publication page: H.264: Advanced video coding for generic audiovisual services (http://www.itu.int/rec/T-REC-H.264)
- MPEG-4 AVC/H.264 Information (http://forum.doom9.org/showthread.php?t=96059) Doom9's Forum
- "Part 10: Advanced Video Coding" (http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=56538). *ISO publication page: ISO/IEC 14496-10:2010 – Information technology — Coding of audio-visual objects*.
- "H.264/AVC JM Reference Software" (http://iphome.hhi.de/suehring/tml/). *IP Homepage*. Retrieved 2007-04-15.
- "JVT document archive site" (http://ftp3.itu.ch/av-arch/jvt-site). Retrieved 2007-05-06.
- "Publications" (http://iphome.hhi.de/wiegand/pubs.htm). Thomas Wiegand. Retrieved 2007-06-23.
- "Publications" (http://iphome.hhi.de/marpe/pub.htm). Detlev Marpe. Retrieved 2007-04-15.
- "Fourth Annual H.264 video codecs comparison" (http://www.compression.ru/video/codec_comparison/mpeg-4_avc_h264_2007_en.html). Moscow State University. (dated December 2007)
- "Discussion on H.264 with respect to IP cameras in use within the security and surveillance industries" (http://www.networkwebcams.com/ip-camera-learning-center/2009/04/03/using-h264-video-compression-in-ip-video-surveillance-systems/). (dated April 2009)
- "Diary Of An x264 Developer: Why so many H.264 encoders are bad" (http://x264dev.multimedia.cx/?p=164). 10/04/2009, – practical implementation issues of H.264
- "Sixth Annual H.264 video codecs comparison" (http://compression.ru/video/codec_comparison/h264_2010/). Moscow State University. (dated May 2010)

Article Sources and Contributors

H.264/MPEG-4 AVC *Source*: http://en.wikipedia.org/w/index.php?oldid=564202076 *Contributors*: 2601:7:100:5E:222:FBFF:FE53:B8EC, A333, ARC Gritt, ASBands, AVB, Ablewisuk, Acdx, Adam Conover, Adamd1008, Aeiuthhiet, Ahoerstemeier, Aims3Bor, Alereon, Algocu, AlistairMcMillan, AlphaPyro, AlyM, Andemedia, Andres, Asided m plane, Asierra, AxG, BMF81, Babyworm, Bald Zebra, Ballplyr86, Barrylb, Barte, Bassower, Beland, Bender235, Berkut, Betaboy, Bevo, Biggiesized, Bluebretagne, Bobble35, Bobblewik, Bongdentoiac, Boothy443, Bork, Briantist, Brindy, Bryan Derksen, Bsautter, CCTVguru, CLW, Can't sleep, clown will eat me, Cantalamessa, Cat5nap, Cctv guy, Cedric.lacrambe, Charliemccay, Chiprunner, Chowells, ChrisRuvolo, Chuck-meister, Closedmouth, Clueless, Cluth, Coinchon, Coldfire82, Corella, Cpufreak2589, Credema, Crissov, Cschalle, Czarkoff, D3, Dadu, Dan aka jack, Daniel C, Daniel G., Daniel.Cardenas, Darimee, Dark Shikari, David Gerard, David Levy, Demitsu, Dewritech, Dgtsyb, DhivehiRaajje, Digital digest, Digital dinosaur, Digitalme, Discospinster, Dittaeva, DmitriyV, DmitryKo, DocWatson42, Doradus, Doubleyouyou, DrJolo, Dreadloco, Dulldull, Dynamo ace, E.P.I.C., EEMIV, EagleOne, Eatrains, Ebohman, Edcolins, Edwtie, Ekyygork, Elange, EliasAlucard, Eliyahu S, Endorf, Erpel13, Esoj zeravla, Esuvorkin, Etiwiki, Etu, FalseAlarm, Farooqn, Father Goose, Ferenczy, Fighuhldz, FirstPrinciples, FleetCommand, Fleminra, Flux.books, FogDevil, Fpmfpm, Franl, Frap, Fred Bauder, Freeone3000, Fudoreaper, Funandtrvl, Fuzheado, G00nsf, GChriss, Gabbe, Gadget850, Gaius Cornelius, Galwhaa, Gerardpons, Ghakko, Gij, Graham87, GrandDrake, Grazfather, Greg L, GregorB, Griessn, Gronky, Gsmgm, GufNZ, Guiddruid, Gurch, Gurnec, HIGH44576I, Hadal, Hairy Dude, Hargrimm, Harryzilber, Hemidemisemiquaver, Hgfernan, Hsj13, Ipsupermarket, Ipsurveillance, Irdepesca572, Isnow, J Darnley, J knowlton tx, J. M., J.delanoy, JSteinseiffer, Janitscharen, Janta, Jantangring, Javsav, Jcarroll, Jeffq, Jesse Viviano, Jesup, Jeysaba, Jim.Callahan,Orlando, Jim.henderson, Jkolak, Jogfalls1947, John254, Jol123, Jonabbey, Jonasmike, Joshua Issac, Jossi, Joz3d, Jrincayc, Judith6yu, Julesd, Justin The Claw, Jvte, Jvting, Jzhang, KAMiKAZOW, Kadavill, Kadin2048, Kakesu, Kaldari, KamasamaK, Kanryu, Katimawan2005, KelleyCook, Kiranwashindkar, Koavf, Kocio, Koenige, Kruador, Kungming2, Kvng, LEADTechnologies, Lemoncurd, Lightmouse, LilHelpa, Limulus, Llloic, Lotje, Luismbs, MER-C, MLeb, Ma Baker, Magioladitis, Mahuhn, MajorVariola, Manabu, Mandolinface, ManiacK, Mannies, Marco Krohn, MarkWahl, Martin196r, Martnym, Matariel, MatthiasBock, Mauls, MaxEnt, McGeddon, Mellum, Mendaliv, Metalim, Mhorchler, Michael Slana, Microwaven, Mintleaf, Monika Ceskova Fourneaux, Mooncow, Morton.lin, Motfellow, MrOllie, Mrand, Msikma, Mufka, Mulligatawny, Muéro, MySchizoBuddy, Nabbia, Nakomaru, Namazu-tron, Nasa-verve, NawlinWiki, Nbarth, Ned Scott, NetRolller 3D, Nextop, Nick.kinson, Nickshanks, Niri.M, Nixdorf, Njuuton, Nonet84, Notmicro, Nowakpl, Nuprin, Onceler, Ooikhaichin, Oosh, OriginalGamer, OroCHU, OverlordQ, Ozhiker, Palosirkka, Pangolin, Passer, Paul1337, Paulirwin, Pawnbroker, Paxse, Pdelong, Pdenapo, Pengvado, Pez Bower, Phatom87, Philwelch, Pig837, Plonk420, Pmsyyz, Poee, Qnaalor, Quadra630, Quahog21, Quarl, Quebec99, Quoth, Qutezuce, Qxz, RBFN, RJASE1, Radiojon, Ravana 999, Rcooley, Regression Tester, Requestion, Rettetast, Rhobite, Richardcavell, Ricky81682, Rjsquirrel, Rjwilmsi, Robert K S, Robert Will, Rockfang, Roguegeek, Ronark, Rrw, Ruud Koot, Rvalles, Ryanaxp, SDC, Salam32, Sapibobo, Savagezhang, Scepia, Schwans, Scientus, Scolen2, Sdornan, SeyedKevin, SharkD, Shawnc, Shirulashem, Sigidagirl, Simaocampos, Simonhorlick, Siway, Slakr, Smadim2, Snacky, Sockatume, Spellmaster, Squilibob, Srleffler, Ssfreitas, Stephan Leeds, Steve.ruckdashel, Stevenj, Stuinzuri, SudoMonas, Sujayt, Svick, Techie007, Teque5, Tetromino, TexasDig, Th3typh00n, Thatdog, The Anome, TheAMmollusc, TheArv, TheMandarin, Thebrid, Thechoirlife2006, Thechristian, Theuns.verwoerd, Thewikipedian, ThreeOneFive, Thue, Thumperward, Ticapix, Tide rolls, Tir-Gwaith, Titoxd, Tom W.M., Topbanana, Tpbradbury, Ttmcmurry, Twistor96, Uncle Milty, Unfree, Unixman83, Vcodex, VictorAnyakin, Videogenie, ViperSnake151, Vitz-RS, VladimirKorablin, VoluntarySlave, VoxLuna, Vpbpi, Vssmike, W.F.Galway, WLGades, Wa2ise, Wallsbk, Wavelength, Wdunaway, WhiplashInferno, Whkoh, Widefox, Wiki alf, WikiBone, Wikifier, Wikilolo, Wikkiwitchh, Wilsone9, Wmahan, Woohookitty, Wookipedian, X264hack, XMog, Yueh, Zachdms, Zhangji88, Zoltan1, Zundark, Ὁ οἶστρος, ねこちゃん, 932 anonymous edits

License