June 2005

MPEG-4 Part 10 AVC (H.264) Video Encoding

Abstract

H.264 has the potential to revolutionize the industry as it eases the bandwidth burden of service delivery and opens the service provider market to new players. This document describes the process of H.264 encoding and transmission over IP.

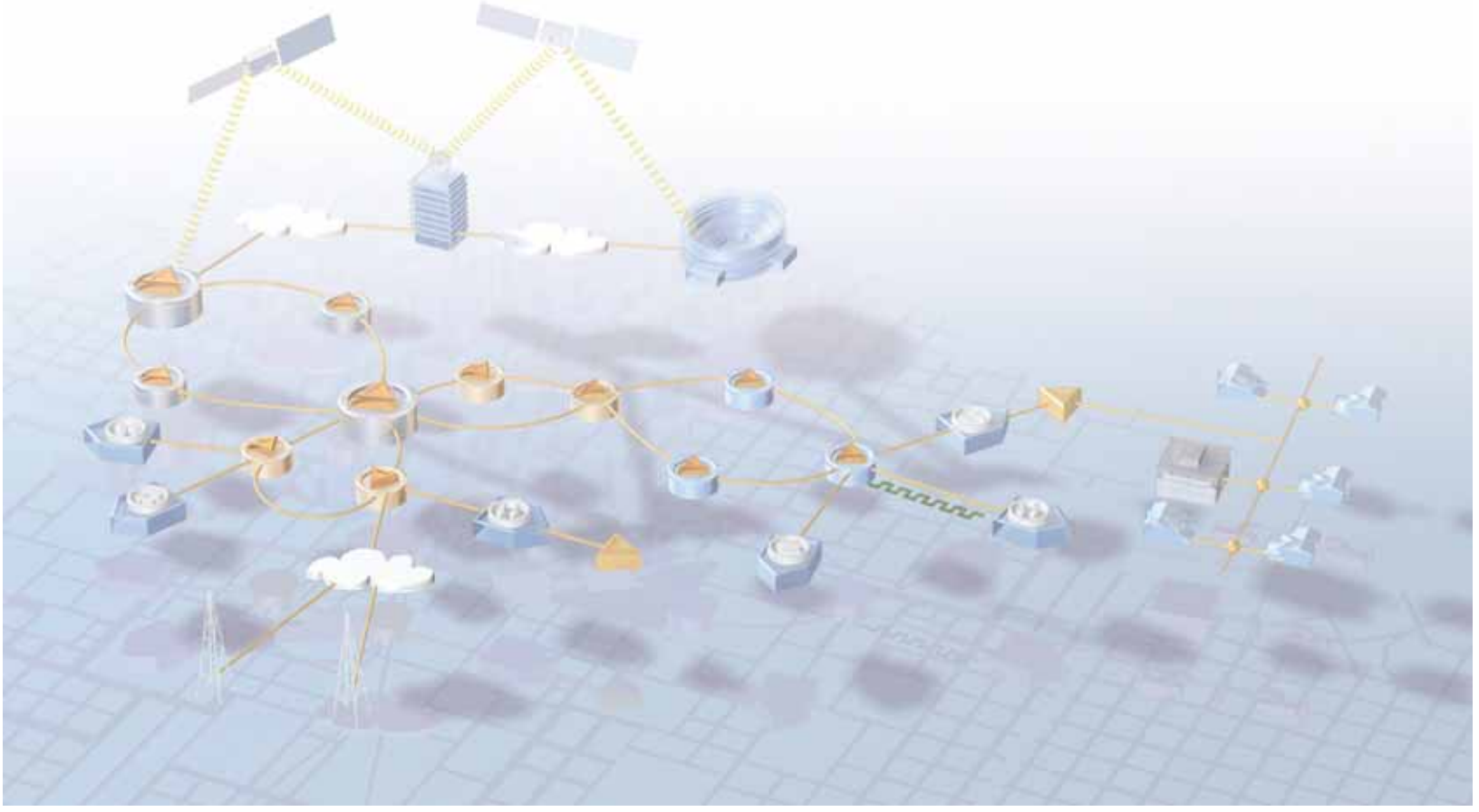

Table of Contents

Introduction

The advent of H.264 (MPEG-4 part 10) video encoding technology has been met with great enthusiasm in the video industry. H.264 has video quality similar to that of MPEG-2, but is more economical with its use of bandwidth. Being less expensive to distribute, H.264 is a natural choice for broadcasters who are trying to find cost effective ways of distributing High Definition Television (HDTV) channels and reducing the cost of carrying conventional Standard Definition channels. In fact, the use of bandwidth has been reduced to the point that it has captured the interest of telephone and data services providers, whose bandwidth limited link to the subscriber had previously not allowed for delivery of bandwidth thirsty television services. H.264 has the potential to revolutionize the industry as it eases the bandwidth burden of service delivery and opens the service provider market to new players. This document describes the process of H.264 encoding and transmission in detail. An overview of Scientific-Atlanta's encoding solution can be found in Appendix B.

H.264 Overview

The main objective behind the H.264 project was to develop a high-performance video coding standard by adopting a "back to basics" approach with simple and straightforward design using well known building blocks.

The ITU-T Video Coding Experts Group (VCEG) initiated the work on the H.264 standard in 1997. Towards the end of 2001, and witnessing the superiority of video quality offered by H.264-based software over that achieved by the (existing) most optimized MPEG-4 part 10 based software, ISO / IEC MPEG joined ITU-T VCEG by forming a Joint Video Team (JVT) that took over the H.264 project of the ITU-T. The JVT objective was to create a single video coding standard that would simultaneously result in a new part (i.e., Part 10) of the MPEG-4 family of standards and a new ITU-T (i.e., H.264) recommendation.

The H.264 standard has a number of advantages that distinguish it from existing standards, while at the same time, sharing common features with other existing standards.

The following are some of the key advantages of H.264:

1. Up to 50% in bit rate savings: Compared to MPEG-2 or MPEG-4 Simple Profile, H.264 permits a reduction in bit rate by up to 50% for a similar degree of encoder optimization at most bit rates.
2. High quality video: H.264 offers consistently good video quality at high and low bit rates.
3. Error resilience: H.264 provides the tools necessary to deal with packet loss in packet networks and bit errors in error-prone wireless networks.
4. Network friendliness: Through the Network Adaptation Layer, that is the same as for MPEG-2, H.264 bit streams can be easily transported over different networks.

The above advantages make H.264 an ideal standard for offering TV services over bandwidth restricted networks, such as DSL networks, or for HDTV. Figure 1 shows a block diagram of the H.264 encoding engine.

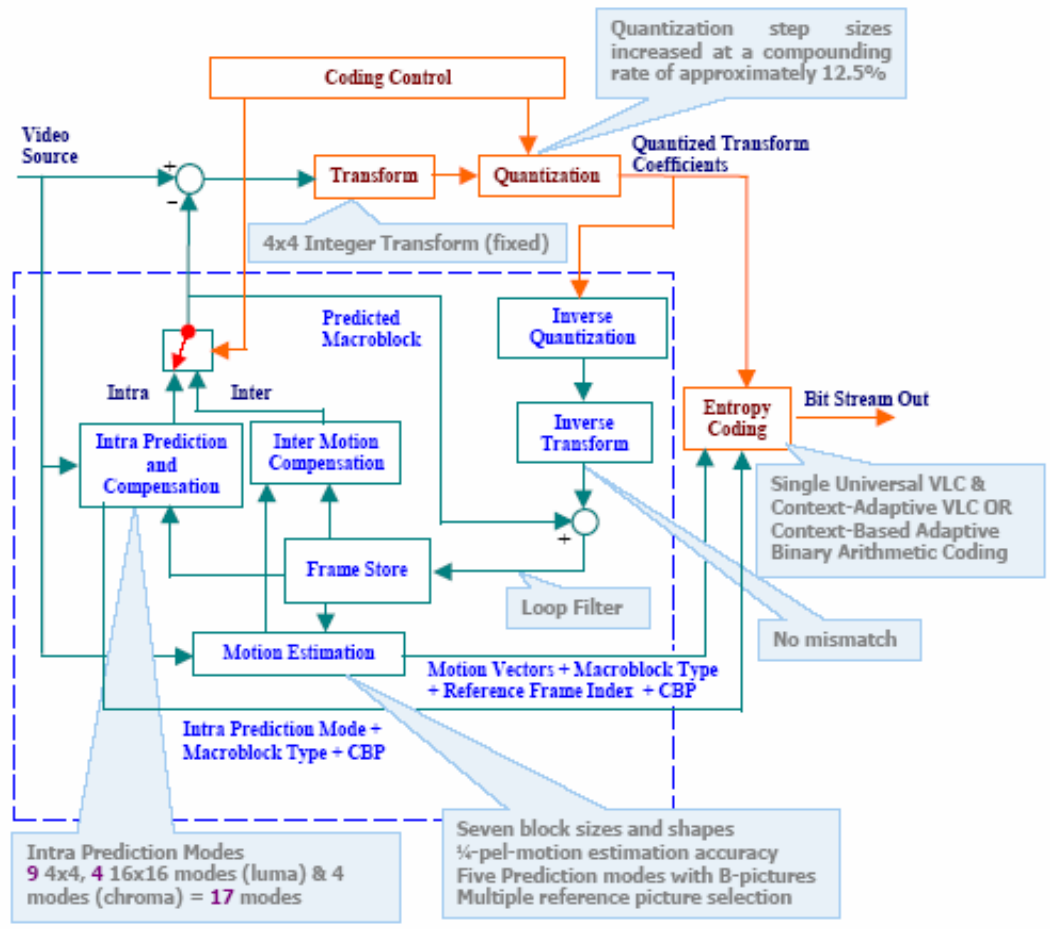

Figure 1. Block Diagram of the H.264 Encoder

H.264 Technical Description

The main objective of the emerging H.264 standard is to provide a means to achieve substantially higher video quality compared to what could be achieved using any of the existing video coding standards. Nonetheless, the underlying approach of H.264 is similar to that adopted in previous standards such as MPEG-2 and MPEG-4 part 2, and consists of the following four main stages:

a. Dividing each video frame into blocks of pixels so that processing of the video frame can be conducted at the block level.

b. Exploiting the spatial redundancies that exist within the video frame by coding some of the original blocks through spatial prediction, transform, quantization and entropy coding (or variable-length coding).

c. Exploiting the temporal dependencies that exist between blocks in successive frames, so that only changes between successive frames need to be encoded. This is accomplished by using motion estimation and compensation. For any given block, a search is performed in the previously coded one or more frames to determine the motion vectors that are then used by the encoder and the decoder to predict the subject block.

d. Exploiting any remaining spatial redundancies that exist within the video frame by coding the residual blocks, i.e., the difference between the original blocks and the corresponding predicted blocks, again through transform, quantization and entropy coding.

On the motion estimation/compensation side, H.264 employs blocks of different sizes and shapes, higher resolution 1/4-pel motion estimation, multiple reference frame selection and complex multiple bi-directional mode selection. On the transform side, H.264 uses an integer based transform that approximates roughly the Discrete Cosine Transform (DCT) used in MPEG-2, but does not have the mismatch problem in the inverse transform.

In H.264, entropy coding can be performed using either a combination of a single Universal Variable Length Codes (UVLC) table with an Context Adaptive Variable Length Codes (CAVLC) for the transform coefficients or using Context-based Adaptive Binary Arithmetic Coding (CABAC).

Organization of the Bit Stream

A given video picture is divided into a number of small blocks referred to as macroblocks. For example, a picture with QCIF resolution (176x144) is divided into 99 16x16 macroblocks as indicated in Figure 2.

A similar macroblock segmentation is used for other frame sizes. The luminance component of the picture is sampled at these frame resolutions, while the chrominance components, Cb and Cr, are down-sampled by two in the horizontal and vertical directions. In addition, a picture may be divided into an integer number of "slices", which are valuable for resynchronization should some data be lost.

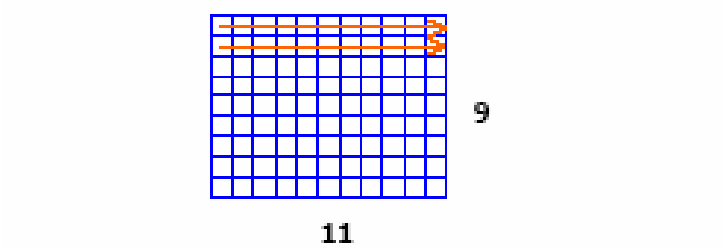

Figure 2. Subdivision of a QCIF picture in 16 x 16 macroblock

A H.264 video stream is organized in discrete packets, called "NAL units" (Network Abstraction Layer units). Each of these packets can contain a part of a slice, that is, there may be one or more NAL units per slice. But not all NAL units contain slice data. There are also NAL unit types for other purposes, such as signaling, headers and additional data. The slices, in turn, contain a part of a video frame. In normal bit streams, each frame consists of a single slice whose data is stored in a single NAL unit. Nevertheless, the possibility to spread frames over an almost arbitrary number of NAL units can be useful if the stream is transmitted over an error prone medium. The decoder may resynchronize after each NAL unit instead of skipping a whole frame if a single error occurs.

H.264 also supports optional *interlaced* encoding. In this encoding mode, a frame is split into two fields. Fields may be encoded using spatial or temporal interleaving. To encode color images, H.264 uses the YCbCr color space like its predecessors, separating the image into luminance (or "luma", brightness) and chrominance (or "chroma", color) planes. It is, however, fixed at 4:2:0 sub-sampling, i.e. the chroma channels each have half the resolution of the luma channel.

H.264 defines five different slice types: I, P, B, SI and SP.

I slices or "Intra" slices describe a full still image, containing only references to itself. A video stream may consist only of I slices, but this implementation is typically not used. The first frame of a sequence always needs to be built out of I slices.

P slices or "Predicted" slices use one or more recently decoded slices as a reference (or "prediction") for picture construction. The prediction is usually not exactly the same as the actual picture content, so a "residual" may be added.

B slices or "Bi-Directional Predicted" slices work like P slices with the exception that former *and future* I or P slices (in playback order) may be used as reference pictures. For this to work, B slices must be decoded *after* the following I or P slice.

SI and SP slices or "Switching" slices may be used for transitions between two different H.264 video streams. This is a very uncommon feature.

The Sequence Parameter Set (abbreviated SPS) and Picture Parameter Set (PPS) contain the basic stream headers. Each of these parameter sets is stored in its own NAL unit, usually occupying only a few bytes. Both parameter sets have their own ID values so that multiple video streams can be transferred in only one H.264 elementary stream.

The most important fields of a sequence parameter set are:

- A profile and level indicator signaling conformance to a profile/level combination specified in H.264 Annex A.
- Information about the decoding method of the picture order.
- The number of reference frames.
- The frame size in macroblocks as well as the interlaced encoding flag.
- Frame cropping information for enabling non-multiple-of-16 frame sizes.
- Video Usability Information (VUI) parameters, such as aspect ratio or color space details.

The most important fields of a picture parameter set are:

- A flag indicating which entropy coding mode is used.
- Information about slice data partitioning and macroblock reordering.
- The maximum reference picture list index.
- Flags indicating the usage of weighted (bi)prediction.
- The initial quantization parameters as well as the luma/chroma quantization parameter offset.
- A flag indicating whether inter-predicted macroblocks may be used for intra prediction or not (“constrained intra prediction”).

Intra Prediction and Coding

Intra coding refers to the case where only spatial redundancies within a video picture are exploited. The resulting frame is referred to as an I-picture. I-pictures are typically encoded by directly applying the transform to the different macroblocks in the frame. Consequently, encoded I-pictures are large in size since a large amount of information is usually present in the frame, and no temporal information is used as part of the encoding process. In order to increase the efficiency of the intra coding process in H.264, spatial correlation between adjacent macroblocks in a given frame is exploited. The idea is based on the observation that adjacent macroblocks tend to have similar properties. Therefore, as a first step in the encoding process for a given macroblock, one may predict the macroblock of interest from the surrounding macroblocks (typically the ones located on top and to the left of the macroblock of interest, since those macroblocks would have already been encoded). The difference between the actual macroblock and its prediction is then coded, which results in fewer bits to represent the macroblock of interest compared to when applying the transform directly to the macroblock itself.

In order to perform the intra prediction mentioned above, H.264 offers nine modes for prediction of 4x4 luminance blocks, including DC prediction (Mode 2) and eight directional modes, labeled 0, 1, 3, 4, 5, 6, 7, and 8 in Figure 3.

Figure 3. Intra prediction modes for 4x4 luminance blocks.

Pixels A to M from neighboring blocks have already been encoded and may be used for prediction. For example, if Mode 0 (Vertical prediction) is selected, then the values of the pixels a to p are assigned as follows:

- a, e, i and m are equal to A,
- b, f, j and n are equal to B,
- c, g, k and o are equal to C, and
- d, h, l and p are equal to D.

For regions with less spatial detail (i.e., flat regions), H.264 supports 16x16 intra coding, in which one of four prediction modes (DC, Vertical, Horizontal and Planar) is chosen for the prediction of the entire luminance component of the macroblock. In addition, H.264 supports intra prediction for the 8x8 chrominance blocks also using four prediction modes (DC, Vertical, Horizontal and Planar). Finally, the prediction mode for each block is efficiently coded by assigning shorter symbols to more likely modes, where the probability of each mode is determined based on the modes used for coding the surrounding blocks.

Inter Prediction and Coding

Inter prediction and coding is based on using motion estimation and compensation to take advantage of the temporal redundancies that exist between successive frames, hence, providing very efficient coding of video sequences. As stated in section 2.1, when a selected reference frame(s) for motion estimation is a previously encoded frame(s), the frame to be encoded is referred to as a P-picture.

When both a previously encoded frame and a future frame are chosen as reference frames, then the frame to be encoded is referred to as a B-picture. Motion estimation in H.264 supports most of the key features adopted in earlier video standards, but its efficiency is improved through added flexibility and functionality.

In addition to supporting P-pictures (with single and multiple reference frames) and B-pictures, H.264 supports a new inter-stream transitional picture called an SP-picture. The inclusion of SP-pictures in a bit stream enables efficient switching between bit streams with similar content encoded at different bit rates, as well as random access and fast playback modes.

Block Sizes

Motion compensation for each 16x16 macroblock can be performed using a number of different block sizes and shapes. These are illustrated in Figure 4. Individual motion vectors can be transmitted for blocks as small as 4x4, so up to 16 motion vectors may be transmitted for a single macroblock.

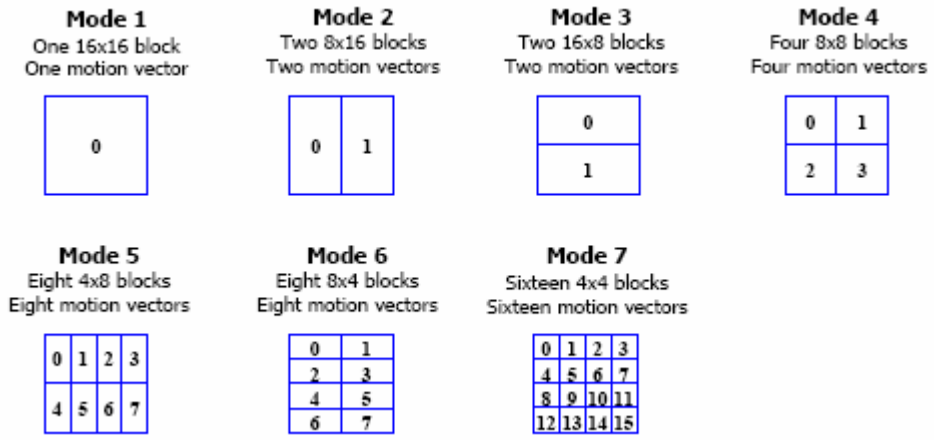

Figure 4. Different modes of dividing a macroblock for motion estimation in H.264

Block sizes of 16x8, 8x16, 8x8, 8x4, and 4x8 are also supported as shown. The availability of smaller motion compensation blocks improves prediction in general, and in particular, the small blocks improve the ability of the model to handle fine motion detail and result in better subjective viewing quality because they do not produce large blocking artifacts. Moreover, through the recently adopted tree structure segmentation method, it is possible to have a combination of 4x8, 8x4, or 4x4 sub-blocks within an 8x8 sub-block. Figure 5 shows an example of such a configuration for a 16x16 macroblock.

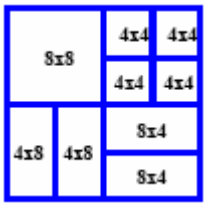

Figure 5. Example of 16x16 macroblock

Motion Estimation Accuracy

The prediction capability of the motion compensation algorithm in H.264 is further improved by allowing motion vectors to be determined with higher levels of spatial accuracy than in existing standards. Quarter-pixel accurate motion compensation is the lowest-accuracy form of motion compensation in H.264 (in contrast with prior standards based primarily on half-pel accuracy, with quarter-pel accuracy only available in the newest version of MPEG-4).

Multiple Reference Picture Selection

The H.264 standard offers the option of having multiple reference frames in inter-picture coding, resulting in better subjective video quality and more efficient coding of the video frame under consideration. Moreover, using multiple reference frames helps make the H.264 bit stream more error resilient. However, from an implementation point of view, there would be additional processing delays and higher memory requirements at both the encoder and decoder.

De-blocking (Loop) Filter

H.264 specifies the use of an adaptive de-blocking filter that operates on the horizontal and vertical block edges within the prediction loop in order to remove artifacts caused by block prediction errors. The filtering is generally based on 4x4 block boundaries, in which two pixels on either side of the boundary may be updated using a different filter. The rules for applying the de-blocking filter are intricate and quite complex, however, its use is optional for each slice (loosely defined as an integer number of macroblocks). Nonetheless, the improvement in subjective quality often more than justifies the increase in complexity.

Integer Transform

The information contained in a prediction error block resulting from either intra prediction or inter prediction is then re-expressed in the form of transform coefficients. H.264 is unique in that it employs a purely integer spatial transform (a rough approximation of the DCT) which is primarily 4x4 in shape, as opposed to the usual floating-point 8x8 DCT specified with rounding-error tolerances as used in earlier standards. The small shape helps reduce blocking and ringing artifacts, while the precise integer specification eliminates any mismatch issues between the encoder and decoder in the inverse transform.

Quantization and Transform Coefficient Scanning

The quantization step is where a significant portion of data compression takes place. In H.264, the transform coefficients are quantized using scalar quantization with no widened dead-zone. Fifty-two different quantization step sizes can be chosen on a macroblock basis – this being different from prior standards. Moreover, in H.264 the step sizes are increased at a compounding rate of approximately 12.5%, rather than increasing it by a constant increment. The fidelity of chrominance components is improved by using finer quantization step sizes compared to those used for the luminance coefficients, particularly when the luminance coefficients are coarsely quantized.

Figure 6. Scan pattern for frame coding in H.264

The quantized transform coefficients correspond to different frequencies, with the coefficient at the top left hand corner in Figure 6 representing the DC value, and the rest of the coefficients corresponding to different nonzero frequency values. The next step in the encoding process is to arrange the quantized coefficients in an array, starting with the DC coefficient. A single coefficient-scanning pattern is available in H.264 (Figure 6) for frame coding, and another one is being added for field coding.

The zigzag scan illustrated in Figure 6 is used in all frame-coding cases, and it is identical to the conventional scan used in earlier video coding standards. The zigzag scan arranges the coefficient in an ascending order of the corresponding frequencies.

Entropy Coding

The last step in the video coding process is entropy coding. Entropy coding is based on assigning shorter codewords to symbols with higher probabilities of occurrence, and longer codewords to symbols with less frequent occurrences. Some of the parameters to be entropy coded include transform coefficients for the residual data, motion vectors and other encoder information. Two types of entropy coding have been adopted. The first method represents a combination of Universal Variable Length Coding (UVLC) and Context Adaptive Variable-Length coding (CAVLC). The second method is represented by Context-Based Adaptive Binary Arithmetic Coding (CABAC).

UVLC/CAVLC

In some video coding standards, symbols and the associated codewords are organized in look-up tables, referred to as variable length coding (VLC) tables, which are stored at both the encoder and decoder. In MPEG-2, a number of VLC tables are used, depending on the type of data under consideration (e.g., transform coefficients, motion vectors).

H.264 offers a single Universal VLC (UVLC) table that is to be used in entropy coding of all symbols in the encoder except for the transform coefficients. Although the use of a single UVLC table is simple, is has a major disadvantage, which is that the single table is usually derived using a static probability distribution model, which ignores the correlations between the encoder symbols.

In H.264, the transform coefficients are coded using Context Adaptive Variable Length Coding (CAVLC). CAVLC is designed to take advantage of several characteristics of quantized 4x4 blocks. First, non-zero coefficients at the end of the zigzag scan are often equal to +/- 1. CAVLC encodes the number of these coefficients (“trailing 1s”) in a compact way. Second, CAVLC employs run-level coding efficiently to represent the string of zeros in a quantized 4x4 block. Moreover, the numbers of non-zero coefficients in neighboring blocks are usually correlated. Thus, the number of non-zero coefficients is encoded using a look-up table that depends on the numbers of non-zero coefficients in neighboring blocks.

Finally, the magnitude (level) of non-zero coefficients increase near the DC coefficient and decrease around the high-frequency coefficients. CAVLC takes advantage of this by making the choice of the VLC look-up table for the level adaptive where the choice depends on the recently coded levels.

H.264 Profiles

H.264 describes two popular profiles: Baseline, mainly for video conferencing and telephony / mobile applications, and Main, primarily for broadcast video applications. Figure 7 shows the common features between the Baseline and Main profiles as well as the additional specific features for each. The Baseline profile allows the use of Arbitrary Slice Ordering (ASO) to reduce the latency in real-time communication applications, as well as the use of Flexible Macroblock Ordering (FMO) and redundant slices to improve error resilience in the coded bit stream. The Main profile enables additional reduction in bandwidth over the Baseline profile through mainly sophisticated Bi-directional prediction (B-pictures), Context Adaptive Binary Arithmetic Coding (CABAC) and weighted prediction.

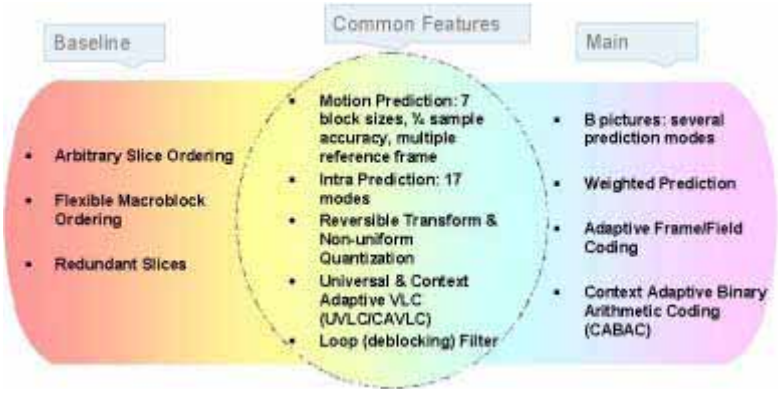

Figure 7. Features for the Baseline and Main profiles

Baseline Profile: Specific Features

Arbitrary Slice Ordering

Arbitrary slice ordering allows the decoder to process slices in an arbitrary order as they arrive to the decoder. Hence the decoder does not have to wait for all the slices to be properly arranged before it starts processing them. This reduces the processing delay at the decoder, resulting in less overall latency in real-time video communication applications.

Flexible Macroblock Ordering (FMO)

Macroblocks in a given frame are usually coded in a raster scan order. With FMO, macroblocks are coded according to a macroblock allocation map that groups, within a given slice, macroblocks from spatially different locations in the frame. Such an arrangement enhances error resilience in the coded bit stream since it reduces the interdependency that would otherwise exist in coding data within adjacent macroblocks in a given frame. In the case of packet loss, the loss is scattered throughout the picture and can be easily concealed.

Redundant Slices

Redundant slices allow the transmission of duplicate slices over error-prone networks to increase the likelihood of the delivery of a slice that is free of errors.

Main Profile: Specific Features

B Pictures

B-pictures provide a compression advantage as compared to P-pictures by allowing a larger number of prediction modes for each macroblock. Here, the prediction is formed by averaging the sample values in two reference blocks, generally, but not necessarily using one reference block that is forward in time and one that is backward in time with respect to the current picture. In addition, "Direct Mode" prediction is supported, in which the motion vectors for the macroblock are interpolated based on the motion vectors used for coding the co-located macroblock in a nearby reference frame. Thus, no motion information is transmitted. By allowing so many prediction modes, the prediction accuracy is improved, often reducing the bit rate by 5-10%.

Weighted Prediction

This allows the modification of motion compensated sample intensities using a global multiplier and a global offset. The multiplier and offset may be explicitly sent, or implicitly inferred. The use of the multiplier and the offset aims at reducing the prediction residuals due, for example, to global changes in brightness, and consequently, leads to enhanced coding efficiency for sequences with fades, lighting changes, and other special effects.

CABAC

Context Adaptive Binary Arithmetic Coding (CABAC) makes use of a probability model at both the encoder and decoder for all the syntax elements (transform coefficients, motion vectors, etc). To increase the coding efficiency of arithmetic coding, the underlying probability model is adapted to the changing statistics within a video frame, through a process called context modeling.

The context modeling provides estimates of conditional probabilities of the coding symbols. Utilizing suitable context models, given inter-symbol redundancy can be exploited by switching between different probability models according to already coded symbols in the neighborhood of the current symbol to encode. The context modeling is responsible for most of CABAC’s 10% savings in bit rate over the VLC entropy coding method (UVLC/CAVLC).

Interlace Support

Interlaced video has two half pictures (fields) in a frame or full picture and they are at different times. The Main profile copes with this by supporting field coding and picture or macroblock adaptive switching between frame and field coding.

Extended Profile

This profile supports all features of the Baseline profile, with the addition of B slices, weighted prediction, field coding and picture or macroblock adaptive switching between frame and field coding. Furthermore it is the only profile to support the SP / SI slice data portioning. It does not support CABAC.

Fidelity Range Extensions (FRExt)

New tools and 4 new profiles have been added to address more demanding applications. The DVD forum and BlueRay disc are very interested in these new profiles.

Tools

New 8x8 transform

Adaptive macro block-level transform (4x4,8x8) switching

Encoder-specified quant scaling matrices

Encoder-specified separate control of chroma quant parameter

Profiles

High Profile (HP) 8 bit video, 4:2:0 chroma sampling

High 10 Profile (Hi10P) up to 10 bit video, 4:2:0 chroma sampling

High 4:2:2 Profile (H422P) up to 10 bit video, 4:2:2 chroma sampling

High 4:4:4 Profile (H444P) up to 10 bit video, 4:4:4 chroma sampling

High Profile: Specific Features

High profile contains the Main profile, a switchable 8x8 transform for residual coding and Scaling matrices for subjective quality optimization. High profile improves *objective* compression quality (significantly for some video, especially HD video). It also improves *subjective* compression capability with support for quantization weighting matrices. High profile *includes Main profile as a subset*, so there is little risk involved in the implementation.

The Main Impacts of the High Profile

Coding efficiency impact (measured as average bit-rate reduction):

HD Film: 12%

HD Video (progressive): 12%

HD Video (interlace): 4% (only 2 test clips)

SD Video (interlace): 6%

Complexity Impact

Implementation beyond Main Profile affects Intra prediction, transform, de-blocking filter control, CABAC decoding. There is very little increase in computational requirements and only a slight increase in memory requirements (CABAC, transform).

High Profile Details

Integer 8x8 Transform (luma only)

Per 8x8 block, same number of adds (64) and 4 extrashifts (20 vs 16) compared with four 4x4 transforms.

$$\begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix} \cdot /8$$

Figure 8. 8x8 Transform

Scanning of the Matrix

Two scans for the 4x4 transform, switched for frame / field coding to support interlaced video.

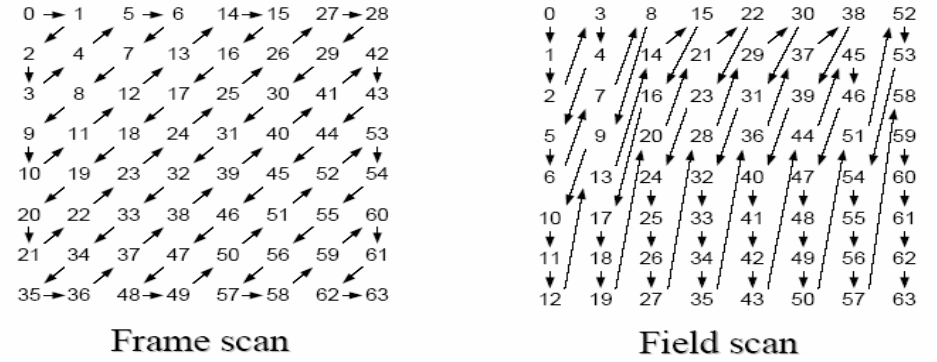

Figure 9. Frame and Field Scan

Intra Prediction

There are nine Intra 8x8 prediction modes similar to the nine modes for the intra 4x4 blocks as stated in section on Intra Prediction and Coding earlier in this document.

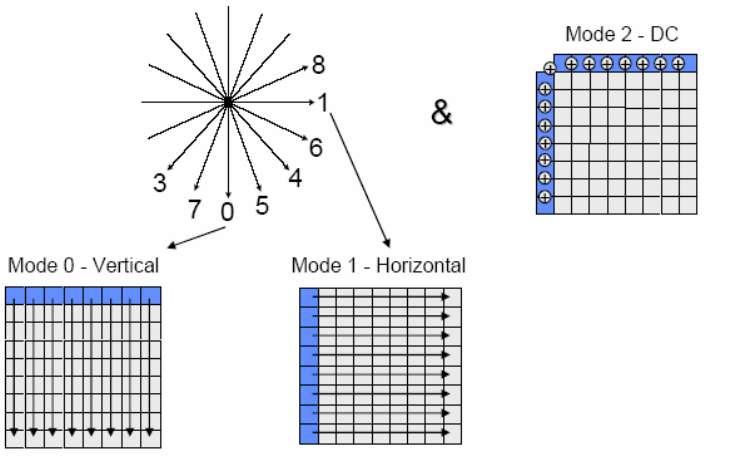

Figure 10. Intra Prediction

Scaling Matrices

Scaling matrices are similar to MPEG-2 video. A matrix can be transmitted in the SPS as well as in the PPS. There is a separate matrix for 4x4 and 8x8 transforms and a separate matrix for Inter frame and Intra frame.

De-blocking Filters, CABAC and Signaling

For the de-blocking filter, only control of the filter is adjusted. It does not filter 4x4 blocks. Furthermore, there is no change to the filter operation itself compared to the Main Profile.

The CABAC, mentioned in the main profile section, contains 61 new contexts and corresponding initialization values, but no change to the CABAC engine. The High Profile adds signaling in the form of the 8x8 transform on / off flag at the PPS level and an 8x8 transform on / off flag per macroblock to allow for adaptive use.

Transport over IP

H.264 bandwidth efficiencies enable video to be compressed to the point that it can be transferred through networks using a small amount of bandwidth. This has created an opportunity for voice and data service providers to offer television services as part of their product portfolio. In order to carry H.264 services to the customer premises, the existing IP infrastructure is used, requiring a method of encapsulating transport streams over IP.

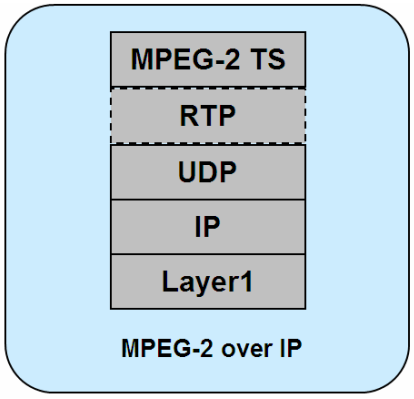

Figure 11. MPEG over IP

For telco type applications, H.264 video and the accompanying audio are put into an MPEG-2 transport stream, which can be sent directly over the User Datagram Protocol (UDP over IP) [IETF RFC 768]. Another method involves the use of the Real-time Transmission Protocol (RTP) [IETF RFC 1889] as an intermediate layer between UDP and the MPEG-2 transport stream.

IP Layer

At the network layer, routing video to the end user from the source can be done through unicast or multicast IP packets. Unicast packets are sent from the source to one destination, whereas multicast packets are sent from one source to all destinations that are part of the multicast group. The latter is useful for routing packets to multiple subscribers from a single encoder source. IP networks inherently introduce delay and jitter to the services they carry. If the delays are somewhat predictable, decoders can compensate for small amounts of jitter by queuing data before it is processed. In order to maximize quality and minimize decoding delays, the IP network connections must be provisioned with acceptable guarantees on quality of service.

UDP Layer

Working on top of the IP layer, UDP is used to provide a connectionless transport from the source to the destination(s). The UDP layer introduces logical communication ports, allowing multiple UDP sessions to be handled between source and destination equipment. Unlike TCP, UDP does not provide an assured data link, so dropped packets are not re-transmitted. The provider of the IP connection must ensure that the quality of the IP link is high in order to minimize or eliminate dropped packets entirely.

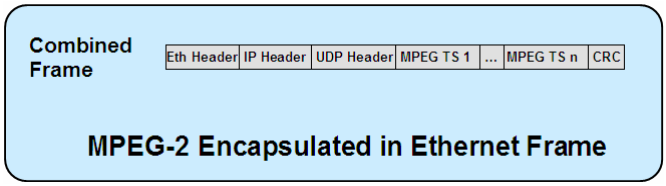

Figure 12. MPEG-2 Ethernet Frame

MPEG-2 Transport Stream

The MPEG-2 transport stream that contains the H.264 video can be mapped into the UDP packet payload, or optionally into an RTP payload. More than one MPEG-2 transport stream frame is generally mapped into a single UDP or RTP packet. This is done for reasons of bandwidth efficiency. MPEG-2 transport stream frames are generally 188 or 204 bytes in length and the length of the combined Ethernet, IP and UDP overhead is 54 bytes. Including an RTP layer adds at least another 12 bytes. Maximizing the number of MPEG-2 transport stream frames you can fit in one UDP or RTP packet improves the efficiency of transporting video with IP.

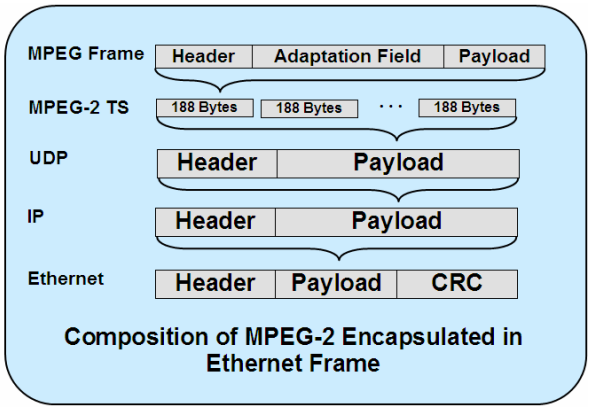

Figure 13. MPEG-2 in Ethernet Composition

Optional RTP Layer

The MPEG-2 transport stream frames can optionally be inserted into an RTP packet before insertion into UDP. RTP adds a sequence number that allows the detection of missing packets and allows the decoder to reorder packets that arrive out of order, as can happen in IP networks with multiple routes between source and destination. It also adds a timestamp that is useful for jitter measurement and synchronization. RTP adds additional overhead to the transportation of MPEG over IP.

Conclusion

H.264 technology reduces the bandwidth requirement for video service distribution without impacting video quality, making delivery of SDTV and HDTV signals less expensive. Service providers can exploit the ubiquity of IP networks to deliver low-bandwidth, high-quality video services to subscribers.

Appendix A - Comparison of H.264 and MPEG-2

Algorithm Characteristic	MPEG-2	H.264
General	Motion compensated predictive, residual, transformed, entropy coded	Same basic structure as MPEG
Intra Prediction	None	Multi-direction, Multi-pattern
Coded Image Types	I, B,P	I, B, P, SP, SI
Transform	8x8 DCT	4x4 DCT-like Integer Transform
Motion Estimation Blocks	16x16	16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4
Entropy Coding	Multiple VLC Tables	Arithmetic Coding and adaptive VLC Tables
Frame Distance for Prediction	+/- 1	Unlimited forward/backward
Fractional Motion Estimation	½ Pixel (MPEG-2) 1/4 Pixel (MPEG-4)	1/4 Pixel
De-blocking Filter	None	Dynamic edge filters

Figure 14. Overview between H.264 and MPEG-2

Appendix B - The D9154 - Scientific-Atlanta's H.264 Encoding Solution

Figure 15. D9154 Encoder Front View

Scientific-Atlanta's D9154 Advanced Compression Encoder (ACE) provides high quality H.264 video encoding for Telco and Broadcast applications using the H.264 Main profile. The D9154 is implemented with high performance DSPs and state of the art FPGAs, providing an encoding platform that can be upgraded with additional functionality in the field.

Bandwidth Efficiency

The D9154 encoder's highly optimized implementation realizes the significant bit rate reductions promised by H.264 technology while achieving video quality that is comparable to MPEG-2 video. Encoded bit rate savings are on average approximately 50%, allowing broadcasters to fit as many as twice the number of channels into a single satellite transponder for distribution. On a per video channel basis, the reduction in encoded bit rate opens the television service provider market to telcos where the bandwidth restrictions imposed by last-mile copper links are no longer a limiting factor.

Pre-processing Filters

D9154 encoder users can make use of pre-processing filters to reduce noise in the signal being encoded. The D9154 applies motion-compensated filtering to reduce noise in successive video frames, reducing noise in the temporal direction while taking into account motion in successive video frames. This pre-processing not only achieves good performance in terms of reducing the occurrence of artifacts, but it also results in encoded bit rate savings, leading to better overall video quality. This is especially useful for telcos and MSOs who are encoding channels at Central Office and headend sites where the source may contain some noise.

Encoder Output

The D9154 encoder is equipped with 10/100 BaseT Ethernet IP outputs as well as traditional ASI outputs. An MPEG-2 transport stream can be transmitted to both types of network simultaneously, allowing distribution over traditional ASI networks as well as IP networks. The D9154 supports multicast and unicast of H.264 encoded video and audio in an MPEG-2 transport stream over a choice of IP/UDP or IP/UDP/RTP.

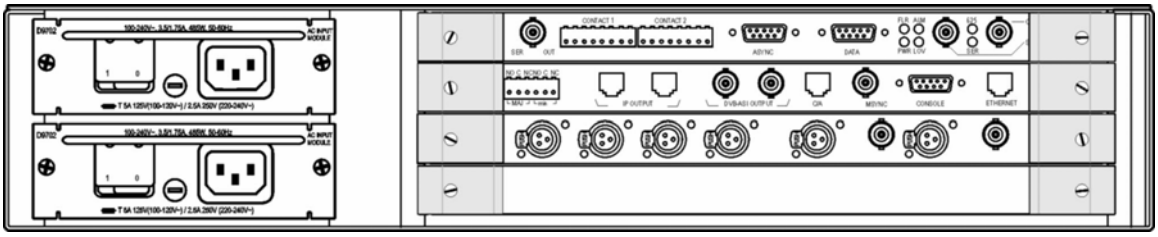

Figure 16. D9154 Advanced Compression Encoder Rear Panel

Scientific-Atlanta, Inc.
770.236.5000
www.scientificatlanta.com

June 2005 Printed in USA
Part Number 7007887 Rev B