International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 6, Number 1 (2013), pp. 57-65

http://www.irphouse.com

# Complexity Control Using Inter Mode Decision In H.264/AVC Baseline Profile

**Patharwalkar Shilpa S[1] and Prof Jadhav P M[2]**

*[1&2]University of Pune, Department of Electronics &Telecommunication Engg, Maharashtra Institute of Technology, Kothrud, Pune, Maharashtra, India*
*[1]E-mail:shilpa.sp16@gmail.com*
*[2]Associate Professor, University of Pune*
*E-mail: jadhav.parul@gmail.com*

**Abstract**

H.264/AVC, the latest video coding standard, outperforms the previous coding standards. It achieves better video compression since it supports new features of video coding such as a large number of intra and inter prediction macro block (MB) modes. H.264/AVC adopts rate-distortion optimization (RDO) technique to obtain the best intra and inter prediction, while maximizing visual quality and minimizing the required bit rate. However, full RD cost calculation for all intra-prediction modes and exhaustive searches for optimal motion vectors for all block sizes increase computational complexity considerably with the number of prediction modes allowed. In order to reduce the complexity, an efficient fast block mode decision algorithm is proposed. The algorithm is based on the spatial correlation of neighboring block and there priority order of occurrence which terminates RDO calculation and reduces the encoding time of video frame with allowable degradation in video quality. Experimental results show that implemented algorithm reduces the encoding time by around 20% under different bandwidth with tolerable PSNR loss of around 0.2db to 0.5 db by using x264 software encoder.

**Keywords:** H.264/AVC, Macroblock , Mode Decision, Rate Distortion Optimization, Motion Estimation

# 1. Introduction

Digital video technology is enabling and generating ever new applications with a broadening range of requirements regarding basic video characteristics such as spatiotemporal resolution, chroma format, and sample accuracy. Application areas today range from videoconferencing over mobile TV and broadcasting of standard/high-definition TV content up to very-high-quality applications such as professional digital video recording or digital cinema/large-screen digital imagery. Prior video coding standards such as MPEG2/H.262, H.263, and MPEG4 Part 2 are already established in parts of those application domains.[1]H.264/MPEG4 Advanced Video Coding (AVC), as the latest entry of international video coding standards, has demonstrated significantly improved coding efficiency, substantially enhanced error robustness, increased flexibility and scope of applicability relative to its predecessors. But this all is at the cost of increased computational complexity. So controlling the complexity is useful when the encoded video must be produced within some timeframe e.g. videoconferencing on mobile phones. If there are no such requirements, it does not matter how much (computational) time encoding takes. In that case, even if the amount of available computational resources changes there is no point in adapting to it.

## 1.1 Complexity

Complexity is the time needed for a computer to execute a program. It can be measured in terms of processor instructions, clock cycles or seconds. Controlling the complexity is identifying and modifying an area of video encoder that achieves complexity. The requirement is that one must be able to reduce the amount of instructions that are executed, while still producing an output bit stream that is valid according to the H.264 format specification(i.e., can be decoded).[3] This cannot be achieved by simply aborting to process a frame. Instead, a part of the encoder must be selected and modified that performs operations which are intended to improve encoding speed but at the cost of degradation of quality i.e. distortion is sure to introduce. For H.264 some methods that can be used to control complexity are Motion Estimation (ME), modifying the Input Picture Stream's Spatial /Temporal Resolution, restricting the number of Hadamard Transforms, restricting the search for intra/inter macro block mode predictions and reduction in no of RDO calculations etc. There are some benefits of complexity control like Graceful Degradation i.e. if the resources needed to do an optimal job are not available, the software can still accommodate the user by providing reduced (but still acceptable) services[6] . In a real-time system, deadlines may be placed on a video encoding task. A complexity controlled encoding task can be made to meet its deadline by default, as the deadline can be translated into a complexity budget that the encoder can be constrained to. In this algorithm, the encoding side of video processing is focused for reducing the computational complexity by targeting the MB mode decision approach by early terminating RDO.

The algorithm is implemented using H.264/AVC open source encoder x264. Its development started in 2003, and it has been used in many popular applications like ffdshow, ffmpeg and MEncoder. In a recent study, x264 showed better quality than several commercial H.264 encoders. The high performance of x264 is attributed

to its rate control, motion estimation, macro block mode decision, and quantization and frame type decision algorithms.[3] The output bit rate and video quality of video encoder depends on several coding parameter such as coding mode and quantization parameter (QP) value.[6]
The rest of the paper is organised as follows, section II of this paper is an overview of inter prediction block mode used in H.264/AVC encoder, section III is an elaboration on proposed algorithm .Section IV shows the experimental results performed with x264 encoder .Finally section V concludes.

## 2. Overview of Inter Prediction Block Mode

### 2.1 H.264 Inter Mode Macroblock Structure

H.264/AVC video coding standards provides three slices inter-mode decision. These slices are I, P, and B slices. I slice do not refer to previous frames, but simply conducts intra frame prediction, which requires more coding bits than the others. P slice refer to the encoded frames in order to perform motion compensation. Furthermore, B slice is similar to P slice, but refer to both previous and current frames. The luminance part of MBs can be divided into one of four block modes, which may be 16x16, 16x8, 8x16, or 4x4. As the block mode is 8x8, the 8x8 sub block can be further divided into one of four block modes, which may be 8x8, 8x4, 4x8, and 4x4. Thus, there are totally 259 segmentation variations for each MB. Another mode which is present in H.264/AVC is SKIP MB. SKIP mode will be directly replaced with the co-located MB in the previous frame. For SKIP mode The optimum motion compensated block size is 16 x16 and The reference frame can be only the adjacent previous frame. Whereas the motion vector is (0, 0) or motion vector is identical to the predictive motion vector and the transformed coefficients will be quantized to all zeros.[2]The inter prediction scheme with variable block size is also called tree structural motion compensation as Fig. 1

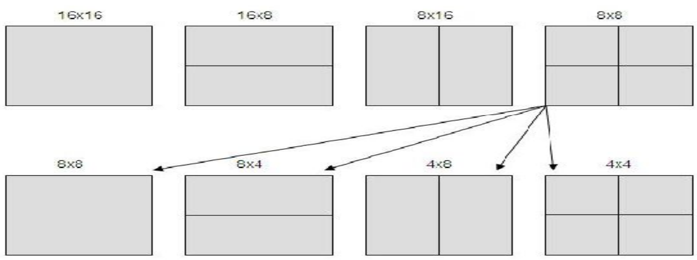



Fig 1. H.264 Inter Block Mode for Tree Structured Motion Compensation

### 2.2 Macroblock Correlation

The proposed algorithm approximates mode-decisions using MB correlation, so as to eliminate the need for brute-force calculations. A good approximation will ensure speedy encoding to H.264/AVC with an allowable sacrifice in the resulting video quality and PSNR (Peak Signal to Noise Ratio). Since there is high correlation between inter blocks, the characteristics of the adjacent blocks are often used to decide the block mode for the current block. In case the upper block is *8x16*, there might exist a vertical edge cross the upper and the current block, i.e. both blocks share the identical block mode with high probability. Similarly, when the left block is 16x8, a horizontal edge might cross the left block and current block, so that the current block mode would be probably the same as the left block mode as shown in Fig. 2.

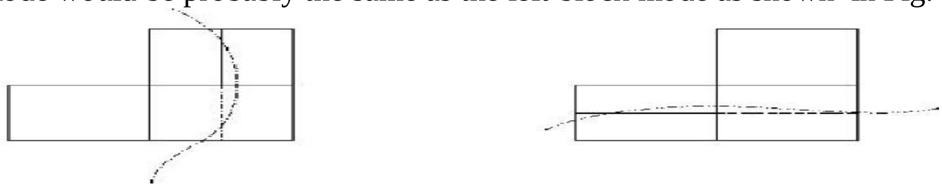


Fig 2 The Block Mode Correlation between Adjacent Blocks [3]

Many standard video sequences have been tested to explore the relationship [3]. Mostly the best mode of current MB is as same as one of the modes of neighboring MBs in most cases .For the implemented algorithm this is achieved by following a priority table of prediction modes of adjacent MBs. Mode decision based on adjacent blocks would first gather statistics of characteristics of the neighboring blocks selected, and then decide a possible block mode based on the statistics. Many standard video sequences have been tested to explore the relationship .Mostly the best mode of current MB is as same as one of the modes of neighboring MBs in most cases. Following Table 1 shows the optimum mode of the current MB for each combination of left MB and up MB. Here M (l) and M (u) indicate the mode of left and up MB respectively.

Table 1 The Optimum Modes of Different Combination of Neighboring Mode[3]

| **M(l)/M(u)** | **Mode 0** | **Mode 1** | **Mode 2** | **Mode 3** | **Mode 8** | **Mode 9** | **Mode 10** |
|---|---|---|---|---|---|---|---|
| **Mode 0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Mode 1** | 0 | 1 | 1 | 1 | 8 | 9 | 1 |
| **Mode 2** | 0 | 1 | 1 | 1 | 8 | 9 | 1 |
| **Mode 3** | 0 | 1 | 1 | 1 | 8 | 9 | 1 |
| **Mode 8** | 0 | 8 | 8 | 8 | 8 | 8 | 8 |
| **Mode 9** | 0 | 1 | 8 | 8 | 8 | 9 | 9 |
| **Mode 10** | 0 | 0 | 0 | 0 | 8 | 9 | 10 |

When the modes of left and up MB are known, the number of modes used in motion estimation can be adjusted.

The algorithm proposed in this paper uses a probability table to relate the prediction modes of adjacent MBs to that of the current MB. For each combination of adjacent MBs, the probability table includes a list of prediction modes in order of expected occurrence such as the most probable mode, next most probable mode etc. The modes priority of current MB based on spatial correlation is shown in Table 2
Here M (l) = Mode of Left MB and M(u)= Mode of Up MB ,Numbers inside the table = Priority Order of Occurrence of Modes.

Table 2 Modes Priority of Current MB Based on Spatial Correlation[3]

| **M(l)/M(u)** | **Mode 0** | **Mode 1** | **Mode 2** | **Mode 3** | **Mode 8** | **Mode 9** | **Mode 10** |
|---|---|---|---|---|---|---|---|
| **Mode 0** | 0,1,8,3,2,10,9 | 0,1,3,2,8,10,9 | 0.1.8.3.2.9.10 | 0,1,3,8,2,9,10 | 0,8,1,3,2,9,10 | 0.1.9.8.2.3.10 | 0.1.10.3.2.9.8 |
| **Mode 1** | 0,1,8,3,2,10,9 | 1,0,8,2,3,9,10 | 1,0,8,2,3,9,10 | 1,0,8,3,2,9,10 | 8,1,3,0,2,9,10 | 9.1.8.3.2.0.10 | 1.0.2.3.8.10.9 |
| **Mode 2** | 0,1,2,8,3,10,9 | 1,8,2,0,3,9,10 | 1,8,3,0,2,9,10 | 1,8,3,0,2,9,10 | 8,1,2,3,0,9,10 | 9.8.1.2.3.0.10 | 1.2.0.8.10.9.3 |
| **Mode 3** | 0,1,8,3,2,9,10 | 1,0,8,3,2,9,10 | 1,0,8,2,3,9,10 | 1,8,3,0,2,9,10 | 8,1,3,0,2,9,10 | 9.8.1.3.2.10.0 | 1.3.0.8.2,9,10 |
| **Mode 8** | 0,1,8,2,3,9,10 | 8,1,2,0,3,9,10 | 8,1,2,0,3,9,10 | 8,1,3,2,0,9,10 | 8,1,3,2,0,9,10 | 8,9,1,2,3,10,0 | 8,1,3,2,9,0,10 |
| **Mode 9** | 0,1,8,3,2,10,9 | 1,8,9,2,3,0,10 | 8,9,1,2,3,0,10 | 8,9,1,3,2,0,10 | 8,9,1,3,2,0,10 | 9,8,1,3,10,2,0 | 9,10,1,3,2,8,0 |
| **Mode 10** | 0,1,10,2,3,9,8 | 0,1,10,2,3,8,9 | 0,1,10,8,2,9,3 | 0,1,3,8,9,10,2 | 8,1,3,0,9,2,10 | 9,8,1,0,3,8,2 | 10,0,1,9,2,3,8 |

## 3. Proposed Algorithm

For video frame *SKIP* block mode take a huge portion. That is because there are a lot of stationary regions existing in natural video. The most common way to detect stationary blocks is to utilize the block differences between inter frames. For non stationary block, the priority order of the candidate block modes based on the block modes of left and upper adjacent blocks is utilized. Besides by observing RD Cost of each mode for every block RD costs of intra block modes is found which is usually more than twice that of the best current mode. Hence, if RD cost of the luminance 16x16 intra block mode is larger than twice that of the best mode at the moment, the calculation of 4x4 inter block mode will not be proceed with. The process of the proposed algorithm is as follows-

**Step1.** Calculate the SAD between the current MB and the collocated MB in the previous frame. If the SAD is less than T1 (set as 2000 for x264) then the best block mode is selected from 16x16 and SKIP. Otherwise, go to step 2.

**Step2.** Define the block mode priority order of the current block based on the block modes of left and upper adjacent blocks. The priority order of the boundary blocks are fixed as 0, 1, 2, 3, 8, 10, and 9.

**Step3.** According to the block mode priority order, the RD Cost of each mode will be calculated in order. If the mode being evaluated belongs to inter prediction mode, then the motion estimation (ME) will be done to get the best MV and then the RD Cost can be calculated. Otherwise, compute the RD Cost of the intra prediction mode.

**Step4.** If the best block mode is unchanged for evaluating two successive block modes in the priority order, the mode decision will be stopped. Otherwise, go to Step 5.

**Step5.** If the current mode is 16x16 intra prediction mode and the RD Cost is larger than twice that of the current best block mode, the mode decision will be stopped. Otherwise, go to Step 6.

**Step6.** After all modes in the priority order are evaluated, the mode decision will be stopped. Otherwise, go to Step 3 and evaluate the block mode of the next priority.

## 4. Experimental Results

The proposed algorithm is implemented on x264 and tested for following type of sequences –QCIF (176x144), CIF (352x288).The simulation conditions of encoder is shown in following Table 3

Table 3 Simulation Conditions of Encoder

| | |
|---|---|
| **Frame Rate (Hz)** | 25 |
| **Total Frames** | 100 |
| **Coding Options Used** | Baseline Profile,IPPPP structure,ME-Hex,Search range +_16,Entropy Coding-CAVLC,No of reference frame=1, No error tool, |

Following are the results showing PSNR and Encoding Time which are obtained at different bit rates viz 32kbps,64kbps,128kbps and 264kbps for different QCIF sequences is shown in table 4

Table 4 Comparison of algorithms for PSNR and Encoding Time (QCIF sequence)

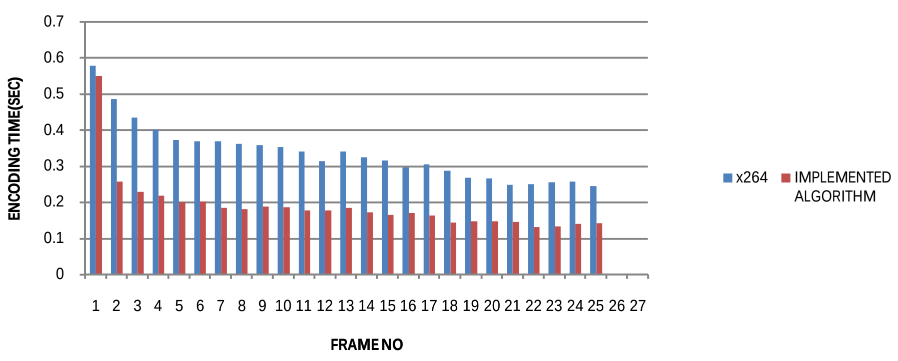
| **Bit rate (kbps)** | **Sr No** | **Sequence** | **PSNR** | | **Encoding Time(sec)** | | |
|---|---|---|---|---|---|---|---|
| | | **QCIF** | **X264** | **Implemented Algorithm** | **X264** | **Implemented Algorithm** | **Δ1 (%)** |
| **32** | 1 | City | 29.59 | 29.46 | 9.02 | 5.79 | 35.80 |
| | 2 | crew | 26.99 | 26.54 | 8.63 | 6.46 | 25.14 |
| | 3 | mobile | 22.31 | 22.45 | 9.20 | 8.12 | 11.73 |
| | 4 | claire | 35.98 | 35.88 | 8.26 | 5.39 | 34.74 |
| **64** | 1 | city | 32.39 | 32.311 | 9.23 | 6.13 | 33.88 |
| | 2 | crew | 29.22 | 29.24 | 9.18 | 6.90 | 33.22 |
| | 3 | mobile | 23.15 | 23.56 | 9.42 | 8.56 | 9.12 |
| | 4 | claire | 39.60 | 39.64 | 8.69 | 5.95 | 31.53 |
| **128** | 1 | City | 35.59 | 35.57 | 9.663 | 8.43 | 12.26 |
| | 2 | crew | 31.77 | 31.76 | 9.56 | 6.94 | 27.40 |
| | 3 | mobile | 27.72 | 27.69 | 9.76 | 8.86 | 9.22 |
| | 4 | claire | 43.44 | 43.37 | 9.12 | 6.17 | 32.34 |
| **256** | 1 | City | 40.33 | 40.24 | 10.899 | 8.38 | 23.68 |
| | 2 | crew | 34.85 | 34.83 | 10.25 | 7.46 | 27.21 |
| | 3 | mobile | 30.49 | 30.507 | 10.45 | 9.05 | 13.39 |
| | 4 | claire | 46.32 | 46.29 | 10.17 | 7.26 | 28.61 |

Different CIF sequences are also tested for PSNR and Encoding Time which are obtained at different bit rates viz 32kbps,64kbps,128kbps and 256kbps is shown in table 5

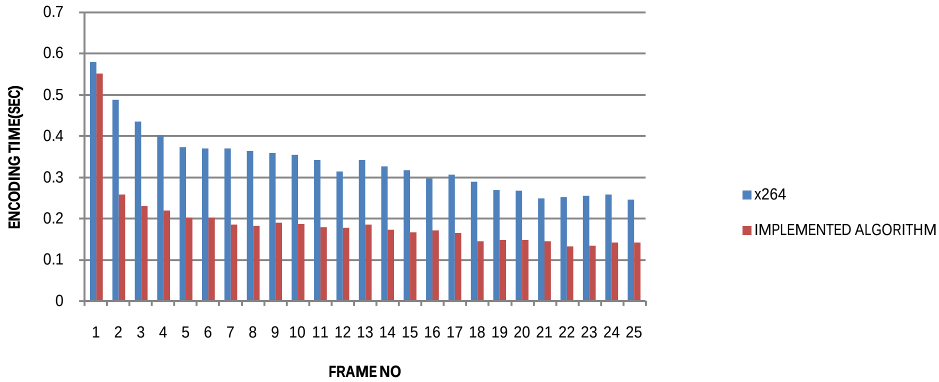
Table 5 Comparison of algorithms for PSNR and Encoding Time (CIF sequence)

| **Bit rate (kbps)** | **Sr No** | **Sequence** | **PSNR** | | **Encoding Time(sec)** | | |
|---|---|---|---|---|---|---|---|
| | | **CIF** | **X264** | **Implemented Algorithm** | **X264** | **Implemented Algorithm** | **Δ1(%)** |
| **32** | 1 | stefan | 21.27 | 21.25 | 23.81 | 18.30 | 23.14 |
| | 2 | paris | 26.51 | 26.43 | 23.59 | 14.38 | 39.04 |
| | 3 | mother | 31.61 | 31.64 | 23.05 | 13.06 | 43.34 |
| | 4 | forman | 25.62 | 25.57 | 24.366 | 15.04 | 38.28 |
| **64** | 1 | stefan | 21.73 | 21.45 | 24.10 | 18.96 | 21.32 |
| | 2 | paris | 29.55 | 29.25 | 23.98 | 14.96 | 37.53 |
| | 3 | mother | 34.11 | 34.57 | 24.56 | 13.86 | 43.56 |
| | 4 | forman | 28.68 | 28.71 | 25.12 | 15.99 | 36.34 |
| **128** | 1 | stefan | 25.63 | 25.43 | 24.90 | 19.26 | 22.65 |
| | 2 | paris | 33.37 | 33.30 | 24.45 | 15.23 | 37.70 |
| | 3 | mother | 36.96 | 36.84 | 24.40 | 14.86 | 39.09 |
| | 4 | forman | 32.25 | 31.14 | 26.18 | 16.56 | 36.74 |
| **256** | 1 | stefan | 28.72 | 28.42 | 25.69 | 20.53 | 20.08 |
| | 2 | paris | 37.16 | 37.10 | 25.02 | 16.42 | 34.37 |
| | 3 | mother | 39.83 | 39.76 | 24.64 | 15.32 | 37.82 |
| | 4 | forman | 35.50 | 35.46 | 27.53 | 18.23 | 30.32 |

Comparison of encoding time of first 25 frames of city_qcif.yuv at 32 kbps and foreman_cif.yuv at 32 kbps is shown in following column graph where encoding time of implemented algorithm is reduced by around 35.80% for first sequence and 38% for second sequence .The encoding time comparison for x264 and proposed algorithm at 32 kbps is shown in Fig 2(a) and (b)



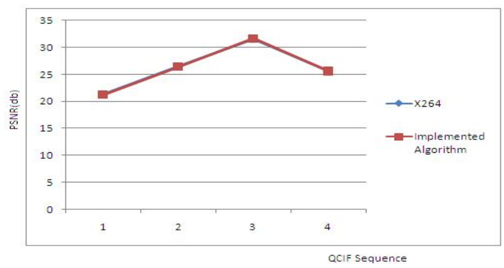

(a)



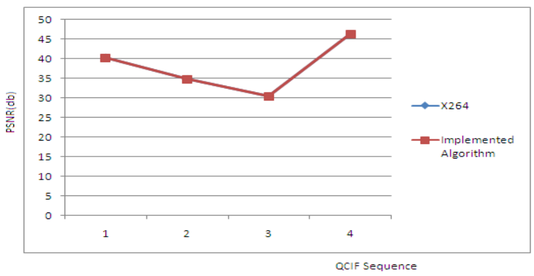

(b)

Fig 2 Comparison of Encoding Time for First 25 Frames of(a) city_qcif.yuv at 32 kbps and (b) foreman_cif.yuv at 32 kbps

Following graph analysis shows comparison of PSNR values for different QCIF and CIF sequences at different bitrates viz 32kbps and 256 kbps. It indicates that there is not much difference in PSNR values of x264 and implemented algorithm. The values remains almost the same is shown in fig 3 (a) , (b) and fig 4(a) , (b)

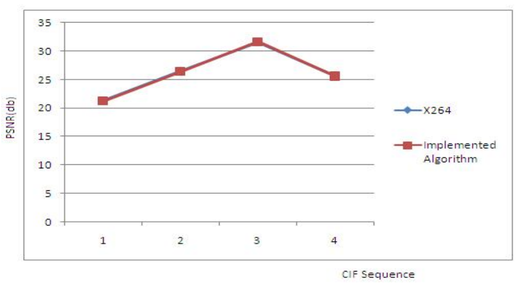



(a)

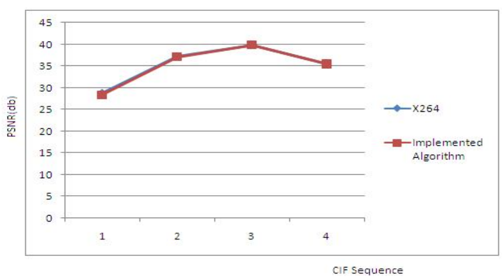



(b)

Fig 3 Comparison of PSNR Values for Different QCIF Sequences at (a) 32kbps (b) 256 kbps




(a)




(b)

Fig 4 Comparison of PSNR Values for Different CIF Sequences at (a) 32kbps (b) 256 kbps

## 5. Conclusions

This algorithm is implemented using x264 encoder. Proposed algorithm reduces encoding time without significantly affecting PSNR for different types of video sequences under different band width (BW).The PSNR loss is around 0.2db to 0.5 db which is tolerable. As the BW is increased the encoding time reduction is on an average 25% to 30% for low motion scene such as city.yuv, claire.yuv, paris.yuv, mother.yuv and it is around 15% to 20% for high motion scene such as mobile.yuv and Stefan.yuv.

## References

[1] Po-Hung Chen, Hung-Ming Chen, Mon-Chau Shie, Che-Hung Su, Wei-Lung Mao, Chia-Ke Huang "Adaptive Fast Block Mode Decision Algorithm for H.264/AVC", 2010 5th IEEE Conference on Industrial Electronics and Applications, Taichung

[2] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, S. Rahardja, and C. C. Ko "Fast Intermode Decision in H.264/AVC Video Coding" IEEE Transactions On Circuits and Systems for Video Technology, vol. 15, no. 6, July 2005

[3] Bin Feng, Guang-xi Zhu, Wen-yu Liu, "Complexity scalable inter modes decision algorithm for H.264 based on spatial correlation, "Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference, Volume 2, 20-22 Feb. 2006, Phoenix Park

[4] Loren Merritt and Rahul Vanam "Improved Rate Control and Motion Estimation for H.264 Encoder" IEEE International Conference on Image Processing, 2007. ICIP 2007, San Antonio, TX

[5] Soon-kak Kwon a, A. Tamhankar b, K.R. Rao c," Overview of H.264/MPEG-4 part 10",Journal of Visual communication and Image Representation 17 (2006) 186–216

[6] Iain E G Richardson "H.264 and MPEG-4 Video Compression Video Coding for Next-generation Multimedia" John Wiley & Sons Ltd, 2003 .