International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:9, No:8, 2015

Enhancing the Performance of H.264/AVC in Adaptive Group of Pictures Mode Using Octagon and Square Search Pattern

S. Sowmyayani, P. Arockia Jansi Rani

Abstract**—**This paper integrates Octagon and Square Search pattern (OCTSS) motion estimation algorithm into H.264/AVC (Advanced Video Coding) video codec in Adaptive Group of Pictures (AGOP) mode. AGOP structure is computed based on scene change in the video sequence. Octagon and square search pattern block-based motion estimation method is implemented in inter-prediction process of H.264/AVC. Both these methods reduce bit rate and computational complexity while maintaining the quality of the video sequence respectively. Experiments are conducted for different types of video sequence. The results substantially proved that the bit rate, computation time and PSNR gain achieved by the proposed method is better than the existing H.264/AVC with fixed GOP and AGOP. With a marginal gain in quality of 0.28dB and average gain in bitrate of 132.87kbps, the proposed method reduces the average computation time by 27.31 minutes when compared to the existing state-of-art H.264/AVC video codec.

Keywords**—**Block Distortion Measure, Block Matching Algorithms, H.264/AVC, Motion estimation, Search patterns, Shot cut detection.

I. Introduction

IN the communication technology, transferring image, video or other data is an important need. All these data need more space in storage devices. In particular, a video sequence with higher resolution takes much of the space in storage devices. Hence it is essential to store the video sequence in compressed format. H.264/AVC is one of the significant video compression standards in the video compression series. In the past half-decade, ISO/IEC MPEG high-performance video coding (HVC) and ITU-T VCEG groups, and the establishment of the Joint Collaborative Team (JCT) aimed at a new generation of video compression standardization process over the current H.264/ AVC [1]. The main aim of the team is to substantially reduce the bit-rate over the current H.264/AVC.

Spatial and temporal redundancies are present in the video sequence. The main task of video compression is to reduce those redundancies while preserving the quality and entropy. Intra and inter prediction methods are used respectively to reduce spatial and temporal redundancy. Intra Prediction is used to find the redundancy between macroblocks within the frame. Inter Prediction aims at reducing the redundancy among frames.

S. Sowmyayani and P. Arockia Jansi Rani are with Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Tirunelveli (e-mail: sowmyayani@gmail.com, jansimsuniv@gmail.com).

This paper contributes to the inter prediction coding method. In this method, first the video sequence is divided into Group of Pictures. In the GOP structure, three different picture types namely I- (Intra), P- (Predicted) and B- (Bi-directionally Predicted) frames are used. I-frame is the first frame in the video sequence which uses intra prediction only. P-frame is the predictive coded picture with inter prediction from the previous I or P frame. B- frame is the bidirectional coded picture with inter prediction from the previous I- or P-frame and future P-frame [2], [3]. There are several methods to decide a GOP structure [4], [5].

In the conventional H.264/AVC, GOP size is fixed. Fixed GOP structures prevent encoders from improving coding efficiency for B frames, since P frames are not chosen according to the motion. In [6], GOP is adaptive according to the changes in the scene. The AGOP structure is calculated using Pearson Correlation Coefficient. The advantage of using the algorithm in [6] is that it achieves better quality and higher bit rate.

In the Inter prediction method, motion estimation algorithm is used to remove temporal redundancy. Motion in the video sequence [7] is estimated using block matching algorithms. B-frames are compared with P-frames in the video sequence using several block matching algorithms.

The block matching algorithms that have been implemented earlier are Full Search (FS) [8], Three Step Search (TSS) [9], New Three Step Search (NTSS) [10], Simple and Efficient TSS (SES) [11], Four Step Search (4SS) [12], Cross Search Pattern [13], Diamond Search (DS) [14], Hexagon-based Search (HEXS) [15]. Several recent algorithms are Unsymmetrical-cross Multi-Hexagon-gird Search (UMHexagonS) [16], Cross Diamond Search (CDS) [17], Cross Hexagon Search (CHS) [18], Enhanced Hexagon Search (EHS) [19], hexagon-diamond search (HDS) [20], Octagon-based Search (OCTBS) [21], Octagon and Triangle Search (OCTTS) [22] etc.

All these algorithms take more computational time [23]. This makes H.264/AVC worse. Hence to reduce computational complexity, we have developed a block matching algorithm using Octagon and square pattern (OCTSS) [24]. The Octagon search pattern is used to find large motion while square search pattern is used to find small motion in the video sequence. Our approach reduces computation time much better than other motion estimation algorithms.

In this paper, we propose to integrate our OCTSS algorithm for motion estimation [24] into the existing H.264/AVC video codec. In addition, we have analyzed the role of GOP structure by comparing the performance of AGOP with that of fixed GOP. We found that the proposed AGOP based H.264/AVC video codec integrated with OCTSS outperforms the state-of-art video codec. The proposed codec achieves better PSNR, bitrate and low computation time.

The rest of the paper is organized as follows. Section II describes the overall system architecture of the proposed method. The AGOP algorithm is discussed in Section III. Section IV elaborates the Octagon and Square Search algorithm. Finally, Section V demonstrates the experimental results followed by conclusion in Section VI.

II. System Architecture

Fig. 1 shows the overall architecture of the proposed method. The detailed description about H.264/AVC video codec is given in [3].

Any frame in the video sequence (F_n) is processed in terms of macro-blocks (MB). Each MB is encoded either in the intra or inter mode and, for each block in the MB, a prediction is formed based on reconstructed frame. In Intra mode, prediction is formed from spatially neighboring samples in the current frame that have previously been transformed, quantized and reconstructed (uF'_n as mentioned in Fig. 1). In the encoding process, neighboring samples are used for Intra prediction, which is simultaneously conducted at the encoder and decoder using the transmitted Intra prediction side information. 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) selected using our adaptive GOP algorithm.

In Fig. 1, the reference frame is shown as the previously encoded picture F'_{n-1} but the prediction reference for each MB partition (in inter mode) may be chosen from a selection of past or future frame (in display order) 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. The encoder decodes (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).

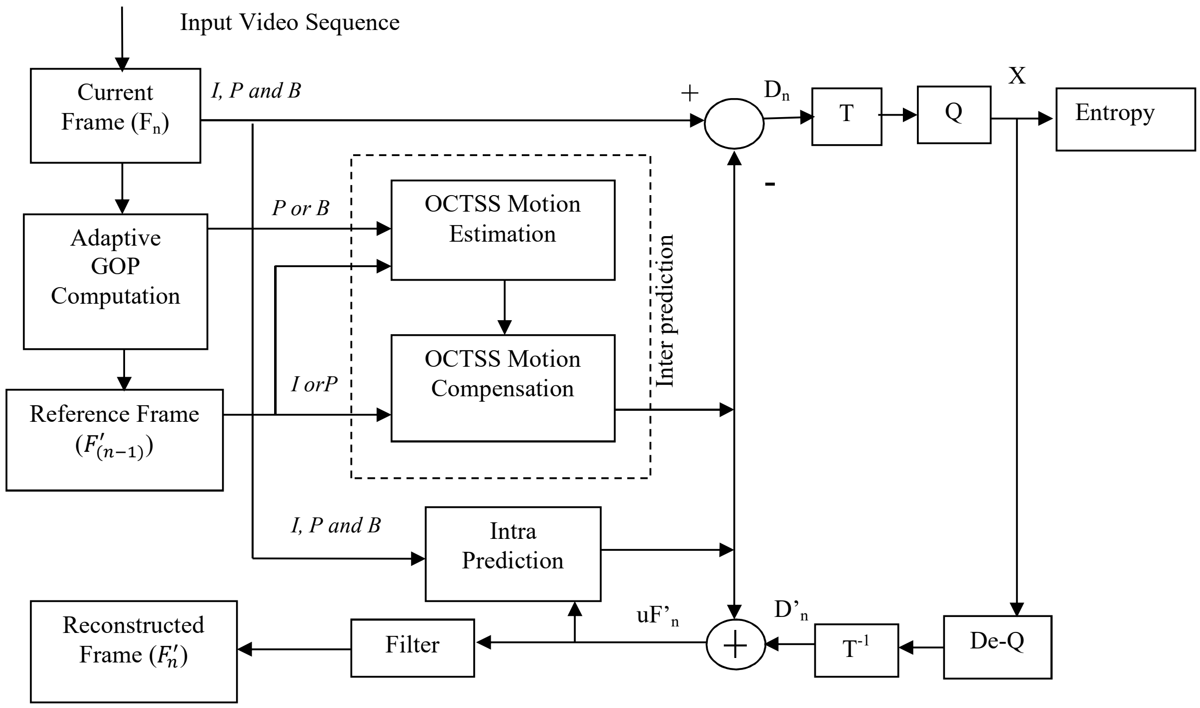

Fig. 1 Proposed video codec illustrating the Integration of Adaptive GOP and OCTSS in H.264/AVC
T – Transformation, Q - Quantization, T^{-1} – Inverse Quantization, De-Q – Quantization

III. Adaptive GOP Structure

This AGOP structuring process employs Pearson Correlation Coefficient (PCC) [25] which is given in (1) to evaluate the similarity among frames. The procedure to compute adaptive GOP structure is described below:

Step1. Let the first frame in the GOP be the Predictive frame

(P- frame) and the last frame in the GOP be the Test frame (T-frame). Initialize the GOP size to 7.

Step2. Compare P- frame with the T-frame using (1). If the correlation between them is above a threshold (T), go to step 5.

Step3. If the correlation is below the threshold (T), then the previous frame in the GOP is set as the T- frame.

Step4. If the T- frame refers to second frame in the GOP, go to step 5; otherwise go to step 2.

Step5. The size of the GOP is set to be number of frames between P- frame and T- frame. Now, the T- frame is selected as the P- frame.

Step6. The P-frame and the consecutive six frames constitute the next GOP structure.

Repeat the algorithm until all frames in the video sequence are processed.

After setting each GOP size, it is given to the motion estimation process.

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

where f, f_p represent pixel intensities of the current frame and the P-frame respectively. M, N are the number of rows and columns in the frame. f^m, f_p^m are mean pixel intensities of the current frame and the P- frame.

IV. Octagon And Square Pattern

We have used our OCTSS algorithm for the block matching process. We have proved that our OCTSS algorithm [24] performs better than other existing algorithms. Block matching uses a value called BDM to rate similarity between two macroblocks. The basic idea is to calculate the mean for the absolute differences of the luminance of pixels located at the same position in the two macroblocks. The Mean Absolute Difference (MAD) which is given in (2) is the least cost function for calculating BDM. OCTSS uses 13 search points for fast motion and 8 search points for slow motion video sequences. The search points are shown in Fig. 2.

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

where M and N is the size of the macroblock, C_{ij} and R_{ij} are the pixels being compared in current macroblock and reference macroblock, respectively. This process is explained below:

Step1. Calculate BDM for center point. If it is zero, go to Step 4

Step2. Calculate BDM for all points (13 for octagon+8 for square).

Step3. Move the center to the point which has minimum BDM.

Step4. The search pattern strategy is square from this step onwards. Repeatedly calculate BDM and move on to the point with minimum BDM until the center point with minimum BDM is reached.

Step5. Stop the search.

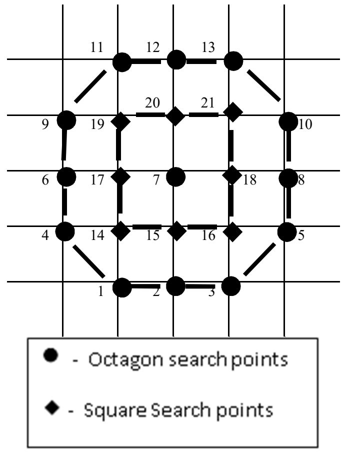

Fig. 2 Octagon and Square Search Pattern

This method is integrated with H.264/AVC JM 19.0 reference software [26] in adaptive mode. The description of H.264 video codec is given in [2]. Following conditions of H.264/AVC encoder were performed for each simulation:

- QP (quantization parameter): 24
- Total number of references: 2
- GOP Size: Adaptive
- Search range: 32
- Entropy coding method: Content-based Adaptive Binary Arithmetic Coding (CABAC)
- Motion Estimation Scheme: Fast Full Search

V. Experimental Results

Experiments were conducted for different types of video such as slow, medium and fast motion video sequences. The proposed method uses B-frames for the experiments which are not used in [27]. The use of B-frames additionally reduces the bitrate [6].

Following are the characteristics of the input video sequences used for experiments:

- Format: YUV
- Size: 144 x 176
- Type: Quad Common Intermediate Format (QCIF 4:2:0 sub-sampling)

TABLE I
Characteristics of Input Video Sequence

Sequence Name	Type	Number of frames
Akiyo	Slow	300
Carphone	Fast	382
Claire	Slow	494
missa	Medium	150
Suzie	Medium	150

The characteristics of the input video sequence are shown in Table I.

The proposed method is compared in terms of data rate, PSNR and computation time. The proposed method is compared with [6] and with the conventional H.264/AVC. The data rate is calculated as:

$$DR = (N_I + n_1 N_B + n_2 N_P) \times frame\_rate \quad (3)$$

where N_I, N_B, N_P are total number of bits used for I-frame, B-frames, P-frames (kb) respectively, n_1, n_2, number of B-frames and P-frames respectively, frame_rate is the number of frames per second.

Table II shows the data rate achieved by the proposed method, fixed GOP H.264/AVC (GOP size = 4) and AGOP based codec. In order to compare the proposed method, various fixed GOP sizes are used in [28]. The GOP size of 4 is selected as it achieves better PSNR than other sizes [6]. As AGOP structure is implemented based on scene change, number of B-frames will be more when compared to H.264/AVC [6]. Hence the data rate achieved by the AGOP codec is less as compared to that of the fixed GOP H.264/AVC. The proposed method results achieve negligibly small improvement in the data rate. This is shown in bar charts in Fig. 3.

TABLE II
DATA RATE ACHIEVED BY FIXED GOP H.264/AVC, AGOP CODEC AND THE PROPOSED METHOD FOR VARIOUS SEQUENCES

Method/ Sequence	Data Rate(kb/s)		
	Fixed GOP H.264/AVC	AGOP Codec	Proposed Method
Akiyo	552.02	428.18	412.23
Carphone	936.65	723.12	709.37
Claire	419.99	301.9	294.5
Missa	478.75	498.6	484.1
Suzie	483.53	417.21	406.4
Average Bitrate	**574.188**	**473.802**	**461.32**

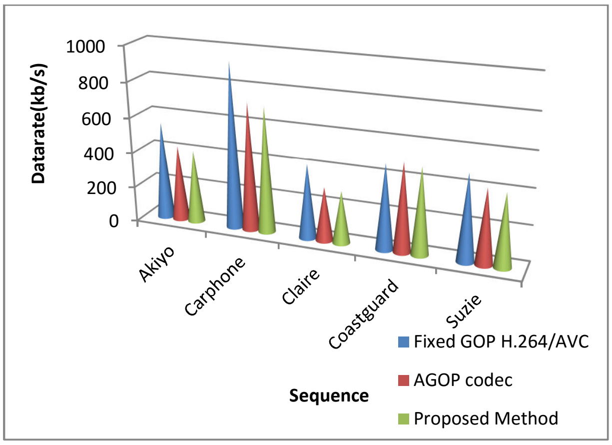

Fig. 3 Bar chart as a function of Data rate achieved by fixed GOP H.264/AVC, AGOP codec and the proposed method for various video sequences

The mathematical formulae for PSNR is

$$PSNR = 10\ log_{10}\left[\frac{(2^b-1)^2}{MSE}\right] \quad (4)$$

where b is the number of bits in a pixel. For 8-bit uniformly quantized video sequence, b = 8. The higher the PSNR value, the better the quality of the reconstructed frame.

Mean Square Error is given by

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(f(M,N) - f'(M,N)\right)^2 \quad (5)$$

where f(M,N) represents the current frame and f' (M,N) is the reconstructed frame with frame size M x N. The PSNR achieved by the H.264/AVC, AGOP codec and the proposed method are shown in Table III. It is observed that the PSNR achieved by the proposed method is appreciably better than the PSNR achieved by H.264/AVC and AGOP codecs.

TABLE III
PSNR ACHIEVED BY FIXED GOP H.264/AVC, AGOP CODEC AND THE PROPOSED METHOD FOR VARIOUS SEQUENCES

Method/ Sequence	PSNR(dB)		
	Fixed GOP H.264/AVC	AGOP Codec	Proposed Method
Akiyo	40.8	40.9	41.25
Carphone	38.51	38.59	38.67
Claire	43.67	43.78	43.9
Missa	37.91	38.14	38.2
Suzie	41.7	41.86	41.98
Average PSNR	**40.518**	**40.654**	**40.8**

The average gain in dB shall be computed as the difference in average PSNR between the existing and the proposed method. From Table III, it is found that the average gain is 0.28 dB and 0.15 dB for the proposed method when compared to fixed GOP and AGOP based codecs respectively.

The average computational time taken by fixed GOP H.264/AVC and AGOP codec is more than one and half hours whereas, for the proposed method, it is less than one hour. This reduction is due to the integration of OCTSS algorithm in H.264/AVC. Table IV shows the average computational time taken by fixed GOP H.264/AVC, AGOP codec and the proposed method.

TABLE IV
AVERAGE COMPUTATIONAL TIME TAKEN BY FIXED GOP H.264/AVC, AGOP CODEC AND THE PROPOSED METHOD FOR VARIOUS SEQUENCES

Method/ Sequence	Computational Time(min)		
	Fixed GOP H.264/AVC	AGOP Codec	Proposed Method
Akiyo	71	65.2	48.26
Carphone	112	98.7	84.9
Claire	58	56.21	39.1
Missa	63	59.6	45.4
Suzie	90	84.3	76.8
Average	**78.8**	**72.802**	**58.892**

VI. CONCLUSION

The proposed method integrates OCTSS algorithm with H.264/AVC in adaptive GOP mode. In the earlier research, the authors have introduced the OCTSS algorithm for block matching and also developed a new adaptive method of grouping frames. In this paper, the aforementioned contributions have been integrated into the existing H.264/AVC video codec. AGOP contributes to the reduction in data rate and OCTSS contributes to the reduction in computation time. The performance of the proposed method is compared with H.264/AVC in fixed GOP and AGOP mode. From the experiments, it is proved that the quality of the video sequence is sustained while reducing the data rate and computation time. In future, any method of segmentation can be integrated to further reduce the data rate.

REFERENCES

[1] Nam Ling, “Expectations and Challenges for Next Generation Video Compression”, 5th IEEE Conf. on Industrial Electronics and Applications, pp. 2339-2344, 2010.

[2] Wiegand, T., Sullivan, G. J., Bjontegard, G., et al., “Overview of the H.264/AVC video coding standard”, IEEE Trans. on CSVT, vol. 13, no. 7, pp. 560-576, 2003.

[3] Sullivan, G. and Wiegand, T., “Video compression - from concepts to the H.264/AVC standard”, Proc. of the IEEE, vol. 93, pp. 18-31, 2005.

[4] Jungwoo Lee et al., “Rate-Distortion Optimized Frame Type Selection for MPEG Encoding”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 3, 1997.

[5] Jiro Katto, “Mathematical analysis of MPEG compression capability and its application to rate control”, Int. Conf. of Image Processing-95, vol. 11, pp. 555-558, 1995.

[6] Sowmyayani, S, Arockia Jansi Rani, P., “Adaptive GOP structure to H.264/AVC based on Scene change”, ICTACT journal on image and video processing: special issue on video processing for multimedia systems, vol. 5, no. 1, pp. 868-872, 2014.

[7] Shilpa Metkar, Sanjay Talbar, “Motion Estimation Techniques for Digital Video Coding” Springer Briefs in Applied Sciences and Technology, Computational Intelligence, 2013

[8] Lin, Y. C. Tai, S. C., “Fast full-search block-matching algorithm for motion-compensated video compression”, IEEE Trans. Communication, vol. 45, no. 5, pp. 527–531, 1997.

[9] Jong-Nam Kim and Tae-Sun Choi, “A Fast Three Step Search Algorithm with Minimum Checking Points”, Proc. of IEEE conf. on Consumer Electronics, vol. 2, no. 4, pp.132-133, 1998.

[10] Li, R., Zeng, B., Liou, M. L., “A new three step search algorithm for block motion estimation”, IEEE Trans. Circuits Syst. Video Technol., vol.4, no.4, pp. 438–442, 1994.

[11] Jianhua Lu, Ming L. Liou, “A Simple and Efficient Search Algorithm for Block-Matching Motion Estimation”, IEEE Trans. Circuits And Systems For Video Technology, vol. 7, no. 2, pp. 429-433, 1997.

[12] Po, L.M., Ma, W.C., “A novel four-step search algorithm for fast block motion estimation”, IEEE Trans. Circuits Syst. Video Technol., vol. 6, no. 3, pp. 313–317, 1996.

[13] Ghanbari, M., “The Cross-Search Algorithm for Motion Estimation”, IEEE Trans. on Communications, vol. 38, no. 1, pp. 950-953, 1990.

[14] Zhu, S., Ma, K. K., “A new diamond search algorithm for fast block matching motion estimation”, IEEE Transactions on Image Processing, vol. 9, pp. 287–290, 2000.

[15] Zhu, C., Lin, X., Chau, L. P., “Hexagon-based search pattern for fast block motion estimation”, IEEE Trans. Circuits Syst. Video Technol., vol. 12, no. 5, pp. 349–355, 2002.

[16] Faizul Haldi Jamil, Ali Chekima, Rosalyn R, et al., “BMA Performance of video coding for motion estimation”, IEEE Trans. Int. Conf. on intelligent system modelling and simulation, 2012.

[17] Cheung, C. H., Po, L. M., “A novel cross-diamond search algorithm for fast block motion estimation”, IEEE Trans. on Circuits and Systems for Video Technology, vol. 12, pp. 1168–1177, 2002.

[18] Kamel Belloulata, Shiping Zhu, Zaikuo Wang, “A Fast Fractal Video Coding Algorithm Using Cross-Hexagon Search for Block Motion Estimation”, Int. Scholarly Research Network.

[19] Bei-li Zou, Cao Shi, Can-Hui Xu, et al., “Enhanced Hexagonal-Based Search Using Direction-Oriented Inner Search tor Motion Estimation”, IEEE Trans. on circuits and systems for video technology, vol. 20, no. 1, 2010.

[20] Cheung, C. H., Po, L. M., “Novel cross-diamond-hexagonal search algorithms for fast block motion estimation”, IEEE Trans. Multimedia, vol. 5, no. 1, pp. 16–22, 2005.

[21] Chau, L. P., Zhu, C., “A fast octagon –based search algorithm for motion estimation”, J. Signal Process, pp. 671–675, 2003.

[22] Chunjiang Duanmu, Yu Zhang, “A New Fast Block Motion Algorithm Based on Octagon and Triangle Search Patterns for H.264/AVC”, Int. Journal of Digital Content Technology and its Applications, vol. 6, no.10, 2012.

[23] Sangeeta Mishra, S., Chittaranjan Pradhan, Alka Singh, “Comparative Study of Motion Estimation Techniques in Video”, Int. Journal of Computer Science and Information Technologies, vol. 5, no. 3, pp. 2982-2989, 2014.

[24] Sowmyayani, S., Arockia Jansi Rani, P., “Block based Motion Estimation using Octagon and Square Pattern”, Int. Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 7, no. 4, pp. 317-324, 2014.

[25] Lenka Krulikovsk´a and Jaroslav Polec, “GOP Structure Adaptable to the Location of Shotcuts”, Int. Journal of Electronics and Telecommunications, vol. 58, no. 2, pp. 129-134, 2012.

[26] 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.

[27] Lenka Krulikovska, Jaroslav Polec, and Michal Martinovic, “Adaptive Group of Pictures Structure Based On the Positions of Video Cuts” World Academy of Science, Engineering and Technology International Journal of Computer, Information Science and Engineering, vol. 7, no. 7, 2013.

[28] Zatt, B., Porto, M., Scharcanski, J., et al., “Gop structure adaptive to the video content for efficient H.264/AVC encoding”, Proc. of 2010 IEEE 17th International Conference on Image Processing, pp.3053-3056, 2010.