SEGMENTATION BASED H.264/AVC USING OCTAGON AND SQUARE SEARCH PATTERN IN ADAPTIVE GROUP OF PICTURES MODE

S. Sowmyayani[1], V. Murugan[2]

[1]*Assistant Professor, Department of Computer Science,*
St. Mary's College (Autonomous), Thoothukudi, Tamilnadu, India
sowmyayani@gmail.com
[2]*Assistant Professor, Department of Computer Science,*
MSU Constituent College of Arts and Science, Kadayanallur, Tamilnadu, India
smv.murugan@gmail.com

1. INTRODUCTION

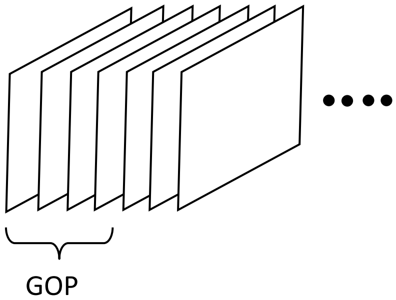
In the digital world, everything we see is being converted into digital form. The communication technology has been widely benefited by the developments of digital signal processing. In the last two decades, communication through internet is spread throughout the world. Digital data is denser than the other data created. All these digital data has to be stored and transmitted. For this data to be stored and transmitted, several spaces are required in discs and bandwidth respectively. It is impossible for storage and transmission if the data are stored as raw data. The process of reducing the size of the data is called data compression. With the increasing popularity of technologies such as video conferencing, broadcasting, video compression has become an essential part in media and other technologies.

A Video occupies more space than images. There are several video compression technologies which yield low data rate for faithful transmission and reproduction of videos. If the video is compressed, it occupies less space. Also, size of the video has an impact on transmission in video conferencing and other broadcasting applications. One among several video compression standards is H.264/AVC. This chapter modifies the architecture of H.264/AVC and developed a framework for video compression.

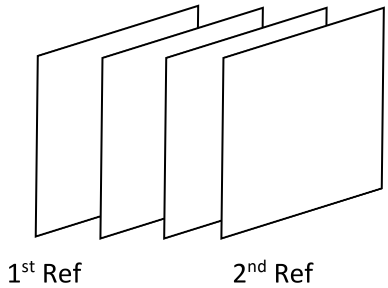
2. H.264/AVC OVERVIEW

The H.264 standard consists of various features and coding tools that contribute to the high compression efficiency, flexibility and robustness. The video sequence is first divided into Group Of Pictures (GOP). Each GOP consists of two keyframes (the first and last frame) as illustrated in Fig. 1. Each GOP consists of fixed number of frames. The architecture of H.264/AVC encoder is shown in Fig. 2.

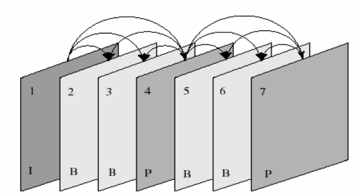
Each frame is divided into macroblocks (MBs). The MBs in the current frame are processed as either intra or inter coded MBs. The encoder consists of a forward path for the encoding and a reverse path for decoding and reconstruction of the current frame. A prediction signal for the MB is calculated using either intra prediction or inter prediction. In intra prediction, the current MB predicted from the neighbouring samples in the current slice which have been already encoded, decoded and reconstructed by the encoder. In inter prediction the prediction signal is obtained through motion estimation and compensation using one (or two) reference picture(s) from the reference picture buffer. The reference picture buffer contains previously coded and decoded pictures that can be selected for inter prediction.



(a) GOP



(b) Keyframes



(c)IBBPBBP Frame Format

Fig. 1 Illustration of Group of Pictures

The prediction MB is then subtracted from the original MB to create a residual MB. The residual MB is transformed, quantized and reordered before entropy coding. Entropy coding is done to remove the statistical redundancy of the data. The entropy coder also processes other information (dotted arrows) necessary for correct decoding of the residual data such as the quantization parameter, MB partition modes, the reference frame(s) used, motion vector information for inter coded MBs and intra mode information for intra coded MBs.

The output of the entropy coder is compressed video bits which are encapsulated in Network Abstraction Layer (NAL) units before transmission or storage. The reverse path (marked by thin arrows) is to reconstruct the lossy coded picture (after quantization) exactly as the decoder. The reconstructed samples of the neighbouring MBs in the current slice may be used for intra prediction of MBs and the current reconstructed picture may be used for inter prediction of future pictures. The exact match between the reconstructions at the encoder and the decoder is essential to avoid any error between the prediction signals at the encoder and the decoder. The prediction signal calculated during the forward path is added to the inverse quantized and inverse transformed residual to create a reconstructed MB (different to the original due to lossy coding). The picture is reconstructed after applying a de-blocking filter in order to reduce the blocking artifacts appearing due to quantization of block transforms.

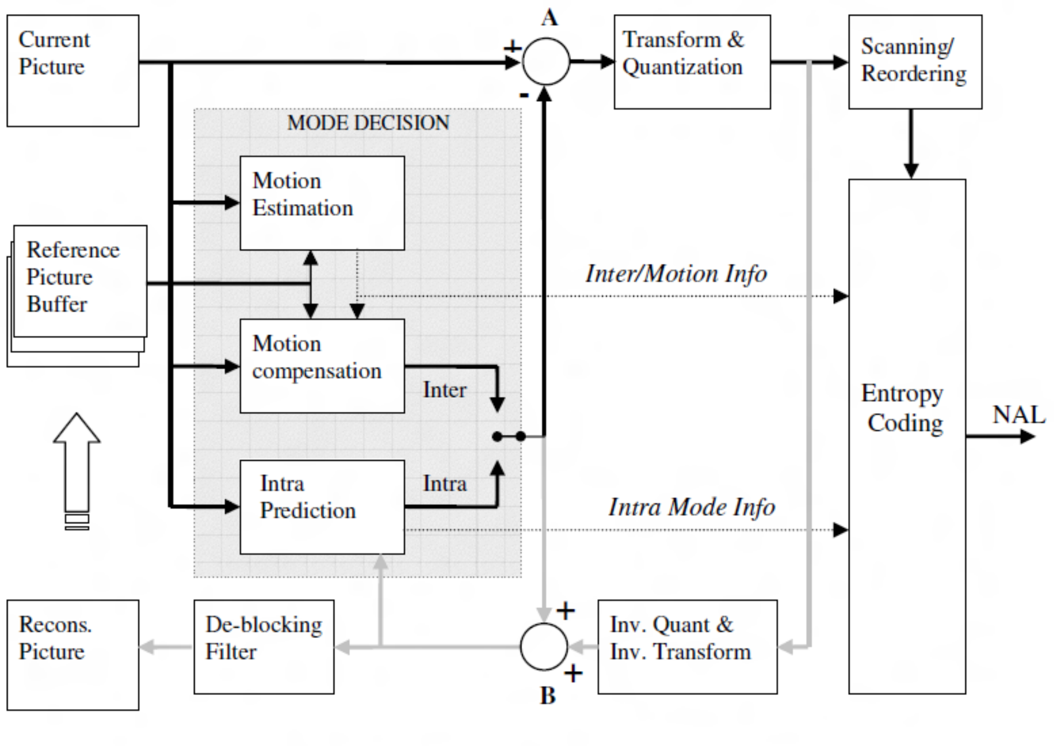


Fig. 2 Block Diagram of H.264/AVC Encoder

The decoder block diagram is shown in Fig. 3. Starting from the right hand side, NAL units are input to the decoder. The NAL units are first entropy decoded to obtain the quantized coefficients and other information necessary to reconstruct the MBs using the quantized coefficients. The coefficients are then rescanned into the luma and chroma arrays of the MB. Similar to the reverse path of the encoder, inverse quantization and inverse transform are applied to the coefficients to produce the residual MB. For inter coded MBs, a prediction is obtained by carrying out motion compensation using the decoded information such as MB partition modes, reference picture(s) and motion vectors. Intra coded MBs are predicted using the decoded intra mode information and previously decoded pixels of neighbouring MBs. The MB is reconstructed by adding the prediction to the residual. The de-blocking filter is applied to reconstruct the current picture. The reconstructed picture is displayed and may also be used as a reference picture for decoding future pictures.

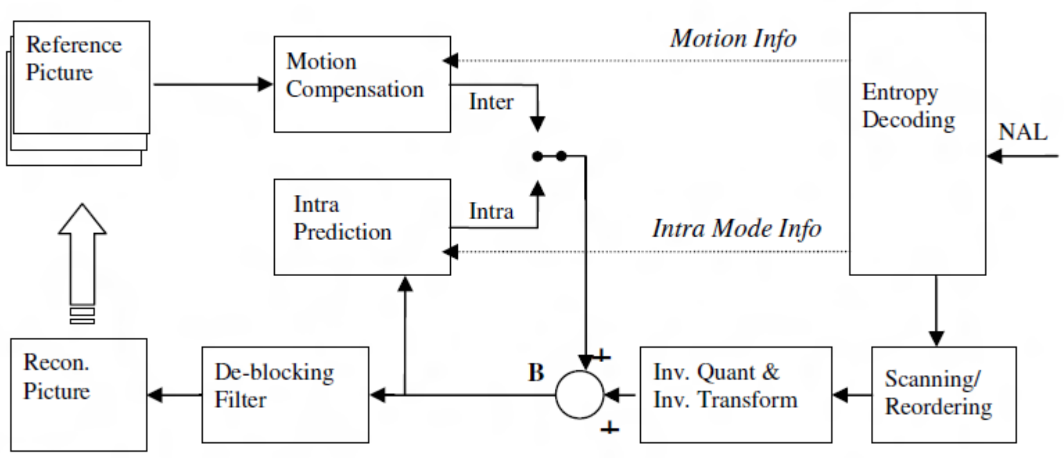


Fig. 3 Block Diagram of H.264/AVC Decoder

3. INTEGRATED FRAMEWORK

The architecture of the integrated framework is shown in Fig. 4. The highlighted blocks in the integrated framework are the modifications done in H.264/AVC architecture. The blocks that are included are: Frame Differencing based Segmentation (FDS), Octagon and Square Search (OSS) pattern and Adaptive GOP (AGOP). The remaining blocks are the same as in H.264/AVC architecture.

The FDS method segments the video sequence before encoding reduces data rate. The OSS method is introduced in intra prediction of H.264/AVC. The size of GOP is made adaptive by using AGOP method.

In Inter mode, prediction for B- or P- frames is formed by motion-compensated prediction from one or multiple reference frame(s) (I- or P- Frames). The current frame is set as either P- frame or B- frame. If it is P- frame, it is used as reference frame for GOP. If it is B-frame, it is segmented using FDS. For motion estimation and compensation, OSS pattern is used for block matching of B- and P- frames.

In Fig. 4, the reference frame is shown as the previously encoded picture F'_{n-1} but the prediction reference for each MB partition may be chosen from a selection of past or future frame that have already been transformed, quantized and reconstructed. The prediction is subtracted from the current block to produce a residual difference block D_n that is transformed and quantized to give X, a set of quantized transform coefficients which are entropy encoded. The entropy-encoded coefficients, together with side information required to decode each block within the MB form the compressed bit stream.

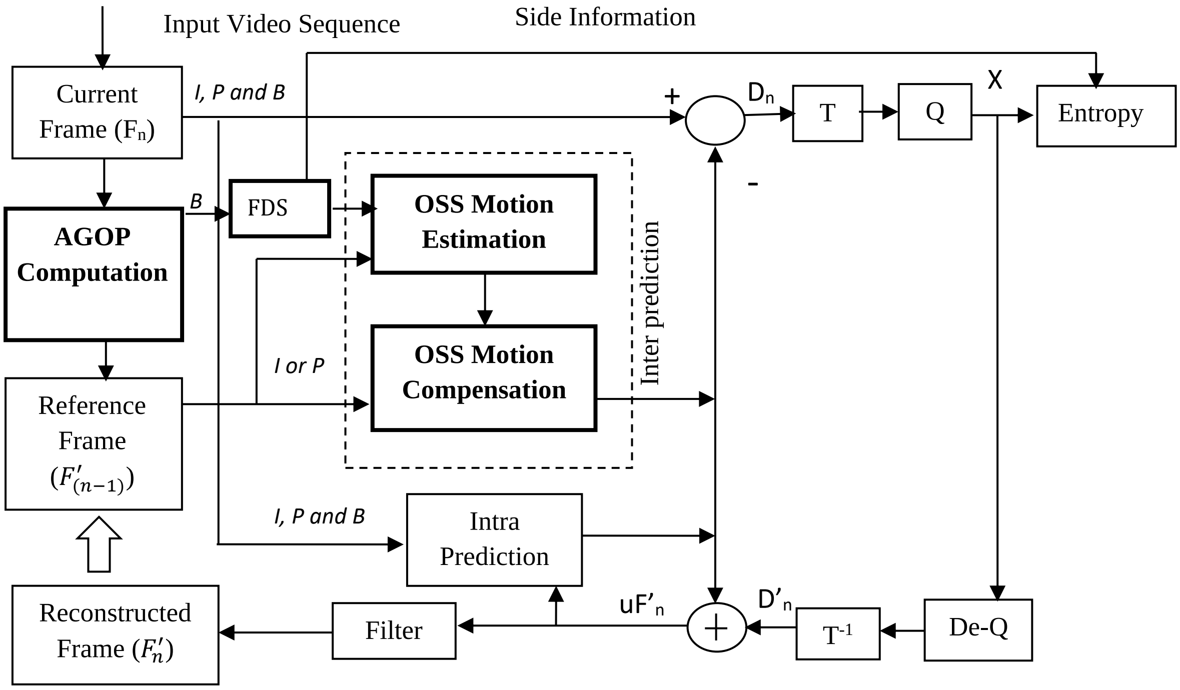


Fig. 4 Integrated Video Coder

T– Transformation, Q - Quantization, T^{-1} – Inverse Transformation, De-Q – De-Quantization

The encoder reconstructs it to provide a reference for further predictions. The coefficients X are de-quantized (De-Q) and inverse transformed (T^{-1}) to produce a difference block D'_n. The prediction block is added to D'_n to create a reconstructed block uF'_n. The reconstructed blocks (uF'_n) are subject to filtering to reduce the blocking distortion. These filtered blocks (F'_n) are combined to generate the reconstructed reference frame (F_n).

The corresponding modification in H.264/AVC decoder is shown in Fig. 5. After decoder reconstruction of the frame, the synthesizer uses this side information to reconstruct the B- frames.

3.1 Frame Differencing based Segmentation (FDS)

The video scenes are classified into static and non-static parts with frame differencing. Segmentation identifies the static regions with no important subjective details and generates coarse masks as well as side information for the synthesizer at the decoder. The synthesizer replaces the static regions by inserting skipped regions.

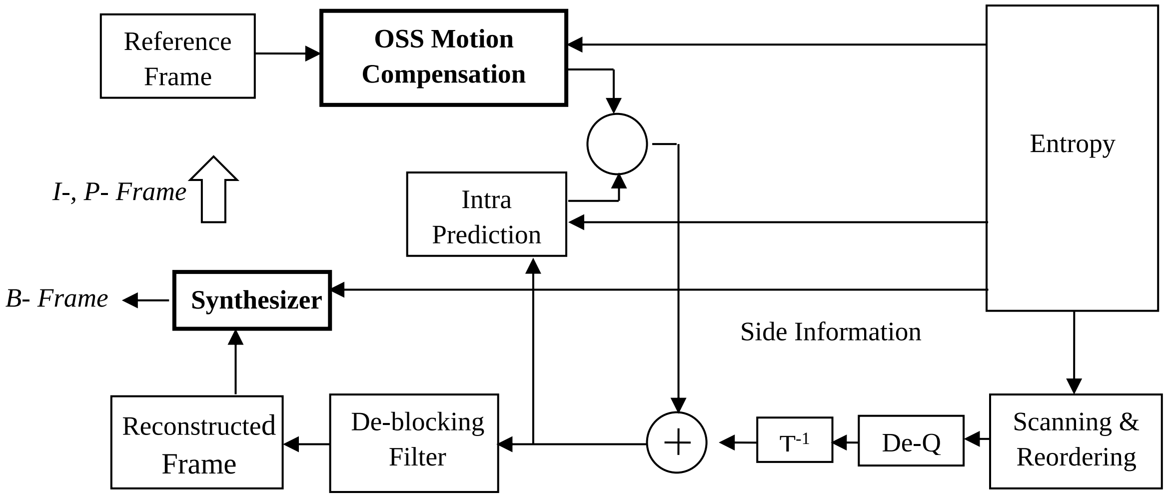


Fig. 5 Integrated Video Decoder

Frame Differencing Technique:

This method uses the image subtraction operator which takes two frames as input and produces segmented output. The two frames are reference frame (I or P frame) and intermediate frames (B frame). Difference is calculated between MB of 1st reference frame and B frame. Again, the difference is calculated between 2nd reference frame and B frame. If the difference is zero for any of the reference frame, then that reference is taken as keyframe for that particular MB. The two frames are reference frame (I or P frame) and intermediate frames (B frame). The output is a segmented frame that is produced after subtracting the intermediate frame pixel values from the keyframe pixel values. It is shown in Fig. 6. The Keyframe which is used for MB skipping will be sent as side information as a control bit. This subtraction is in a single pass.

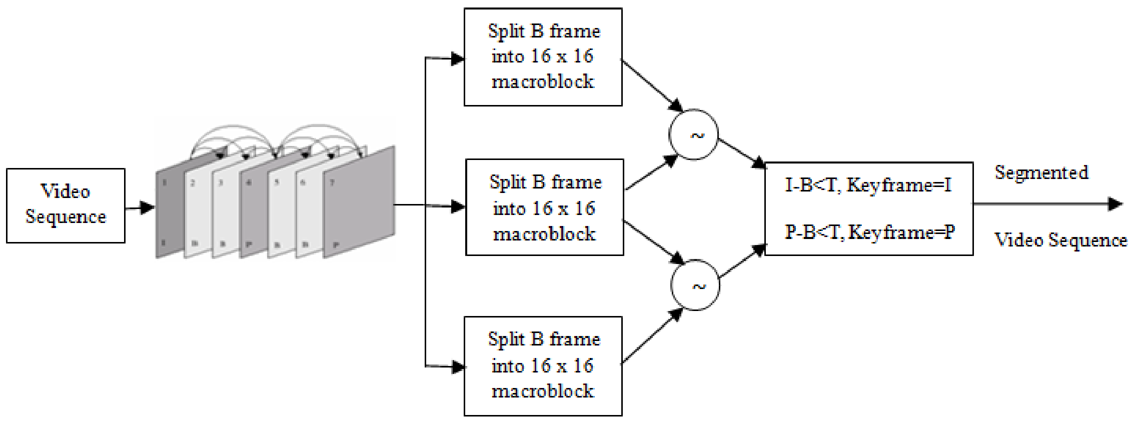


Fig. 6 Illustration of Frame Differencing technique

3.2 Adaptive GOP

In Adaptive GOP method, initially GOP is set to a fixed size. Frames are compared within that GOP using correlation. According to the correlation, GOP is changed within that fixed size. So, there will be no GOP size greater than that fixed size. This method does not calculate any threshold. Hence the time needed to calculate global or local threshold is eliminated.

The input video sequence is given to encoder for encoding I^{th} Frame. GOP Size is determined by the Adaptive GOP algorithm from the input sequence during encoding process. Adaptive GOP algorithm sends side information to the decoder which contains GOP size for the entire video sequence. H.264 Decoder uses this side information to decode the video sequence.

Frame Comparison Logic:

The logic of frame comparison is illustrated in Fig. 7. In this method, no two consecutive frames are compared. Initially, GOP size is fixed to 7. This is because, if the GOP size is set to adaptive, there is a chance to get larger size of GOP which leads to lower PSNR. This method will have adaptive GOP size but not greater than 7.

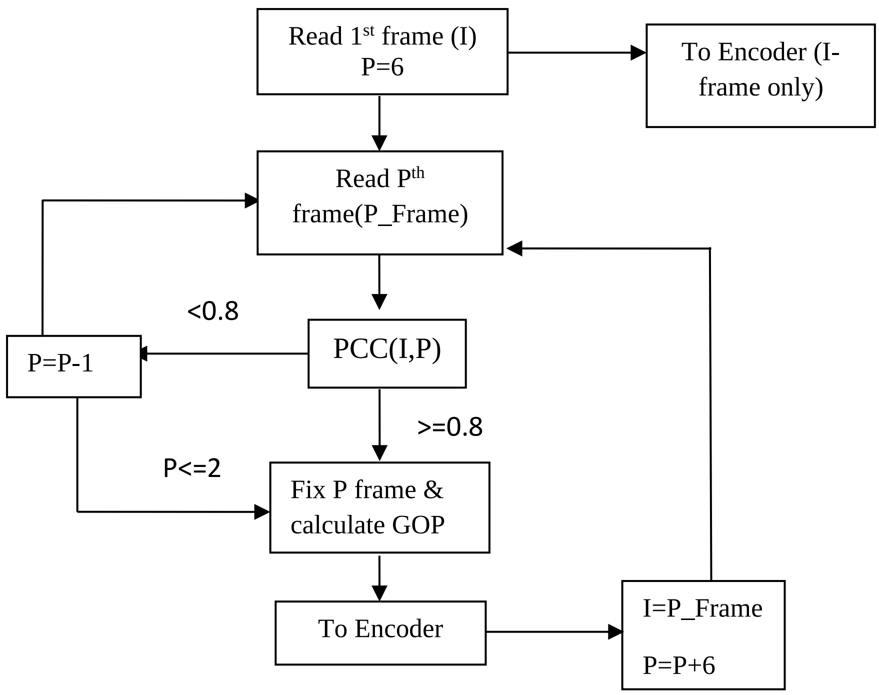


Fig. 7 Frame comparison process

For evaluating the similarity of frame and its prediction, Pearson Correlation Coefficient (PCC) is chosen. It is widely used to measure the similarity of two frames for cut detection. The value of PCC can fall between 0(no correlation) and 1(perfect correlation).

Correlations above 0.80 are considered as really high and lowest values will be determined as cuts. The PCC is expressed as follows.

$$PCC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(f(i,j)-f^{m})(f_p(i,j)-f_p^{m})}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}(f(i,j)-f^{m})^2(f_p(i,j)-f_p^{m})^2}} \quad (1)$$

3.3 Octagon and Square Search Pattern (OSS)

The H.26/4/AVC consists of two subsystems: Motion Estimation and Motion Compensation as shown in Fig. 2 and 3. In Motion Estimation, the video sequence is divided into frames. Each current frame is again subdivided into MBs. For each current MB, the motion vector is estimated using OSS pattern. The Motion Estimation process uses the reference frame for estimating the motion for the current frame. In the Motion Compensation process, each MB in the current frame is reconstructed with its reference frame using motion vectors. The quality is estimated with the original frame and its reconstructed frame.

Each MB in the current frame is checked with candidate MB in a particular range in the reference frame which is called search window. The maximum displacement in the search window is ±7 pixels in both the horizontal and the vertical directions for 16 x 16 MB size. OSS uses 13 search points for large motion and 8 search points for small motion. The search points are shown in Fig. 8.

In the searching procedure of the OSS algorithm, initially, Block Distance Measure (BDM) is calculated with respect to the center point of the MB. If it is zero, there is no motion in the frame. So, the motion vector is set to (0,0) and then the next MB is processed. Otherwise, BDM should be calculated for all points (13 for octagon and 8 for square pattern) in OCTSS pattern. The search pattern is then switched from Octagon pattern to Square pattern. Square search is repeated until the minimum BDM occurs at the center point.

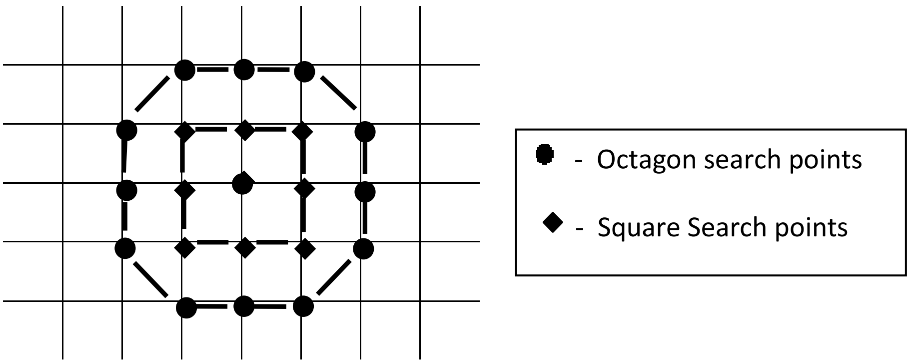


Fig. 8 Octagon and Square Search Pattern

Matching of one macro block with the other is based on the output of the BDM. The candidate MB with minimum BDM is the one that matches closest to the current block. Among the various cost functions, the one that is less computationally expensive is the Mean Absolute difference (MAD) and is given by the formula:

$$MAD = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left|C_{ij} - R_{ij}\right| \quad (2)$$

where M and N is the size of the macro block, C_{ij} and R_{ij} are the pixels being compared in current macro block and reference macro block, respectively.

4. EXPERIMENTAL RESULTS

The performance of the integrated framework is evaluated by comparing it with the recent version of H.264/AVC JM 19.0 [2]. It is also compared with FDS method [3], AGOP method [4] and OSS method [5] which are integrated separately into JM 19.0 with the configuration given below. Experiments were conducted for all types of input video sequences given in Table 1. The video format used for this experiment is QCIF format (144 x 176).

Table 1 Test Video Sequences

Sequence Name	No. of frames	Type
Akiyo	300	Slow
Claire	494	Slow
Grandma	870	Slow
Mother-daughter	300	Slow
Mobile	300	Slow
Miss America	150	Medium
Soccer	150	Medium
Suzie	150	Medium
Foreman	300	Large
Carphone	382	Large

The parameters of H.264/AVC are given in Table 2. The performance of this method can be measured using data rate and Peak Signal to Noise Ratio (PSNR). Table 3 shows the data rate comparison of the integrated framework.

Table 2 Parameters for H.264/AVC Experimental Setup

Parameter	Value
Quantization Parameter	24
Search Range	32
GOP size	4
Motion Estimation	Fast Full Search
Entropy coding	CABAC
Profile	High
Frame Rate	30

Table 3 Data Rate Achieved by H.264/AVC and Individual Integration Methods

Video Sequence Type	Method/ Sequence	Data rate(kb/s)				
		H.264/ AVC	FDS	AGOP	OSS	Integrated Framework
Slow Motion	Akiyo	37.73	34.55	36.8	34.63	34.21
	Claire	49.87	41.36	45.63	45.36	40.5
	Grandma	50.32	49.5	47.25	49.32	49.23
	Mother-daughter	70.78	65.89	70.3	68.12	65.22
	Mobile	606.95	599.01	602.69	605.2	598.97
Medium Motion	Miss America	50.98	49.26	50.21	50.97	48.12
	Soccer	462.62	459.7	461.23	462.18	457.85
	Suzie	138.28	113.25	115.32	130.87	111.26
Fast Motion	Foreman	207.91	193.56	204.77	206.5	192.25
	Carphone	276.03	266.33	274.36	275.68	265.18

From Table 3, it is observed that the integrated framework provides better data rate in comparison with that of all other individual methods. The OSS algorithm reduces data rate only to some extent. The data rate of the integrated framework is reduced due to the use of FDS method. The PSNR achieved by H.264/AVC and the individual integration methods for slow motion video sequences is shown in Table 4.

Table 4 PSNR Gain Achieved by H.264/AVC and the Individual Integration Methods for slow motion video sequences

Method/ Sequence	PSNR (dB)				
	H.264/AVC	FDS	AGOP	OSS	Integrated Framework
Akiyo	43.05	42.1	42.84	42.06	43.15
Claire	43.27	42.25	42.9	42.3	44.12
Grandma	41.2	40.7	40.95	40.96	41.41
Mother-daughter	42.88	41.4	42.78	42.1	42.96
Mobile	38.06	37.14	38.01	37.14	39.12

For slow motion video sequences, the integrated framework achieves better PSNR than all other individual methods and the H.264/AVC. But, for medium and large motion video sequences, the integrated framework achieves negotiable PSNR loss of 0.02dB. Table 5 shows the computation time of H.264/AVC and the individual methods.

Table 5 Computation Time Comparison of H.264/AVC and the Individual Integration Methods

Video Sequence Type	Computation Time (min)				
	H.264/AVC	FDS	AGOP	OSS	Integrated Framework
Slow Motion	185.44	180.24	175.69	162.05	161.05
Medium Motion	66.36	60.5	58.63	53.7	53.68
Fast Motion	140.16	135.26	131.89	124.15	123.97
Average Computation Time	130.65	126.81	121.47	113.3	112.9

The average computation time for the integrated framework is minimal when compared to the H.264/AVC method and all other individual methods. This reduction is due to the use of OSS block matching algorithm. The OSS algorithm reduces computation time when compared to H.264/AVC. In the integrated framework, the computation time is still reduced due to the segmentation before encoding.

5. CONCLUSION

A framework for video compression is designed by integrating FDS, OSS and AGOP into H.264/AVC. Experiments were conducted for all types of video sequences and compared with the recent version of H.264/AVC JM 19.0. The experimental results proved that the integrated framework is better than H.264/AVC in terms of computation time, PSNR and data rate. The integrated framework reduces computation time and data rate for all types of video sequences with very negligible loss in PSNR for medium and large motion video sequences.

References:

[1] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6). H.264/14496-10 AVC Reference Software Manual (revised for JM 19.0). June 2015.

[2] Miaohui Wang, King Ngi Ngan, and Long Xu. Efficient H.264/AVC Video Coding with Adaptive Transforms. IEEE Transactions on Multimedia. June 2014, Vol. 16, No. 4, pp. 933-946.

[3] Sowmyayani, S. and Rani, P.A.J., 2014. Block based Motion Estimation using Octagon and Square Pattern. International Journal of Signal Processing, Image Processing and Pattern Recognition, 7(4), pp.317-324.

[4] Sowmyayani, S., Rani, J. and Arockia, P., 2014. Adaptive GOP structure to H. 264/AVC based on Scene change. ICTACT Journal on Image & Video Processing, 5(1), pp. 868-872.

[5] Sowmyayani, S. and Rani, P.A.J., 2016. Frame differencing-based segmentation for low bit rate video codec using H. 264. International Journal of Computational Vision and Robotics, 6(1-2), pp.41-53.