INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2020, VOL. 66, NO. 3, PP. 571-582
Manuscript received June 15, 2019; revised July, 2020. DOI: 10.24425/ijet.2020.134014

Efficient H.264 Intra Frame CODEC with Best Prediction Matrix Mode Algorithm

Sara Hamdy, Abdelhalim Zekry, and Wael A. Mohamed

***Abstract*—The continuous growth of smart communities and ever-increasing demand of sending or storing videos, have led to consumption of huge amount of data. The video compression techniques are solving this emerging challenge. However, H.264 standard can be considered most notable, and it has proven to meet problematic requirements. The authors present (BPMM) as a novel efficient Intra prediction scheme. We can say that the creation of our proposed technique was in a phased manner; it's emerged as a proposal and achieved impressive results in the performance parameters as compression ratios, bit rates, and PSNR. Then in the second stage, we solved the challenges of overcoming the obstacle of encoding bits overhead. In this research, we try to address the final phase of the (BPMM) codec and to introduce our approach in a global manner through realization of decoding mechanism. For evaluation of our scheme, we utilized VHDL as a platform. Final results have proven our success to pass bottleneck of this phase, since the decoded videos have the same PSNR that our encoder tells us, while preserving steady compression ratio treating the overhead. We aspire our BPMM algorithm will be adopted as reference design of H.264 in the ITU.**

***Keywords*—Best Prediction Matrix Mode (BPMM), Compression efficiency, H.264/AVC, Huffman coding, Intra Prediction**

I. INTRODUCTION

THE demand for video communication services in very various scenarios is becoming a challenging matter, in particular, those associated with the Internet and Smart Communities and many applications under their umbrella such as video conferencing, broadcasting, video telephony, video surveillance, and digital camcorders [1]. One of the most important video coding standards, video coding standard called H.264/AVC. This standard remarkably has better coding efficiency than its predecessors, and has attracted considerable interest from the academic and developer communities [2,3].

Fig. 1. illustrates the whole system architecture of the platform utilized for this article and data flow of the proposed H.264 decoder. Initially, a circular buffer receives the compressed H.264 bitstream from RAM after resetting the system.

A circular buffer plays an important role in fetching bitstream to the decoder to extract the information that used to the decoding process and create a sequence of video [4,5].

The decoder has two critical paths of the baseline architecture, bitstream parser, and reconstruction data path.

S. Hamdy, and W. A. Mohamed are now with Electrical Engineering Department, Benha Faculty of Engineering, Benha University, Egypt (e-mail: sara.hamdy@bhit.bu.edu.eg, wael.ahmed@bhit.bu.edu.eg).

A. Zekry was with Electronics and Communications Department, Faculty of Engineering, Ain Shams University, Egypt (e-mail: aaazekry@hotmail.com).

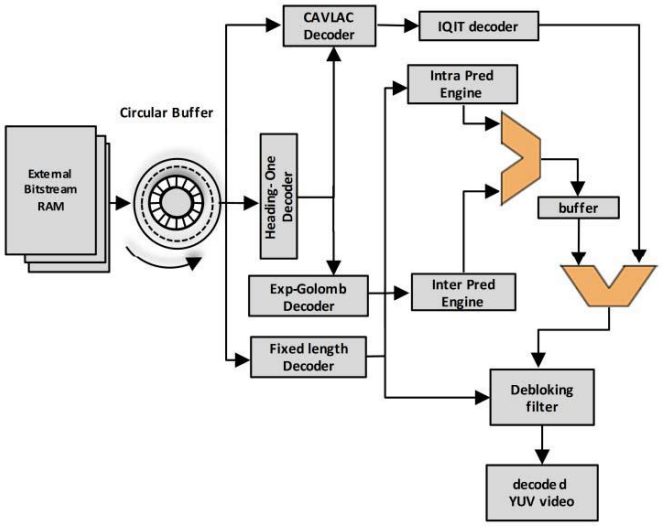

Fig. 1. The hardware architecture of the H.264 Decoder

The key components of the bitstream parser path include Heading One Detector, context adaptive variable length codes (CAVLC) decoder, Exponential Golomb decoder, and Fixed Length decoder, where they are all in charge of reverse coded syntax elements and then extracts quantized transform coefficients and prediction information. The reconstruction data path consists of Intra Prediction, Inter Prediction, and deblocking Filter[6].

The current macroblock type is regarded as the main factor in invoking either Intra or Inter prediction block. Finally, the output of intra/inter block is fetched to the deblocking filter. The main objective of this path is the creation of prediction like the one generated by the encoder for each macroblock, based on prior decoded frames in the event of Inter prediction or prior decoded pixel in the current frame in a state of Intra prediction [7].

Concurrently, the reconstructed macroblock displayed as part of the video frame when it's created from the adding of prediction macroblock with a decoded residual [8].

Fairly recently, the authors proposed the architectural framework for Best Prediction Matrix Mode (BPMM) [9]. This mode of prediction has relied on an established fact that each of DC, vertical and horizontal modes are most used than the others.

More details on (BPMM) are covered in the next section. Our proposed technique had gone through a series of three main phases as follows: phase I the proposal stage in which the BPMM achieved better compression ratios, bit rates, and PSNR depending on improving the Intra prediction scheme of H.264 and reducing the residual information by selecting best values for each pixel [9,10].

Phase II is the challenges and development stage, where we discussed the high overhead bits problems caused by prediction mode signaling and how those obstacles had been overcome.

Finally, phase III which we are discussing now since the realization of our technique lacked the complementary operation, effective decompression or decoding mechanism to recover original video prior to display. That will pose new challenges, so the authors are motivated by offering that goal in this paper.

The rest of paper is organized as follows: We present several recent approaches related to the enhancement of intra frame coding in Section II. Section III gives the outlined of Intra prediction as the main core in H.264 encoder with a brief overview of The Best Prediction Matrix Mode methodology. Section IV discusses Open Problems and Challenges in the proposed algorithm. A new approach for solving the problems is presented in Section V. Section VI contains the design validation and analysis of the results. Finally, in Section VII, we draw out conclusions.

II. RELATED WORK

Several recent approaches exist regarding enhancement of intra frame coding in order to try reducing the prediction error by exploitation of predicted pixels that are more closer to the coding pixel while maintaining or enhancing the performance level of PSNR and reducing execution time, power consumption and area.

For instance, Mukherjee et al. [11]. proposed CAVLC decoder using VLSI design with a view for enhancing transmission of HD videos without prejudice to the area, with the possibility of integration with the rest of the H.264 video coder blocks.

In [12] the authors claimed that H.264 intra coding generally considers the main process to achieve a higher compression ratio and great PSNR. Therefore, for providing the foregoing, hardware implementation of H.264 intra encoder/decoder scheme is proposed.

The study in [13] is based on the decoding of intra frame using 4x4 pipelined architecture which is implemented by using Verilog HDL. The author was able to enhance the speed of decoding but did not mention the power analysis and PSNR.

In another study [14] the Verilog hardware language is used to implement dual modes video decoder chip based on Intra prediction analysis.

The authors claimed that this approach improves the processing speed and frequency simultaneously while satisfying both timing and area for the chip design. The study in [15] proposed an approach for the Intra prediction unit based on area-efficient hardware and a configurable highthroughput. The baseline point of this proposal is to minimize the redundancy in addition operations and the area hardware implementation by reuse of data paths.

While [16] aimed at strengthening complexity reduction of traditional decoder through modifications into subunits of Inverse Quantization (IQ) and Inverse Transform (IT) units. This approach has already led to decreased complexity by 40% but on the other hand, it causes PSNR degradation.

In article [17] the authors demonstrate Matlab code that can act as a basis for hardware implementation of (CAVLC) encoder/decoder to process High Definition video sequences. The aforementioned research [11,15] tried working on a reduction of computational complexity, based on Innovative approaches. But they did not mention the depth of PSNR changes and Power consumption. Similarly, the approach proposed in [16,17] resulted in obvious degradation in the PSNR ratio.

All of the above papers, have motivated the authors to concentrate this research to achieve impressive results in all performance parameters of the intra frame codec of PSNR, compression ratios, and bit rates.

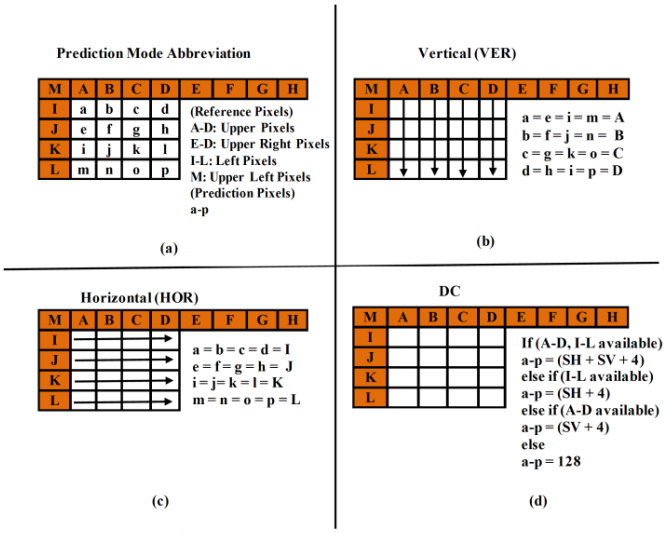

Fig. 2. Equations of luma 4 x 4 mode prediction pixels (vertical, horizontal, and DC) [18]

III. BEST PREDICTION MATRIX MODE

Intra prediction is the primary factor for the eradication of the spatial redundancy in the frame. The prediction modes are illustrated in Fig. 2. The coding of each 4×4 sub-blocks may be performed by the use of nine prediction modes, these modes should be ready to be transported to the decoder. Some or all of 13 neighboring pixels (A-M) are a prerequisite for building the nine prediction modes. The arrows in Fig. 2. illustrate the use of the neighboring pixels to create each of vertical, horizontal and DC modes.

The newly proposed technique is clarified in this section. It has been proved as a promising paradigm to upgrade the bit rate, the compression capability, and PSNR of the H.264 encoder. Statistics point out that the DC, horizontal, and vertical prediction modes are repeatedly used than other modes. The reason for this is owing to the fact that a high correlation between the predicted pixels and the reference samples exists [19].

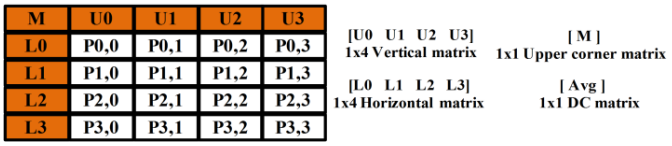

Fig. 3. Construction matrices of our BPMM Intra prediction algorithm

This stimulated us to design our BPMM, where the essential idea underlying the proposed technique is to aggregate the DC,

horizontal and vertical modes in a single new intra 4 × 4 prediction mode. The tradeoffs between values of the four matrices: horizontal, vertical, upper-left corner and DC matrices are a basic approach to achieve our BPMM calculations. Fig. 3. illustrates the construction matrix, which comprises the above-mentioned four matrices.

Fig. 4. shows the newly proposed Intra prediction technique. Pi,j symbolizes the pixel to be predicted in the ith row and jth column of the current 4 × 4 block where 0 <= i, j<= 3, Li and Uj symbolize the reference samples reconstructed from left and upper blocks respectively.

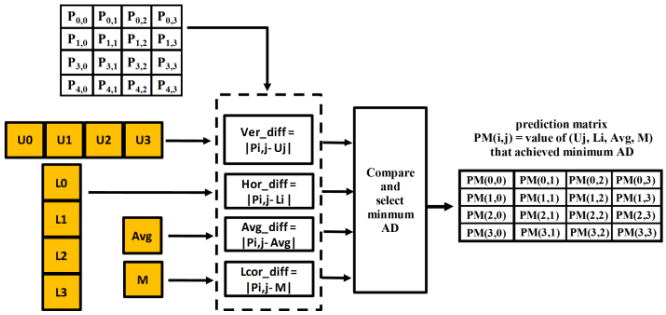

Fig. 4. The proposed Intra prediction technique

Table I summarizes the procedure for generating the best prediction matrix. Pi,j symbolizes the pixel to be predicted in the **i**th row and **j**th column of the current 4 × 4 block where 0<=i,j<= 3, Liand Uj symbolize the reference samples reconstructed from left and upper blocks respectively. M and Avg symbolize the values of upper-left corner pixel and the average of the neighboring pixels respectively. Ver_diff, Hor_diff, Avg_diff, and Lcor_diff symbolize the absolute difference values of the vertical, horizontal, average, and the left corner.

TABLE I
PSEUDO CODE FOR COMPUTING BPMM

Algorithm 1

1: Read original intra 4x4 block ;
2: Prepare construction matrices;
3: Compute the absolute difference as follows:
4: **for** (i , j = 0 : 3) **do** ►step = 1
5: Ver_diff= $|P_{i,j}- U_j|$; Hor_diff = $|P_{i,j}- L_i|$;
6: Avg_diff =$|P_{i,j}-$ Avg|; Lcor_diff = $|P_{i,j}- M|$;
7: Find the minimum value of the absolute differences;
8: Put the value of one of (U_j, L_i, Avg, M) that
Corresponds to this minimum in the prediction matrix (i,j).
9: **end for**
10: Get the best prediction matrix.

As demonstrated in Fig. 5., the newly proposed Intra prediction scheme is explained through a numerical example. As shown in the example the proposed scheme is executed through three steps wherein the first step, as the traditional way of H.264 standard, we will construct four prediction blocks with a dimension of 4 x 4 for each mode of vertical, horizontal, average, and moreover than the standard also the left corner. As the standard in the 2nd step, we are starting to calculate the absolute difference for every pixel in each 4 x 4 mode separately not the sum of absolute difference as the standard. In this step, we will detect the nearest value for each pixel that constructs the original sub-block from the surrounding pixels that achieves a minimum absolute difference which are circled in red as shown in Fig. 5. Within the 3rd step, we shall elect these best prediction pixels and combine them together to get finally the best prediction matrix. It should be noted that the best prediction matrix is an excellent substitute of the standard prediction modes and we will rely on it only to achieve the prediction. Further details regarding the proposed algorithm can be found in [9,10].

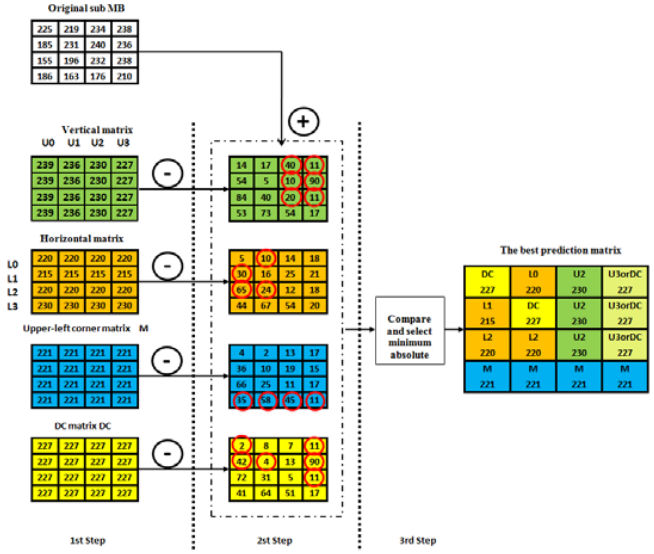

Fig. 5. A numerical example of applying the BPMM

TABLE II
EXPERIMENTAL RESULTS OF BPMM COMPARED TO HENSON[20]

Sequence CIF	PSNR increase (dB)	Comp. Ratio increase %
City	+0.6	38.58
Mother-Daughter	+0.6	28.24
Crew	+0.71	24.47
Water	+0.6	42.97
Average	+0.627	33.56

As shown in Table II the preliminary findings have shown that the (BPMM) had achieved a leading compression ratio with an average of 33.56%. Also we have an increase in PSNR with an average of 0.627dB, bearing in mind we are using CIF videos within variable QP values.

But we must never forget that the obtained results are based on using the headers of H.264 standard. These headers would not be suitable to signal the prediction mode for each pixel, but limit its work to represent the prediction mode and sending it for each block. In the next section, we will discuss this issue in detail.

IV. Open Problems and Challenges

As soon as the Intra prediction modes for each 4 × 4 luma sub-block are established the decoder needs to receive these modes. But this brings a heavy burden to exhaust a huge number of bits. There is a recognized fact that the neighboring 4 × 4 luma sub-blocks are often highly correlated in the same macroblock especially for their modes. For further clarification, a prediction block for current sub-block Z is calculated based on the neighboring previously encoded blocks X and Y and let 's say mode 2 is used to predict both previously encoded blocks X and Y as shown in Fig. 6.

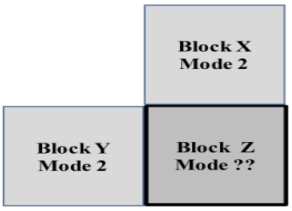

Fig. 6. Intra mode prediction example [8]

Probably that the best mode for current sub-block Z is mode 2 also. To benefit from this information, the predictive coding expects 4 × 4 sub-blocks modes so that the most probable prediction mode to current4 × 4 sub-block is calculated in the encoder and decoder defined as the smallest prediction modes among X and Y. The DC prediction mode will be selected in case of non-availability of these neighboring blocks [8].

As mentioned above, Intra prediction modes must be signaled to the decoder. Consequently, the H.264 encoder sends two flags for each 4×4 sub-block, prev_intra 4×4_pred_mode (one bit) is set to 1 if the most probable prediction mode is used, and rem_intra4×4_pred_mode (three bit) is sent in the case of the prev_intra4×4_pred_mode flag is 0 to indicate that there's a change in mode. In view of the foregoing explanation, the best cases, if the sixteen 4 × 4 sub-blocks which represent the fully MB have the same prediction modes, the overhead bits for establishing the modes will be:

$$\textit{Total Overhead} = \textit{Num of bits for (prev\_mode\_flag)} \times \textit{Num of sub-blocks} \quad (1)$$

$$= 1\left(\frac{bit}{sub\ block}\right) \times 16\left(\frac{sub\ block}{MB}\right)$$

$$= 16\left(\frac{bit}{MB}\right)$$

However, if the modes of sixteen 4 × 4 sub-blocks are different from each other at the worst the overhead bits are then:

$$\textit{Total Overhead} = [\textit{Num of bits for (prev\_mode\_flag)} + \textit{Num of bits for (rem\_mode\_flag)}] \times \textit{Num of sub-blocks} \quad (2)$$

$$= \left(1\left(\frac{bit}{sub\ block}\right) + 3\left(\frac{bit}{sub\ block}\right)\right) \times 16\left(\frac{sub\ block}{MB}\right)$$

$$= 64\left(\frac{bit}{MB}\right)$$

The key feature provided by this traditional approach depends on the whole 4 × 4 sub-block has only one prediction mode. In contrast, each pixel in the 4 × 4 sub-block has its own mode in our proposed technique. Consequently, there are sixteen modes for each 4 × 4 sub-block must be signaled to the decoder and this could certainly contain a tremendous number of bits.

For greater clarity, at best the minimum overhead bits can be achieved if the full sixteen pixels forming the sub-block have the same prediction modes as follows:

$$\textit{Total Overhead} = \textit{Num of bits for (prev\_mode\_flag)} \times \textit{Num of pixel fo rsub-blocks} \times \textit{Num of sub-blocks} \quad (3)$$

$$= 1\left(\frac{bit}{pixel}\right) \times 16\left(\frac{pixel}{sub\ block}\right) \times 16\left(\frac{sub\ block}{MB}\right)$$

$$= 256\left(\frac{bit}{MB}\right)$$

The overhead bits are described below, in the worst case when the full sixteen pixels modes in the sub-block are different from each other:

$$\textit{Total Overhead} = [\textit{Num of bits for (prev\_mode\_flag)} \times \textit{Num of pixel for sub-blocks} + \textit{Num of bits for (rem\_mode\_flag)} \times \textit{Num of pixel for sub-blocks}] \times \textit{Num of sub-blocks} \quad (4)$$

$$= \left(1\left(\frac{bit}{pixel}\right) \times 16\left(\frac{pixel}{sub\ block}\right) + 3\left(\frac{bit}{pixel}\right) \times 16\left(\frac{pixel}{sub\ block}\right)\right) \times 16\left(\frac{sub\ block}{MB}\right)$$

$$= 1024\left(\frac{bit}{MB}\right)$$

Based on the above facts, the implementation of the Best Prediction Matrix Mode (BPMM) in practice is not a frivolous task. The key daunting trouble is that we have a high overhead, a result of the existence from 240 to 960 extra bits in each sub-block mode would provoke a heavy loss of bandwidth. Accordingly, the decreasing overhead is a bottleneck for (BPMM) in practice [21].

V. Proposed Methodology for Reducing the Overhead

The Prediction Matrix Mode (BPMM) and its challenges have been introduced in previous sections in this article, where signaling of prediction mode bits requires high overhead of bits. A survey of the existing literature led us for some methodologies to overcome this issue. Huffman Encoding algorithm is one of the most famous lossless variable length coding algorithms. So, we enlisted the help of this coding algorithm to reduce overhead. Other lossless compression can be used equally well such as CABAC.

A. Huffman Algorithm

Despite the promising benefits of Best Prediction Matrix Mode technique, high overhead is a critical issue, which significantly impedes its practical use. Huffman coding has been increasingly accepted as an effective way to get hold of how bit rate of each block header information did not increase even we coded each pixel with different modes which expectedly causes an increase in bit rate.

We first review the hierarchy of the Network Abstraction Layer (NAL) as illustrated in Fig. 7. The Macroblock layer was a major area of our focus and it will be a starting point for our methodology.

This layer contains all headers and data needed by the decoder as following:

- mB_type specified the types of the macroblock whether it I, SI, P or B.

- mb_pred, which indicates the type of prediction and we can extract the mode that was chosen for each sub-blocks by using this header.
- coded_block_pattern which is implemented to know the numbers of non-zero transform coefficients whether there are one or more.
- Finally, the data of QP and residual will come successively.

The key feature of the mb_pred header is that it holds all data of prev_mode_flag and rem_mode_flag for the sixteen 4 × 4 sub-blocks that are combined together to produce one completed macroblock. As we have said previously in Section IV the length of this header is dynamic and it will lie between two boundary values from 16 Bit/MB if the sixteen sub-blocks representing the full MB have the same prediction mode, to 64 Bit/MB when the modes of this sub-blocks are different from each other.

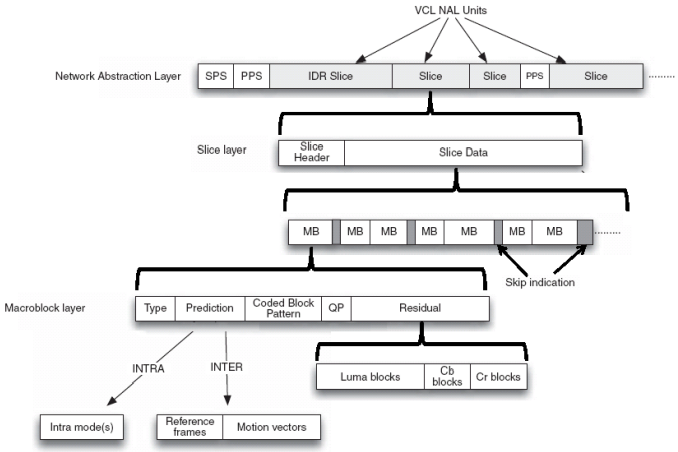

Fig. 7. The Syntax overview [8]

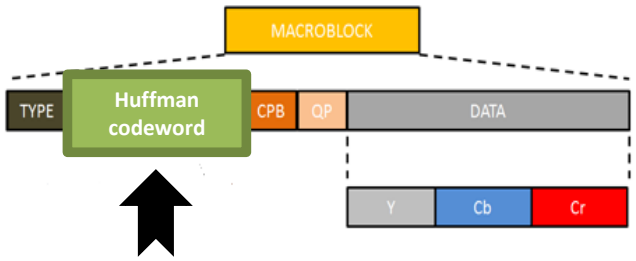

Fig. 8. Huffman codeword as a syntax of the macroblock

In order to solve the higher header overhead of the best prediction matrix we will replace the mb_pred header with a Huffman codeword and send it as a syntax of the macroblock layer in the H.264 encoded bitstream as shown in Fig. 8. Each Huffman codeword will be representing unique prediction modes of fully MB (16 × 16 best prediction matrices). Then, we can extract the prediction mode of each pixel in the MB using a Huffman code dictionary. The length of the Huffman codeword has varied between 10, 13, 14, and 15 Bits/MB after it was varied from 256 to 1024 Bits/MB. This means that we avoided the overhead without exceeding the permitted limit of the standard mb_pred header (16~64) Bits/MB.

The Pseudo code that has been applied to code the prediction mode information for every pixel using Huffman coding is provided by Algorithm in Table III.

TABLE III
PSEUDO CODE FOR COMPUTING HUFFMAN CODING

Algorithm 2 Pseudo code for Huffman Coding

1: After we calculate the best prediction matrix of 4×4 sub-block, then we get sixteen 4×4best prediction matrices for each MB are being combined to construct the best prediction matrix with 16×16 dimensions,

2: Determine the repetition of unique 16×16 best prediction matrices,

3: Arrange the unique 16×16 best prediction matrices indescending order according to its repetition, and

4: Generate a Huffman code dictionary based on theprobability of each 16 $\times$ 16 best prediction matrix that's where assigned fewer numbers of bits to the most frequent symbol.

Recent work indicated that the Huffman coding achieved that there is no similarity in the Huffman encoded code words for each macroblock and the code words can be inserted in the H.264 bitstream and transmitted together easily without additional prefix codeword. Consequently, the overhead challenges can be relaxed.

The H.264 bit stream which has code words of prediction mode information for every pixel can be decoded with every standard H.264 decoder. Only in order to decode it, it is needful to use the proper decoder, which contains additional components so it could achieve the decryption operation and then extract prediction mode information from Huffman code word. As a result, the decoding procedure requires a Huffman code dictionary and must recognize the code length as well as the symbol itself. Then it decodes it and gets the prediction mode information [21]. Table IV summarizes this procedure as follows:

TABLE IV
PSEUDO CODE FOR COMPUTING HUFFMAN DECODING

Algorithm 3 Pseudo code for Huffman decoding

1: Detection of the Huffman code length that can be extended up to 16 bits, then extract codeword from H.264 bit stream ,

2: We should do a search through Huffman code dictionary to find prediction mode information corresponding to the extracted codeword,

3: creation of best prediction matrix of 4×4 sub-block,

4: Finally, we get sixteen 4×4 best prediction matrices for each MB are being combined to construct the best prediction matrix with 16×16 dimensions,

VI. FPGA IMPLEMENTATION OF THE PROPOSED CODEC

The evaluation of our new Intra prediction technique and its experimental results are addressed in this section based on these aspects: First, create of checking model to verify the requirements and specifications of our technique. Second, reviewing some analysis of timing, area, and power. Third, We are going to use Henson hardware as a standard H.264 encoder reference [20]. The VHDL implementation of our new Intra prediction technique decoder is done through H.264 hardware

decoder of Nova [22] as the platform for testing our proposal. Lastly, the results of PSNR, compression ratio, and bit rates are discussed for more accurate evaluating of our prediction approach.

A. Nearest to the Best (NTB) as a model checking

The model checking method is the most popular and efficient way to verify the requirements and specifications of systems. This technique is generally used as a credentialing tool whether the final system satisfies an expected requirement and desired specification. If the model checking outputs are true this signifies that the tested system satisfies given specifications. Otherwise, a counter example will be generated to identify the exporter of the error in the model, error correction, and try one more time. The main object is to increase our confidence in the correctness of the proposed system by guaranteeing that the model satisfies all of the system requirements.

Based on the above, we proposed the Nearest to the Best approach as a model checking of our proposed technique (BPMM). How to achieve successfully selection of the best matching block between the original and predicted macroblocks is the main subject of Intra prediction. That choice depends essentially on calculating the sum of absolute differences (SAD) between pairs of original and predicted macroblocks and choosing the smallest one. The (SAD) equation is as follows:

$$SAD = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |C(i,j) - R(i,j)| \quad (5)$$

where **C**(i, j) and **R**(i, j) are the (i, j)th elements of the current original block **C** and the reconstructed block **R** [8].
The proposed NTB for model checking events relies on that, the part in (SAD) equation which has the current original block will be replaced with BPMM block. so that the (SAD) equation would read as follows:

$$\mathbf{SAD} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |\mathbf{BPMM}(i,j) - \mathbf{R}(i,j)| \quad (6)$$

Where **BPMM**(i, j) and **R**(i, j) are the (i, j)th elements of the current **BPMM** block and the reconstructed block **R**.
To guarantee the success of the model checking, the encoding process will be reinstated again by relying on the new form of (SAD) equation to explore which mode is Nearest to the Best and satisfy the smallest SAD. If we get PSNR and compression ratio identical to those we get in the standard, it means that model checking outputs are true. This indicates that the tested BPMM algorithm satisfies standard specifications. Otherwise, it means that we have an error in the proposed algorithm needs to be corrected.

The experimental results illustrate that the result of comparability process between two operations of video compression one using traditional (SAD) formula, and the other based on Nearest to the Best (SAD) formula. The results demonstrate that our proposed technic has passed the model checking test and has proven accurate every time.

B. Timing

The MB prediction state is the key cycle to generate the three phases of exploring prediction modes which is simplified and demonstrated in Fig. 9. The MB prediction state in the traditional H.264 decoder is composed of prev_mode_state, rem_mode_state, and intra_chroma_state, whereas in our approach its relies on Huffman detection state, a Prediction mode detection state, and intra_chroma_state.

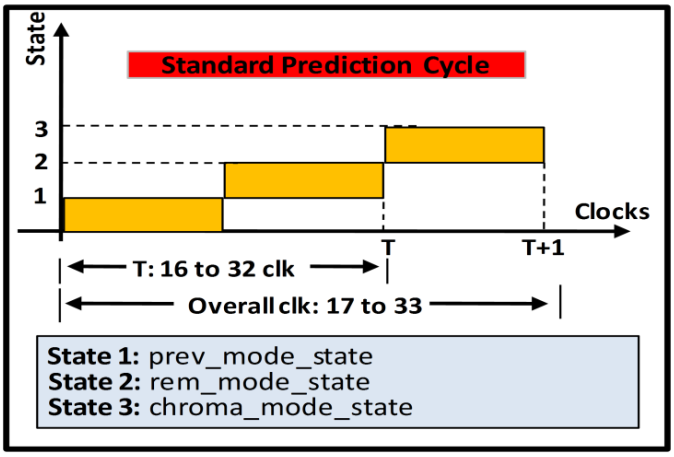

Fig. 9. The MB prediction state in the standard H.264 decoder

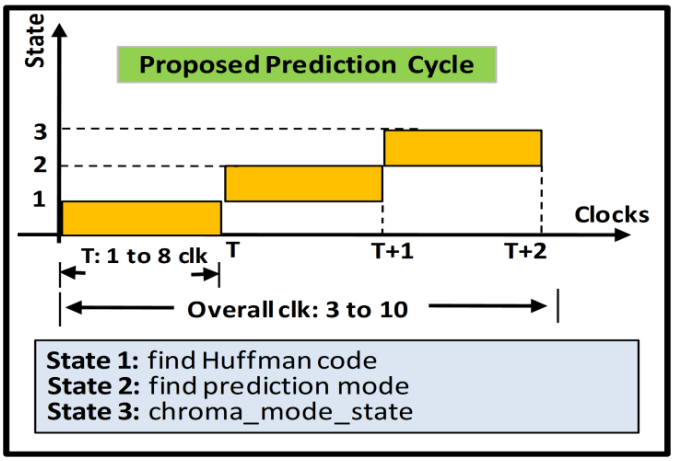

Fig. 10. The MB prediction state in the proposed H.264 decoder

As shown in Fig. 10, an important feature in the MB prediction state architecture is that the time consumed in the standard H.264 decoder and our proposed technics are different. As noted earlier in section 4, the number of clocks that MB prediction state would need to achieve a prediction process is ranged from 17 to 33 clocks/MB dependent on whether or not all of the sixteen sub-blocks that shape the MB have the same prediction modes or not. Whilst 3 to 10 clocks/MB are enough for our approach to provide good prediction performance in the same stage.

It is worth observing that BPMM decoder can now run much faster than the standard one, along with the Huffman algorithm that there is a significantly higher chance to encounter overhead suggesting very promising performance enhancements.

C. Power& Area

In the same context, and in order to provide validation of the new scheme, Power consumption analysis and Area analysis are two bottlenecks that our approach must pass both of them. The evaluation of the new algorithm with regard to Power consumption is shown in Table V, where it is clear that BPMM new algorithm achieved a decrease in the dynamic power on the average by 20.21% compared to the H.264 decoder testbed of Nova.

On the other hand, we used Xilinx ISE 14.2i software to accomplish the synthesis of the new prediction algorithm. To evaluate the accuracy of our obtained results the calculation of area usage is done through loading the proposed algorithm in Xilinx Vertix-4 FPGA. The results of area utilization are demonstrated in Table V for our BPMM algorithm, Nova, and Mukherjee et al. It may be noted that the number of occupied slices is smaller by 10.13%, LUTs area is smaller by 9.6%, and FFs count is smaller by 19.7% compared to Nova. Seeking for more judgment we compared it with newly developed algorithm of Mukherjee et al. The results indicate that the number of occupied slices is less by 31.35%, LUTs area is less by 40.68%, FFs count is less by 67.82%, and the authors in this paper did not give the Dynamic power.

TABLE V
POWER CONSUMPTION AND FPGA AREA ANALYSIS COMPARISON BETWEEN THE PROPOSED APPROACH , NOVA[22], AND MUKHERJEE ET AL[11]

Attributes	Proposed	Nova	Mukherjee et al.
Dynamic power (W)	0.075	0.094	The authors did not mention the power
Number of occupied slices	1,835	2,042	2,673
Number of 4-input LUTs	3,171	3,508	5,346
FFs	573	714	1,781
Number of IOBs	111	110	1,63

D. Huffman Encoder Architecture Realization

Our hardware architecture of Huffman encoder is shown in Fig. 11. It consists of five main modules to realize Huffman encoder architecture, where we can store the modes of current MB pixels using current MB buffer module, then the generated Huffman dictionary is stored within dictionary buffer module. Also, counter, controller, and a comparator are needed. Finally, the handshaking is provided by “STROBE” and “READY” signals. The prime object of the Huffman encoder is to provide each MB Pixels mode with a unique Huffman code. A request will be sent to the ram which holds the 16×16 best prediction matrix when the READY signal set high, then the Current MB Pixels mode will be extracted and received by Current MB Buffer to load it. Now we get two outputs: one is”STROBE” signal and the other is the Huffman code. The loaded current MB Pixels Mode will be compared with the MB Pixels Mode that stored in Huffman Dictionary Buffer. If the comparison process result is identical the comparator will set ”STROBE” to "1", referring to that Huffman code is valid. On the other hand in the case when each of the current MB pixels mode and the MB Pixels Mode that stored in Huffman dictionary Buffer are different the controller will increment the counter to the next address looking for Huffman code in the rest of Huffman Dictionary Buffer locations.

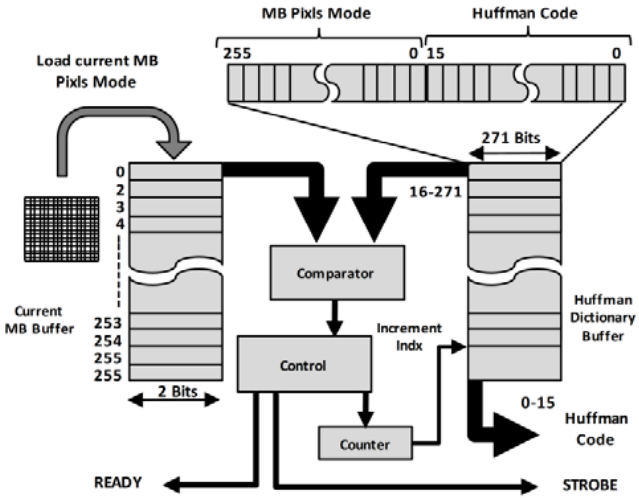

Fig.11. The hardware architecture of Proposed Huffman Encoder

E. HuffmanDecoder Architecture Realization

The Huffman decoder hardware is integrated into the H.264 decoder as a sub-module of its Intra Prediction module. As shown in Fig.12, the realization of Huffman decoder architecture is depending on five core modules, namely Current Huffman Buffer to store the Current Huffman code, MB Dictionary Buffer in which the generated MB Pixels mode Dictionary is stored, counter, controller and comparator. Also two signals namely “STROBE” and “READY” are acting as handshaking signals. The key function of the Huffman decoder is to get the Current MB Pixels mode of each Huffman code to be decoded. When the READY level is high,it means the ram holding the Current Huffman code has received a request to load them in the Current Huffman Buffer.

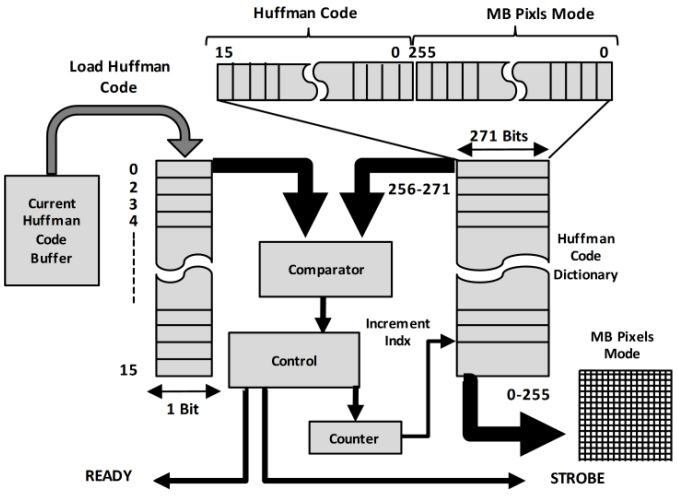

Fig. 12.The hardware architecture of Proposed Huffman decoder

We can get two outputs from the Huffman decoder: One is the Current MB Pixels mode and the other is so-called "STROBE" signal. The main comparison process would occur between the loaded Current Huffman code and the stored Huffman codes in MB Dictionary Buffer.Once "STROBE" signal set by "1" this indicates that MB Pixels mode is considered as valid.But then again if both Current Huffman code and the stored MB Pixels mode in MB Dictionary Buffer are drastically different, then we will be accessing next addressing MB Dictionary Buffer looking for MB Pixels mode by increments the counter through the controller signal.

F. Performance Evaluation of our Proposed Approach

A systematic evaluation of the proposed Best Prediction Matrix Mode decoder is presented in this part based on the findings and results gained through subjugation the aforementioned scenarios to our experimental testbed evaluation. This will be achieved by using the sequences of (QCIF) and (CIF) video formats for a case study. Comparisons were made by four QCIF videos namely Salesman, Foreman, Mother-Daughter and Silent as well as three CIF videos namely City, Water, and Crew. To demonstrate the superiority of our approach, we explore the results of evaluating our prediction algorithm in the following performance parameters: compression ratio, PSNR and bit rates.

A prototype of H.264 decoder, including our Intra prediction algorithm and proposed method of Huffman design, is implemented using Nova hardware decoder as a platform of testing.

For an accurate judgment comparison and evaluation, we configure both of encoder and decoder to use various values of a quantization parameter within range 20 to 45 with a step of 5. Then, the development ofBPMM encoder/decoder faces the main challenge which concentrated on that the result of the comparison between the PSNR of the reconstructed video in encoder-side and the other PSNR of decoded video in decoder-side must be identical.Accordingly, Fig. 13.demonstrates our success to pass this bottleneck through a sample of the video. Also, the rest of videos showed similar behavior.

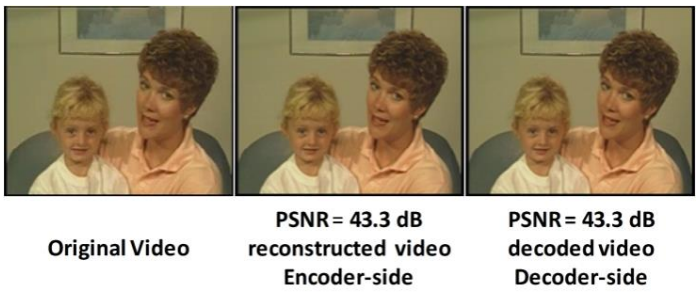

Fig. 13. The PSNR Mother-Daughter video

TableVI summarizes our experimental results of developed (BPMM) with and without Huffman compared to Henson, which indicates that our codec yielded a number of improvements in terms of compression ratiothat increases by 50.97 %, the bit rate decreases by 29.19 %, and the PSNR is increased on the average by 0.627dB. The increasing of compression ratio after we are using Huffman is evident due to the header length will be varying between (10,13,14, and 15) Bits/MB after we supposed using the best cases of the standard header(16 bits for each Macroblock).

For further confirmation, the newly proposed prediction technique is tested with QCIF video sequences and the results are summarized in Table VII. Compared to Henson the newly proposed codec achieves enhancing the compression ratio by 54.82 %, the bit rate by 28.27 %, and the average PSNR by 0.627dB.

TABLE VI
EXPERIMENTAL RESULTS OF DEVELOPEDBPMMWITH AND WITHOUT HUFFMAN COMPARED TO HENSON[20]

Video Cif	PSNR increase (dB)	Comp. Ratio %		Comp. Ratioinc rease %	Bit rate decrease %
		Proposed without Huffman	Proposed with Huffman		
Water	+ 0.6	+42.97	+56.198	+30.78	-29.506
Crew	+0.71	+24.47	+40.625	+66.01	-29.399
City	+0.6	+38.58	+55.9	+44.89	-26.932
Mother-Daughter	+0.6	+28.24	+45.81	+62.21	-30.95
Average	0.627	+33.565	+49.63	+50.97	-29.19

TABLE VII
EXPERIMENTAL RESULTS OF DEVELOPEDBPMM COMPARED TO HENSON[21]

Video Qcif	PSNR increase (dB)	Comp. Ratioincrease %	Bit rate decrease %
Silent	+0.5	+52.54	-27.203
Mother-Daughter	+0.7	+48.648	-29.096
Foreman	+0.68	+54	-27.859
Salesman	+0.63	+64.102	-28.96
Average	+0.627	+54.82	-28.27

Figs 14-20 clearly indicate that the proposed BPMM encoder constantly outperforms the standard prototype of H.264 (Henson platform) in terms of improving the compression ratio. As a consequence, there is a reduction in the bit rate also the PSNR is marginally increased.

Water CIF video follows the same tendency in enhancing the compression ratio on average by 56.198% and as a result, the bit rate is decreasing on average by 29.506% while the PSNR is slightly increased on average by 0.6 dB and the predictor repeats itself for the other CIF videos as depicted in Figs 14-16.

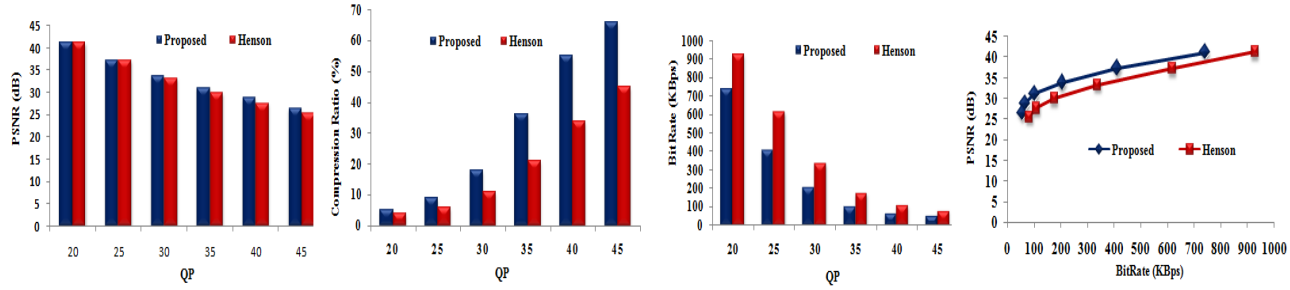

Fig. 14. Evaluation of Developed BPMM Using Water CIF Video.

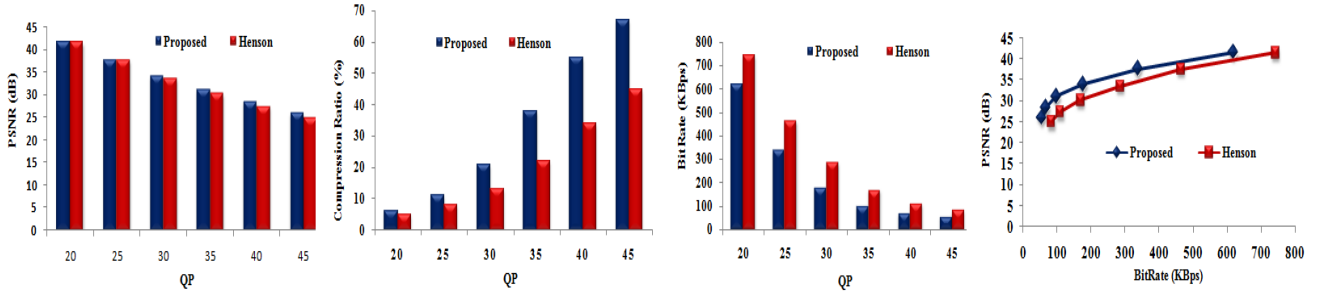

Fig. 15. Evaluation of Developed BPMM Using City CIF Video.

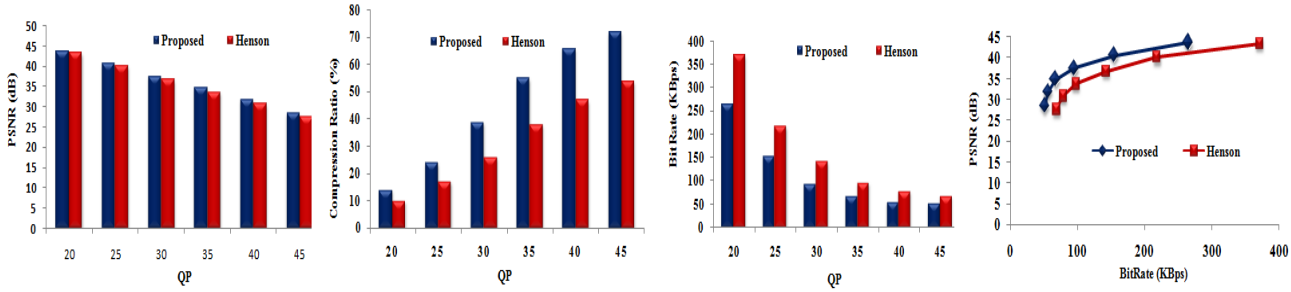

Fig. 16. Evaluation of Developed BPMM Using Crew CIF Video.

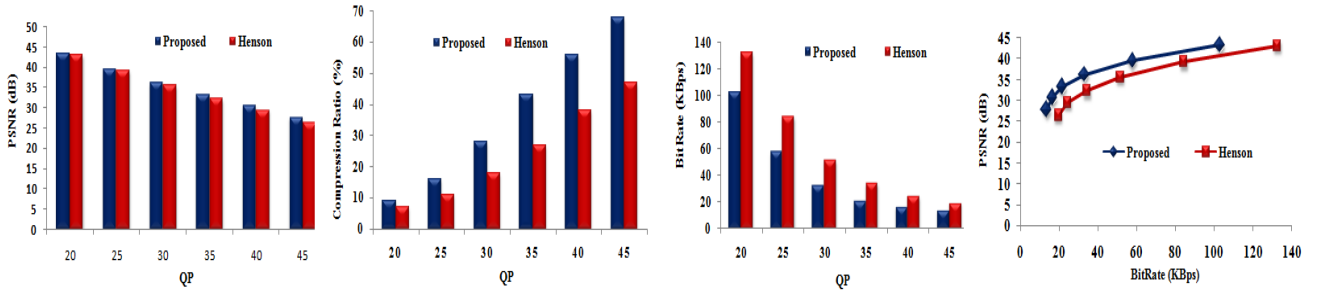

Fig. 17. Evaluation of Developed BPMM Using Mother-DaughterQCIF Video.

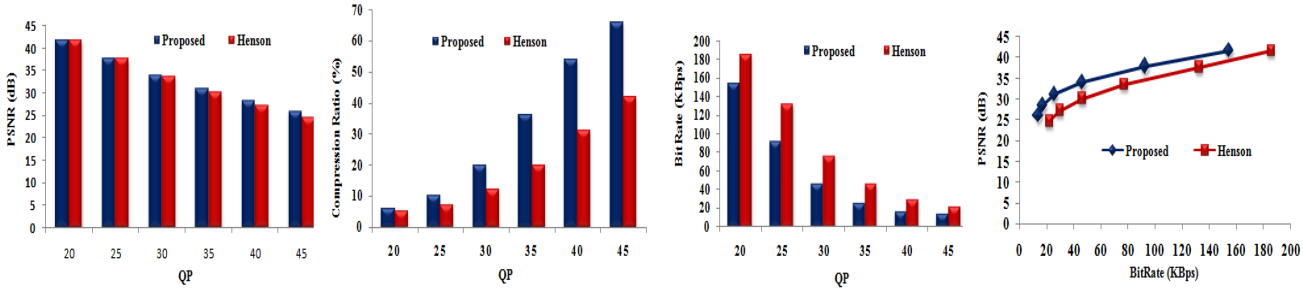

Fig. 18. Evaluation of Developed BPMM Using Salesman QCIF Video.

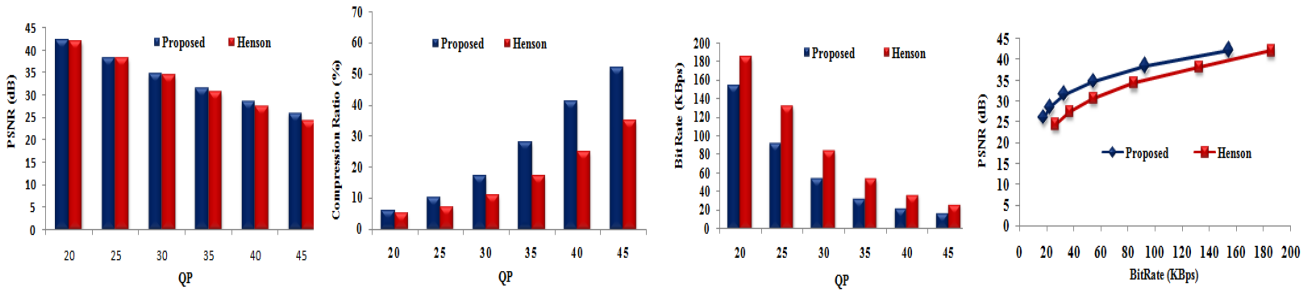

Fig. 19. Evaluation of Developed BPMM Using Foreman QCIF Video.

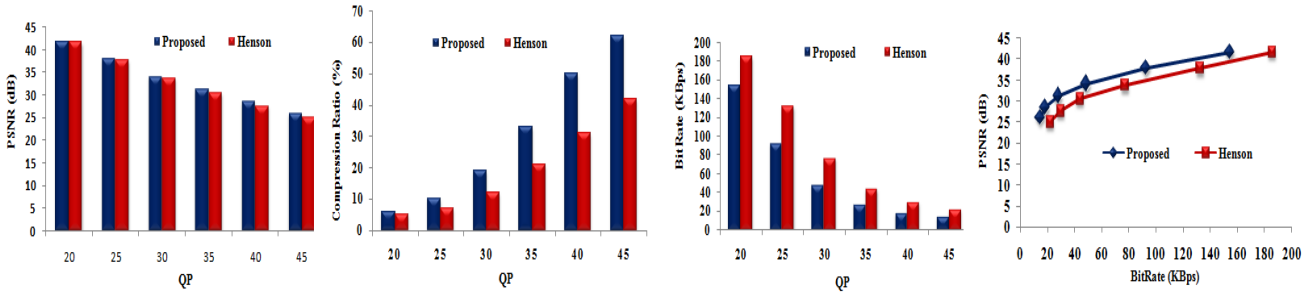

Fig. 20. Evaluation of Developed BPMM Using Silent QCIF Video.

TABLE VIII
EXPERIMENTAL RESULTS OF DEVELOPEDBPMM COMPARED TO TIAN SONG ET AL[23].

Sequence CIF	Method	Parameter	Quantization Parameter				Bitrate Reduction	PSNR Increase (dB)
			24	26	28	30		
Mobile	Tian Song et al	PSNR(dB)	38.21	36.44	34.85	33.01	-7.25 %	+ 0.01
		Bit rate (kbps)	2614	1987.5	1507.3	1006.4		
	Proposed	PSNR(dB)	38.125	36.47	34.88	33.07		
		Bit rate (kbps)	2053.71	1767.89	1516.82	1260.88		
Coastguard	Tian Song et al	PSNR(dB)	38.62	36.7	35.38	33.47	-36.98 %	+ 1.59
		Bit rate (kbps)	1871.5	1342.03	1046.26	731.21		
	Proposed	PSNR(dB)	39.6	38.21	37	35.72		
		Bit rate (kbps)	1071.66	857.86	686.60	529.27		
Football	Tian Song et al	PSNR(dB)	40.96	39.42	38.08	36.7	-57.29 %	- 0.95
		Bit rate (kbps)	2308.89	1796.75	1434.15	1136.71		
	Proposed	PSNR(dB)	39.78	38.5	37.33	35.725		
		Bit rate (kbps)	891	742.5	677.57	540.03		
BUS	Tian Song et al	PSNR(dB)	38.79	36.86	35.25	33.54	-21.14 %	+ 0.77
		Bit rate (kbps)	1796.69	1324.39	1013.65	730.17		
	Proposed	PSNR(dB)	39.05	37.56	36.17	34.72		
		Bit rate (kbps)	1247.67	1037.19	861.4	690.27		
Tempete	Tian Song et al	PSNR(dB)	38.31	36.56	34.98	33.27	- 15.98 %	+ 0.50
		Bit rate (kbps)	1982.31	1480.5	1078.81	732.03		
	Proposed	PSNR(dB)	38.53	37.05	35.63	33.9		
		Bit rate (kbps)	1454.98	1200.98	991.33	783.52		
Average							-27.728 %	+ 0.384

TABLE IX
EXPERIMENTAL RESULTS OF DEVELOPEDBPMM COMPARED TO JM18.6&N MANJANAIK ET AL[24].

Sequence QCIF	Method	Parameter	Quantization Parameter			Bit rate N Manjanaik et al Method	Bit rate Proposed Method	Inference
			24	28	30			
Coastguard	JM18.6	PSNR(dB)	38.64	36.51	34.95	-21.5 %	-54.31 %	Our proposed method achieved more Bit rate Reduction on the average by 24.94 % Compared With N Manjanaik et al Method & PSNR is increased on the average by 0.96 dB
		Bit rate (kbps)	1821.16	1522.08	1260.84			
	Proposed	PSNR(dB)	39.68	37.16	35.85			
		Bit rate (kbps)	990	626.02	487.5			
	N Manjanaik et al	PSNR(dB)	34.89	34.9	34.93			
		Bit rate (kbps)	1354.33	1194.44	1065.13			
Foreman	JM18.6	PSNR(dB)	39.88	38.64	37.43	-25.5 %	-32.77 %	
		Bit rate (kbps)	1256.08	1044.88	870.64			
	Proposed	PSNR(dB)	39.9	37.37	36.06			
		Bit rate (kbps)	954.84	653.2	524.12			
	N Manjanaik et al	PSNR(dB)	38.92	38.93	38.98			
		Bit rate (kbps)	914.77	774.05	673.72			
FOOTBALL	JM18.6	PSNR(dB)	38.77	37.08	35.48	-27.45 %	-30.89 %	
		Bit rate (kbps)	1996.08	1678.4	1409.2			
	Proposed	PSNR(dB)	38.25	35.16	33.68			
		Bit rate (kbps)	1616.25	1098.75	798.28			
	N Manjanaik et al	PSNR(dB)	35.44	35.05	35.51			
		Bit rate (kbps)	1373.11	1218.29	1096.57			
BUS	JM18.6	PSNR(dB)	38.64	36.78	35	-44.51 %	-68.66 %	
		Bit rate (kbps)	2968.12	2546.24	2163.84			
	Proposed	PSNR(dB)	40.08	37.06	35.71			
		Bit rate (kbps)	1069.69	742.5	594			
	N Manjanaik et al	PSNR(dB)	35.6	35.6	35.62			
		Bit rate (kbps)	1568.49	1406.57	1284.89			
Average						- 29.74 %	- 46.657 %	

However, the newly proposed prediction technique would be tested with QCIF video sequences. For the Mother-Daughter QCIF video, there is an evident increase in the compression ratio on average by 48.648% and this significantly boosted the bit rate reduction within an average of 29.096%. Meanwhile, the PSNR increases marginally by 0.7dB. Also, the rest of the QCIF videos have the same tendency as illustrated in Figs 17-20.

For further asserting, the newly proposed prediction technique, itis compared with previous works both of Tian Song method[23]by using five CIF videos and N Manjanaik et al Method[24] by using four QCIF videos. Table VIII demonstrates that ours achieved significantreduction in bit rate on average by 27.728% while the PSNR is slightly increased on average by 0.384 dB compared With Tian Song et al method.

Moreover, as illustrated in the Table IX, our proposed prediction technique scored a significant reduction in bit rate on average by 46.657% compared with JM 18.6 reference algorithmwhile previous work by N Manjanaik et al achievedonly a 29.74% reduction in bit rate on average compared with JM 18.6 reference algorithm.

Undoubtly, our proposed prediction technique has been proved to have better performance than comparable Intraprediction techniques under various test conditions.

VII. CONCLUSIONS

This article proposes a novel Intraprediction codec for the H.264 standard. Our scheme introduces interesting results in terms of PSNR, compression ratio and execution time. Our proposed algorithm so-called Best Prediction Matrix Mode (BPMM) has faced several challenges.Perhaps the most important problems are the high overhead bits and decoding mechanism to recover original video prior to display. After overcoming these challenges, (BPMM) has been subjected intensive testing and performance comparisons compared with the H.264 standard. Many experimental results have proved that our proposed method is more efficient than the standard one.

Our future work will involve the recognition of our Intra predictioncodec to be considered as a reference design of H.264. Also, this intraframe codec will be extended to the higher video standards such as H.

REFERENCE

[1] G. Sundari, V. J. K. K. Sonti, and T. Bernatin, "FPGA Implementation of Modified Intra-Frame Prediction for H.264 Video Codec," International Journal of Pure and Applied Mathematics,vol. 118, no. 17, pp. 135–151, Jan. 2018.

[2] F. Golaghazadeh, S. p. Coulombe, F. X. Coudoux, and P. Corlay, "Low complexity H. 264 list decoder for enhanced quality real-time video over IP," IEEE, , pp. 1-6, 2017.

[3] A. Elyousfi, A. El Hachimi, and H. Hamout, "Texture complexity based fast and efficient intra block mode decision algorithm for H. 264/AVC," IEEE, pp. 1-4, 2017.

[4] I. A. Ali, S. Moiron, M. Fleury, and M. Ghanbari, "Data Partitioning Technique for Improved Video Prioritization," Computers., vol. 6, no. 3, p. 23, 2017.

[5] M. Al-Jammas and R. Nabeel, "IBBB Inter Frame Prediction of H.264 /AVC," International Journal of Computer Applications, vol. 181, no. 14, pp. 0975-8887 , Sept.2018.

[6] I. J. Barge and C. Ababei, "H. 264 video decoder implemented on FPGAs using 3x3 and 2x2 networks-on-chip," IEEE, pp. 1-6, 2017.

[7] Chahrazed Adda and Abou Elhassen Benyamina, "Design of the H264 application and Implementation on Heterogeneous Architectures,"

International Journal of Computer Applications, vol. 180 , no. 7 Sept.2017.
[8] I. E. Richardson, The H. 264 advanced video compression standard John Wiley & Sons, 2011.
[9] S. Hamdy, M. E. Ibrahim, and A. Zekry, "VHDL realization of efficient H. 264 intra prediction scheme based on best prediction matrix mode," Int. J. Comput. Appl., vol. 77, pp. 1-7, 2013.
[10] A. S. Hamdy, B. M. E. Ibrahim, and C. M. B. Abdelhalim, "Efficient H.264 intra prediction scheme based on best prediction matrix mode," IEEE, pp. 363-367, 2013.
[11] R. Mukherjee, A. Banerjee, I. Chakrabarti, P. K. Dutta, and A. K. Ray, "Efficient VLSI design of CAVLC decoder of H. 264 for HD videos," IEEE, pp. 1-4, 2017.
[12] M. H. Al-Jammas, and N. N. Hamdoon, "FPGA Implementation of Intra Frame for H. 264/AVC Based DC Mode," International. Journal of Computer. Engineering. and Information. Technology, vol. 9, no. 11, pp. 264-270, 2017.
[13] La-Gou Wu, Yu-Kun Song, Gao-Ming Du, and uo-Li Zhang, "A 4x4 Pipelined Intra Frame Decoder for H.264," International Conference on Anti-counterfeiting, Security, and Identification in Communication, vol. 3rd June.2009.
[14] S. Xu, Y. Xing, and X. Wei, "Design of intra prediction module in H. 264 and AVS dual modes video decoder chip,", 2 ed IEEE, 2011, pp. 635-638.
[15] M. Nadeem, S. Wong, and G. Kuzmanov, "An efficient hardware design for intra-prediction in H. 264/AVC decoder," IEEE, pp. 1-6, 2011.
[16] M. S. Alizadeh and M. Sharifkhani, "Low complexity downsizer H. 264 decoder," IEEE, pp. 604-607, 2014.
[17] G. S. Yogesh and S. Ramachandran, "Context adaptive variable length decoder of H. 264," IEEE, pp. 286-290, 2014.
[18] C.-Y. K. H.-C. a. J.-W. C. Youn-Long Steve Lin, "VLSI Design for Video Coding:H.264/AVC Encoding from Standard Specification to Chip," Springer Publishing Company, 2010.
[19] L. Wang, L. M. Po, Y. M. S. Uddin, K. M. Wong, and S. Li, "A novel weighted cross prediction for H. 264 intra coding," Citeseer, pp. 165-168, 2009.
[20] A. H. Henson, "264 Hardware Encoder in VHDL," Tech. Rep. , Zexia. Access. Ltd., 2008.
[21] S. Hamdy, A. Zekry, and W. A.Mohamed, "Development of the Best prediction matrix mode algorithm for intra prediction in H.264 encoder," International Journal of Engineering & Technology, pp. 13595-13601, Dec.2018.
[22] K.Xu, "Nova: a H.264/AVC Baseline Decoder Specification Rev.0.1",OpenCores, 2008.
[23] T. Song, A. Shimamoto, T. Bando, and W. Zhao, "Novel Intra modes with Temporal-Spatial Predictionfor H.264/AVC," IEEE International Conference on Consumer Electronics-Berlin(ICCCE-B), pp.62-65, Sept. 2011.
[24] N. Manjanaik, B. D. Parameshachari, S. N. Hanumanthappa, and R. Banu, " Intra Frame Coding In Advanced Video Coding Standard (H.264) to Obtain Consistent PSNR and Reduce Bit Rate for Diagonal Down Left Mode Using Gaussian Pulse," IOP Conference Series: Materials Science and Engineering, vol. 225, pp. 1-9, 2017