IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 1, Ver. II (Jan. 2014), PP 53-68
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Study of H.264/Avc Algorithm and It's Implementation In Matlab

Samitha T[1], Prasanth C. R[2], Lekshmi P.R[3],Shanti K.P[4]
[1]*Assistant Professor , Dept. Of Electronics,College of Engg Cherthala*
[2]*M.Tech Scholar Dept. Of Electronics,College of Engg Cherthala*
[3]*M.Tech Scholar Dept. Of Electronics,College of Engg Cherthala*
[4]*M.Tech Scholar Dept. Of Electronics,College of Engg Cherthala*

Abstract : *H.264/AVC(Advanced Video Coding) is a video coding standard of the ITU-T(International Telecommunication Union for Telecommunication Standardization Sector) Video Coding Experts Group and the ISO/IEC(International Standards Organization/International Electrotechnical Commission) Moving Pic- ture Experts Group. The main goals of the H.264/AVC standardization effort have been enhanced compression performance and provision of a network friendly video representation ad- dressing conversational (video telephony) and no conversational (storage, broadcast, or streaming) applications. H.264/AVC has achieved a significant improvement in rate-distortion efficiency relative to existing standards. This article provides an overview of the technical features of H.264/AVC, describes profiles and applications for the standard, and compared it with H.261 and H.263.*

Keywords: *Advanced video coding (AVC),intraprediction, interprediction.*

I. Introduction

H.264/AVC is the international video coding standard. It is approved by ITU-T as Recommendation H.264 and by ISO/IEC as International Stan- dard (MPEG(Motion Picture Expert Group)-4 part 10) AVC. It is widely used for the transmission of standard definition (SD) and high definition (HD) TV signals over satellite, cable, and terrestrial emission and the storage of high-quality SD video signals onto DVDs. However, an increasing number of services and growing popularity of high definition TV are creating greater needs for higher coding efficiency. Moreover, other transmission media such as Cable Modem, xDSL (Digital Subscriber Line), or UMTS (Universal Mo- bile Telecommunications System) offer much lower data rates than broadcast channels, and enhanced coding efficiency can enable the transmission of more video channels or higher quality video representations within existing digital transmission capacities.Digital video is a sequence of still images or frames and represents scenes in motion. A video signal is a sequence of two dimensional (2D) im- ages projected from a dynamic three dimensional (3D) scene onto the image plane of a video camera. Video coding is the process of compressing and decompressing a digital video signal. Coding of video is performed picture by picture. Each picture to be coded is first partitioned into a number of slices (it is possible to have one slice per picture also). Slices are individual coding units in this standard as compared to earlier standards as each slice is coded independently. The hierarchy of video data organization is as follows: picture-slices-macroblocks-submacroblocks-blocks-pixels.

II. H.261 STANDARD.

H.261 is a codec designed by ITU-T for video conferencing over PSTN (Public Switched Telephone Network). H.261 describes the video coding and decoding methods for the moving picture component of audiovisual ser- vices at the rate of p x 64 kbit/s, where p is in the range 1 to 30. It de- scribes the video source coder, the video multiplex coder and the transmission coder. Figure 2.3 represents an overview of the H.261 CODEC(Coder De- coder).H.261 encoding is based on the discrete cosine transform (DCT) and allows for fully-encoding only certain frames (INTRA-frame) while encoding the differences between other frames (INTER-frame). The main elements of the H.261 source coder are prediction, block transformation (spatial to frequency domain translation), quantization, and entropy coding. Loop fil- tering provides a noticeable improvement in video quality but demands extra processing power.Two types of image frames are defined: Intra-frames (I-frames) and Inter- frames (P-frames) I frames are treated as independent images.

2.1.1 Intra Frame Coding

The term intra frame coding refers to the fact that the various lossless and lossy compression techniques are performed relative to information that is contained only within the current frame, and not relative to any other frame in the video sequence. In other words, no temporal processing is performed

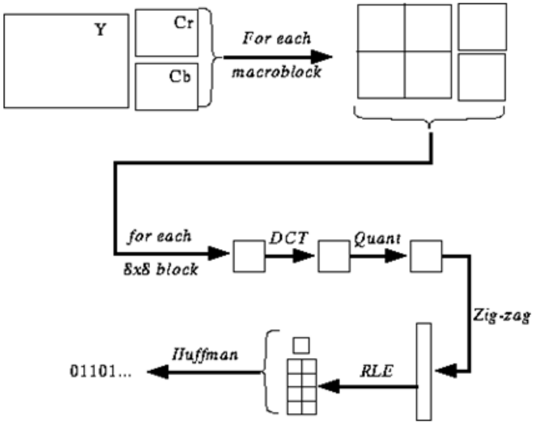


Figure 2.1: Basic video encoder for intra frames only.

outside of the current picture or frame. This mode will be described first because it is simpler, and because non-intra coding techniques are extensions to these basics. Figure 2.1 shows a block diagram of a basic video encoder for intra frames only. It turns out that this block diagram is very similar to that of a JPEG(Joint Picture Expert Group) still image video encoder, with only slight implementation detail differences.

2.1.2 Inter-frame(P-frame) Coding

A block diagram of the basic encoder with extensions for non-intra frame coding techniques is given in Figure 2.2. Of course, this encoder can also support intra frame coding as a subset. Starting with an intra, or I frame, the encoder can forward predict a future frame. This is commonly referred to as a P frame, and it may also be predicted from other P frames, although only in a forward time manner. As an example, consider a group of pictures that lasts for 6 frames. In this case, the frame ordering is given as I,P,P,P,P,P,I,P,P,P,P,...

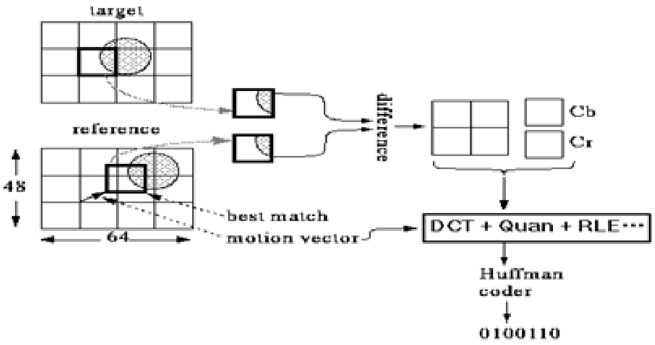


Figure 2.2: P frame Coding

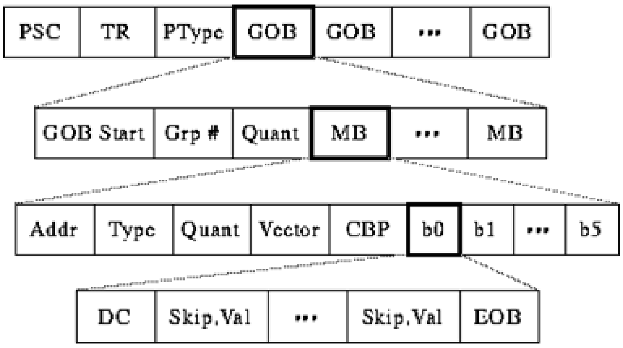


Figure 2.3: H.261 Bit stream structure

The H.261 Bitstream structure may be summarised as follows:

- Need to delineate boundaries between pictures, so send Picture Start Code - PSC
- Need timestamp for picture (used later for audio synchronization), so send Temporal Reference - TR
- Need timestamp for picture (used later for audio synchronization), so send Temporal Reference - TR
- Is this a P-frame or an I-frame? Send Picture Type- P Type
- Picture is divided into regions of 11x3 macroblocks called Groups of Blocks - GOB
- Might want to skip whole groups, so send Group Number (Grp #)
- Might want to use one quantization value for whole group, so send Group Quantization Value - GQuant
- Overall, bitstream is designed so we can skip data whenever possible while still unambiguous.

III. H.261

H.263 is designed for very low bit-rate application. The coding algorithm of H.263 is similar to that used by H.261.Besides the CIF and QCIF that H.261 supports, it also supports SQCIF (Sub Quarter Common Intermediate Format), 4CIF, and 16CIF. 4CIF is 4 times the res- olution of CIF, and 16CIF is 16 times the resolution. SQCIF is about half the resolution of QCIF. This means that H.261 compares well with the MPEG standards. SQCIF is approximately half the resolution of QCIF. 4CIF and16CIF are 4 and 16 times the resolution of CIF respectively. The support of 4CIF and 16CIF means the codec could then compete with other higher bitrate video coding standards such as the MPEG standards

3.1 Main features with respect to H.261

Unrestricted Motion Vector mode: In this optional mode, motion vectors are allowed to point outside the picture. The edge pixels are used as prediction for the ”non-existing” pixels. With this mode a significant gain is achieved if there is movement across the edges of the picture, especially for the smaller picture formats. Additionally, this mode includes an extension of the motion vector range so that larger motion vectors can be used. This is especially useful in case of camera movement and large picture formats.

Syntax-based Arithmetic Coding mode: In this optional mode, arithmetic coding is used instead of variable length coding. The SNR(Signal to Noise Ratio) and reconstructed pictures will be the same, but significantly fewer bits will be produced.

Advanced Prediction mode: In this optional mode, Overlapped Block Motion Compensation (OBMC) is used for the luminance part of P-pictures.Four 8x8 vectors instead of one 16×16 vector are used for some of the macroblocks in the picture. The encoder has to decide which type of vectors to use. Four vectors use more bits, but give better prediction. The use of this mode generally gives a considerable improvement. A subjective gain is achieved because OBMC results in less blocking artifacts.
PB-frames mode: A PB-frame consists of two pictures being coded as one unit. The name PB comes from the name of picture types in ITU-T Rec. H.262 where there are P-pictures and B-pictures. Thus a PB-frame consists of one P-picture which is predicted from the previous decoded P-picture and one B-picture which is predicted from both the previous decoded P-picture and the P-picture currently being decoded. The name B-picture was chosen because parts of B-pictures may be bidirectionally predicted from the past and future pictures. With this coding option, the picture rate can be in- creased considerably without substantially increasing the bit rate. However, an Improved PB-frames mode is also provided. The original PB-frames mode is retained herein only for purposes of compatibility with systems made prior to the adoption of the Improved PB-frames mode.
In H.263 has seven basic picture types (of which only the first two are

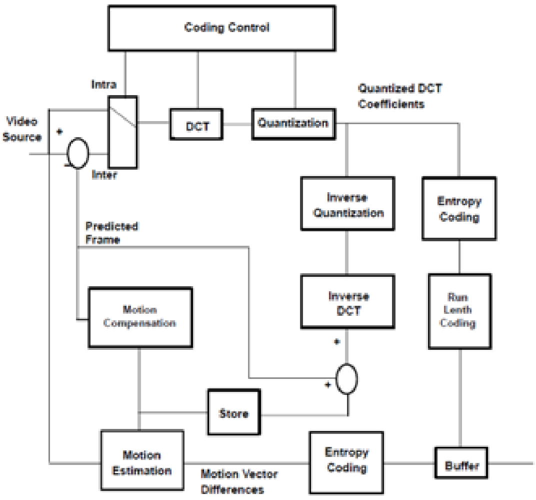


Figure 2.6: H.263 Encoder Structure

mandatory) which are defined primarily in terms of their prediction structure:

1. INTRA: A picture having no reference picture(s) for prediction (also called an I-picture);
2. INTER: A picture using a temporally previous reference picture (also called a P-picture);
3. PB: A frame representing two pictures and having a temporally previ- ous reference picture
4. Improved PB: A frame functionally similar but normally better than a PB-frame;
5. B: A picture having two reference pictures, one of which temporally precedes the B-picture and one of which temporally succeeds the B- picture and has the same picture size;
6. EI: A picture having a temporally simultaneous reference picture which has either the same or a smaller picture size; and
7. EP: A picture having two reference pictures, one of which temporally precedes the EP-picture and one of which is temporally simultaneous and has either the same or a smaller picture size.

3.2 Applications

H.261 is the most important application in the video conferencing and video communications systems. Its uses include studio based video conferenc- ing, desktop video conferencing, surveillance, monitoring, computer training, and tele-medicine.

IV. H.264

4.1 Features of H.264

- Variable block size motion compensation with small block sizes: This standard supports more flexibility in the selection of motion compensation block sizes and shapes than any previous standard, with a min- imum luma motion compensation block size as small as 4x4.
- Quarter sample accurate motion compensation: Most prior standards enable halfsample motion vector accuracy at most. The new designimproves up on this by adding quarter-sample motion vector accuracy, as first found in an advanced profile of the MPEG 4 Visual standard, but further reduces the complexity of the interpolation processing com- pared to the prior design.
- Motion vectors over picture boundaries: While motion vectors in MPEG-
2 and its predecessors were required to point only to areas within the previously-decoded reference picture, the picture boundary extrapola- tion technique first found as an optional feature in H.263 is included in H.264/AVC.
- Multiple reference picture motion compensation: The new design ex- tends upon the enhanced reference picture selection technique found in to enable efficient coding by allowing an encoder to select, for motion compensation purposes, among a larger number of pictures that have been decoded and stored in the decoder.
- Decoupling of referencing order from display order:H.264/AVC allows encoder to choose the ordering of pictures for referencing and display
purposes with a high degree of flexibility constrained only by a total memory capacity bound imposed to ensure decoding ability. Removal of the restriction also enables removing the extra delay previously as- sociated with bi-predictive coding.
- Decoupling of picture representation methods from picture referencing capability: In prior standards,

pictures encoded using some encoding methods (namely bi-predictively-encoded pictures) could not be used as references for prediction of other pictures in the video sequence. By removing this restriction, the new standard provides the encoder more flexibility and, in many cases, an ability to use a picture for referencing that is a closer approximation to the picture being encoded.

- Weighted prediction: A new innovation in H.264/AVC allows the motion- compensated prediction signal to be weighted and offset by amounts specified by the encoder. This can dramatically improve coding effi- ciency for scenes containing fades, and can be used flexibly for other purposes as well.
- Improved skipped and direct motion inference: In prior standards, a skipped area of a predictively-coded picture could not motion in the scene content. This had a detrimental effect when coding video con- taining global motion, so the new H.264/AVC design instead infers mo- tion in skipped areas. For bi-predictively coded areas (called B slices), H.264/AVC also includes an enhanced motion inference method known as direct motion com-pensation, which improves further on prior direct pre-diction designs found in and MPEG-4 Visual.
- Directional spatial prediction for intra coding: A new technique of ex- trapolating the edges of the previously-de-coded parts of the current picture is applied in regions of pictures that are coded as intra (i.e., coded without ref-erence to the content of some other picture). This im-proves the quality of the prediction signal, and also allows predic- tion from neighboring areas that were not coded using intra coding (something not enabled when using the transform-domain prediction method found in and MPEG-4 Visual).
- In the loop deblocking filtering: Block-based video coding produces artifacts known as blocking artifacts. These can originate from both the prediction and residual difference coding stages of the decoding process. Application of an adaptive deblocking filter is a well known method of improving the resulting video quality, and when designed well, this can improve both objective and sub-jective video quality. Building further on a concept from an optional feature of , the deblocking filter in the H.264/AVC design is brought within the motion-compensated prediction loop, so that this improvement in quality can be used in inter-picture prediction to improve the ability to predict other pictures as well. In addition to improved prediction methods, other parts of the design were also enhanced for improved coding efficiency, including the following.
- Small block size transform: All major prior video coding standards used a transform block size of 8x8, while the new H.264/AVC design is based primarily on a 4x4 transform. This allows the encoder to repre- sent signals in a more locally adaptive fashion, which reduces artifacts known colloquially as ringing. The smaller block size is also justified partly by the advances in the ability to better predict the content of the video using the techniques noted above, and by the need to pro- vide transform regions with boundaries that correspond to those of the smallest prediction regions.
- Hierarchical block transform:While in most cases, using the small 4x4 transform block size is perceptually beneficial, there are some signals that contain sufficient correlation to call for some method of using a repre-sentation with longer basis functions. The H.264/AVC standard enables 4x4 transform in two ways: 1) by using a hierarchical trans- form to extend the effective block size use for low-frequency chroma information to an 8x8 array and 2) by allowing the encoder to select a special coding type for intra coding, enabling extension of the length of the luma transform for low-frequency information to a 16x16 block size in a manner very similar to that applied to the chroma.
- Short word length transform: All prior standard designs have effec- tively required encoders and decoders to use more complex processing for transform computation. While previous designs have generally re- quired 32 bit processing, the H.264/AVC design requires only 16 bit arithmetic.
- Exact match inverse transform: In previous video coding standards, the transform used for representing the video was generally specified only within an error tolerance bound, due to the impracticality of obtaining an exact match to the ideal specified inverse transform. As a result, each decoder design would produce slightly different decoded video, causing a drift between encoder and decoder representation of the video and reducing effective video quality. Building on a path laid out as an optional feature in the effort, H.264/AVC is the first standard to achieve exact equality of decoded video content from all decoders.
- Arithmetic entropy coding: An advanced entropy coding method known as arithmetic coding is included in H.264/AVC. While arithmetic coding was previously found as an optional feature of H.263, a more ef-

fective use of this technique is found in H.264/AVC to create a very powerful entropy coding method known as CABAC (context adaptive binary arithmetic coding).

- Context adaptive entropy coding: The two entropy coding methods ap- plied in H.264/AVC, termed CAVLC (context adaptive variable length coding) and CABAC, both use context-based adaptivity to improve performance relative to prior standard designs. Robustness to data errors/losses and flexibility for operation over a variety of network environments is enabled by a number of design aspects new to the H.264/AVC standard, including the following highlighted features.
- Parameter set structure: The parameter set design provides for robust and efficient conveyance header information. As the loss of a few key bits of information (such as sequence header or picture header information) could have a severe negative impact on the decoding process when using prior standards, this key information was separated for handling in a more flexible and specialized manner in the H.264/AVC design.
- NAL unit syntax structure: Each syntax structure in H.264/AVC is placed into a logical data packet called a NAL unit. Rather than forcing a specific bitstream interace to the system as in prior video coding standards, the NAL unit syntax structure allows greater customization of the method of carrying the video content in a manner appropriate for each specific network.
- Flexible macroblock ordering (FMO): A new ability to partition the pic- ture into regions called slice groups has been developed, with each slice becoming an independently decodable subset of a slice group. When used effectively, flexible macroblock ordering can significantly enhance robustness to data losses by managing the spatial relationship between the regions that are coded in each slice. (FMO can also be used for a variety of other purposes as well.)
- Arbitrary slice ordering (ASO): Since each slice of a coded picture can be (approximately) decoded independently of the other slices of the picture, the H.264/AVC design enables sending and receiving the slices of the picture in any order relative to each other. This capability, first found in an optional part of , can improve end to end delay in real time applications, particularly when used on networks having out of order delivery behavior (e.g., internet protocol networks).
- Redundant pictures: In order to enhance robustness to data loss, the H.264/AVC design contains a new ability to allow an encoder to send redundant representations of regions of pictures, enabling a (typically somewhat degraded) representation of regions of pictures for which the primary representation has been lost during data transmis-sion.
- Data Partitioning: Since some coded information for representation of each region (e.g., motion vectors and other prediction information) is more important or more valu-able than other information for purposes of representing the video content,H.264/AVC allows the syntax of each slice to be separated into up to three different partitions for transmis- sion, depending on a categorization of syntax elements. This part of the design builds further on a path taken in MPEG-4 Visual and in an optional part of H.263++. Here, the design is simplified by having a single syntax with partitioning of that same syntax controlled by a spec-ified categorization of syntax elements.
- SP/SI synchronization/switching pictures: The H.264/AVC design in- cludes a new feature consisting of picture types that allow exact syn-chronization of the decoding process of some decoders with an ongoing video stream produced by other decoders without penalizing all de- coders with the loss of efficiency resulting from sending an I picture. This can enable switching a decoder between representations of the video content that used different data rates, recovery from data losses or errors, as well as enabling trick modes such as fast-forward, fast- reverse, etc.

4.2 Network Adaptation Layer (NAL)

The H.264 standard is designed in two distinct layers: a video coding layer (VCL), and a network adaptation layer (NAL). The NAL is designed in order to provide network friendliness to enable simple and effective customization of the use of the VCL for a broad variety of systems. Some key concepts of the NAL are NAL units, byte stream, and packet format uses of NAL units, parameter sets, and access units.

A. NAL Units

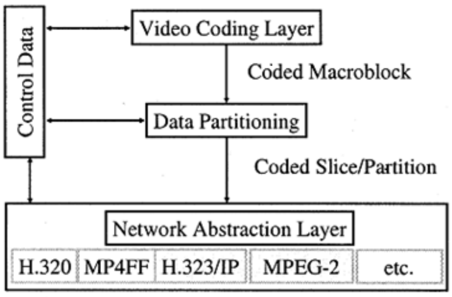


Figure 4.1: Structure of H.264/AVC video encoder

The coded video data is organized into NAL units, each of which is ef- fectively a packet that contains an integer number of bytes. The first byte of each NAL unit is a header byte that contains an indication of the type of data in the NAL unit, and the remaining bytes contain payload data of the type indicated by the header. NAL units can be carried in data packets without start code prefixes.

B. NAL Units in Packet-Transport System Use

In other systems (e.g., IP(Internet protocol)/RTP(Real Time Protocol) systems), the coded data is carried in packets that are framed by the system transport protocol, and identification of the boundaries of NAL units within the packets can be established without use of start code prefix patterns. In such systems, the inclusion of start code prefixes in the data would be a waste of data carrying capacity, so instead the NAL units can be carried in data packets without start code prefixes.

C. VCL and Non-VCL NAL Units

NAL units are classified into VCL and non VCL NAL units. The VCL NAL units contain the data that represents the values of the samples in the video pictures, and the non-VCL NAL units contain any associated additional information such as parameter sets.

D. Parameter Sets.

A parameter set is supposed to contain information that is expected to rarely change and offers the decoding of a large number of VCL NAL units. There are two types of parameter sets: sequence parameter sets, which apply to a series of consecutive coded video pictures called a coded video sequence; picture parameter sets, which apply to the decoding of one or more individual pictures within a coded video sequence.

E. Access Units

A set of NAL units in a specified form is referred to as an access unit. The decoding of each access unit results in one de-coded picture. Each access unit contains a set of VCL NAL units that together compose a primary coded picture. It may also be prefixed with an access unit delimiter to aid in locating the start of the access unit.

4.3 Video Coding Layer (VCL).

A. Pictures, Frames, and Fields

A coded video sequence in H.264/AVC consists of a sequence of coded pictures. A coded picture in can represent either an entire frame or a single field. Generally, a frame of video can be considered to contain two interleaved fields, a top and a bottom field. The top field contains even numbered row and the bottom field contains the odd numbered rows (starting with the second line of the frame). If the two fields of a frame were captured at different time instants, the frame is referred to as an interlaced frame, and otherwise it is referred to as a progressive frame. The coding representation in H.264/AVC based primarily on geometric concepts rather than being based on timing.

B. YCbCr Color Space and 4:2:0 Sampling

The video color space used by H.264/AVC separates a color representation into three components called Y, Cb, and Cr. Com-ponent Y is called luma, and represents brightness. The two chroma components Cb and Cr represent the extent to which the color deviates from gray toward blue and red, re- spectively. Because the human visual system is more sensitive to luma than

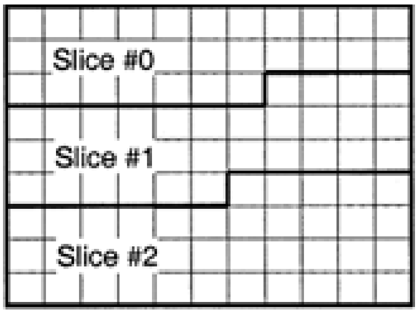


Figure 4.2: Subdivision of a picture into slices when not using FMO

chroma, H.264/AVC uses a sampling structure in which the chroma compo- nent has one fourth of the number of samples than the luma component (half the number of samples in both the horizontal and vertical dimensions). This is called 4:2:0 sampling with 8 bits of precision per sample. The sampling structure used is the same as in MPEG-2 Main profile video.

C. Division of the Picture Into Macroblocks

A picture is partitioned into fixed-size macroblocks that each cover a rectangular picture area of 16x16 samples of the luma component and 8x8 samples of each of the two chroma components. Macroblocks are the basic building blocks of the standard for which the decoding process is specified. The basic coding algorithm for a macroblock is described after we explain how macroblocks are grouped into slices.

D. Slices and Slice Groups

Slices are a sequence of macroblocks which are processed in the order of a raster scan when not using FMO. A picture maybe split into one or several slices as shown in Figure 4.2. A picture is therefore a collection of one or more slices in H.264/AVC. Slices are self contained in the sense that given the active sequence and picture parameter sets, their syntax elements can be parsed from the bit stream and the values of the samples in the area of the picture that the slice represents can be correctly decoded without use of data from other slices provided that utilized reference pictures are identical at encoder and decoder. Some information from other slices maybe needed to apply the deblocking filter across slice boundaries.

The case when FMO is not in use can be viewed as the simple special case of FMO in which the whole picture consists of a single slice group. Using FMO, a picture can be split into many macroblock scanning patterns such as interleaved slices, a dispersed macroblock allocation, one or more foreground slice groups and a leftover slice group, or a checker-board type of mapping. The latter two are illustrated in Figure 4.3.

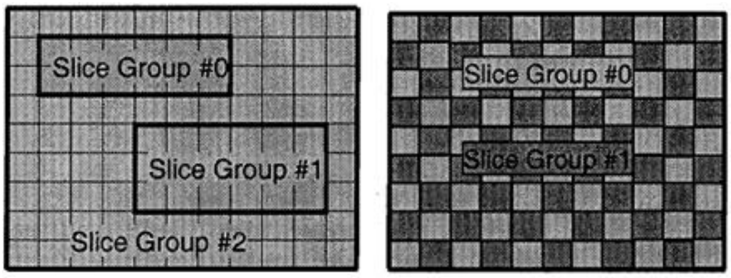


Figure 4.3: Subdivision of a QCIF frame into slices when utilizing FMO

Each slice can be coded using different coding types as follows.

- I slice: A slice in which all macroblocks of the slice are coded using intra prediction.
- P slice: In addition to the coding types of the I slice, some mac- roblocks of the P slice can also be coded using inter prediction with at mostonemotion-compensated predic-tion signal per prediction block.
- B slice:In addition to the coding types available in a P slice, some macroblocks of the B slice can also be coded using inter prediction withtwo motion-compensated pre-diction signals per prediction block. The above three coding types are

very similar to those in previous stan- dards with the exception of the use of reference pic-tures as described below. The following two coding types for slices are new.

- SP slice:A so-called switching P slice that is coded such that efficient switching between different pre-coded pic-tures becomes possible.
- SI slice:A so-called switching I slice that allows an exact match of a macroblock in an SP slice for random access and error recovery purposes.

E. Encoding and Decoding Process for Macroblocks

All luma and chroma samples of a macroblock are either spatially or temporally predicted,and the resulting prediction residual is encoded using transform coding. For transform coding purposes, each color component of the prediction residual signal is subdivided into smaller 4x4 blocks. Each block is transformed using an integer transform, and the transform coeffi- cients are quantized and encoded using entropy coding methods.Figure 3.4 shows a block diagram of the VCL for a macroblock. The input video sig- nal is split into macroblocks, the association of macroblocks to slice groups and slices is selected, and then each macroblock of each slice is processed as shown. An efficient parallel processing of macroblocks is possible when there are various slices in the picture.

F. Adaptive Frame/Field Coding Operation

In interlaced frames with regions of moving objects or camera motion, two adjacent rows tend to show a reduced degree of statistical dependency when compared to progressive frames in. In this case, it may be more efficient to compress each field separately. To provide high coding efficiency, the H.264/AVC design allows encoders to make any of the following decisions when coding a frame.

1. To combine the two fields together and to code them as one single coded frame (frame mode).
2. To not combine the two fields and to code them as sepa-rate coded fields (field mode).
3. To combine the two fields together and compress them as a single frame, but when coding the frame to split the pairs of two vertically adjacent macroblocks into either pairs of two field or frame macroblocks before coding them.

The choice between the three options can be made adaptively for each frame in a sequence. The choice between the first two options is referred to as picture-adaptive frame/field (PAFF) coding. When a frame is coded.

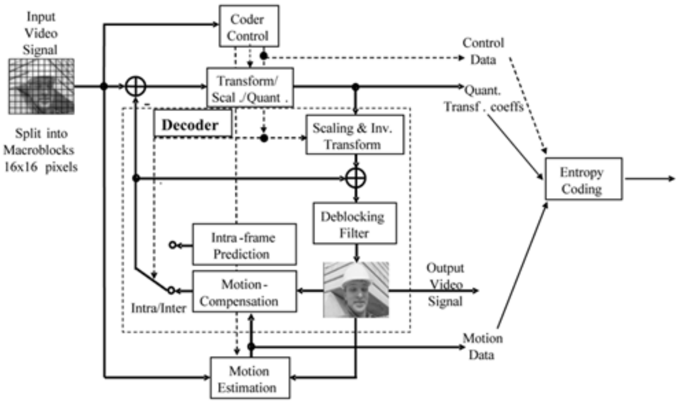


Figure 4.5: Basic coding structure for H.264/AVC for a macroblock.

as two fields, each field is partitioned into macroblocks and is coded in a manner very similar to a frame, with the following main exceptions:

- motion compensation utilizes reference fieldsrather than reference frames;
- the zig-zag scan of transform coefficients is different;.
- the strong deblocking strength is not used for filtering horizontal edges of macroblocks in fields, because the field rows are spatially twice as far apart as frame rows and the length of the filter thus covers a larger spatial area.

If a frame consists of mixed regions where some regions are moving and others are not, it is typically more efficient to code the nonmoving regions in frame mode and the moving regions in the field mode.

There- fore, the frame/field encoding decision can also be made independently for each vertical pair of macroblocks (a 16x32 luma region) in a frame. This coding option is referred to as macroblock-adaptive frame/field (MBAFF) coding.

G. Intra-Frame Prediction Each macroblock can be transmitted in one of several coding types depending on the slice-coding type. In all slice- coding types, the following types of intra coding are supported, which are denoted as Intra 4x4 or Intra 16x16 together with chroma predic- tion and IPCM(Intra Pulse Code Modulation) prediction modes. The Intra 4x4 mode is based on predicting each 4x4 luma block separately and is well suited for coding of parts of a picture with significant de- tail. The Intra 16x16 mode, on the other hand, performs prediction of the whole 16x16 luma block and is more suited for coding very smooth areas of a picture. In addition to these two types of luma prediction, a separate chroma prediction is conducted. As an alternative to Intra 4x4 and Intra16x16, the IPCM coding type allows the encoder to simply bypass the prediction and transform coding processes and instead di- rectly send the values of the encoded samples. The IPCM mode serves the following purposes.

1. It allows the encoder to precisely represent the values of the sam- ples.
2. It provides a way to accurately represent the values of anomalous picture content without significant data expansion
3. It enables placing a hard limit on the number of bits a decoder must handle for a macroblock without harm to coding efficiency

The prediction signal for each predictive-coded MxN luma block is obtained by displacing an area of the corresponding reference pic- ture, which is specified by a translational motion vector and a picture reference index. Thus, if the macroblock is coded using four 8x8 par- titions and each 8x8 partition is further split into four 4x4 partitions, a maximum of 16 motion vectors may be transmitted for a single P macroblock. The accuracy of motion compensation is in units of one quarter of the distance between luma samples. In case the motion vector points to an integer-sample position, the prediction signal con- sists of the corresponding samples of the reference picture; otherwise the corresponding sample is obtained using interpolation to generate

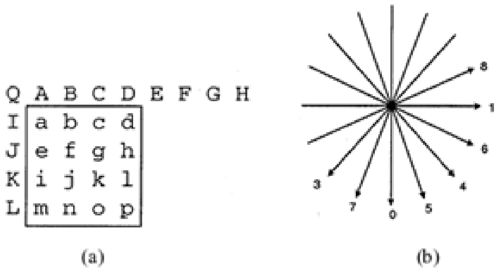


Figure 4 .5: (a) Intra 4x4 prediction is conducted for samples a-p of a block using samples A-Q. (b) Eight prediction directions for Intra 4x4 prediction.

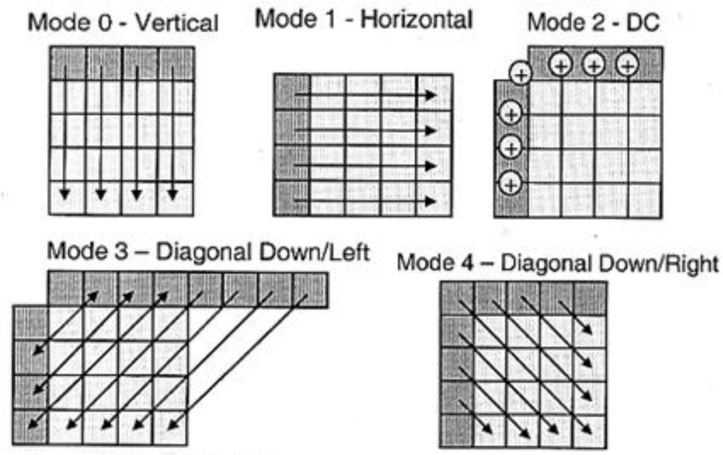


Figure 4.6: Five of the nine Intra 4x4 prediction modes

noninteger positions. The prediction values at half-sample positions are obtained by applying a one-dimensional 6-tap FIR filter horizon- tally and vertically. Prediction values at quarter-sample positions are generated by averaging samples at integer- and half-sample positions. Figurre 3.7 illustrates the fractional sample interpolation for samples a-k and n-r.The samples at half sample positions labeled b and h are derived by first calculating intermediate values b_1 and h_1 and, respec- tively by applying the 6-tap filter as follows:

Those modes are suitable to predict directional structures in a picture such as edges at various angles. Figure 4.6 shows five of the nine In- tra4x4 prediction modes. For mode 0 (vertical prediction), the samples above the 4x4 block are copied into the block as indicated by the ar- rows. Mode 1 (horizontal prediction) operates in a manner similar to vertical prediction except that the samples to the left of the 4x4 block are copied. For mode 2 (DC prediction), the adjacent samples are aver- aged as indicated in Figure 4.7. The remaining six modes are diagonal prediction modes which are called diagonal-down-left, diagonal-down- right, vertical-right, horizontal-down, vertical left, and horizontal up prediction. As their names indicate, they are suited to predict textures with structures in the specified direction.

The first two diagonal prediction modes are also illustrated in Figure 4.5. When samples E-H (Figure 4.5) that are used for the diagonal- down-left prediction mode are not available (because they have not yet been decoded or they are outside of the slice or not in an intra-coded macroblock in the constrained intra-mode), these samples are replaced by sample D. Note that in earlier draft versions of the Intra 4x4 prediction mode the four samples below sample L were also used for some prediction modes. However, due to the need to reduce memory access, these have been dropped, as the relative gain for their use is very small. When utilizing the Intra16x16 mode, the whole luma component of a macroblock is predicted. Four prediction modes are supported. Prediction mode 0 (vertical prediction), mode 1 (horizontal prediction), and mode 2 (DC prediction) are specified similar to the modes in Intra 4x4 prediction except that instead of 4 neighbors on each side to predict a 4x4 block, 16 neighbors on each side to predict a 16x16 block are used. For the specification of prediction mode 4 (plane prediction). The chroma samples of a macroblock are predicted using a similar prediction technique as for the luma component in Intra 16x16 macroblocks, since chroma is usually smooth over large areas.Intra prediction (and all other forms of prediction) across slice boundaries is not used, in order to keep all slices independent of each other.

H. Inter-Frame Prediction

1) Inter-Frame Prediction in P Slices: In addition to the intra mac- roblock coding types, various predictive or motion-compensated coding types are specified as P macroblock types. Each P macroblock type cor- responds to a specific partition of the macroblock into the block shapes used for motion-compensated prediction. Partitions with luma block sizes of 16x16, 16x8, 8x16, and 8x8 samples are supported by the syn- tax. In case partitions with 8x8 samples are chosen, one additional syn- tax element for each 8x8 partition is transmitted. This syntax element specifies whether the corresponding 8x8 partition is further partitioned into partitions of 8x4, 4x8, or 4x4 luma samples and corresponding chroma samples.

The accuracy of motion compensation is in units of one quarter of the distance between luma samples. In case the motion vector oints to an integer-sample position, the prediction signal consists of the corre- sponding samples of the reference picture; otherwise the corresponding sample is obtained using interpolation to generate noninteger positions. The prediction values at half-sample positions are obtained by apply- ing a one-dimensional 6-tap FIR filter horizontally and vertically. Pre- diction values at quarter-sample positions are generated by averaging samples at integer and half sample positions.

Figurre 3.9 illustrates the fractional sample interpolation for samples a-k and n-r. The samples at half sample positions labeled b and h are derived by first calculating intermediate values b_1 and h_1 , respectively by applying the 6-tap filter as follows:

$$b_1 = (E - 5F + 20G + 20H - 5I + J) \qquad (4.1)$$

$$h_1 = (A - 5C + 20G + 20M - 5R + T) \qquad (4.2)$$

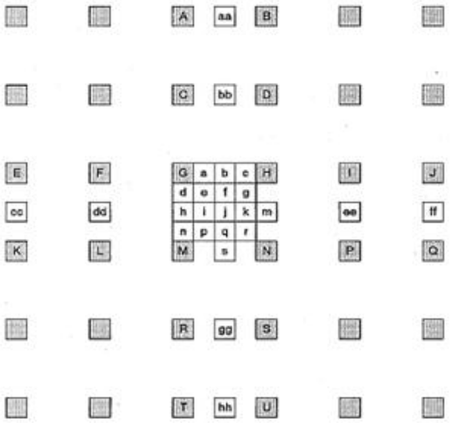


Figure 4.7: Filtering for fractional-sample accurate motion compensation. Upper-case letters indicate samples on the full-sample grid, while lower case samples indicate samples in between at fractional-sample positions

The final prediction values for locations b and h are obtained as follows and clipped to the range of 0-255:

$b = (b_1 + 16) >> 5$ (4.3)

$h = (h_1 + 16) >> 5$ (4.4)

The samples at half sample positions labeled as j are obtained by

$j_1 = \alpha - 5dd + 20h_1 + 20m_1 - 5ee + ff$ (4.5)

where intermediate values denoted as cc,dd,ee,m_1 and ff are obtained in manner similar to h_1 .The final prediction value j is then computed as

$j = (j_1 + 512) >> 10$ (4.6)

The samples at quarter sample positions labeled as a, c, d, n, f, i, k, and q are derived by averaging with upward rounding of the two nearest samples at integer and half sample positions as, for example, by The samples at quarter sample positions labeled as e, g, p, and r are derived by averaging with upward rounding of the two nearest samples at half sample positions in the diagonal direction as, for example, by

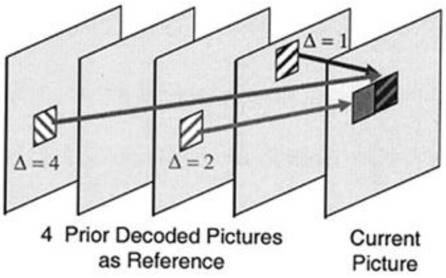
$a = (G + b + 1) >> 1$ (4.7) The samples at quarter sample
positions labeled as e, g, p, and r are
derived by averaging with upward rounding of the two nearest samples
at half sample positions in the diagonal direction as, for example, by

$e = (b + h + 1) >> 1$ (4.8)

The prediction values for the chroma component are always obtained by bilinear interpolation. Since the sampling grid of chroma has lower res- olution than the sampling grid of the luma, the displacements used for chroma have one-eighth sample position accuracy. The more accurate motion prediction using full sample, half sample and one-quarter sam- ple prediction represent one of the major improvements of the present method compared to earlier standards for the following two reasons.

1. The most obvious reason is more accurate motion representation.
2. The other reason is more flexibility in prediction filtering.

Full sample, half sample and one-quarter sample prediction represent different degrees of low pass filtering which is chosen automatically in the motion estimation process. In this respect, the 6-tap filter turns out to be a much better tradeoff between necessary prediction loop filtering and has the ability to preserve high-frequency content in the prediction loop. The syntax allows so-called motion vectors over picture bound- aries, i.e., motion vectors that point outside the image area. In this case, the reference frame is extrapolated beyond the image boundaries by repeating the edge samples before interpolation. The motion vector components are differentially coded using either median or directional prediction from neighboring blocks. No motion vector component pre- diction (or any other form of prediction) takes place across slice bound- aries. The syntax supports multipicture motion-compensated predic- tion . That is, more than one prior coded picture can be used as reference for motion-compensated prediction. Figure 4.8 illustrates the concept. Multiframe motion-compensated prediction requires both encoder and decoder to store the reference pictures used for inter prediction in a multipicture buffer



.

Figure 4.8: Multiframe motion compensation. In addition to the motion vector, also picture reference parameters Δ are transmitted. The concept is also extended to B slices

The decoder replicates the multipicture buffer of the encoder according to memory management control operations specified in the bitstream. Unless the size of the multipicture buffer is set to one picture, the index at which the reference picture is located inside the multipicture buffer has to be signalled. The reference index parameter is transmitted for each motion-compensated 16x16, 16x8, 8x16, or 8x8 luma block. Motion compensation for smaller regions than 8x8 use the same reference in- dex for prediction of all blocks within the 8x8 region. In addition to the motion-compensated macroblock modes described above, a P mac- roblock can also be coded in the so-called P Skip type. For this coding type, neither a quantized prediction error signal, nor a motion vector or reference index parameter is transmitted. The reconstructed signal is obtained similar to the prediction signal of a P 16x16 macroblock type that references the picture which is located at index 0 in the multipicture buffer. The motion vector used for reconstructing the P Skip macroblock is similar to the motion vector predictor for the 16x16 block. The useful effect of this definition of the P Skip coding type is that large areas with no change or constant motion like slow panning can be represented with very few bits.

2) Inter-Frame Prediction in B Slices:In comparison to prior video cod- ing standards, the concept of B slices is generalized in H.264/AVC. For example, other pictures can reference pictures containing B slices for motion-compensated prediction, depending on the memory man- agement control operation of the multipicture buffering. Thus, the substantial difference between B and P slices is that B slices are coded in a manner in which some macroblocks or blocks may use a weighted average of two distinct motion-compensated prediction values for build- ing the prediction signal. B slices utilize two distinct lists of reference pictures, which are referred to as the first (list 0) and second (list 1) ref- erence picture lists, respectively. Which pictures are actually located in each reference picture list is an issue of the multipicture buffer control and an operation very similar to the conventional MPEG-2 B pictures can be enabled if desired by the encoder.

In B slices, four different types of inter-picture prediction are supported: list 0, list 1, bi-predictive, and direct prediction. For the bi-predictive mode, the prediction signal is formed by a weighted average of motion- compensated list 0 and list 1 pre-diction signals. The direct prediction mode is inferred from previously transmitted syntax elements and can be either list 0 or list 1 prediction or bi-predictive.

J. Entropy Coding.

In H.264/AVC, two methods of entropy coding are supported. The sim- pler entropy coding method uses a single infinite extent codeword ta- ble for all syntax elements except the quantized transform coefficients. Thus, instead of designing a different VLC table for each syntax el- ement, only the mapping to the single codeword table is customized according to the data statistics. The single codeword table chosen is an exp-Golomb code with very simple and regular decoding properties.

For transmitting the quantized transform coefficients, a more efficient method called Context-Adaptive Variable Length Coding (CAVLC) is employed. In this scheme, VLC tables for various syntax elements are switched depending on already transmitted syntax elements. Since the VLC tables are designed to match the corresponding conditioned statistics, the entropy coding performance is improved in comparison to schemes using a single VLC table.

In the CAVLC (Context-Adaptive Variable Length Coding) entropy coding method, the number of nonzero quantized coefficients (N) and the actual size and position of the coefficients are coded separately. After zig-zag scanning of transform coefficients, their statistical distri- bution typically shows large values for the low frequency part decreas- ing to small values later in the scan for the high-frequency part. An example for a typical zig-zag scan of quantized transform coefficients could be given as follows: 7,6,2,0,1,0,0,1,0,0,0,0,0,0,0,0.

Based on this statistical behavior, the following data elements are used to convey information of quantized transform coefficients for a luma
4x4 block.

1. Number of Nonzero Coefficients (N) and Trailing 1s:Trailing 1s(T1s) indicate the number of coefficients with absolute value equal to 1 at the end of the scan. In the example T1s =2 and the number of coefficients is N=5. These two values are coded as a combined event. One out of 4 VLC tables is used based on the number of coefficients in neighboring blocks.
2. Encoding the Value of Coefficients: The values of the coefficients are coded. The T1s need only sign specification since they all are equal to +1 or -1. Please note that the statistics of coefficient values has less spread for the last nonzero coefficients than for the first ones. For this reason, coefficient values are coded in reverse scan order. In the examples above, -2 is the first coefficient value to be coded. A starting VLC is used for that. When coding the next coefficient (having value of 6 in the example) a new VLC may be used based on the just coded co-efficient. In this way adaptation is obtained in the use of VLC tables. Six exp-Golomb code tables are available for this adaptation.

Sign Information: One bit is used to signal coefficient sign. For T1s, this is sent as single bits. For the other coefficients, the sign bit is included in the exp-Golomb codes. Positions of each nonzero coefficient are coded by specifying the positions of 0s before the last nonzero coefficient. It is split into two parts:

4. TotalZeroes: This codeword specifies the number of zeros between the last nonzero coefficient of the scan and its start. In the ex- ample the value of TotalZeros is 3. Since it is already known that N=5, the number must be in the range 011. 15 tables are available for N in the range 115. (If there is no zero coefficient.)
5. RunBefore: In the example it must be specified how the 3 zeros are distributed. First the number of 0s before the last co-efficient is coded. In the example the number is 2. Since it must be in the range 03 a suitable VLC is used. Now there is only one 0 left. The number of 0s before the second last coefficient must therefore be 0 or 1. In the example the number is 1. At this point there are no 0s left and no more information is coded. The efficiency of entropy coding can be improved further if the Context-Adaptive Binary Arithmetic Coding (CABAC) is used . On the one hand, the usage of arithmetic coding allows the assignment of a noninteger number of bits to each symbol of an alphabet, which is extremely beneficial for symbol probabilities that are greater than 0.5. On the other hand, the usage of adaptive codes permits adaptation to non sta- tionary symbol statistics. Another important property of CABAC is its context modeling. The statistics of already coded syntax elements are used to estimate conditional probabilities. These conditional probabilities are used for switching several estimated probability models. In H.264/AVC, the arithmetic coding core engine and its associated probability estimation are specified as multiplication-free low-complexity methods using only shifts and table look-ups. Compared to CAVLC, CABAC typically provides a reduction in bit rate between 5

Figure 3.9: Five of the nine Intra 4x4 prediction modes.

K. In-Loop Deblocking Filter

One particular characteristic of block-based coding is the accidental production of visible block structures. Block edges are typically recon- structed with less accuracy than interior pixels and blocking is gener- ally considered to be one of the most visible artifacts with the present compression methods. For this reason, H.264/AVC defines an adaptive in-loop de-blocking filter, where the strength of filtering is controlled by the values of several syntax elements.

Whether the samples p_0 and q_0 as well as p_1 and q_1 are filtered is determined using quantization parame- ter (QP) dependent thresholds $\alpha(QP\ )$ and $\beta(QP\ )$ and. Thus, filtering of and only takes place if each of the following conditions is satisfied:

1.$|p_0| - |q_0| < \alpha(QP\ )$ (3.9)

2.$|p_0| - |p_0| < \beta(QP\ )$ (3.10)

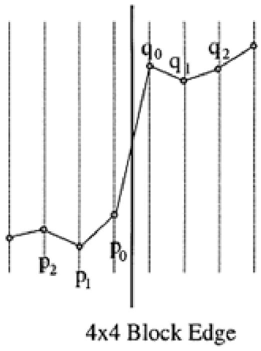


Figure 4.8 illustrates the principle of the deblocking filter using a visualization of a one-dimensional edge.

3.$|q_1| - |q_0| < \beta(QP)$ (3.11)

where the $\alpha(QP)$ is considerably smaller than $\beta(QP)$. Accordingly, filtering of p_1 or q_1 takes place if the corresponding following condition is satisfied:

$|p_2| - |p_0| < \beta(QP)$ or $|q_2 q_0| < \beta(QP)$ (3.12)

4.4 Profiles and Levels

Profiles and levels specify conformance points. These conformance points are designed to facilitate interoperability between various ap- plications of the standard that have similar functional requirements. A profile defines a set of coding tools or algorithms that can be used in generating a conforming bitstream, whereas a level places constraints on certain key parameters of the bitstream. All decoders conforming to a specific profile must support all features in that profile. Encoders are not required to make use of any particular set of features supported in a profile but have to provide conforming bitstreams, i.e., bitstreams that can be decoded by conforming decoders. In H.264/AVC, three profiles are defined, which are the Baseline, Main, and Extended Profile.

The Baseline profile supports all features in H.264/AVC except the following two feature sets:

– Set 1: B slices, weighted prediction, CABAC, field coding, and picture or macroblock adaptive switching between frame and field coding.

– Set 2: SP/SI slices, and slice data partitioning

The first set of additional features is supported by the Main profile. However, the Main profile does not support the FMO, ASO, and re- dundant pictures features which are supported by the Baseline profile

Thus, only a subset of the coded video se-quences that are decodable by a Baseline profile decoder can be decoded by a Main profile decoder. (Flags are used in the se-quence parameter set to indicate which profiles of decoder can decode the coded video sequence).

The Extended Profile supports all features of the Baseline profile, and both sets of features on top of Baseline profile, except for CABAC.

In H.264/AVC, the same set of level definitions is used with all profiles, but individual implementations may support a different level for each supported profile. There are 15 levels defined, specifying upper limits for the picture size (in macroblocks) ranging from QCIF to all the way to above 4k 2k, decoder-processing rate (in macroblocks per second) ranging from 250k pixels/s to 250M pixels/s, size of the multipicture buffers, video bit rate ranging from 64 kbps to 240 Mbps, and video buffer size.

4.5 Applications

The increased compression efficiency of H.264/AVC offers to enhance existing applications or enables new applications. A list of possible application areas is provided below.

– Conversational services which operate typically below 1Mbps with low latency requirements. The ITU-T SG16 is currently mod- ifying its systems reommendations to sup-port H.264/AVC use in such applications, and the IETF is working on the design of an RTP payload packetization. In the near term, these services would probably utilize the Baseline profile (possibly progressing over time to also use other profiles such as the Extended profile). Some specific applications in this category are given below.

⋆ H.320 conversational video services that utilize circuit switched
ISDN-based video conferencing.

⋆ 3GPP conversational H.324/M services.

⋆ H.323 conversational services over the Internet with best effort
IP/RTP protocols.

⋆ 3GPP conversational services using IP/RTP for trans-port
and SIP for session setup.

– Entertainment video applications which operate between 18+ Mbps with moderate latency such as 0.5 to 2 s. The H.222.0/MPEG-2
Systems specification is being modified to support these applica- tion. These applications would probably utilize the Main profile and include the following.

⋆ Broadcast via satellite, cable, terrestrial, or DSL.

⋆ DVD for standard and high-definition video.

⋆ Video-on-demand via various channels.

– Streaming services which typically operate at 50150 kbps and have
2 s or more of latency. These services would probably utilize the Baseline or Extended profile and may be distinguished by whether they are used in wired or wire-less environments as follows:

⋆ 3GPP streaming using IP/RTP(Internet Protocol/Real Time
Transfer Protocol) for transport and RTSP(Real Time Stream- ing Protocol) for session setup. This extension of the 3GPP specification would likely use Baseline profile only. Streaming over the wired Internet using IP/RTP protocol and RTSP for session setup. This domain which is currently dominated by powerful proprietary solutions might use the Extended profile and may require integration with some future system designs.

– Other services that operate at lower bit rates and are dis-tributed via file transfer and therefore do not impose delay constraints at all, which can potentially be served by any of the three profiles depending on various other systems requirements are:

⋆ 3GPP multimedia messaging services;

⋆ video mail.

V. Conclusion

The emerging H.264/AVC video coding standard hasbeen developed and standardized collaboratively by both theITU-T VCEG and ISO/IEC MPEG organizations. H.264/AVCrepresents a number of advances in standard video coding technology, in terms of both coding efficiency enhancementand flexibility for effective use over a broad variety of networktypes and application domains. Its VCL design is based on conventional block-based motion-compensated hybrid videocoding concepts, but with some important differences relativeto prior standards. We thus summarize some of the important differences:

- enhanced motion-prediction capability;
- use of a small block-size exact-match transform;
- adaptive in-loop deblocking filter;
- enhanced entropy coding methods.

When used well together, the features of the new design provide approximately a 50% bit rate savings for equivalent perceptualquality relative to the performance of prior standards (especially for higher-latency applications which allow some use ofreverse temporal prediction).

References

[1] Overview of the H.264/AVC Video Coding Standard- ThomasWiegand, Gary J. Sullivan, Senior Member, IEEE, Gisle Bjntegaard, and Ajay Luthra, Senior Member, IEEE , IEEE transactions on circuits and systems for video technology, vol. 13, no. 7, July 2003.

[2] Signal Processing: Image Communication 19 (2004) 793849 Video coding using the H.264/MPEG-4 AVC compression standard Atul Puria, Xuemin Chenb, Ajay Luthrac.

[3] The H.264 advance video compression standard Second Edition Iain E. Richardson Vcodex Limited, UK,A John Wiley and Sons, Ltd., Publication 2010.

[4] Performance comparison of the emerging H.264 video coding stan- dard with the existing standards Nejat Kamaci, Yucel Altunbasak Center for Signal and Image Processing Georgia Institute of Tech- nology, Atlanta, GA, USA 2003.

[5] Kalva, H. (2006): The H.264 video coding standard, IEEE mul- timedia, 13(4), pp. 86–90.

[6] Marpe, D. and Wiegand, T. and Sullivan, G.J. (2006): The H.264/MPEG4 advanced video coding standard and its applica- tions: IEEE Communications Magazine, 44(8), pp. 134-143.