Volume 3, Issue 3, March 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Comparison of H.264 and MPEG-4 Codec Based on PSNR-Peak Signal to Noise Ratio Algorithm

Deepak Dembla
AIET, Jaipur
India

Biren Patel
M.Tech. Student
India

Ashish Kumar
AIET,Jaipur
India

Dr. Yogesh Bhomia
AIET, Jaipur
India

Abstract:* Over the past decades, digital video compression technologies have become an integral part of the way we create, communicate and consume visual information. Digital video communication can be found today in many application sceneries such as broadcast services over satellite and terrestrial channels, digital video storage, wires and wireless conversational services and etc. The data quantity is very large for the digital video and the memory of the storage devices and the bandwidth of the transmission channel are not infinite, so it is not practical for us to store the full digital video without processing. For instance, we have 720 x 480 pixels per frame, 30 frames per second, total 90 minutes full color video, and then the full data quantity of this video is about 167.96 Gigabytes. Thus, several video compression algorithms had been developed to reduce the data quantity and provide the acceptable quality as possible as can. This paper starts with an explanation of the basic concepts of video compression algorithms and then introduces and performs video compression standards H.264 and MPEG4. In paper highly flexible approach of H.264 & MPEG4 concentrates specifically on efficient compression of video frames base on PSNR. Key features of the standard include compression efficiency (providing significantly better compression than any previous standard), and a focus on popular applications of video compression.

***Keywords:* MPEG4, H.264, PSNR, Frames, Gigabytes.**

I. INTRODUCTION

The upcoming H.264/MPEG-4 AVC video compression standard promises a significant improvement over all previous video compression standards. In terms of coding efficiency, the new standard is expected to provide at least 2x compression improvement over the best previous standards and substantial perceptual quality improvements over both MPEG-2 and MPEG-4. The standard, being jointly developed by ITU-T and ISO/IEC, will address the full range of video applications including low bit-rate wireless applications, standard-definition and high-definition broadcast television, video streaming over the Internet, delivery of high-definition DVD content, and the highest quality video for digital cinema applications. The ITU-T name for the standard is H.264 (previously called H.26L), while the ISO/IEC name is MPEG-4 Advanced Video Coding (AVC) which will become Part 10 of the MPEG-4 standard. Since AVC is an extension to the current MPEG-4 standard, it will benefit from MPEG-4’s well-developed infrastructure tools (e.g. system layer and audio). It is expected that MPEG-4 AVC will be selected over the current MPEG-4 video compression standard, known as MPEG-4 Advanced Simple Profile (ASP), for the majority of applications that demand the highest compression and quality levels. Digital video shares all the features of other digital formats, including lossless transmission, lossless storage, and ease of editing. To ease storage / transmission requirements, compression is commonly performed on the video. Modern lossy compression techniques allow tremendous storage savings with little visible degradation. The MPEG-2 video compression standard [1, 2] has allowed the success of DVD-video and digital high definition television. New advancements in digital video compression technology have led to the recently. Finalized H.264 video compression standard [3], which is poised to follow the success of the highly accomplished MPEG-2 standard. The goal of this paper is to compare the two standards and highlight the differences between the two standards. Although both follow the same general framework, there are several fundamental key advancements in the H.264/AVC standard including a new integer transform, advanced arithmetic entropy coding, and the inclusion of an in-loop filter. The first part of this paper will provide an introduction to some basics of video compression. The next part will go into more detail of the technical Differences between the MPEG-4 and H.264/AVC video coding standards. Finally, we will discuss some experiments performed to illustrate the compression gains offered by H.264/AVC over MPEG-4.

II. WHAT IS CODEC?

A **codec** is a device or computer program capable of encoding or decoding a digital data stream or signal. The word *codec* is a portmanteau of "**co**der-**dec**oder" or, less commonly, "**co**mpressor-**dec**ompressor". A *codec* (the *program*) should not be confused with a coding or compression *format* or *standard* – a format is a document (the standard), a way of storing data, while a codec is a program (an *implementation*) which can read or write such files. In practice, however, "codec" is sometimes used loosely to refer to formats [7]. A codec encodes a data stream or signal for transmission, storage or encryption, or decodes it for playback or editing. Codecs are used in video conferencing, streaming media and video editing applications. A video camera's analog-to-digital converter (ADC) converts its analog signals into digital signals, which are then passed through a video compressor for digital transmission or storage. A receiving device, then, runs the signal through a video decompressor, then a digital-to-analog converter (DAC) for analog display. The term *codec* is also used as a generic name for a videoconferencing unit [7].

A. *About MPEG4*

MPEG-4Visual (Part 2 of the MPEG-4 group of standards) was developed by the Moving Picture Experts Group (MPEG), a working group of the International Organization for Standardization (ISO). These groups of several hundred technical experts (drawn from industry and research organizations) meet at 2–3 month intervals to develop the MPEG series of standards. MPEG-4 (a multi-part standard covering audio coding, systems issues and related aspects of audio/visual communication) was first conceived in 1993 and Part 2 was standardized in 1999.

B. *About H.264/AVC*

The H.264/AVC standardization effort was initiated by the Video Coding Experts Group (VCEG), a working group of the International Telecommunication Union (ITU-T) that operates in a similar way to MPEG and has been responsible for a series of visual telecommunication standards. The final stages of developing the H.264/AVC standard have been carried out by the Joint Video Team, a collaborative effort of both VCEG and MPEG, making it possible to publish the final standard under the joint auspices of ISO/IEC (as MPEG-4 Part 10) and ITU-T (as Recommendation H.264/AVC) in 2003.

C. *How video compression works*

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. To play the compressed file, an inverse algorithm is applied to produce a video that shows virtually the same content as the original source video. The time it takes to compress, send, decompress and display a file is called latency. The more advanced the compression algorithm, the higher the latency, given the same processing power. A pair of algorithms that works together is called a video codec (encoder/decoder). Video codecs that implement different standards are normally not compatible with each other; that is, video content that is compressed using one standard cannot be decompressed with a different standard. For instance, an MPEG-4 Part 2 decoder will not work with an H.264/AVC encoder. This is simply because one algorithm cannot correctly decode the output from another algorithm but it is possible to implement many different algorithms in the same software or hardware, which would then enable multiple formats to be compressed. Different video compression standards utilize different methods of reducing data, and hence, results differ in bit rate, quality and latency. Results from encoders that use the same compression standard may also vary because the designer of an encoder can choose to implement different sets of tools defined by a standard. As long as the output of an encoder conforms to a standard’s format and decoder, it is possible to make different implementations. This is advantageous because different implementations have different goals and budget. Professional non-real-time software encoders for mastering optical media should have the option of being able to deliver better encoded video than a real-time hardware encoder for video conferencing that is integrated in a hand-held device. A given standard, therefore, cannot guarantee a given bit rate or quality. Furthermore, the performance of a standard cannot be properly compared with other standards, or even other implementations of the same standard, without first defining how it is implemented. A decoder, unlike an encoder, must implement all the required parts of a standard in order to decode a compliant bit stream. This is because a standard specifies exactly how a decompression algorithm should restore every bit of a compressed video. The graph below provides a bit rate comparison, given the same level of image quality, among the following video standards: Motion JPEG, MPEG-4 Part 2 (no motion compensation), MPEG-4 Part 2 (with motion compensation) and H.264/AVC (baseline profile).

III. BASIC VIDEO COLOR SPACE CONVERSATION

Numerous YUV formats are defined throughout the video industry. This article identifies the 8-bit YUV formats that are recommended for video rendering in Windows. Decoder vendors and display vendors are encouraged to support the formats described in this article. This article does not address other uses of YUV color, such as still photography [8].
The formats described in this article all use 8 bits per pixel location to encode the Y channel (also called the luma channel), and use 8 bits per sample to encode each U or V chroma sample. However, most YUV formats use fewer than 24 bits per pixel on an average, because they contain fewer samples of U and V than of Y. This article does not cover YUV formats with 10-bit or higher Y channels [9].
Note: For the purposes of this article, the term U is equivalent to Cb, and the term V is equivalent to Cr.

This article covers the following topics:

- YUV Sampling. (describes the most common YUV sampling techniques)
- Surface Definitions. (describes the recommended YUV formats)
- Color Space and Chroma Sampling Rate Conversions.(provides some guidelines for converting between YUV and RGB formats and for converting between different YUV formats)
- Identifying YUV formats in Media Foundation.(explains how to describe YUV format types in Media Foundation).

IV. CHROMINANCE SUB-SAMPLING

Chroma channels can have a lower sampling rate than the luma channel, without any dramatic loss of perceptual quality. A notation called the "A: B: C" notation is used to describe how often U and V are sampled relative to Y:

- 4:4:4 means no downsampling of the chroma channels.
- 4:2:2 means 2:1 horizontal downsampling, with no vertical downsampling. Every scan line contains four Y samples for every two U or V samples.
- 4:2:0 means 2:1 horizontal downsampling, with 2:1 vertical downsampling.
- 4:1:1 means 4:1 horizontal downsampling, with no vertical downsampling. Every scan line contains four Y samples for each U and V sample. It is less common than other formats and is not discussed in detail in this article.

The following diagrams shows how chroma is sampled for each of the downsampling rates. Luma samples are represented by a cross and chroma samples are represented by a circle.

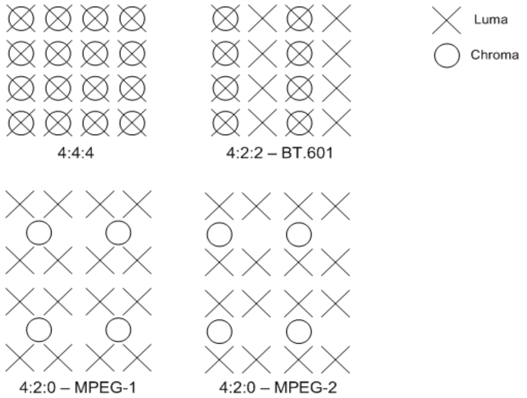

Fig. 1 Sample of Luma and Chroma.

The dominant form of 4:2:2 sampling is defined in ITU-R Recommendation BT.601. There are two common variants of 4:2:0 sampling. One of these is used in MPEG-2 video, and the other is used in MPEG-1 and in ITU-T Recommendations H.261 and H.263. Compared with the MPEG-1 scheme, it is simpler to convert between the MPEG-2 scheme and the sampling grids defined for 4:2:2 and 4:4:4 formats. For this reason, the MPEG-2 scheme is preferred in Windows and should be considered as the default interpretation for 4:2:0 formats [10, 11].

V. UNDERSTANDING ABOUT I-FRAMES, P-FRAME AND B-FRAMES

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 Reference both preceding and succeeding I- or P-frames.[13,14]

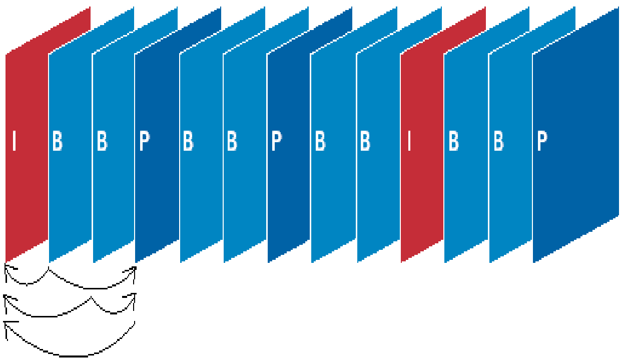

Fig. 2: A typical sequence with I-, B- and P-frames.

When a video decoder restores a video by decoding the bit stream frame by frame, decoding must always start with an I-frame. P-frames and B-frames, if used, must be decoded together with the reference frame(s).

In the H.264/AVC baseline profile, only I- and P-frames are used. This profile is ideal for network cameras and video encoders since low latency is achieved because B-frames are not used. [13, 14].

VI. SUBJECTIVE QUALITY MEASUREMENT PSNR

The PSNR block computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed or reconstructed image.

The *Mean Square Error (MSE)* and the *Peak Signal to Noise Ratio (PSNR)* are the two error metrics used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower the error.

To compute the PSNR, the block first calculates the mean-squared error using the following equation:

$$PSNR(dB) = 10 * \log(\frac{255^2}{MSE})$$

$$MSE = \sum_{i=1}^{x}\sum_{j=1}^{y}\frac{(|A_{ij} - B_{ij}|)^2}{x * y}$$

The range of PSNR value should be 1db t 100db.In Normal codec currently PSNR value will be nearly 50 db.If it is 100db means both file entered for processing are same, but it not possible[15,16].

VII. SYSTEM REQUIREMENT SPECIFICATION

- Original & generated raw files should be of same file format.(e.g. Both files should have .YUV format).
- Original & generated raw files should have same number of frames.
- In case, YUV and PCM files do not contain header information, there is a need to get required information from the user.
- It should display output as Frame by Frame for video and sample as Block by Block (sample size is 1024) for audio. In case, if data are not equal to frame size or sample Block size, then it discards that frame or block.

VIII. FLOW OF RAW FILE (*.YUV) & PROCESSED FILES (*.YUV)[EXPERIMENTAL SETUP]

Now, Deals with only original raw files and generated raw files of any Codec. So it should work with any Codec.

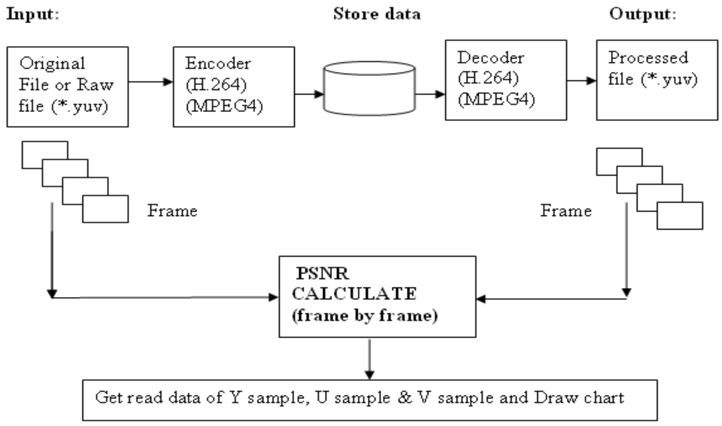

Fig.3 Comparison of I/P & O/P file base on PSNR.

IX. RESULTS AND ANALYSIS

Now in this part, Show bellow table, it shows testing for 1 to 30 (*.yuv (4:2:0)) frames and reading about input raw file & generated raw file.

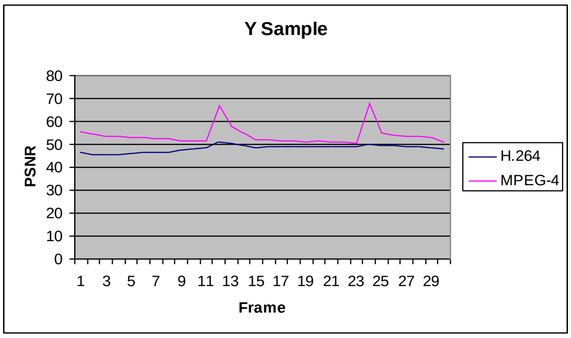

Figure 4. Comparison of Y samples frames of of H.264/AVC and MPEG-4

Table 1 PSNR ratio of Input file/output file which is decoded by H.264/AVC codec

	H.264/AVC (Kbps)		
Frame Number	Y Sample	U Sample	V Sample
1	46.394039	53.996105	53.960239
2	45.490421	52.379498	52.67915
3	45.558582	52.31337	52.632984
4	45.626572	52.256119	52.971619
5	46.016186	52.633461	53.199165
6	46.525066	53.072243	53.385113
7	46.661331	53.100918	53.474575
8	46.680725	53.23785	53.463364
9	47.327545	52.922421	53.013885
10	48.199886	53.446217	53.747978
11	48.510048	53.598457	53.995232
12	51.058014	56.124378	56.917618
13	50.453037	55.10667	55.68462
14	49.531715	53.82954	54.601696
15	48.377861	52.510666	53.460659
16	49.13126	52.998768	53.605232
17	48.955276	52.881874	53.409069
18	48.806713	52.888805	53.390995
19	48.99382	52.921059	53.249226
20	48.844971	53.223209	53.555061
21	49.071579	52.920208	53.397266
22	48.915665	52.852592	53.118
23	48.904636	52.556797	53.034824
24	50.091148	54.425903	55.320705
25	49.558479	53.583546	54.404991

Table 2 PSNR ratio of Input file/output file which is decoded by MPEG-4 codec

	MPEG-4 (Kbps)		
Frame Number	Y Sample	U Sample	V Sample
1	55.468803	65.050385	62.511284
2	54.318981	62.425312	61.731377
3	53.681393	61.127552	60.438801
4	53.474285	60.660854	60.390961
5	53.050465	59.83023	59.578423
6	52.881157	59.411217	59.330711
7	52.508533	58.9986	58.915836
8	52.44199	58.661243	58.868668
9	51.7272	57.458427	57.761833
10	51.628216	56.930038	57.109093
11	51.475655	56.659435	56.816589
12	67.18438	71.232536	74.731323
13	57.996479	63.70266	64.284622
14	55.157059	59.568172	59.762192
15	52.114658	57.021049	57.86858
16	51.876286	56.144016	56.942921
17	51.412262	55.621651	56.314453
18	51.292988	55.318043	55.840641
19	51.119892	55.21685	55.721802
20	51.405136	55.341484	55.789219
21	51.018719	55.07494	55.490685
22	50.812237	54.501984	55.064877
23	50.687057	54.43845	55.047882
24	68.042641	72.638725	74.336235
25	55.16898	60.458084	60.999794
26	53.965385	58.052902	58.75359
27	53.380753	57.124763	57.925919
28	53.349094	56.86871	57.894207
29	53.243076	56.752716	57.915142
30	50.940994	54.465359	56.243202

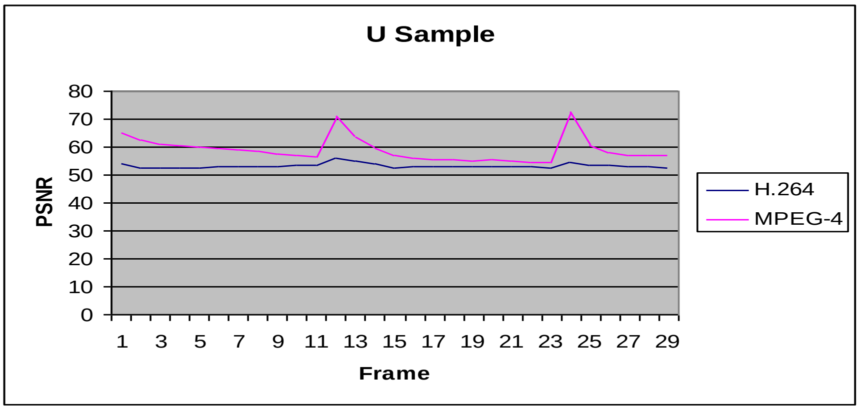

Figure 5. Comparison of U samples of H.264/AVC and MPEG-4

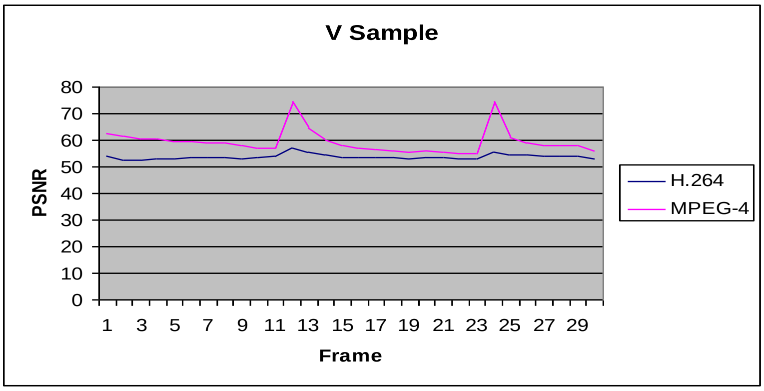

Fig. 6 Comparison of V samples of H.264/AVC and MPEG-4

X. CONCLUSIONS

Different choices during the design of a CODEC and different strategies for coding control can lead to significant variations in compression and computational performance between CODEC implementations. However, the best performance that may be achieved by a CODEC is limited by the available coding tools. The performance examples presented here and many other studies in the literature indicate that H.264/AVC has the ability to out-perform MPEG-4Visual convincingly (which in turn performs significantly better than MPEG-4). Performance is only one of many factors that influence whether a new technology is successful in the marketplace and we will examine some of the other issues that are currently shaping the commercial market for video coding.

REFERENCES

[1] "*Generic Coding of Moving Pictures and Associated Audio Information*", ITU-T and ISO/IEC JTC 1 Std. 13 818-2, 1994.

[2] B. G. Haskell, A. Puri, and A. N. Netravalli, "*Digital Video: An Introduction to MPEG-2*", New York, NY: Chapman and Hall, 1997.

[3] "*Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding*", ITU-T and ISO/IEC JTC 1 Std. ITU-T Recommendation H.264 and ISO/IEC 13 818-10 (MPEG-4), 1994.

[4] "*Television – 1280 × 720 Progressive Image Sample Structure – Analog and Digital Representation and Analog Interface*", Society of Motion Picture and Television Engineers Std. 296M, 2001.

[5] J. Taylor, "*DVD Demystified*", Dallas, TX: McGraw-Hill, 1998.

[6] K. B. Benson and D. G. Fink, "*HDTV: Advanced Television for the 1990s*". Dallas,TX : McGraw-Hill, 1990.

[7] Introduction about Codec: http://en.wikipedia.org/wiki/Codec

[8] K. Jack, "*Video Demystified*". San Deigo, CA: HighText Interactive, 1996.

[9] C. A. Poynton, "*A Technical Introduction to Digital Video*", New York, NY:Wiley, 1996.

[10] G. Sullivan, "Approximate theoretical analysis of RGB to YCbCr to RGB conversion error," *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG*,Doc. JVT-I017, July 2003.

[11] A. M. Tekalp, "*Digital Video Processing*", Englewood Cli.s, NJ: Prentice-Hall, 1995

[12] "*Encoding Parameters of Digital Televisions for Studios*", International Telecommunication Union Std. Recommendation ITU-R BT.601-4, 1990.

[13] White paper**:** "H.264 video compression standard".Axis Communication. ©2008 Axis Communications AB, ETRAX, ARTPEC.

[14] White paper: "Comparing Media Codecs for Video Content" Jeremiah Golston (Distinguished Member Technical Staff, Texas Instruments) j-golston@ti.com#250 at Embedded Systems Conference – San Francisco 2004

[15]. C. J. van den Branden Lambrecht and O. Verscheure, "Perceptual quality measure using a spatiotemporal model of the Human Visual System, Digital Video Compression Algorithms and Technologies", *Proc. SPIE*, **2668**, San Jose, 1996.

[16] Quality measures base on PSNR: http://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio.

