

## Fast Inter Mode Decision Algorithm for H.264/AVC Video Encoding System Based on Correlation of Macroblock

Byung-Gyu Kim, Badrul Hilmi and Hyo-Sung Kim  
 Department of Computer Science and Engineering,  
 Sun Moon University, Republic of Korea

**Abstract:** To more effectively reduce temporal and spatial redundancy in the MPEG-4 Part-10 AVC/H.264, motion compensation uses variable block sizes and directional inter prediction investigates all available coding modes to determine the best one. However, these functions are performed for all variable block sizes, high complexity results due to the large number of combinations of coding modes. The researchers propose a fast inter mode decision to reduce the number of combinations of candidate modes using predictive data from macroblock correlations of adjacent frames. An adaptive thresholding scheme using the rate-distortion costs of neighboring blocks is applied for an early decision of the best macroblock in the inter-mode search process.

**Key words:** H.264/AVC video, inter-mode decision, macroblock correlation, variable, complexity, Korea

### INTRODUCTION

In H.264/AVC, there are a total of 7 different block sizes (16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4) that can be used for inter frame motion estimation/compensation (Wiegand *et al.*, 2003). These different block sizes actually form a two-level hierarchy inside the MB. The first level comprises block sizes of 16×16, 16×8 or 8×16 and is called the Large Macroblock type. In the second level, the MB is specified as the P8×8 type where each 8×8 block can be one of the subtypes 8×8, 8×4, 4×8 or 4×4. These are referred to as the Sub-Macroblock type. The relationship between these different block sizes is shown in Fig. 1.

In addition, many natural video sequences contain stationary regions. In the H.264/AVC encoder, the RD cost to decide the best inter-prediction mode is computed as follows (Sullivan and Wiegend, 1998):

$$J(s,c,MODE \setminus QP) = SSD(s,c, MODEQP) + \lambda_{MODE} \cdot R(s,c, MODE \setminus QP)$$

where, QP is the quantization parameter,  $\lambda_{MODE}$  is the Lagrangian multiplier and SSD is the sum of the squared differences between the original block and its reconstruction.  $R(s,c,MODE|QP)$  is the number of bits associated with the mode currently selected for the MB.

To find the best coding parameters for each macroblock, H.264/AVC reference software encodes all possible combinations of parameters and calculates the rate and distortion of a given macroblock for each combination. Thus, the encoder computes the RD costs of all possible coding options and chooses the coding mode of a given macroblock that has the minimum RD cost. This optimal decision causes the problem of a high degree of computational complexity because for each macroblock, the encoder must have information about the required bits and the resulting distortion of the current coding mode to choose the best coding mode. This information is available only after finishing the encoding process. Therefore, the current H.264/AVC reference software performs this complex process to find the best coding mode.

H.264/AVC also uses a 4×4 integer transform in which the RD cost of a macroblock is computed 16 times under the assumption that the RD cost is computed on 4×4 block units. For 8×8 sub-partitions each sub-block can be motion-compensated independently with the variable

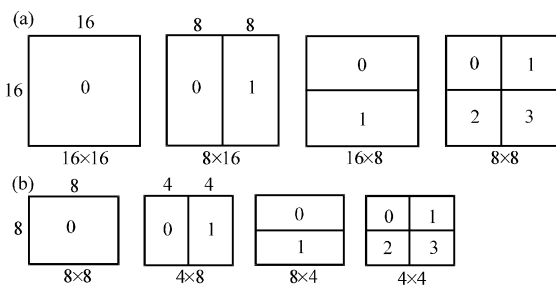


Fig. 1: Different partitions in the MB; (a) Macroblock partition, (b) Sub-Macroblock partition

block sizes of  $8 \times 8$ ,  $8 \times 4$ ,  $4 \times 8$  and  $4 \times 4$ . Thus, for each  $8 \times 8$  sub-block the RD cost is computed to determine the best block mode.

To choose the minimum RD cost of an  $8 \times 8$  sub-partition, the RD cost of each sub-partition is calculated 16 times. Thus, the RD cost is computed 64 times for an  $8 \times 8$  sub-partition mode for a given macroblock. In total, the RD cost is calculated 128 times for all variable block size modes of inter macroblocks. A crucial consideration for application of mode prediction for a fast mode decision is to make sure that the predicted mode has the smallest RD cost for a given MB. However, during experiments on various video sequences using H.264/AVC video coding, we observed that homogeneous regions prevail in video sequences. Thus, many parts of a video sequence only require a search with a large block type even in SKIP mode so using the sub-macroblock in the search process is not necessary. Use of the sub-macroblock type wastes computational time and effort.

Many fast mode decision algorithms have been proposed to reduce the computational complexity without any degradation in image quality. Jeon's (Choi *et al.*, 2006; Hilmi *et al.*, 2010) selective intra-coding and early SKIP is based on the Average sum Boundary Error (ABE) and the Average Rate (AR), Jing and Chau (2004) fast inter-mode decision scheme uses both the frame difference and the MB difference. The method of Wu *et al.* (2005) uses the spatial homogeneity of a video object's textures and the temporal stationary characteristics in video sequences. Salgado's (Sullivan and Wiegand, 1998) method uses temporal correlation and a sequential mode search with a high speed-up ratio for the baseline profile in the encoding time. The efficient thresholding scheme of Kim (Zeng *et al.*, 2009; Badrul Hilmi *et al.*, 2010) for early termination of the mode search is based on the RD cost of the most correlated MB. The cost can be found using a simple MB tracking scheme with a  $P-16 \times 16$  block type in the previous frame. Chia's (Grecos and Yang, 2007) hierarchical decision method reduces the number of search points in the transform process.

Kim and Kim (2008) and Kim *et al.* (2010) proposed two algorithms for a fast mode decision. The first algorithm reduces the inter-mode decision complexity using a direct prediction based on block correlation. The second algorithm is for B-picture coding group candidate modes that have similar average SAD values. Grecos and Yang (2007) algorithm uses two heuristics for prediction of a small set decidable mode then uses a set of skip mode conditions for P and B slices.

Zeng *et al.* (2009) proposed an algorithm based on the motion activity of macroblocks which were divided into classes from no motion to highly-textured regions with fast motion or with a scene change. Liu *et al.* (2009) proposed an efficient inter-mode decision based on

motion homogeneity that evaluated a normalized motion vector field from motion estimation of a  $4 \times 4$  block size. With three directional motion homogeneity measures derived from the normalized motion vector  $\vec{f}_i$   $W_i$ , candidate inter-modes for each macroblock are determined and the RD cost for other modes is unnecessary, thus reducing calculations.

In this study, we propose a fast inter-mode decision to reduce the number of combinations of candidate modes using predictive data from macroblock correlations of the adjacent frame.

An adaptive thresholding scheme using the rate-distortion costs of neighboring blocks is applied for an early decision of the best macroblock in the inter-mode search process.

### FAST INTER-MODE DECISION BASED ON MACROBLOCK CORRELATIONS

#### Early termination of candidate modes using a co-located

**macroblock:** Strong correlations between adjacent inter-frames exist in video sequences. Some algorithms were proposed based on information from a co-located macroblock. Experiments have shown that a co-located macroblock is highly correlated with the current macroblock (Table 1). Thus, correlations are especially

Table 1: The conditional probabilities of the current MB with the previous MB (%)

Sequences	Foreman	Bus	Football
<b>SKIP(MB_Direct)</b>			
SKIP(MB_Direct)	74.84	66.90	81.13
MB16×16	15.94	23.62	14.91
MB16×8	1.77	3.72	1.47
MB8×16	2.60	3.52	1.77
MB8×8	0.95	2.87	0.70
<b>MB16×16</b>			
SKIP(MB_Direct)	64.45	45.25	50.71
MB16×16	21.77	32.16	28.41
MB16×8	5.17	7.77	8.87
MB8×16	5.37	6.69	7.31
MB8×8	3.21	8.11	4.61
<b>MB16×8</b>			
SKIP(MB_Direct)	37.84	27.57	19.22
MB16×16	29.49	33.98	30.79
MB16×8	14.88	16.55	24.15
MB8×16	7.50	8.38	15.07
MB8×8	10.27	13.49	10.70
<b>MB8×16</b>			
SKIP(MB_Direct)	44.50	26.86	14.96
MB16×16	27.02	34.04	30.90
MB16×8	7.01	10.94	13.72
MB8×16	16.54	15.54	28.75
MB8×8	4.91	12.56	12.14
<b>MB8×8</b>			
SKIP(MB_Direct)	28.48	18.60	8.75
MB16×16	26.84	21.49	21.78
MB16×8	16.93	17.38	20.25
MB8×16	12.11	14.81	24.21
MB8×8	15.62	27.69	24.85

strong for the SKIP (mode 0), 16×16 (mode 1) and P8×8 (mode 8) modes. To determine the initial search mode, the best mode information of a correlated MB in a time-successive frame is used. This direct prediction algorithm only uses a large size block mode (Modes 0, 1, 2, 3). Otherwise in the case of a sub-macroblock, a full mode search is used. This direct prediction algorithm for the initial search can be described as (Initial mode search):

- Case 0, If the MB correlated: SKIP (mode 0)  
Set the initial MB current: SKIP (mode 0)
- Case 1, If the MB correlated: 16×16 (mode 1)  
Set the initial MB current: SKIP (mode 0),  
16×16 (mode 1)
- Case 2, If the MB correlated: 16×8 (mode 2)  
Set the initial MB current: SKIP (mode 0),  
16×8 (mode 2)
- Case 3, If the MB correlated: 8×16 (mode 3)  
Set the initial MB current: SKIP (mode 0),  
8×16 (mode 3)
- Case 4, If the MB correlated: Sub-block  
Set the initial MB current: Full mode search

The early termination method to determine the initial search mode is based on the experimental results in Table 1. In Case 0, if the correlated MB is encoded in SKIP mode, the probability that the current MB will be encoded in SKIP mode is 70.15%. Based on this high probability we can predict the candidate search mode.

**Macroblock correlation and adaptive threshold of error correction:** Early termination of the candidate mode decision process will cause a significant speed up but when we only use this process the image quality will decrease. To maintain good image quality we need an additional process to determine whether an additional search mode is needed. The researchers use an adaptive thresholding scheme based on the rate-distortion cost of the correlated MB.

Generally if an object's motion is increased between successive macroblocks, the RD cost is also increased and vice versa. The RD cost represents the motion characteristics of the macroblock. For example, if the RD of the current macroblock is less than the RD cost of the co-located macroblock, the researchers can assume that the current macroblock contains less detail than the co-located macroblock and the probability that the current MB is encoded in the same mode as the co-located macroblock mode is high. Therefore, in this process if the rate distortion cost of the current macroblock is higher than the RD cost of the co-located macroblock, this

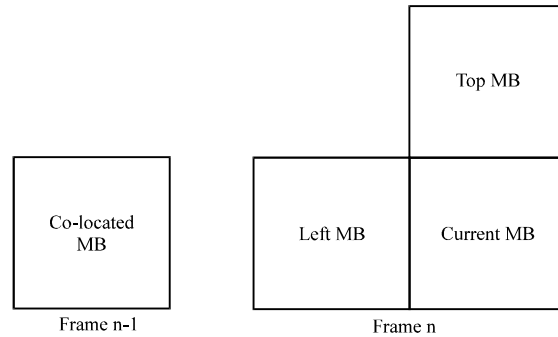


Fig. 2: Correlations with the current macroblock

macroblock requires an additional search mode. This process using properties of neighboring macroblock is shown in Fig. 2.

This additional search mode is divided into two groups. The first is the SKIP and 16×16 modes and the second is the 16×8 and 8×16 modes. For SKIP and 16×16, if the left MB and the top MB are a large block size then the additional search mode will be a Large Block type. Otherwise, the Fullsearch mode is used. In the next case for 16×8 and 8×16, we use the average of the RD costs of the co-located macroblock. The researchers check whether the left macroblock (current frame) and top macroblock (current frame) require an additional process. Then, for the 16×8 mode, if the top and left MBs are 16×8, then the additional search mode is SKIP, 16×16 or 16×8. The current RD cost is compared with the average RD cost and if it is still higher then the fullsearch mode is applied. This process is also used for the 8×16 mode. This additional search mode can be described as follows:

- For SKIP and 16×16 mode:  
If Top and Left MB is Large MB  
Set additional search mode: Large MB  
Otherwise: Fullsearch
- For 16×8 mode:  
If Top and Left MB is 16×8  
Set additional search mode: SKIP, 16×16 and  
16×8  
Check Current R-Dcost > Average R-Dcost  
Top and Left MB  
If satisfied: Fullsearch  
Otherwise: Next MB
- For 8×16 mode:  
If Top and Left MB is 8×16  
Set additional search mode: SKIP, 16×16 and  
8×16  
Check Current R-Dcost > Average R-Dcost  
Top and Left MB  
If satisfied: Fullsearch  
Otherwise: Next MB

**EXPERIMENTAL RESULTS**

To verify the proposed scheme, a comprehensive set of experiments for a variety of video sequences with different motion characteristics was performed. The researchers used Jeon (Choi *et al.*, 2006), Kim’s (Salgado and Nieto, 2006; Grecos and Yang, 2007) methods for an objective comparison of the encoding performance. These two methods provide good quality performance for a fast mode decision.

All test video sequences used were well known in video compression testing. They included various MPEG standard sequences with CIF and QCIF sizes. Analyses were performed with encoding frames = 100, RD optimization enabled, QP = 24, 28 and 32, IPPP sequence types in the main profile using CAVLC with a search range of MV = ±16 and the number of reference frames = 1. FME was used as a default and Hadamard transform was enabled.

JM 11.0 reference software of the JVT (Joint Video Team) was used as a reference code for evaluation of the encoding performance. We defined three measures for evaluating the encoding performance including average ΔPSNR, average ΔBits and an encoding-time saving factor ΔT. The average ΔPSNR was defined as the difference in decibels between the average PSNR of the proposed method and the corresponding value of another method. As performance improved, this criterion became smaller:

$$\Delta\text{PSNR} = \text{PSNR}_{\text{Proposed}} - \text{PSNR}_{\text{Original}}$$

The average ΔBits was defined as the bit-rate difference as a percentage between the compared methods and the encoding-time saving factor was defined for a complexity comparison as:

$$\Delta\text{Bitrate} = \frac{\Delta\text{Bitrate}_{\text{Proposed}} - \Delta\text{Bitrate}_{\text{Original}}}{\Delta\text{Bitrate}_{\text{Original}}} \times 100\%$$

$$\Delta\text{Time} = \frac{\Delta\text{Time}_{\text{Proposed}} - \Delta\text{Time}_{\text{Original}}}{\Delta\text{Time}_{\text{Original}}} \times 100\%$$

Figure 3 shows RD curves for several sequences. Two algorithms were used to verify the performance for the fast mode decision schemes. The RDO performance of the proposed method and the performance of JM Reference software using full inter-mode search was similar. Kim’s method also showed similar performance compared with a full inter-mode search and also had a large speed up factor for the encoding time. Grecos’s algorithm provided worse image quality than both the proposed method and Kim’s method with a slower speed-up factor.

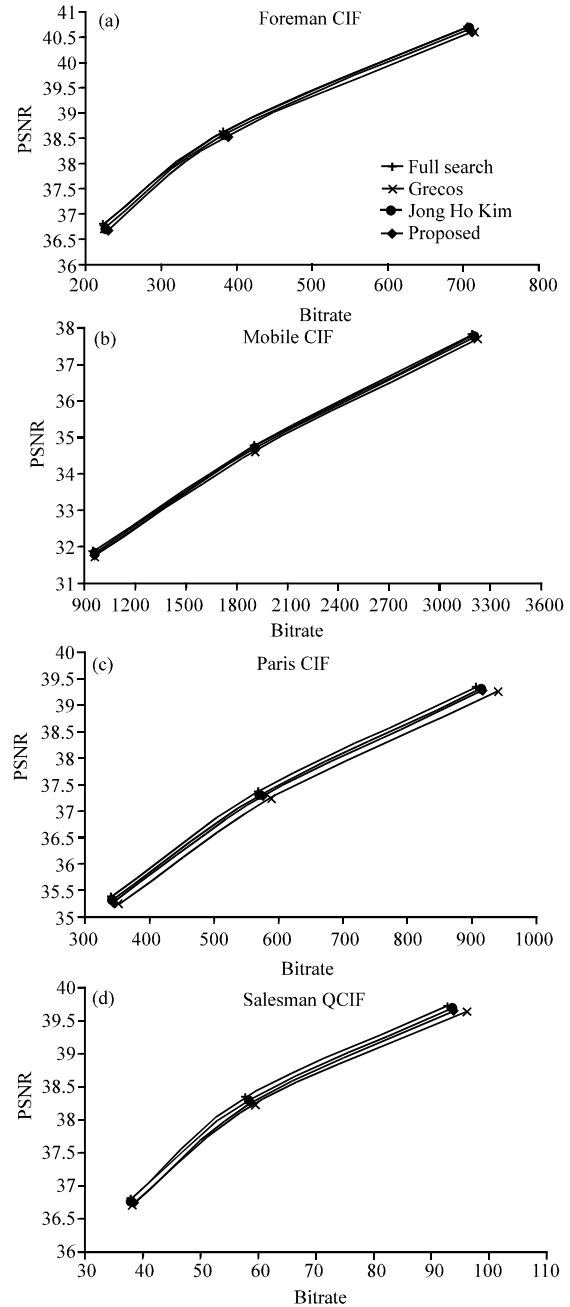


Fig. 3: RDO performance for IPPP sequences: (a) Foreman, (b) Mobile, (c) Paris and (d) Salesman sequences

Table 2 shows a comparison between all tested algorithms for IPPP sequences. For image quality, the average PSNR loss of the proposed algorithm was 0.067 dB. The PSNR loss was also stable in all video sequence types. Despite Kim’s and Grecos’s algorithms providing better image quality performance, this result for

Table 2: Comparison results for IPPP sequences

Sequence	QP	Grecos			Jong Ho Kim			Proposed		
		PSNR	Bitrate	Time	PSNR	Bitrate	Time	PSNR	Bitrate	Time
foreman_cif	24	-0.093	-6.351	-46.009	-0.034	0.316	-67.192	-0.0800	0.960	-72.945
	28	-0.063	-6.963	-43.699	-0.045	0.793	-70.521	-0.0840	1.601	-76.761
	32	-0.017	-6.300	-46.068	-0.070	0.972	-73.586	-0.1130	2.604	-79.691
mobile_cif	24	-0.084	-2.225	-44.789	-0.051	0.093	-64.917	-0.0550	0.347	-66.793
	28	-0.106	-2.079	-46.119	-0.050	-0.112	-65.555	-0.0640	0.241	-68.689
	32	-0.132	-1.513	-52.508	-0.060	-0.668	-66.669	-0.0890	-0.472	-71.440
paris_cif	24	-0.040	-4.639	-53.225	-0.036	0.747	-74.111	-0.0630	1.246	-79.059
	28	-0.016	-4.906	-54.853	-0.045	0.752	-75.574	-0.0660	1.481	-80.426
	32	0.024	-5.451	-56.408	-0.039	0.719	-75.959	-0.0800	1.326	-81.358
mother and daughter_cif	24	-0.028	-6.300	-55.312	-0.019	-0.119	-75.522	0.0620	0.987	-81.900
	28	0.010	-6.454	-56.249	-0.034	0.653	-77.086	-0.0530	1.109	-83.870
	32	0.000	-7.742	-57.344	-0.027	0.315	-78.724	0.0360	1.158	-85.118
hall_cif	24	0.023	-5.747	-52.725	-0.048	-1.067	-73.382	-0.0980	-1.903	-78.056
	28	0.008	-4.341	-54.186	-0.057	-1.062	-76.900	-0.0930	-2.588	-83.247
	32	-0.039	-6.560	-55.801	-0.016	-2.179	-78.593	-0.0550	-1.980	-85.628
akiyo_cif	24	-0.007	-4.755	-55.363	-0.017	0.059	-77.125	-0.0400	0.356	-85.380
	28	0.008	-5.450	-56.949	-0.014	0.355	-78.048	-0.0490	0.434	-86.354
	32	-0.005	-8.140	-56.688	-0.060	-0.850	-78.928	0.0860	-0.705	-87.021
stefan_qcif	24	-0.057	-4.493	-48.008	-0.041	0.082	-68.732	-0.0450	0.263	-70.154
	28	-0.045	-4.251	-49.732	-0.057	-0.030	-68.558	-0.0590	0.114	-70.677
	32	-0.034	-3.202	-51.664	-0.054	-0.602	-69.497	-0.0660	0.079	-72.628
salesm an_qcif	24	-0.039	-1.047	-51.200	-0.018	0.871	-77.368	-0.0580	1.289	-82.985
	28	-0.078	-1.580	-52.012	-0.027	1.291	-77.465	-0.0730	1.820	-83.571
	32	-0.080	-3.336	-54.174	-0.014	-0.437	-77.127	-0.0530	0.992	-84.108
Average		-0.039	-4.743	-52.128	-0.039	0.037	-73.589	-0.0670	0.448	-79.077

the algorithm is good considering the speed-up factor we achieved. The proposed algorithm causes a negligible bitrate increment in low motion video sequences while high motion sequences had a bit increment of >1%. The proposed algorithm had a 0.448% bit increment compared with Kim's and Grecos's algorithms and Grecos's method exhibited a small bitrate saving for all video sequence types.

The last performance evaluation in the fast mode decision is the speed-up factor or time saving for the encoding process. Good speed up performance should not affect the other quality factors, meaning that a method can achieve a minimum image quality loss and a negligible bitrate increment with a good speed-up factor. The proposed method achieved an average time saving of 79.07% with little image quality loss and a negligible bitrate increment. The proposed method had better speed-up performance than both Kim's method (73.58% on average) and Grecos's method (52.128% on average).

The proposed algorithm exhibited a small quality loss (-0.067 dB) and a small bit increment (0.448%) with an average speed-up factor of 79.077%. Degradation in both PSNR and the bit rate were slightly more in comparison to Kim's method. The proposed algorithm achieved a speed-up factor of >6% compared to Kim's method. The proposed method reduced the encoding time with a negligible loss in image quality and bit-rate performance. Future research will focus on an adaptive search range control mechanism to obtain a larger speed-up factor while maintaining a small quality loss. The magnitude of the motion vector is inclined to be larger for the sub-block type (P8x8) due to a detailed texture and

partial motion components. Thus, the search range can be controlled using an adaptive mathematical mode. Use of this characteristic can increase the speed of the encoding system.

### CONCLUSION

Researchers proposed a fast block mode decision algorithm for the H.264/AVC video standard based on two-step processing. First, the proposed fast inter-mode decision algorithm causes an early termination of the search process based on correlations of the macroblock. Adaptive thresholding is used to provide an additional search process when necessary to achieve better image quality. Based on the rate-distortion curve, the proposed method shows performance similar to the full search method by reducing the total encoding time with little image quality degradation and a small bit increment. A 79.07% time saving factor was achieved with a 0.448% bit increment and a 0.067 dB PSNR loss in image quality.

### ACKNOWLEDGEMENT

This research was supported by the Sun Moon University Research Grant of 2009.

### REFERENCES

Choi, I., J. Lee and B Jeon, 2006. Fast coding mode selection with rate-distortion optimization for MPEG-4 part-10 AVC/H.264. IEEE Trans. Circuits Syst. Video Technol., 16: 1557-1561.

- Grecos, C. and M. Yang, 2007. A framework for fast mode decision in the H264 video coding standard. *Digital Signal Process.*, 17: 652-664.
- Hilmi, B., C.S. Park and B.G. Kim, 2010. A survey of fast inter mode decision algorithms for H.264/AVC video encoding system. *Int. J. Soft Comput.*, 5: 128-148.
- Jing, X. and L.P. Chau, 2004. Fast approach for H.264 inter mode decision. *Electron. Lett.*, 40: 1050-1052.
- Kim, J.H. and B.G. Kim, 2008. Fast block mode decision algorithm in H.264/AVC video coding. *J. Visual Commun. Image Represent.*, 19: 175-183.
- Kim, J.H., H.S. Kim, B.G. Kim, H.Y. Kim, S.Y. Jeong and J.S. Choi, 2010. Fast block mode decision scheme for B-picture coding in H.264/AVC. *Proceedings of the International Conference on Image Processing*, Sept. 26-29, Hong Kong, pp: 2873-2876.
- Liu, Z., L. Shen and Z. Zhang, 2009. An efficient intermode decision algorithm based on motion homogeneity for H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.*, 19: 128-132.
- Salgado, L. and M. Nieto, 2006. Sequence independent very fast mode decision algorithm on H.264/AVC baseline profile. *Proceedings of the International Conference on Image and Processing*, Oct. 8-11, Atlanta, GA, pp: 41-44.
- Sullivan, G. and T. Wiegand, 1998. Joint video team, rate-distortion optimization for video compression. *IEEE Sig. Process. Mag.*, 11: 74-90.
- Wiegand, T., G.J. Sullivan, G. Bjontegard and A. Luthra, 2003. Overview of the H.264/AVC video coding standard. *IEEE. Trans. Circuits Syst. Video Technol.*, 13: 560-576.
- Wu, D., F. Pan, K.P. Lim, S. Wu and Z.G. Li *et al.*, 2005. Fast intermode decision in H.264/AVC video coding. *IEEE Trans. Circuits Syst. Video Technol.*, 15: 953-958.
- Zeng, H., C. Cai and K.K. Ma, 2009. Fast mode decision for H.264/AVC based on macroblock motion activity. *IEEE Trans. Circuits Syst. Video Technol.*, 19: 491-499.