RESEARCH ARTICLE OPEN ACCESS

# Performance Analysis of Multi-view Video Delivery over HTTP with and Without H.264 Coding

Cherlo Indrasena Reddy, Mc. Mamatha
Final Year M.Tech(DSCE) Student, Dept. of ECE, BIT Institute of Technology, Hindupur, Anantapur (A.P) – INDIA.
Assistant Professor, Dept. of ECE, BIT Institute of Technology, Hindupur, Anantapur (A.P) – INDIA.

**Abstract**
This paper puts light on different video delivery algorithms with the H.264 coding and without it in HTTP channel. The H.264 coding and decoding algorithm is developed and tested on the Matlab Paltform and the HTTP channel is developed using Java platform. This paper deals with the performance analysis of the speed of delivery with and without H.264 coding on the video. The performance measure is the time taken for the delivery with the H.264 coding and without it and the comparative quality of delivery. The compressed video had given good speed of delivery and also good quality of quality of delivery.

**Index Terms**— H.264 coding, Video Compression, HTTP protocol

## I. Introduction

In the present day the world has changed into the so called “digital age” or “electronic age”, where the mobile phones are called as smart phones because using them we can make not only calls but they are used for online browsing, sending emails, viewing / capturing videos , transfer data, navigation purposes, as camera and with many apps. Digital television sets have become lot more compact with accessibility of regional and international channels with High-Definition(HD) quality. The stored data on re-writable hard disks, DVDs and Blu-ray discs which are light weight, portable with massive space for storage. Internet connection is tremendous fast with wireless routers and modems operating with high speeds [1]. In this fast rising world of communications, data compression is still one of the most valuable and essential components in any multimedia system. The Modern data compression techniques offer the opportunity to store or transmit the huge amounts of data that is necessary to represent digital images and degital videos in an effective and robust way.

Compression is the procedure where redundant information is removed and data is represented with less number of bits than the original information would use. The consumption of expensive resources such as data storage on hard disks/servers and transmission bandwidths can be reduced by using this techinique. To enable real-time data transmission using less resources reasearches are being continued on compression technique.Hence thistechniques are being categorized as lossless or lossy. Lossless compression can be made as the largest part of real-world data are statistically redundant. On the other hand,When the data has been losslessly compressed, the original data can be recovered without any loss. Lossless compression exploits statistical redundancy and represents data with lot more reliability and less error[4]. This compression is advantageous in areas like text compression and audio compression. In Lossy compression data cannot be recovered exactly as it involves some loss of information . lossy compression is used in areas where data distorion is acceptable like video compression, image compression and also in some types of audio compression. In digital cameras,to enhance storage capacity with less degradation of picture quality lossy image compression is applied.Similarly lossy video compression is used on DVDs, Blu-ray disks [38],Internet telephony by means of MPEG-2 [39], H.264 video codec.

Video sequences includesa significant amount of statistical and subjective redundancies within and between frames. crucial goal of a video source coding is reduction in bit rate for storage and tranismission by exploring both statistical (spatial) and subjective (temporal) redundancies and to encode a “minimum set” of information by applying entropy coding tecniques[5].The size of data in multimedia signals is very high for example, for representing 2 minutes of CD-quality music (44,100 samples per second, 16 bits per sample)It requires more than 84 million bits. In case of video signals to represent 1 second of video without compression (using CCIR 601 format) [40], more than 20 Mbytes or 16Mbits is required.[6]The importance of compression for multimedia signals are indicated by this data

This paper takes up the performance analysisbetween video delivery with and without H.264 compression.Matlab based implementation of the H.264 compression and the Java based HTTP implementation is carried out and the results are analyzed for the quality of delivery and the time taken for video delivery.

Section II deals with the H.264 algorithm for video coding. Section III about the HTTP implementation Section IV discusses the results and discussion. Section V concludes the paper about the performance analysis of the methods used.

## I. H.264 video coding algorithm

Depending upon the applications and end-usage there are numerous video compression standards both open source and proprietary. The moving pictures experts group (MPEG) and video coding experts group (VCEG) united together to form joint video team (JVT) in 2001, which developed the ITU-T Rec. H.264 | ISO/IEC 14496-10,usually referred as “H.264” , “MPEG-4 Part 10” and “Advanced Video Coding (AVC) ” published by the International Telecommunication Union (ITU) and the International Standards Organisation (ISO) [15]. .

This approach left the room for innovating in encoding algorithm development.As a result of effort in this thesis,A low complexity mode selection encoding algorithm is being focused .

As H.264/AVC has very efficient compression methods,it compress video a lot more efficiently than older standards and provide more flexibility for application to a large variety of network enviornments. To accomplish highly efficient compression, the computational cost linked with it is also very high. Due to this reason,these enlarged compression effeciencies cannot be exploited across all application domains. Resource constrained devices like cell phones and other embedded systems use simple encoders or simpler profiles of the codec to substitute compression efficieny and quality for reduced complexity [6]. Decoding process and bitstream syntax of the compressed video are being specified by the video coding standards .Encoding process orprocess of producing a standard complaint video is not specified.

In Intra (spatial) prediction compression uses only current frame for prediction.Any movement from the neighboring blocks is being predicted by it. It reuses the mode information from adjacent blocks. Intra frame prediction is frequently used in uniform zones of the picture where there is a small amount of movement.

To remove redundancy a reference frame is used,which is strictly coded with raw pixel values so has full information stored. A reference frame is also well-known as a I-frame, it can be either a past or a future frame from the video sequence. To find a similar block from the reference frame to the one it is encoding a block matching algorithm is used. probably, the encoder will find such a block, but it may not be the perfect match and can introduce prediction error,hence algorithm takes difference of the reference frame block and the block it is encoding and calculates the prediction error. The algorithm searches for another reference frame block with matching characteristics and calculates prediction error if the prediction error is above certain threshold value. As and when a matching block with minimum prediction error is found, the encoder only transmits a vector, called as motion vector , which containsco-ordinates that summit to the block from the reference frame.Finally,the encoder takes transform of the difference between reference and predicted frames, calculates the transform coefficients followed by entropy coding which would be adequate for reproducing the video frame at the decoder. A GOP (group of pictures) always begins with an I-frame. It contains full information hence any errors within the GOP structure can be corrected using the I-frame. B-frames are first and foremostly used for compression efficiency but they also propogate errors in H.264. P-frames includes motion compensated difference from the previous I-frame or P-frame. Distance between two I-frames can be expressed as GOP length.

Key features that make H.264/AVC a highly efficient codec are :

- Variable block size motion compensation with block sizes from 16x16 to 4x4, enabling particular segementation of moving regions.
- Sharper subpixel motion compensation is derived from Six tap filtering .Where as Quarter-pixel motion is derived from linear interpolation.
- ,Allowing encoder to specify the scaling and offset through Weighted prediction .
- Lossless macroblock coding
- An in-loop deblocking filter
- Loss resilence features like network abstraction layer (NAL), flexible macroblock ordering (FMO) , redundant slices (RS) and data partitioning (DP)
- An entropy coding design including context adaptive binary arithmetic coding (CABAC) , context adaptive variable length coding (CAVLC) and variable length coding (VLC)
- Switching slices like SI and SP slices.

H.264 Encoder:

The working mechanism of H.264 encoder consists of same principles as that of any other codec. Figure 2.6 shows the basic building blocks of H.264 video codec.

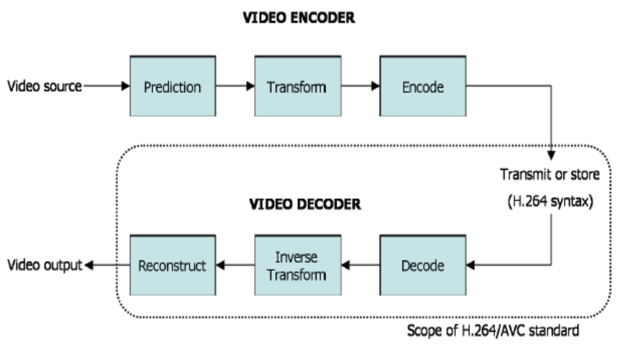

Figure 2.6 H.264 video coding and decoding process [4].

Input to the encoder in general is an intermediate format stream, which goes all the way through the prediction block; the prediction block will execute intra and inter prediction (motion estimation and compensation) and exploit the redundancies that are being present within the frame and between successive frames. The output given by the prediction block is then transformed and quantized. An integer fairly accurate to the discrete cosine transform(DCT) is used for transformation [12]. It uses 4x4 or 8x8 integer transform, and the output is given as a set of coefficients each of which is a weighted value for a standard basis pattern. Those coefficients are then quantized i.e. each coefficient is divided with an integer value.The precision of the transform coefficients according to the quantization parameter (QP)can be reduced through quantization. Naturally, the result is a block in which nearly all or all the coefficients are zero, with a few non-zero coefficients. After that, the coefficients are encoded into a bit stream.A number of parameters that must be encoded to form a compressed bit stream is being created by the video coding process [13]. These values include:

- Quantized transform coefficients.
- Information to re-create prediction.
- Information about the structure of compressed data and the compression tools used under encoding.

Variable length coding or arithmetic coding are used to convert this parameters into binary codes. Each and every encoding method produces an efficient, compact binary representation of information. The encoded bit stream can now be transmitted or stored.

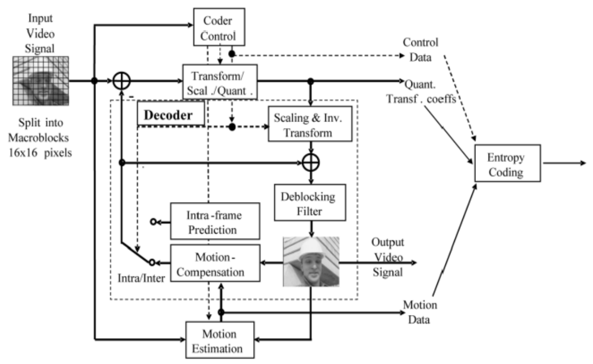

Figure 2.7 Basic coding structure of H.264/AVC for a macroblock [9]

H.264 Decoder:

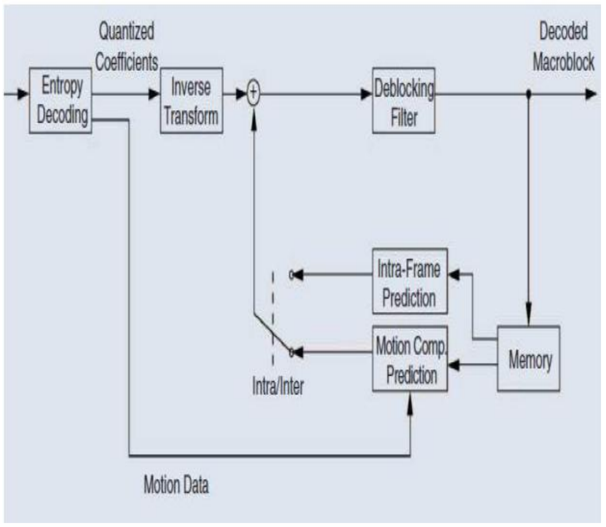

Figure 2.17 Basic coding structure H.264/AVC video decoder [4].

The process of Decoding is exactly opposite to the encoding process. A video decoder receives the compressed H.264 bitstream then decodes the syntax elements and extracts information such as quantized transform coefficients, prediction information etc. To recreate the video sequence this data is used The quantization parameter multiplies the quantized transform coefficients.Quantization parameter is an integer value. once the transform coefficients are rescaled, inverse transform combines the standard basis pattern,which is weighed by the rescaled coefficients, to re-establish each and every block of residual data. To form the residual data macroblock these blocks are combined together. For each macroblock, prediction identical is performed by the decoder to the one created by the encoder. Then this is added to the decoded residual data to rebuild a decoded macroblock which can be displayed as part of a video frame subsequently.

## II. HTTP Implementation

H.264/AVC supports a ample range of prediction options – intra prediction can be done using data within the current frame, inter prediction can be done using motion compensated prediction from formerly coded frames, multiple prediction block sizes, multiple reference frames. An I Macro block (I MB) could be predicted with intra prediction from neighboring samples within the current frame.Samples in a previously-coded frame that are prior to the current picture in display ordercan be predicted by using a P Macro block(P MB).Now let's start by creating a class called "SocketClient". We will put this into a package called bdn. are java.net and java.io are the only packages that we need in this example. If youdid not deal with the java.net. package before, as implied by it's name, java.net contains the basic classes and methods you need for network programming.

One of the significant things about java is the regular use of Input and OutputStreams to read and write I/O, despite of the device. In other words, you can almost be secured that An InputStream is used if you are reading from any input source and An OutputStreamis used to write for output source. Itmeans reading and writing through a network is almost as same as reading and writing files.Because of this reason, we should import java.io package into our program.

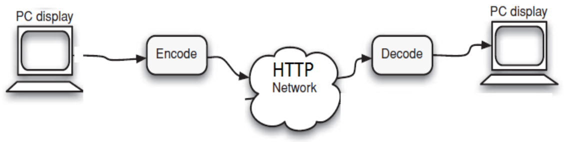

Figure 1: Basic Block diagram

A couple of pieces of information should be known to make a socket connection . First you need a host to connect with. Here in this example we are going to run client(s) and the server on the same machine. A String host is defined as **localhost**.

Note: we had used TCP/IP address **127.0.0.1** as a replacement for **localhost**.

Now we have to know about the TCP/IP port that is going to be communicating with the program . TCP/IP uses ports as it is assumed that servers will be doing over one network function at a time and ports make a way available to maange this. For example: a server might be serving up web pages (port 80), it may have a FTP (port 21) server, and it may be handling a SMTP mail server (port 25).Server assigns ports. It is the responsibility of the client to know what port should be used for a particular service. In the TCP/IP world, every computer with a TCP/IP address has contact with 65,535 ports. It should be noted that ports are not physical devices such as serial, parallel, or USB port. They are an abstraction in computer's memory running under the TCP/IP transport layer protocol.

Note: Ports 1 – 1023 are held in reserve for services such as HTTP, FTP, email, and Telnet.

Lets go back to the code. We will create an int called **port**. The server that is built later in the article will be listening on port 19999. for this result we initialize **port** to 19999.

A couple of other items that we define here are a StringBuffer **instr** and String **TimeStamp**

A StringBuffer **instr** should be used for reading our InputStream. String **TimeStamp** will be used to communicate with the server. finally, System.out.println() a message to make sure that the program has begun…this is not actually necessary , but I found rarely a status message about logging a program gives people a state of peace that a program is actually doing something.

We must first acquire the server's 32-bit IP address to establish a connection with a server. IP address can be obtained by invoking the InetAddress. Get By Name () method. As it is described, we pass this method the name of the host weare looking to connect to. It returns an InetAddress object address that contains the host name/IP address pair (i.e. Using localhost in the getByName() method will return localhost/127.0.0.1 in the InetAddress object).

Once the InetAddress object is obtained, a socket connection with our server can readily be established.a Socket called **connection** can be createdby instantiating a new Socket object with InetAdress object address and with our earlier created int **port**. As and when the server is not responding on the port we are searching for, we'll get a "java.netConnectException: Connection refused:.." message.

Finally connection is established. Now some information should be written to the server. As mentioned earlier,Java treats reading and writing sockets like reading and writing files. afterwards we start by establishing an OutputStream object. In general buffer will be used by TCP stacks to improve performance within the network. And, although it's not required necessary, BufferedInputStreams and BufferedOutputStreams can be used while reading and writing data across the network. We instantiate a BufferedOutputStream object **bos** by requesting an OutputStream from our socket connection.

BufferedOutputStream.write() method should be used to write bytes across the socket. I preferably use OutputStreamWriter objects to write on because I am usually dealing in multiple platforms and would like to control the character encoding.

## III. SIMULATION, RESULTS and discussions

Matlab implementation of the H.264 algorithm was carried out and the result of the encoded video was sent through the HTTP server to the client and the at the client side the encoded bit stream are received and the bit stream are decoded using the H.264 decoder .

The same process is carried out without encoder and decoder.It was found that the speed of delivery was better in the H.264 encoded and decoded video delivery and also the video quality was maintained to nearly the same as when not compressed

The tabular column would infer that the H.264 algorithm be introduced for the HTTP based video delivery. Because the time taken for the video delivery without the H.264 is higher compared to that with H.264 and the PSNR is in the acceptable range it could be suggested that even for the hyperspectral videos this method would ne successful.

| Parameters | With H.264 | Without H.264 |
|---|---|---|
| Time Taken | .5 secs | 5 secs |
| PSNR | High | High |

Table 1.Results with and without H.264 coding

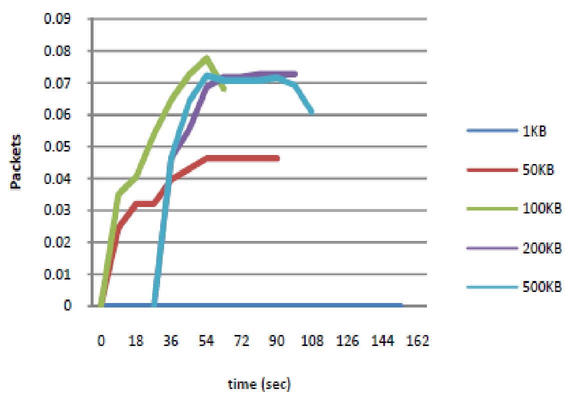

Figure 2.8. Average retransmission of packets when there is no interference

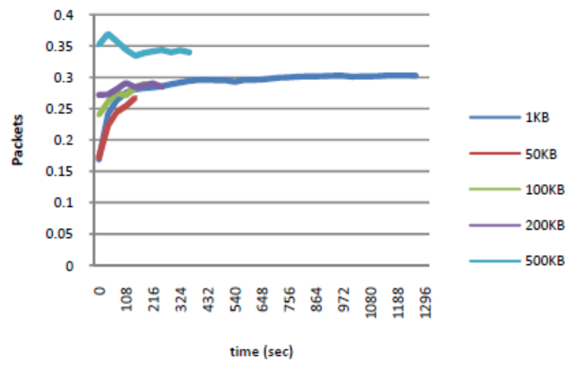

Figure 2.9. Average retransmission of packets when interference is introduced

## IV. Conclusion

The HTTP protocol was implemented with the H.264 and without it and the performance was carried out for sending the video through the protocol. The compressed video had given good speed of delivery and also good quality of quality of delivery.This would hold good for even the hyperspectral videos.

## References

[1.] I.E.Richardson,"TheH.264 advanced video compression standard",2nd Edition ,Wiley 2010.
[2.] Open source article,"Blu-ray discs",http://www.blu-ray.com/info/
[3.] Open source article," Coding of moving pictures and audio" http://mpeg.chiariglione.org/standards/mpeg-2/mpeg-2.htm
[4.] Open source article,"Blu-ray discs",http://www.blu-ray.com/info/
[5.] Open source article," Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios", http://www.itu.int/rec/R-REC-BT.601/
[6.] Open source article," Coding of moving pictures and audio" http://mpeg.chiariglione.org/standards/mpeg-2/mpeg-2.htm
[7.] JM reference software http://iphome.hhi.de/suehring/tml/
[8.] JM reference software http://iphome.hhi.de/suehring/tml/
[9.] T. Wiegand and G. J. Sullivan, "The H.264 video coding standard", IEEE Signal Processing Magazine, vol. 24, pp. 148-153, March 2007.
[10.] S. Kwon, A. Tamhankar and K.R. Rao, "Overview of H.264 / MPEG-4 Part 10", J. Visual Communication and Image Representation, vol. 17, pp.186-216, April 2006.