International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-3, Issue-2)

Research Article | **February 2014**

Video Compression using H.264 AVC Standard

Rucha Bahirat, Amit Kolhe
Electronics and Telecommunication,
CSVTU University, India

Abstract*— Video compression is about reducing and removing redundant video data so that a digital video file can be effectively sent and stored. The process involves applying an algorithm to the source video to create a compressed file that is ready for transmission or storage.The latest video compression standard, H.264 (also known as MPEG-4 part 10/AVC for advanced video coding) promises a significant improvement over all previous video compression standards. H.264 is an open, licensed standard that supports the most efficient video compression techniques available today. Without compromising image quality, an H.264 encoder can reduce the size of digital video file by more than 80% compared with the motion JPEG format and as much as 50% more than with the MPEG-4 part-2 standard. In this paper we describe the overview of H.264 compression standard with motion compensated prediction. We describe a basic method called block-based coding employed by most video coding standards. We use graphical illustration to show the functionality.

Keywords*— H.264, MPEG, video compression, motion estimation, motion compensation, BMA, motion vector.

I. INTRODUCTION

Standardization has been an essential requirement for any technology that is supported by a large number of manufacturers. It makes it easier for the equipment from participating manufacturers to interact correctly and removes the need of customize the basic drives of each manufacturer's hardware and software. International study groups, VCEG (Video Coding Experts Group) of ITU-T (International Telecommunication Union—Telecommunication sector), and MPEG (Moving Picture Experts Group) [1] of ISO/IEC, have researched the video coding techniques for various applications of moving pictures since the early 1990s. Since then, ITU-T developed H.261 as the first video coding standard for videoconferencing application. MPEG-1 video coding standard was accomplished for storage in compact disk and MPEG-2 [2] (ITU-T adopted it as H.262) standard for digital TV and HDTV as extension of MPEG-1 [3]. Also, for covering the very wide range of applications such as shaped regions of video objects as well as rectangular pictures, MPEG-4 part 2 [4] standard was developed. This includes also natural and synthetic video/audio combinations with interactivity built in. On the other hand, ITU-T developed H.263 [5] in order to improve the compression performance of H.261 and the base coding model of H.263 were adopted as the core of some parts in MPEG-4 part 2. MPEG-1, -2, and -4 also cover audio coding. To provide better compression of video compared to previous standards, H.264/MPEG-4 part 10 [6] video coding standard was recently developed by the JVT (Joint Video Team) [7] consisting of experts from VCEG and MPEG. H.264 fulfils significant coding efficiency, simple syntax specifications, and flawless integration of video coding into all current protocols and multiplex architectures. Thus, H.264 can support various applications like video broadcasting, video streaming, video conferencing over fixed, and wireless networks and over different transport protocols. H.264 video coding standard has the same basic functional elements as previous standards (MPEG-1, MPEG-2, MPEG-4 part 2, H.261, and H.263) [3], i.e., transform for reduction of spatial correlation, quantization for bit rate control, motion compensated prediction for reduction of temporal correlation, and entropy encoding for reduction of statistical correlation.

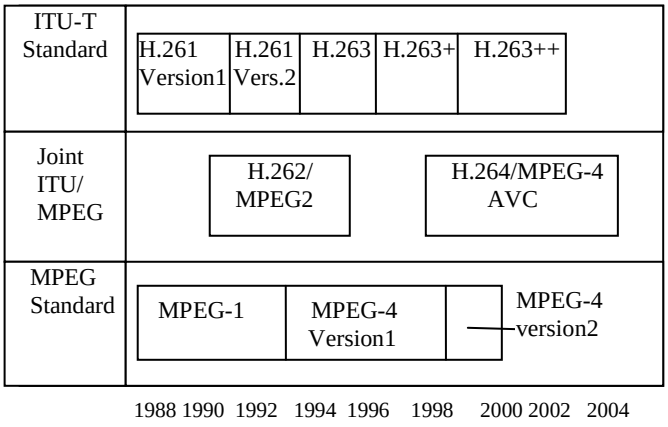

Fig.1. History of video standard

First section describes the various video coding standards. The rest of this paper is organized as following. In section 2 we present an overview of H.264/MPEG-4 AVC standard and the typical H.264 video codec. Section 3 introduces motion compensated prediction and motion vectors. Next section gives the result and Last section describes conclusion.

II. H.264/AVC

All paragraphs H.264/MPEG-4 Part 10 or AVC (Advanced Video Coding) is a video compression format, and is currently one of the most commonly used formats for the recording, compression, and distribution of video content. H.264/AVC/MPEG-4 Part 10 contains a number of new features that allow it to compress video much more effectively than older standards and to provide more flexibility for application to a wide variety of network environments. H.264 has the flexibility to support a wide variety of applications with very different bit rate requirements. For example, in entertainment video applications—which include broadcast, satellite, cable and DVD—H.264 will be able to deliver a performance of between 1 to 10Mbit/s with high latency, while for telecom services, H.264 can deliver bit rates of below 1Mbit/s with low latency.

H.264 has seven profiles, each targeting a specific class of applications. Each profile defines what feature set the encoder may use and limits the decoder implementation complexity. H.264 has 11 levels or degree of capability to limit performance, bandwidth and memory requirements. Each level defines the bit rate and the encoding rate in macroblock per second for resolutions ranging from QCIF to HDTV and beyond. The higher the resolution, the higher the level required. The basic coding structures of this standard is similar to that of earlier standards and is commonly referred to as motion-compensated— transform coding structure. 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. As in earlier standards, a slice consists of a sequence of macroblocks with each macroblock (MB) consisting of 16x16 luminance (y) and associated two chrominance (Cb and Cr) components. Each macroblock's 16x16luminance is partitioned into 16x16, 16x8, 8x16, and 8x8, and further, each 8x8 luminance can be sub-partitioned into 8x8, 8x4, 4x8 and 4x4. The 4x4 sub-macroblock partition is called a block. This partitioning is shown in fig.2. Therefore, variable block-size motion estimation uses smaller block size for moving objects and larger block size for background, which increase the video quality and the coding efficiency. The hierarchy of video data organization is as follows [8]:

Picture [slices {macroblocks (submacroblocks (blocks (pixels)))}]

A digitized video signal consists of a periodical sequence of images called frame. Each frame consists of a two dimensional array of pixels. Each pixel consists of three color components, R, G and B. Usually, pixel data is converted from RGB to another color space called YUV in which U and V components can be sub-sampled. A block-based coding approach divides a frame into macroblocks each consisting of say 16x16 pixels. In a 4:2:0 format, each MB consists of 16x16 = 256 Y components and 8x8 = 64 U and 64 V components. Each of three components of an MB is processed separately [9].

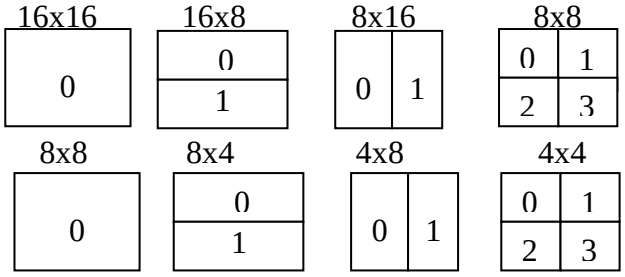

Fig. 2. Segmentation of macroblock for motion compensation

As can be seen in the "H.264/MPEG-4 AVC – Overview Block Diagram", the new standard is composed of several processing stages:

- Motion Estimation
- Transform (and Inverse Transform)
- Quantization (and Inverse Quantization)
- Loop Filter
- Entropy Coding

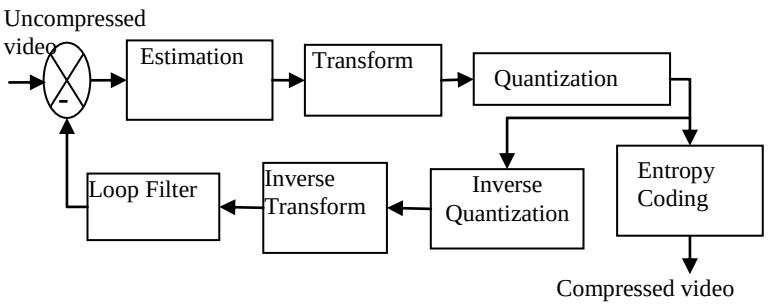

Fig. 3. H.264 AVC block diagram

1. *Estimation-*

H.264 distinguishes itself from the other MPEG standards mainly during motion estimation and its two components, motion compensation and motion vectors. Motion estimation is the process by which image information is assessed for similarities that can be reused in subsequent frames. This ultimately reduces the amount of data that is encoded and therefore reduces the bit rate.

Initially, an H.264 or MPEG-2/4 encoder receives either frames (progressive video) or fields (interlaced video) from a camera or other video source. At the start of motion estimation, these images are divided up in a raster of macroblocks that are organized into arbitrarily shaped slices. The raster is one aspect that sets H.264 apart. While MPEG-2/4 separate input frames or fields into a fixed raster of blocks containing 8x8 pixels, H.264 allows block sizes to vary. An H.264 encoder's raster can therefore include block sizes of 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, or 4x4 pixels. So, less detailed areas, such as a clear blue sky, may use a 16x16 block while more detailed areas, such as the edges of moving vehicles, will probably use the smaller,4x4 block size. Adjusting the block size as necessary not only makes H.264 encoding more efficient but it also improves the perceived quality of the image. Fixed gridlines are more jarring to the eye than jumbled blocks or chaotic patterns. As a result, most would agree that H.264 also noticeably improves the apparent video view. Deciding which block size to use and where is not something that is defined in the H.264 standard. This allows engineers to creatively compete for the most accurate and efficient motion estimation process. Consequently, a proficient motion estimation process can be what either makes or breaks an H.264 encoder [10].

2. *Transform*

In AVC, the transform coding always utilizes predictions to construct the residuals, even in the case of Intra MBs. That is, the pixel values in a MB are always predicted, either from neighbouring pixels in the same picture (in the case of Intra MBs), or from pixels in one or two previously decoded reference pictures (in the case of Inter MBs). The difference between the actual and predicted data is called residual error data. Discrete Cosine Transform (DCT) is a popular block-based transform for image and video compression. It transforms the residual data from time domain representation to frequency domain representation. Because most image and video are low frequency data, DCT can centralize the coding information [11].

3. *Quantization*

The coefficients from the transform stage are quantized, which reduces the overall precision of the integer coefficients and tends to eliminate high frequency coefficients, while maintaining perceptual quality. The quantizer is also used for constant bit rate applications where it is varied to control the output bit rate. The main functionality of quantization is to scale down the transformed coefficients and to reduce the coding information. Because human visual system is less sensitive to high frequency image component, some video and image compression standards may use higher scaling-value (quantization parameter) for high frequency data [11].

4. *Loop Filter*

The H.264/MPEG-4 AVC standard defines a de-blocking filter that operates on both 16x16 macroblocks and 4x4 block boundaries. In the case of macroblocks, the filter is intended to remove artifacts that may result from adjacent macroblocks having different estimation types (e.g. motion vs. intra estimation), and/or different quantizer scale. In the case of blocks, the filter is intended to remove artifacts that may be caused by transform/quantization and from motion vector differences between adjacent blocks. The loop filter typically modifies the two pixels on either side of the macroblock/block boundary using a content adaptive non-linear filter.

5. *Entropy Coding*

Before entropy coding can take place, the 4x4 quantized coefficients must be serialized. Depending on whether these coefficients were originally motion estimated or intra estimated, a different scan pattern is selected to create the serialized stream. The scan pattern orders the coefficients from low frequency to high frequency. Then, since higher frequency quantized coefficients tend to be zero, run-length encoding is used to group trailing zeros, resulting in more efficient entropy coding. The entropy coding stage maps symbols representing motion vectors, quantized coefficients, and macroblock headers into actual bits. Entropy coding improves coding efficiency by assigning a smaller number of bits to frequently used symbols and a greater number of bits to less frequently used symbols.

There are two major types of entropy coding: Variable Length Coding (VLC) and Context Adaptive Binary Arithmetic Coding (CABAC). CABAC offers superior coding efficiency over VLC by adapting to the changing probability distribution of symbols, by exploiting correlation between symbols, and by adaptively exploiting bit correlations using arithmetic coding. H.264 also supports Context Adaptive Variable Length Coding (CAVLC) which offers superior entropy coding over VLC without the full cost of CABAC [12].

III. MOTION COMPENSATED PREDICTION

Successive frames in a video sequence often are very similar. Coding their differences can reduce temporal redundancies and provide significant compression. However when a sequence of frames contains rapidly moving objects, sudden scene changes, or fade-ins and fade- outs the similarity between neighboring frames is reduced and compression is affected negatively. Temporally based predictive coding works best with certain kinds of inputs- namely a sequence of images with significant temporal redundancy. When used on image with little temporal redundancy, data expansion can occur. Video compression system avoids the problem of data expansion in two ways:

1. By using block-oriented 2-D transform, like JPEG's DCT-based coding when there is insufficient interframe cor relation (similarity between frames).
2. By tracking object movement and compensating it during the prediction and differencing process.

When there is insufficient interframe correlation to make predictive coding effective, the block-oriented 2-D transform is used. In block transform coding we divide an image into small non-overlapping blocks of equal size (e.g. 8X8) and processing the blocks independently using 2-D transform. In this coding, a reversible, linear transform (such as Fourier transform) is used to map each block or sub image into a set of transform coefficients, which are then quantized and coded. Frames compressed in this way (without a prediction residual) are called Intraframes or Independent frames (I frames) [13]. Depending on the H.264 profile, different types of frames such as I-frames, P-frames and B-frames, may be used by an encoder. An I-frame, or intra frame, is a self-contained frame that can be independently decoded without any reference to other images. The first image in a video sequence is always an I-frame. I-frames are needed as starting points for new viewers or resynchronization points if the transmitted bit stream is damaged. I-frames can be used to implement fast-forward, rewind and other random access functions. An encoder will automatically insert I-frames at regular intervals or on demand if new clients are expected to join in viewing a stream. The drawback of I-frames is that they consume much more bits, but on the other hand, they do not generate many artifacts. A P-frame, which stands for predictive inter frame, makes references to parts of earlier I and/or P frame(s) to code the frame. P-frames usually require fewer bits than I-frames, but a drawback is that they are very sensitive to transmission errors because of the complex dependency on earlier P and I reference frames. A B-frame, or bi-predictive inter frame, is a frame that makes references to both an earlier reference frame and a future frame [14].

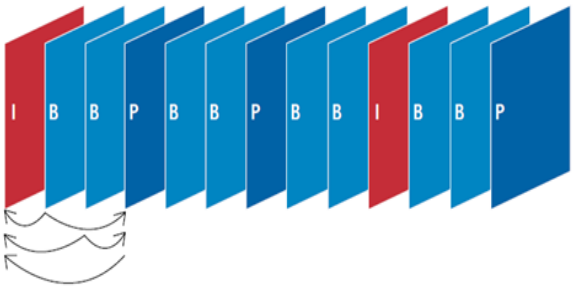

Fig. 5. Typical sequence with I,P and B frames

Fig.6. illustrate the block based motion compensated prediction. Each video frame is divided into non-overlapping rectangular regions- typically of size 4X4 to 16X16- called macroblocks. (Only macroblocks are shown in fig.6.)

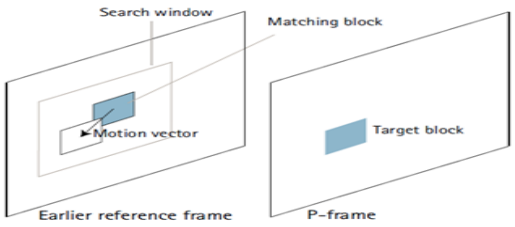

Fig.6. Illustration of block based motion compensation

The movement of each macroblock with respect to its most likely position in the previous (or subsequent) video frame, called the reference frame, is encoded with motion vector. The vector describes the motion by defining the horizontal and vertical displacement from the most likely position. The displacement typically is specified to nearest pixel, half pixel, or quarter pixel precision. If sub-pixel precision is used, the prediction must be interpolated from the combination of pixels in the reference frame. An encoded frame is based on previous frame (forward prediction) is called predictive frame (P-frame); one that based on subsequent frame (backward prediction) is called bidirectional frame (B- frame) [15].

During the motion estimation, the motion of object is measured and encoded into motion vectors. The search for the best motion vector requires that criterion of optimality be defined. For example, motion vectors may be selected on the basis of maximum correlation or minimum error between macrbolock pixels and the predicted pixels from the chosen reference frame. One of the most commonly used error measure is mean absolute distortion (MAD) [12].

MAD(x,y)=

$$\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}|f(x+i,y+j)-p(x+i+dx,y+j+dy)|$$

Where x and y are the coordinates of the upper left pixel of m x n macroblock being coded, dx and dy are the displacement from the reference frame and p is an array of predicted macroblock pixel values. Given the selection criterion like that of above equation, motion estimation is performed by searching for the dx and dy that minimize MAD(x,y) over the allowed range of motion vector displacement including sub pixel displacement. This process often is called block matching [16].

The H.264 standards use block-matching technique for motion estimation /compensation. The block matching algorithm uses an exhaustive search procedure to find the block that has the least error. More specifically, it compares the present block with every possible block in a predefined range surrounding the current position. For each candidate block, it calculates the prediction error. After all possible candidates have been investigated, the one with the least error.

To generate the prediction residual, we divide target frame into non-overlapping 16x16 macroblocks and compared each macroblock against every 16x16 region in the reference frame that fell within ± 16 pixels of the macroblock's position in target frame. We use the MAD to determine the best match by selecting displacement (dx,dy) with the lowest MAD [16]. The resulting displacement are the x and y components of the motion vectors shown in Fig. 7 (d).

IV. RESULT

Fig.7. (a) and (b) are taken from the flower video sequence and they are successive frames, Fig.7.(a) is the anchor frame and Fig. 7(b) is the target frame. Fig. 7(c) shows the motion compensated prediction residuals and Fig. (d) shows the estimated motion vector.

Fig. 7. (a) Anchor frame

Fig.7. (b) Target frame

PSNR 30db

Fig.7. (c) Predicted anchor frame

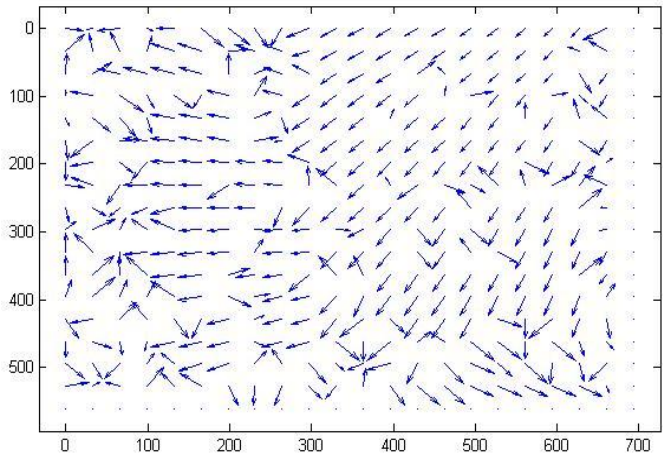

Fig. 7. (d) motion field for anchor frame

V. CONCLUSIONS

H.264 presents a huge step forward in video compression technology. It offers techniques that enable better compression efficiencies due to more accurate prediction capabilities, as well as improved resilience to errors. It provides new possibilities for creating better video encoders that enable higher quality video streams, higher frame rates and higher resolutions at maintained bit rates (compared with previous standards), or, conversely, the same quality video at lower bit rates.

.

REFERENCES

[1] MPEG website: http://www.mpeg.org.
[2] MPEG-2: ISO/IEC JTC1/SC29/WG11 and ITU-T, ISO/IEC 13818-2: Information Technology-Generic Coding of Moving Pictures and Associated Audio Information: Video, ISO/IEC and ITU-T1994.
[3] M. Ghanbari, Standard Codecs: Image Compression to Advanced Video Coding, IEE, Hertz, UK, 2003.
[4] MPEG-4: ISO/IEC JTCI/SC29/WG11, ISO/IEC 14 496:2000-2: Information on Technology-Coding of Audio-Visual Objects-Part 2: Visual, ISO/IEC, 2000.
[5] H.263: International Telecommunication Union, Recommendation ITU-T H.263: Video Coding for Low Bit Rate Communication, ITU-T, 1998.
[6] H.264: International Telecommunication Union, Recommendation ITU-T H.264: Advanced Video Coding for Generic Audiovisual Services, ITU-T, 2003.

[7] JVT website: ftp://standards.polycom.com.
[8] Atul Puri, Xuemin Chen, and Ajay Luthra, "Video coding using the H.264/MPEG-4 AVC compression standard" Elsevier Signal Processing: Image Communication 19 (2004) 793–849.
[9] Jian-Wen Chen, Chao-Yang Kao, and Youn-Long Lin, "Introduction to H.264 Advanced Video Coding" 0-7803-9451-8/06/2006 IEEE.
[10] file:///E:/motionalgo/h.264.htm.
[11] H. Malvar, A. Hallapuro, M. Karczewicz, and L.Kerofsky, "Low-Complexity Transform and Quantization in H.264/AVC",*in IEEE Transactions on Circuits and Systems for Video Technology*, pp: 598-603, Jul. 2003.
[12] P. List, A. Joch, J. Lainema, G. Bjntegaard, and M.Karczewicz, "Adaptive deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 614-619, 2003.
[13] Prabhat K. Andleigh and Kiran Thakrar, " Multimedia System Design"2011.
[14] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing"3rd ed.
[15] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard", *IEEE Transactions on Circuit and System for Video Technology, pp. 560-576, Jul. 2003.*
[16] T. Wedi, "Motion Compensation in H.264/AVC", *in IEEE Transactions on Circuits and Systems for Video Technology*
[17] M. Flierl, T. Wiegand, and B. Girod, "A Locally Optimal Design Algorithm for Block-Based Multi-Hypothesis Motion-Compensated Prediction" *in Data Compression Conference*, pp: 239-248, 1998.