H.264: An Emerging Standard for Advanced Video Coding

Article · April 2008

5 authors, including:

Anand Mankodia
Ganpat University

Vedvyas Jayprakash Dwivedi
Gokul Global University Siddhpur, Gujarat, India

Proceeding of the National Level Conference On Emerging Trends in Information Communication Technology

25th May 2008

Conveners

Ms. N. J. Vaghela

Mr. M. K . Joiser

(Lecturer CE/IT Department)

- **Computer Engineering Department**
- **Information Technology Department**

C. U. Shah College Of Engineering & Technology,
Wadhwan City

H.264:
An Emerging Standard for Advanced Video Coding

Anand Mankodia[1], Sunil Bhatt[2], Rakesh Soni[1]
Hiren Mevada[3], Ved Vyas Dwivedi[4]
[1] PG Students, CSE, V. T. Patel Dept. of EC Engineering.
[2]Lecturer, C.U.Shah College Of Engineering,Wadhwan.
E-Mail:sunil24_bhatt@yahoo.co.in,ph. +91 - 9428009859
[3]Asst. Professor, V. T. Patel Dept. of EC Engineering
[4]Head, V. T. Patel Dept. of EC Engg., CIT-Changa. Petlad, Anand, Gujarat
E-mail:mankodia@gmail.com, Ph. +91-9428460868.

Abstract -Broadcast television and home entertainment have been revolutionized by the advent of digital TV and DVD-video. These applications and many more were made possible by the standardization of video compression technology. The next standard in the MPEG series, MPEG4, is enabling a new generation of internet-based video applications whilst the ITU-T H.263 standard for video compression is now widely used in videoconferencing systems.[1]

After finalizing the original H.263 standard for video telephony in 1995, the ITU-T Video Coding Experts Group (VCEG) started work on two further development areas: a "short-term" effort to add extra features to H.263 (resulting in Version 2 of the standard) and a "long-term" effort to develop a new standard for low-bitrate visual communications. The long-term effort led to the draft "H.26L" standard, offering significantly better video compression efficiency than previous ITU-T standards. In 2001, the ISO Motion Picture Experts Group (MPEG) recognized the potential benefits of H.26L and the Joint Video Team (JVT) was formed, including experts from MPEG and VCEG. JVT's main task is to develop the draft H.26L "model" into a full International Standard. In fact, the outcome will be two identical standards: ISO MPEG4 Part 10 of MPEG4 and ITU-T H.264. The "official" title of the new standard is Advanced Video Coding (AVC); however, it is widely known by its old working title, H.26L and by its ITU document number, H.264.[1]

The goal of H.264 standard is to enhance compression performance, to provide network friendly video representation and to improve the distortion effects.

This paper illustrates the fundamentals of H.264 standard and its motion prediction process. It also gives some comparison between different standards. The paper finally list out some of the applications of H.264 standard.

Keywords- H.264, JVT, AVC, Intra prediction, Inter prediction, motion compensation, motion vector prediction, bit stream encoding/decoding.

I. INTRODUCTION

H.264 is an industry standard for video compression, the process of converting digital video into a format that takes up less capacity when it is stored or transmitted. Video compression (or video coding) is an essential technology for applications such as digital television, DVD-Video, mobile TV, videoconferencing and internet video streaming. Standardizing video compression makes it possible for products from different manufacturers (e.g. encoders, decoders and storage media) to inter-operate. An **encoder** converts video into a compressed format and a **decoder** converts compressed video back into an uncompressed format.

Recommendation H.264: Advanced Video Coding is a document published by the international standards bodies ITU-T (International Telecommunication Union) and ISO/IEC (International Organization for Standardization / International Electrotechnical Commission). It defines a format (syntax) for compressed video and a method for decoding this syntax to produce a displayable video sequence. The standard document does not actually specify how to encode (compress) digital video – this is left to the manufacturer of a video encoder – but in practice the encoder is likely to mirror the steps of the decoding process. Fig.1 shows the encoding and decoding processes and highlights the parts that are covered by the H.264 standard.

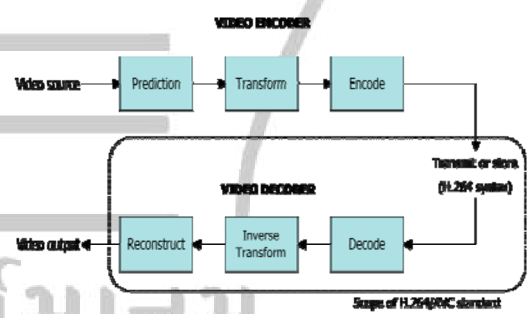


Fig.1 The H.264 video coding and decoding[2]

The H.264/AVC standard was first published in 2003. It builds on the concepts of earlier standards such as MPEG-2 and MPEG-4 Visual and offers the potential for better compression efficiency (i.e. better-quality compressed video) and greater flexibility in compressing, transmitting and storing video.

II. How does an H.264 codec work?

An H.264 video encoder carries out prediction, transform and encoding processes (see Fig.1) to produce a compressed H.264 bitstream. An H.264 video decoder carries out the complementary processes of decoding, inverse transform and reconstruction to produce a decoded video sequence.

III. Encoder Process

A. Prediction:

The encoder processes a frame of video in units of a **Macroblock** (16x16,8x8 displayed pixels). It forms a **prediction** of the macroblock based on previously-coded data, either from the current frame (**intra** prediction) or from other frames that have already been coded and transmitted (**inter** prediction). The encoder subtracts the prediction from the current macroblock to form a **residual**.

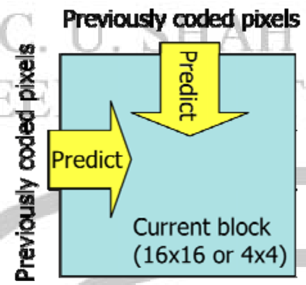


Fig2. Intra prediction[3]

The prediction methods supported by H.264 are more flexible than those in previous standards, enabling accurate predictions and hence efficient video compression. Intra prediction uses 16x16 and 4x4 block sizes to predict the macroblock from surrounding, previously-coded pixels within the same frame (Fig. 2).

Inter prediction uses a range of block sizes (from 16x16 down to 4x4) to predict pixels in the current frame from similar regions in previously-coded frames (Fig. 3).

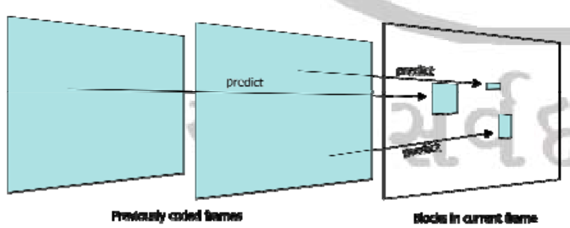


Fig. 3 Inter Prediction[3]

B. Prediction of Inter Macroblocks in P-slices[3]:

B1: Introduction:

Inter prediction creates a prediction model from one or more previously encoded video frames. The model is formed by shifting samples in the reference frame(s) (motion compensated prediction). The AVC CODEC uses block-based motion compensation, the same principle adopted by every major coding standard since H.261. Important differences from earlier standards include the support for a range of block sizes (down to 4x4) and fine sub-pixel motion vectors (1/4 pixel in the luma component).

B2: Tree structured motion compensation[3]:

AVC supports motion compensation block sizes ranging from 16x16 to 4x4 luminance samples with many options between the two. The luminance component of each macroblock (16x16 samples) may be split up in 4 ways as shown in Fig.4: 16x16, 16x8, 8x16 or 8x8. Each of the sub-divided regions is a macroblock partition. If the 8x8 mode is chosen, each of the four 8x8 macroblock partitions within the macroblock may be split in a further 4 ways as shown in Fig.5: 8x8, 8x4, 4x8 or 4x4 (known as macroblock sub-partitions). These partitions and sub-

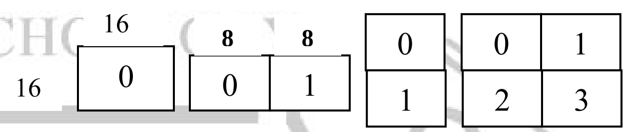


Fig.4 Macroblock partitions: 16x16, 8x16, 16x8,8x8[3]

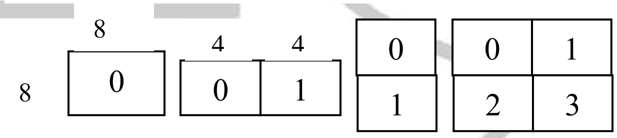


Fig.5 Macroblock partitions: 8x8, 4x8, 8x4, 4x4[3]

partitions give rise to a large number of possible combinations within each macroblock. This method of partitioning macroblocks into motion compensated sub-blocks of varying size is known as **tree structured motion compensation**.

A separate motion vector is required for each partition or sub-partition. Each motion vector must be coded and transmitted; in addition, the choice of partition(s) must be encoded in the compressed bitstream. Choosing a large partition size (e.g. 16x16, 16x8, 8x16) means that a small number of bits are required to signal the choice of motion vector(s) and the type of partition; however, the motion compensated residual may contain a significant amount of energy in frame areas with high detail. Choosing a small partition size (e.g. 8x4, 4x4, etc.) may give a lower-energy residual after motion compensation but requires a larger number of bits to signal the motion vectors and choice of partition(s). The choice of partition size therefore has a significant impact on compression performance. In general, a large partition size is appropriate for homogeneous areas of the frame and a small partition size may be beneficial for detailed areas.

The resolution of each chroma component in a macroblock (Cr and Cb) is half that of the luminance (luma) component. Each chroma block is partitioned in the same way as the luma component, except that the partition sizes have exactly half the horizontal and vertical resolution (an 8x16 partition in luma corresponds to a 4x8 partition in chroma; an 8x4 partition in luma corresponds to 4x2 in chroma; and so on). The horizontal and vertical components of each motion vector (one per partition) are halved when applied to the chroma blocks.

Example: Fig.6 shows a residual frame (without motion compensation). The AVC reference encoder selects the "best" partition size for each part of the frame, i.e. the partition size that minimizes the coded residual and motion vectors. The macroblock partitions chosen for each area are shown superimposed on the residual frame. In areas where there is little change between the frames (residual appears grey), a 16x16 partition is chosen; in areas of detailed motion (residual appears black or white), smaller partitions are more efficient.

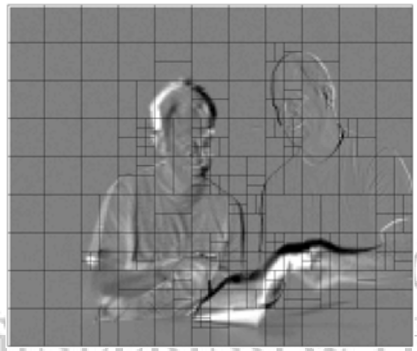


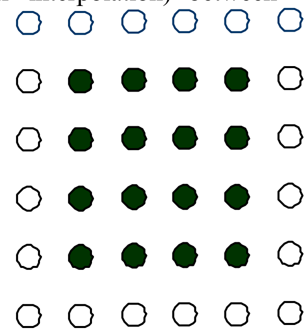
Fig. 6 Residual (without MC) showing optimum choice of partitions[2]

B3:Sub-pixel motion vectors:

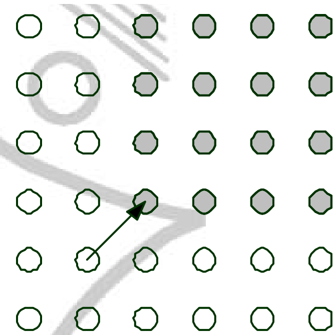
Each partition in an inter-coded macroblock is predicted from an area of the same size in a reference picture. The offset between the two areas (the motion vector) has ¼-pixel resolution (for the luma component). The luma and chroma samples at sub-pixel positions do not exist in the reference picture and so it is necessary to create them using interpolation from nearby image samples. Fig.7 gives an example. A 4x4 sub-partition in the current frame of fig.7(a) is to be predicted from a neighbouring region of the reference picture. If the horizontal and vertical components of the motion vector are integers (fig.7b), the relevant samples in the reference block actually exist (grey dots). If one or both vector components are fractional values (fig.7c), the prediction samples (grey dots) are generated by interpolation between adjacent samples in the reference frame (white dots).

Sub-pixel motion compensation can provide significantly better compression performance than integer-pixel compensation, at the expense of increased complexity. Quarter-pixel accuracy outperforms half-pixel accuracy.

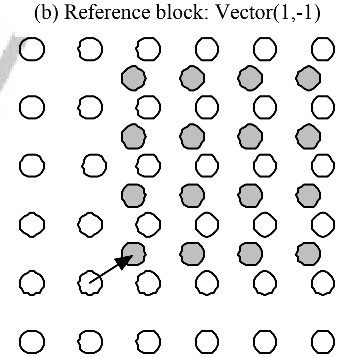
In the luma component, the sub-pixel samples at half-pixel positions are generated first and are interpolated from neighbouring integer-pixel samples using a 6-tap Finite Impulse Response filter. This means that each half-pixel sample is a weighted sum of 6 neighbouring integer samples. Once all the half-pixel samples are available, each quarter-pixel sample is produced using bilinear interpolation between neighbouring half- or integer-pixel samples. If the video source sampling is 4:2:0, 1/8 pixel samples are required in the chroma components (corresponding to ¼-pixel samples in the luma). These samples are interpolated (linear interpolation) between integer-pixel chroma samples.



(a) 4x4 block in current frame



(b) Reference block: Vector(1,-1)



(c) Reference block: Vector(0.75,-0.5)

Fig: 7 Integer & Subpixel prediction[2]

B4: Motion vector predictions:

Encoding a motion vector for each partition can take a significant number of bits, especially if small partition sizes are chosen. Motion vectors for neighbouring partitions are often highly correlated and so each motion vector is predicted from vectors of nearby, previously coded partitions. A predicted vector, MVp, is formed based on previously calculated motion vectors. MVD, the difference between the current vector and the predicted vector, is encoded and transmitted. The method of forming the prediction MVp depends on the motion compensation partition size and on the availability of nearby vectors. The "basic" predictor is the median of the motion vectors of the macroblock partitions or sub-partitions immediately above, diagonally above and to the right, and immediately left of the current partition or sub-partition. The predictor is modified if (a) 16x8 or 8x16 partitions are chosen and/or (b) if some of the neighbouring partitions are not available as predictors. If the current macroblock is skipped (not transmitted), a predicted vector is generated as if the MB was coded in 16x16 partition mode.

At the decoder, the predicted vector MVp is formed in the same way and added to the decoded vector difference MVD. In the case of a skipped macroblock, there is no decoded vector and so a motion- compensated macroblock is produced according to the magnitude of MVp.

C. Transform and quantization

A block of residual samples is transformed using a 4x4 or 8x8 **integer transform**, an approximate form of the

Discrete Cosine Transform (**DCT**). The transform outputs a set of **coefficients**, each of which is a weighting value for a standard basis pattern. When combined, the weighted basis patterns re-create the block of residual samples. Figure 4 shows how the inverse DCT creates an image block by weighting each basis pattern according to a coefficient value and combining the weighted basis patterns.

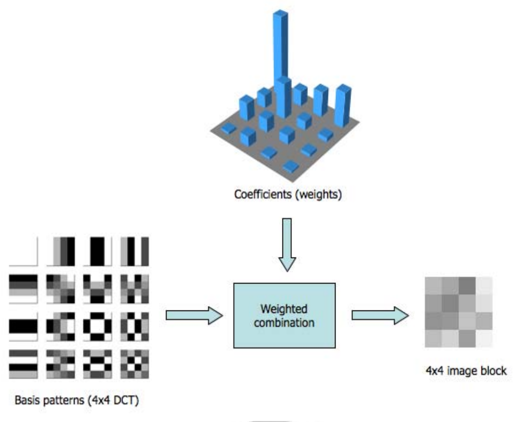


Fig. 8 Inverse transform: combining weighted basis patterns to create a 4x4 image block[3]

The output of the transform, a block of transform coefficients, is **quantized**, i.e. each coefficient is divided by an integer value. Quantization reduces the precision of the transform coefficients according to a quantization parameter (QP). Typically, the result is a block in which most or all of the coefficients are zero, with a few non-zero coefficients. Setting QP to a high value means that more coefficients are set to zero, resulting in high compression at the expense of poor decoded image quality. Setting QP to a low value means that more non-zero coefficients remain after quantization, resulting in better decoded image quality but lower compression.

Each residual macroblock is transformed, quantized and coded. Previous standards such as MPEG-1, MPEG-2, MPEG-4 and H.263 made use of the 8x8 Discrete Cosine Transform (DCT) as the basic transform. The “baseline” profile of H.264 uses three transforms depending on the type of residual data that is to be coded: a transform for the 4x4 array of luma DC coefficients in intra macroblocks (predicted in 16x16 mode), a transform for the 2x2 array of chroma DC coefficients (in any macroblock) and a transform for all other 4x4 blocks in the residual data. If the optional “adaptive block size transform” mode is used, further transforms are chosen depending on the motion compensation block size (4x8, 8x4, 8x8, 16x8, etc).

D. Bit stream encoding

The video coding process produces a number of values that must be **encoded** to form the compressed bit stream. These values include:

- Quantized transform coefficients
- Information to enable the decoder to re-create the prediction
- Information about the structure of the compressed data and the compression tools used during encoding
- Information about the complete video sequence.

These values and parameters (**syntax elements**) are converted into binary codes using **variable length coding** and/or **arithmetic coding**. Each of these encoding methods produces an efficient, compact binary representation of the information. The encoded bitstream can then be stored and/or transmitted.

IV. DECODER PROCESSES[3]

A. Bit stream decoding

A video decoder receives the compressed H.264 bitstream, decodes each of the syntax elements and extracts the information described above (quantized transform coefficients, prediction information, etc). This information is then used to reverse the coding process and recreate a sequence of video images.

B. Rescaling and inverse transform

The quantized transform coefficients are **re-scaled**. Each coefficient is multiplied by an integer value to restore its original scale. An inverse transform combines the standard basis patterns, weighted by the re-scaled coefficients, to re-create each block of residual data. These blocks are combined together to form a residual macroblock.

C. Reconstruction

For each macroblock, the decoder forms an identical prediction to the one created by the encoder. The decoder adds the prediction to the decoded residual to **reconstruct** a decoded macroblock which can then be displayed as part of a video frame.

V. H.264 IN PRACTICE[3]

A. Performance

Perhaps the biggest advantage of H.264 over previous standards is its compression performance. Compared with standards such as MPEG-2 and MPEG-4 Visual, H.264 can deliver:

1. Better image quality at the same compressed bitrate, or
2. A lower compressed bitrate for the same image quality.

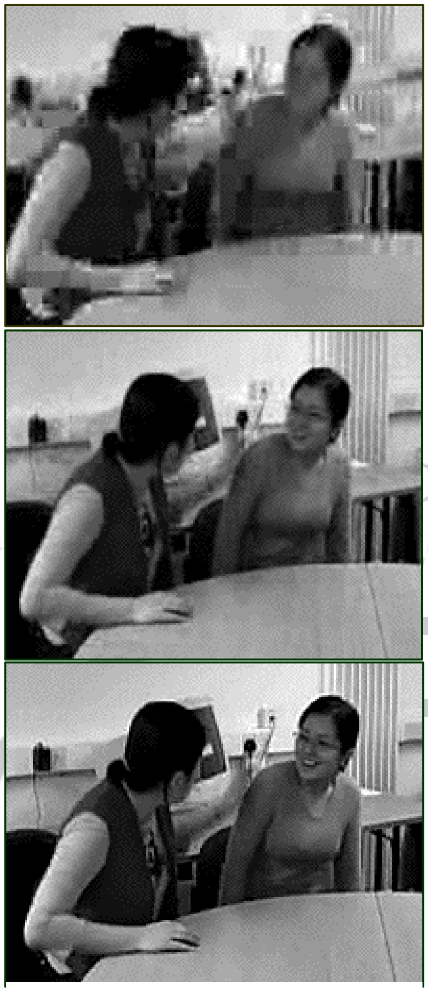


Fig. 9 A Video frame compressed at the same bitrate using MPEG-2 (first), MPEG-4 visual (Second) and H.264 compression (third)[3]

For example, a single-layer DVD can store a movie of around 2 hours' length in MPEG-2 format. Using H.264, it should be possible to store 4 hours or more of movie-quality video on the same disk (i.e. lower bitrate for the same quality). Alternatively, the H.264 compression format can deliver better quality at the same bitrate compared with MPEG-2 and MPEG-4 Visual (Fig.9). The improved compression performance of H.264 comes at the price of greater computational cost. H.264 is more sophisticated than earlier compression methods and this means that it can take significantly more processing power to compress and decompress H.264 video.

VI. Applications[2]

As well as its improved compression performance, H.264 offers greater flexibility in terms of compression options and transmission support. An H.264 encoder can select from a wide variety of compression tools, making it suitable for applications ranging from low-bitrate, low-delay mobile transmission through high definition consumer TV to professional television production. The standard provides integrated support for transmission or storage, including a packetised compressed format and features that help to minimize the effect of transmission errors.

H.264/AVC is being adopted for an increasing range of applications, including:

- High Definition DVDs (HD-DVD and Blu-Ray formats)
- High Definition TV broadcasting in Europe
- Apple products including iTunes video downloads, iPod video and MacOS
- NATO and US DoD video applications
- Mobile TV broadcasting
- Internet video
- Videoconferencing

VII. Conclusion

H.264 is the emerging standard for video compression. H.264 builds on the concepts of earlier standards such as MPEG-2 and MPEG-4 Visual and offers the potential for better compression efficiency (i.e. better-quality compressed video) and greater flexibility in compressing, transmitting and storing video. It is the industry standard for HDTV, Internet Video, Video Conferencing, DVD recording etc. So, carry on compressing!

Acknowledgements

The authors would like thanks EC Department of CIT-Changa for providing us useful laboratory facilities. We also thanks Hiren Mevada, Asst. Professor, EC department, CIT-Changa for critically review the paper and his necessary suggestions.

References

[1] ITU-T Rec. H.264 / ISO/IEC 11496-10, "*Advanced Video Coding*", Final Committee Draft, Document JVT-E022, September 2002
[2] ITU-T Rec. H.264 / ISO/IEC 11496-10, "*Advanced Video Coding"*, Final Committee Draft, Document JVT-G050, March 2003
[3] ITU-T Rec. H.264 / ISO/IEC 11496-10, *"Advanced Video Coding"*, Final Committee Draft, Document JVT-F100, December 2002
[4] I. Richardson, *"Video CODEC Design"*, John Wiley & Sons, 2002.
[5] D. Marpe, G Blättermann and T Wiegand, *"Adaptive Codes for H.26L"*, ITU-T SG16/6 document VCEG-L13, Eibsee, Germany, January 2001
[6] A. Hallapuro and M. Karczewicz, *"Low complexity transform and quantization – Part 1: Basic Implementation"*, JVT document JVT-B038, February 2001
[7] L. Chiariglione, *"The development of an integrated audiovisual coding standard: MPEG"*, Proceedings of the IEEE, vol.83, no 2, February 1995
[8] CARR, M.D., *"New video coding standard for the 1990s"*, Electronics & Communication Engineering Journal, June 1990, 2, (3), pp. 119-124.
[9] WELLS, N.D., *"Component codec standard for high-quality digital television"*, Electronics & Communication Engineering Journal, August 1992, 4, (4), pp. 195-202.
[10] http://www.mpeg.org
[11] http://www.vcodex.com