A COMPARATIVE ANALYSIS OF H.264 VIDEOCOMPRESSION STANDARD USING DIFFERENT ENTROPY CODING METHODS

Logashanmugam .E[1] and Ramachandran .R[2]
[1]Research Scholar,Sathyabama University ,Rajiv Gandi Road, Tamilnadu, India.
[2]Sri Venkateshwara Engineering College, Chennai, India
E-mail : [1]logu999@yahoo.com

Abstract

In this paper, we present a comparative performance analysis of low power hardware architecture for real time implementation of context adaptive binary arithmetic coding algorithm and Context-based adaptive Variable Length Coding algorithm (CAVLC) used in H.264/ MPEG part 10 video coding standards. The video compression efficiency achieved in H.264 standard is not a result of any single feature but rather a combination of a number of encoding tools. CAVLC requires considerably more processing power to decode. The encoding process is simulated by MATLAB which compress the input video sequence and ensures better performance in video compression. The results that are obtained reveal better performance of both entropy coding methods.

Key words: H.264, CABAC, CAVLC, Huffman coding, Arithmetic coding.

I. INTRODUCTION

The increasing demand to incorporate video data into telecommunication series, the corporate environment, the entertainment industry and even at home have made digital video technology as an important one. A problem however is that still image and digital video data rates are very large typically in the range of 150Mbits/sec. Data rates of this magnitude would consume a lot of bandwidth, storage and computing resources in the typical personal computer. For this reason Video compression standards have been developed to eliminate picture redundancy, allowing video information to be transmitted and stored in a compact and efficient manner. Uncompressed multimedia (graphics, audio and video) data requires considerable storage capacity and transmission bandwidth. Despite rapid progress in mass-storage density, processor speeds, and digital communication system performance, demand for data storage capacity and data-transmission bandwidth continues to outstrip the capabilities of available technologies. The recent growth of data intensive multimedia-based web applications have not only sustained the need for more efficient ways to encode signals and images but have made compression of such signals to efficient storage and communication.

A new International standard for video compression is developed to improve the performance of the existing applications and to enable the applicability of video compression to new real-time applications. This new standard is developed with the collaboration of ITU and ISO standardization organizations, offering significantly better video compression efficiency than previous International Standards and it is called in two different names namely H.264 and MPEG4 Part 10.

The objectives of this work are as follows. One objective has been to compile an introduction to the subject of video compression techniques. There exist a number of studies of various parts of the encoder and decoder, but complete treatments on a technical level are not common. Another objective has been to search for algorithms that can be used to implement the most demanding components of a Video compression standard. They include, Discrete Wavelet Transform (DWT) & Inverse Discrete Wavelet transform (IDWT), Scaling and Quantization, Entropy encoding & decoding. A third objective is to evaluate their performance with regard to PSNR, memory requirements and complexity. These properties were chosen because they have the greatest impact on the implementation effort and the computation demands. The final objective is to simulate MPEG/H.264 encoder.

II. OVERVIEW OF H.264

H.264 is a standard for video compression. It is also known as MPEG-4 Part 10, or MPEG-4 AVC (for Advanced Video Coding). It was written by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC Moving Picture Experts Group (MPEG) as the product of a partnership effort known as the Joint Video Team (JVT). The ITU-T H.264 standard and the ISO/IEC MPEG-4 Part 10 standard are jointly maintained so that they have identical technical content. Video compression systems are used in many commercial products, from consumer electronic devices such as digital camcorders, cellular phones to video teleconferencing systems. These applications make the video compression hardware devices an inevitable part of many commercial products.

This new standard, offering significantly better video compression efficiency than previous video compression standards, is developed with the collaboration of ITU and ISO standardization organizations. Hence it is called with two different names, H.264 and MPEG4 Part 10. H.264 video coding standard has a much higher coding efficiency potential (capable of saving up to 50% of bit rate at the same level of video quality) than the previous standards.

Due to its high coding efficiency, flexibility and robustness to different communication environments, in the near future, H.264 is expected to be widely used in many applications such as digital TV, DVD, video transmission in wireless networks and video conferencing over the internet. The video compression efficiency achieved in H.264 standard is not a result of any single feature but rather a combination of a number of encoding tools. The top-level block diagram of an H.264 Encoder is shown in Fig 1.

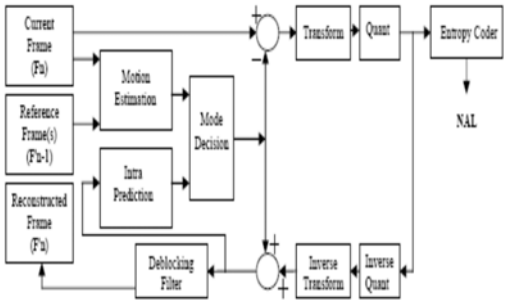


Fig: 1. H.264 CODEC Block Diagram

The H.264 standard includes a Video Coding Layer (VCL), which efficiently represents the video content, and a Network Abstraction Layer (NAL), which formats the VCL representation of the video and provides header information in a manner suitable for transportation by particular transport layers or storage media [1].

As shown in Fig 2, an H.264 encoder has a forward path and a reconstruction path. The forward path is used to encode a video frame by using intra and inter predictions and to create the bit stream. The reconstruction path is used to decode the encoded frame and to reconstruct the decoded frame. Since a decoder never gets original images, but rather works on the decoded frames. Reconstruction path in the encoder ensures that both encoder and decoder uses identical reference frames for intra and inter predictions. This avoids possible encoder – decoder mismatches [1, 3, and 4].

In the forward path the input frames are partitioned in to macro block (MB). Each MB is encoded into intra or inter mode depending on the mode decision. In both intra and inter modes, the current MB is predicted from the reconstructed frame. Intra mode generates the predicted MB based on spatial redundancy, whereas inter mode, generates the predicted MB based on temporal redundancy. Mode decision compares the required amount of bits to encode a MB and the quality of the decoded MB for both of these modes and chooses the mode with better quality and bit-rate performance. In either case, intra or inter mode, the predicted MB is subtracted from the current MB to generate the residual MB.

Residual MB is transformed using 4x4 and 2x2 integer transforms. Transformed residual data is quantized and quantized transform coefficients are re-ordered in a zigzag scan order. The reordered and quantized transform coefficients are entropy encoded. The entropy-encoded coefficients together with header information, such as MB prediction mode and quantization step size, form the compressed bit stream. The compressed bit stream is passed to network abstraction layer (NAL) for storage or transmission [1,3,4].

Reconstruction path begins with inverse quantization and inverse transform operations. The quantized transform coefficients are inverse quantized and inverse transformed to generate the reconstructed residual data. Since quantization is a lossy process, inverse quantized and inverse transformed data are not identical to the original residual data. The reconstructed residual data are added to the predicted pixels in order to create the reconstructed frame. A deblocking filter is applied to reduce the effects of blocking artifacts in the reconstructed frame. Another application area for H.264 intra frame coder is in motion picture production, editing and archiving, where video frames are coded as I-frames in order to allow random access for each individual picture. For such applications, H.264 is shown to be superior at lower resolutions. The wavelet transform (WT) has gained widespread acceptance in signal processing and image compression. Recently the JPEG committee has released its new image coding standard, JPEG-2000, which has been based upon DWT. Wavelet transform, decomposes a signal into a set of basis functions. These basis functions are called wavelets. Wavelets are obtained from a single prototype wavelet y(t) called mother wavelet by dilations and shifting. The discrete wavelet transform has a huge number of applications in science, engineering, mathematics and computer science. Most notably, it is used for signal coding, to represent a discrete signal in a more redundant form, often as a preconditioning unit for data compression.

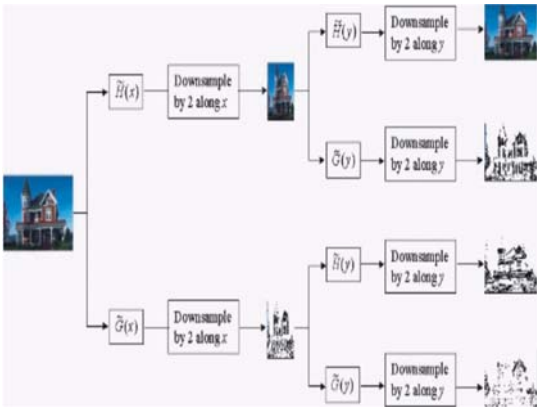


Fig: 2. Discrete Wavelet Transform

The function of the coder is to transmit the DWT coded data to the decoder, in a bit rate efficient manner, so that it can perform the inverse transform to reconstruct the image. It has been observed that the numerical precision of the DWT coefficients may be reduced while still maintaining good image quality at the decoder. Quantization is used to reduce the psycho visual redundancy in turn it reduces the required number of bits to be transmitted.

The degree of quantization applied to each coefficient is weighted according to the visibility of the resulting quantization noise to a human observer. In practice, this results in the high-frequency coefficients being more coarsely quantized than the low-frequency coefficients. Note that the quantization noise introduced by the coder is not reversible in the decoder, during the decoding process. For a typical block of pixels, most of the coefficients produced by the DCT are close to zero. The quantizer module reduces the precision of each coefficient so that the near-zero coefficients are set to zero and only a few significant non-zero coefficients are left. This is done in practice by dividing each coefficient by an integer scale factor and truncating the result. It is important to realize that, the quantizer "throws away" information.

The basic idea of Vector Quantization based image compression is to divide the image into blocks. Typically, some blocks (hopefully many) are similar to other blocks although usually not identical. The encoder identifies a class of similar blocks and replaces these with a "generic" block representative of the class of similar blocks. The encoder encodes a lookup table that maps short binary codes to the "generic" blocks. Typically, the shortest binary codes represent the most common classes of blocks in the image.

The Vector Quantization (VQ) decoder uses the lookup table to assemble an approximate image comprised of the "generic" blocks in the lookup table. Note that this is inherently a lossy compression process because, the actual blocks are replaced with a generic block that is a "good enough" approximation to the original block.

The encoding process is slow and computationally intensive because, the encoder must accumulate statistics on the frequency of blocks and calculate the similarity of blocks in order to build the lookup table. The decoding process is very quick because, it is lookup table based. In Vector Quantization, the lookup table may be called a codebook. The binary codes that index into the table may be called code words.

Vector Quantization is prone to blocking artifacts as compression is increased. Vector Quantization is an entire sub-field in signal and image processing. It goes well beyond the brief description above and is applied to other uses than video compression. Quantization is the lossy component of video compression. After the DCT operation, quantization is applied to the resultant frequency matrix. Since human vision is less sensitive to color than to brightness, the quantization of the color values can be greater than that of the intensity values. This provides a higher degree of compression for the color components, without causing much perceptible image degradation. Similarly, given that human vision is less sensitive to high frequency details than to low frequency details, the quantization of high frequency values can be greater, gaining extra compression without sacrificing perceived image quality. Because it compresses the overall range of the data, a quantization operation on the frequency matrix will output an altered matrix which contains more frequent and longer runs of the same value. In particular, a quantized matrix typically contains many more zeros. The quantized matrix can be inverted to approximate the original full-range frequency matrix. However, the finer granulations between values that were eliminated have been permanently lost.

Inverse Quantization is an algorithm that allows an image to be displayed with a limited set of colors that associate each color with its nearest representative. The image is represented in compressed form by mathematical components.

III. CABAC AND CAVLC

The algorithm of context-based adaptive binary arithmetic coding (CABAC) has been developed within the joint standardization activities of ITU-T and ISO/IEC for the design and specification of H.264/AVC as the latest member in the family of International video coding standards. CABAC is a form of entropy coding used in H.264/MPEG-4 AVC video encoding. As such it is an inherently lossless compression technique. It is notable for

providing considerably better compression than most other encoding algorithms used in video encoding and is considered one of the primary advantages of the H.264/AVC encoding scheme.

CABAC is supported in main and higher profiles and requires a considerable amount of processing to decode, when compared to other similar algorithms. As a result, Context-adaptive variable-length coding (CAVLC), a lower efficiency entropy encoding scheme, is sometimes used instead to increase performance on slower playback devices.

A. CABAC ENCODING

These elements are illustrated as the main algorithmic building blocks of the CABAC encoding block diagram.

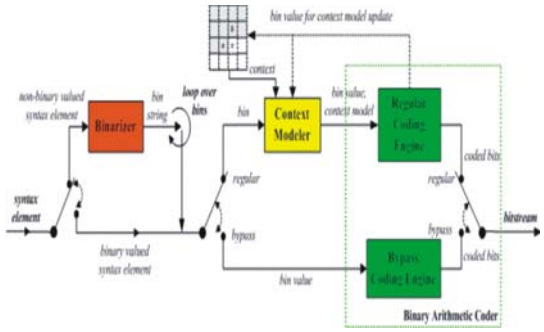


Fig :3. CABAC encoding block diagram

After a block has been predicted and DWT transformed, the resulting prediction modes and DWT coefficients must be stored in a stream. To minimize the number of bits needed to store these values, the Context Adaptive Binary Arithmetic Coding (CABAC) algorithms from the H.264 standard is used.

CABAC has multiple probability modes for different contexts. It first converts all non-binary symbols to binary. Then, for each bit, the coder selects the appropriate probability model to be used, then uses information from nearby elements to optimize the probability estimate. Arithmetic coding is then applied to compress the data.

B. Binarization

Binarization is a form of pre-processing stage (before coding) that operates on syntax elements and generates a unique intermediate binary codeword for a given syntax element. This intermediate codeword is called bin string and each binary value of it is called as a bin. This stage effectively reduces the alphabet size of the syntax element and allows more efficient operation of context modeling stage. There are four main binarization scheme employed in this stage: unary (U) binarizer, truncated unary (TU) binarizer, k-th order exp-Golomb (EGk) Binarizer and fixed-length (FL) binarizer. Each one of them applies a different mathematical transformation to the syntax element. Also, two combinations of these binarization schemes are used for some syntax elements.

C. Context Modelling

Context modeler is the heart of context-adaptive capability of CABAC that differentiate it from other entropy coding techniques. It assigns a model probability distribution to given symbols which are used for generating the code at the subsequent coding stage. This model determines the code itself and controls the efficiency of the coding. It consists of a table of 399entries in which each consists of a 6-bit probability value and a 1-bit MPS (Most Probable Symbol). The table is accessed and updated by binary arithmetic coding stage; it is initialized with some predefined values (3 variations of initial table exist that depending on encoding parameters, one is selected for an encoding scenario) at the beginning of each slice.

D. Binary Arithmetic Coder

Binary arithmetic coder is another differentiating feature of CABAC. It is based on recursive interval subdivision. The interval and its location are tracked at any time by two integer values. Based on the statistical property of the symbol being coded, the interval is divided in to two regions proportional to probability of LPS and MPS. Updation of this interval produces 0 or more output bits to be appended to the output stream. A context-model entry (associated with the symbol) provides the statistics of the symbol being coded. The 6-bit of context entry is the probability estimate of the Least Probable Symbol (LPS) while the 1-bit of the entry shows the polarity of MPS .

Since the sub-interval size is reduced after each coding, a renormalization operation rescales the interval range and location to proper range. Actually this renormalization process generates the output bit as part of its rescaling process.

E. ARITHMETIC CODING

Arithmetic coding is a method for lossless data compression. Normally, a string of characters such as the words "hello here" is represented using a fixed number of bits per character, as in the ASCII code. Like Huffman coding, arithmetic coding is a form of variable-length entropy encoding that converts a string into another representation that represents frequently used characters using fewer bits and infrequently used characters using more bits, with the goal of using fewer bits in total. As opposed to other entropy encoding techniques that separate the input message into its component symbols and replace each symbol with a code word, arithmetic

coding encodes the entire message into a single number, a fraction n, where n is defined as (0.0 ⩽n < 1.0).

F. Context Based Adaptive Variable-length Coding - Cavlc

When the CODEC is in this mode, the residual data is coded using CAVLC but other data are coded using simple Exp-Golomb codes. These data are first appropriately mapped to the Exp-Golomb codes depending on the data type (e.g. MB headers, MVs, etc.), and then the corresponding code words are transmitted.

The zigzag scanned quantized coefficients of a residual block are coded using Context Adapting VLC tables. The already coded information of the neighboring blocks (i.e. upper and left blocks) and the coding status of the current block determine the context. Optimized VLC tables are specifically provided for each context to efficiently code the coefficients in different statistical conditions.

Context-based adaptive variable-length coding (CAVLC) is a new and important feature of the latest video coding standard, H.264/AVC. The VLSI implementation of CAVLC modified from the conventional run-length coding architecture will lead to low throughput and utilization. In this paper, an efficient CAVLC design is proposed. The main concept is the two-stage block pipelining scheme for parallel processing. When one block is processed by the scanning engine to collect the required symbols, its previous block is handled by the coding engine to translate symbols into bit stream. Our dual-block-pipelined architecture doubles the throughput and utilization of CAVLC at high bit rates. Moreover, a zero skipping technique is adopted to reduce up to 90% of cycles at low bit rates. Last but not least, Exp-Golomb coding for other general symbols and bit stream encapsulation for the network abstraction layer are integrated with CAVLC as a complete H.264/AVC baseline profile entropy coder.

In H.264, the Context-based adaptive variable-length coding (CAVLC) is used for lossless compression. Direct table-lookup implementation requires higher cost because it employs a larger memory to produce the encoded results. In this paper, we present a more efficient technique for CAVLC implementation. Compared with those previous CAVLC chips, our design requires the lowest hardware cost.

Low power architecture for realizing the CAVLC decoder is proposed. In traditional VLC decoding algorithms, we have to search a level in Huffman coding tree per operation. Therefore, the throughput rate is limited by the searching level. The CAVLC algorithm takes the advantage of the trend among AC coefficients in each block to predict the next codeword. The prediction mechanism can significantly improve the decoding efficiency. Hence, we suggest two efficient approaches, table partitioning and prefix pre decoding, to reduce the power consumption in decoding the VLC codes. The proposed low-power CAVLC architecture achieves the real-time requirement for 720p HD (1280/spl times/720) format, while the clock is operated at 125 MHz. CAVLC algorithm is used to encode transformed and quantized residual luminance and chrominance blocks in a macro block in the order.

The block 1 is formed by the DC coefficients of 4×4 luminance blocks only for the macro blocks that are coded in 16×16 intra mode. Blocks 16 and 17 are formed by the DC coefficients of 4×4 chrominance blocks for all the macro blocks. All the transformed and quantized 4×4 and 16×16 for the entire macro block are given as inputs to CAVLC algorithm.

CAVLC algorithm processes each 4×4blocks in zigzag scan order and each 4×4block in raster scan order.

IV. SIMULATION RESULTS

The simulations are done in MATLAB.Figure 4 shows the Video Sequence from the File. Fig 5. shows the Conversion of video sequencein to frames. Fig 6 shows the result of Image Encoding . Fig 7 shows the results of Compression using arithmetic encoder

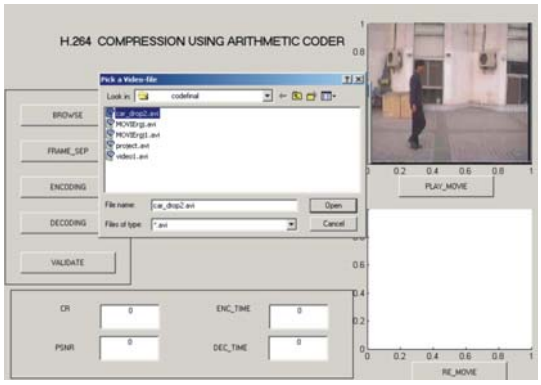


Fig: 4. Obtaining Video Sequence from the File

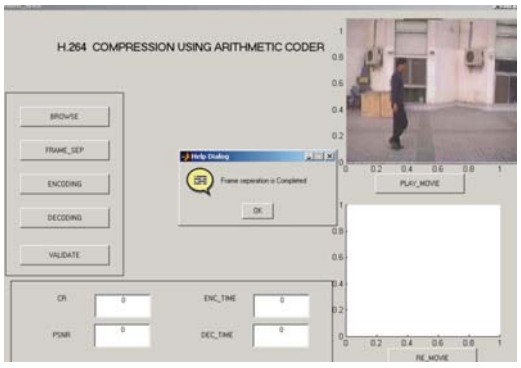


Fig: 5. Conversion of Video Sequence in to Frames

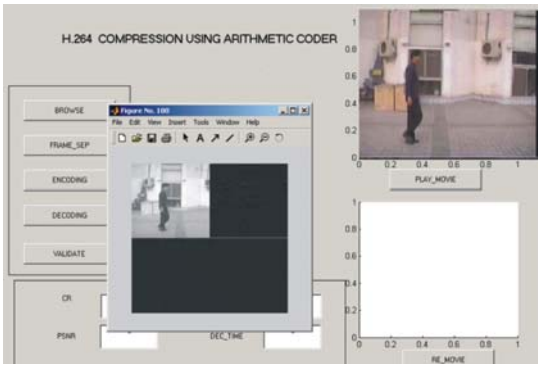


Fig: 6. Results of Image Encoding

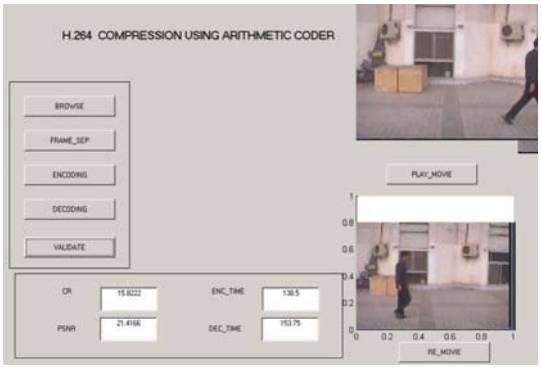


Fig: 7. Compression Using Arithmetic Encoder

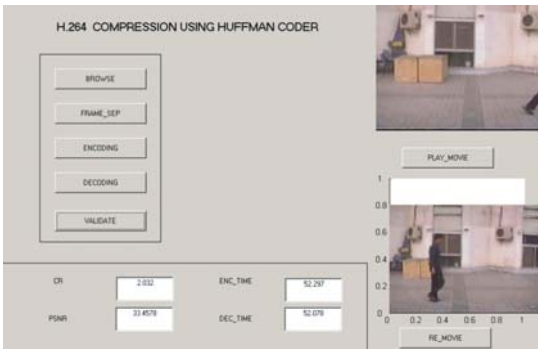


Fig: 8. Compression Using Huffman Encoder

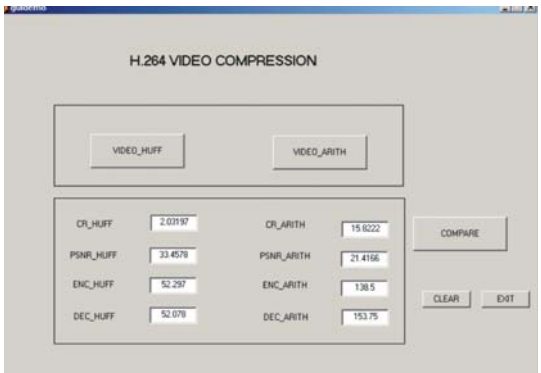


Fig: 9. Comparison of Huffman and Arithmetic Encoder

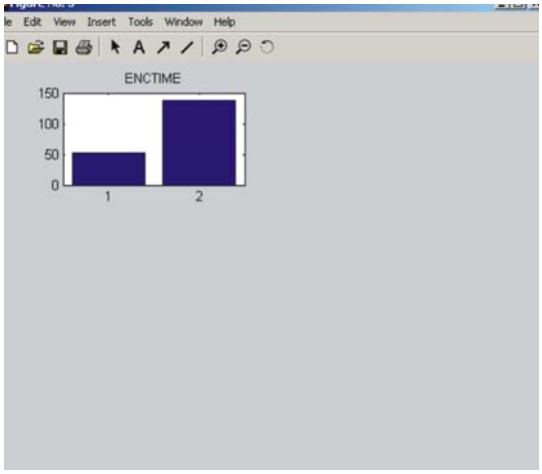


Fig: 10.Comparison of Huffman and Arithmetic Encoding Techniques Based on Their Processing Speed

The compression ratio (CR) and signal to noise ratio (PSNR) of the chosen file using Huffman encoder is displayed in the above Fig 8. Fig 9 shows the Comparison of Huffman and Arithmetic encoder.

Fig 10 shows the Comparison of Huffman and Arithmetic encoding techniques based on their processing speed. Fig 11 shows the Comparison of Huffman and Arithmetic decoding time. Fig 12 shows the Comparison of Huffman and Arithmetic coding based Compression Ratio and PSNR.

The Huffman and Arithmetic coding based encoding and decoding techniques are compared in the figures 10 and 11 in terms of their processing speed

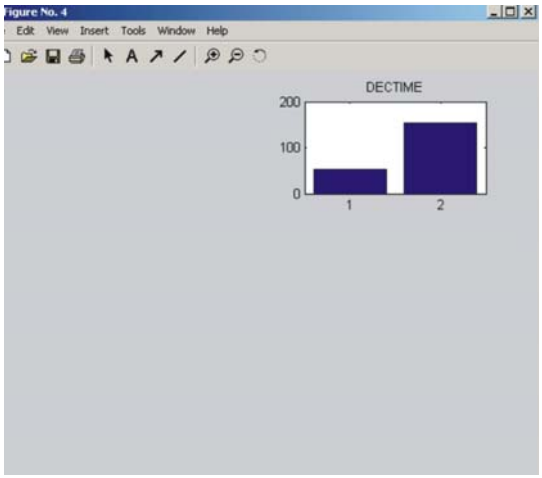


Fig: 11. Comparison of Huffman and Arithmetic Decoding Time

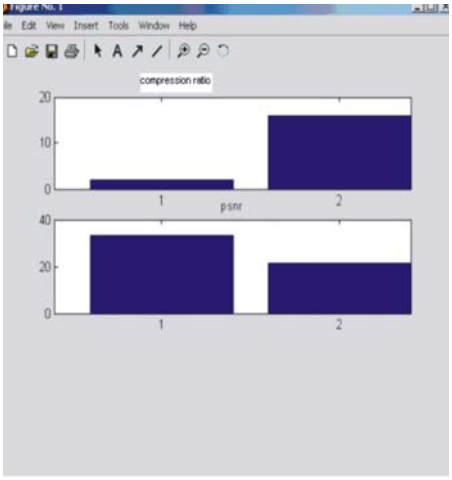


Fig 12. Comparison of Huffman and Arithmetic Coding Based Compression Ratio and PSNR

V. CONCLUSION

This paper focuses on developing H.264 approach for efficient compression of gray scale images. The encoded file size is much smaller than the original file size due to resizing. The H.264 decoding reconstructs the matrix very close to the original matrix that was given as input to the H.264 encoding. The peak signal to noise ratio has been obtained as 33.1 dB and the compression factor is about 15:1. The data rate is 0.4 bits per pixel.

As a conclusion, H.264/AVC is introduced with significant enhancement both in compression efficiency and error resilience. Compared with former video coding standards such as MPEG2 and MPEG4 part 2, it saves more than 40% in bit rate and provides important characteristics such as error resilience, stream switching, fast forward/backward etc. It is believed to be the most competitive video coding standard in this new era. However, the improvement in performance also introduces increase in computational complexity, which requires higher speed both in hardware and software.

The comparative analysis of H.264 video compression standard for different entropy coding methods like CABAC and CAVLC is quite evident from the figures 9,10,11and 12.

A. Future Scope

There will be possible improvement of the existing H. 264 spatial scalable coding by adding a low pass filter after images are up sampled from base layer. This can improve the video quality by decreasing the block effect. The FPGA-based H.264 intra frame coder implementation can be modified as an ASIC implementation and prototypes can be fabricated.

A complete H.264 video encoder hardware can be implemented by integrating motion estimation, motion compensation, de-blocking filter, intra vs. inter mode decision and rate control modules to the H.264 intra frame coder hardware.

REFERENCES

[1] R. Schäfer, T. Wiegand and H. Schwarz, January 2003. "The Emerging H.264/AVC Standard",*EBU Technical Review.*

[2] Personal Communication with Hasan Ateş.

[3] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra July 2003. "Overview of the H.264/AVC Video Coding Standard", *IEEE Trans. on Circuits and Systems forVideo Technology vol*. 13, no. 7, pp. 560–576.

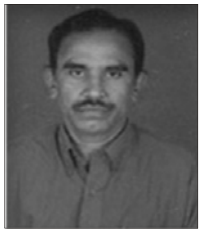
[4] I. Richardson, 2003. "H.264 and MPEG-4 Video Compression*",* Wiley.

[5] May 2003. Joint Video Team (JVT) of ITU-T VCEG and ISO/IEC MPEG, Draft ITU-T."Recommendation and Final Draft International Standard of Joint VideoSpecification", ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC.

[6] O.Tasdizen, I. Hamzaoglu, September 2005. "A High Performance and Low Cost Hardware Architecture for H.264 Transform and Quantization Algorithms", 13th European Signal Processing Conference.

[7] Digital Image processing GONAZLEZ

Prof.E.Logashanmugam received Masters in Electronics engineering from Anna University, Chennai, India. Currently he is pursuing Ph.D in Sathyabama University , Chennai, India. He is working as Head of Department of Electronics and Communication Engineering in Sathyabama University. His field of interest is Image processing and VLSI Design .He is a member of IEEE and ISTE.



Dr.R.Ramachandran received Ph.D from Anna University, Chennai, India. He is working as Principal in Sri Venkateswara College of Engineering , Chennai, India. His field of interest is Image processing and VLSI Design .He is a member of IEEE and Life fellow of IETE.