International Journal of Soft Computing 8 (4): 231-238, 2013
ISSN: 1816-9503

Fast Inter Mode Decision Algorithm for H.264/AVC Based on Motion Complexity Analysis

Kalyan Goswami, Chan-Seob Park, Jong-Hyuk Lee and Byung-Gyu Kim
Department of Computer Engineering, SunMoon University, Asan, Republic of Korea

Abstract: In H.264/AVC video standard, inter-mode prediction technique is used for reducing temporal as well as spatial redundancies by employing seven variable size of MacroBlocks (MBs). To determine the best MB among the different sizes the rate distortion cost is used as a determining factor. This process requires relatively large amount of computation to encounter all the combinations of different sizes of Mbs. Hence, a fast inter-mode decision algorithm is required to reduce the computational complexity in H.264/AVC. In this study, a new fast inter-mode decision algorithm is proposed in this context. The proposed method reduces the number of candidate modes using direct information from the co-located macroblock (s). A novel motion complexity analysis scheme is also discussed in this study which is strong enough to categorize different moving objects inside a MB and to make a proper decision for the best candidate mode. Experiment results show that the proposed algorithm can achieve speed-up factor up to 78.70% on average with negligible loss of image quality and quite good bit rate performance.

Key words: H.264/AVC, inter-mode decision, macroblock correlation, motion complexity analysis, MAD, Korea

INTRODUCTION

Video compression is basically about the reduction and removal of redundant data from a raw video stream to make an efficient and effective video file which can be sent or stored easily. This process involves a set of algorithms which are applied to a raw video sequence to create a compressed stream that is suitable for transmission or storage. On the other hand to play the compressed stream, an inverse set of algorithms are required in the decoder side to produce a video sequence which shows virtually the same content as the original one. The required time to compress, send, decompress and display a video stream is called latency. Generally, more advanced compression algorithm takes higher latency. Different video compression standards utilize different methods to remove or reduce the redundant data and hence, results differ in bit rate, quality and latency. Now a days, to create a user friendly consumer electronics product is it very important to make the device as fast as possible. Moreover, from the user's point of view it is better to have a real time performance device. Hence, the main challenge for the video codec algorithm developers in recent time is to make a suitable algorithm which gives fast (i.e., very low latency) as well as good performance in terms of quality, bit rate, etc.

In the domain of video compression, generally two types of prediction techniques are considered in any latest coding standards. These are inter and intra-frame predictions which are mainly responsible to eliminate spatial and temporal redundancies, respectively. In this research, researchers are concentrating on the inter-frame prediction technique in H.264/AVC. This prediction technique is used to eliminate the large amount of temporal redundancy that exists in video sequences which is expressed in terms of one or more neighboring frames. In conventional predictive coding technique the difference between current and predicted frames (based on the previous frame) is coded and transmitted. For the better prediction technique the prediction error as well as the transmission bit rate are small. The inter-coding of a macroblock uses motion estimation and compensation techniques which are highly computational intensive subroutines of H.264/AVC. It is found from literatures that 50-80% of the total encoding time is required for motion estimation and compensation.

In the H.264/AVC video coding standard, 7 different block shapes are used for inter frame motion estimation and compensation (Wiegand *et al*., 2003; Richardson, 2003). These different block sizes actually form a two-level of hierarchy inside a Macro Block (MB). The first level comprises block sizes of 16×16, 16×8 or 8×16 that is called a large MB 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

Corresponding Author: Kalyan Goswami, Department of Computer Engineering, SunMoon University, Asan, Republic of Korea

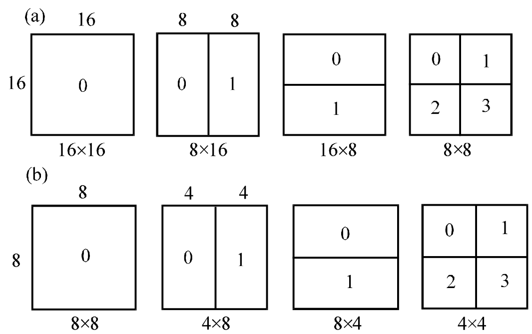

Fig. 1: Different partitions in macroblock: a) Large Macroblock and b) Sub-macroblock

different block sizes is shown in Fig. 1. However, to find the best coding parameter (s) for a MB inside a video frame the H.264/AVC reference software encodes all possible combinations of parameters and calculates the Rate-Distortion (RD) cost for each combination. After that it is required to choose the best coding mode among all possible coding options for each macroblock. Generally, the structure among these 7 options which has the minimum RD cost is considered as the best one. Hence, the information about the best mode is only available after completion of the whole encoding process. Therefore, the current reference software for H.264/AVC performs a computationally complex process to find the best coding mode for a MB in the inter-frame prediction technique.

From the experiments on various benchmark video sequences using H.264/AVC environment, it is observed that any natural video sequence consists spatial homogeneous and temporal stationary regions. These two types of regions are mostly encoded in bigger block sizes (such as 16×16) or sometimes need not required any encoding (SKIP mode). Therefore, before encoding if researchers can decide that a macroblock is homogeneous and or temporal stationary then researchers can safely eliminate the RD-cost calculation for all other modes and encode the macroblock using larger block size such as 16×16 or 8×8.

The spatial homogeneous and temporal stationary regions in two benchmark video sequences are shown in Fig. 2. It can be observed from Fig. 2a that the background, black suit of the man can be considered as the homogenous areas and are coded using 16×16 block size. Although, the boundary area of the lady's suit is non-homogenous with strong edges but the object remains still during some time interval. For this reason, this region is considered as the temporal stationary region and also coded using 16×16 block size. On the other hand, the dancers in the upper part of the image are relatively smaller and contain much motion.

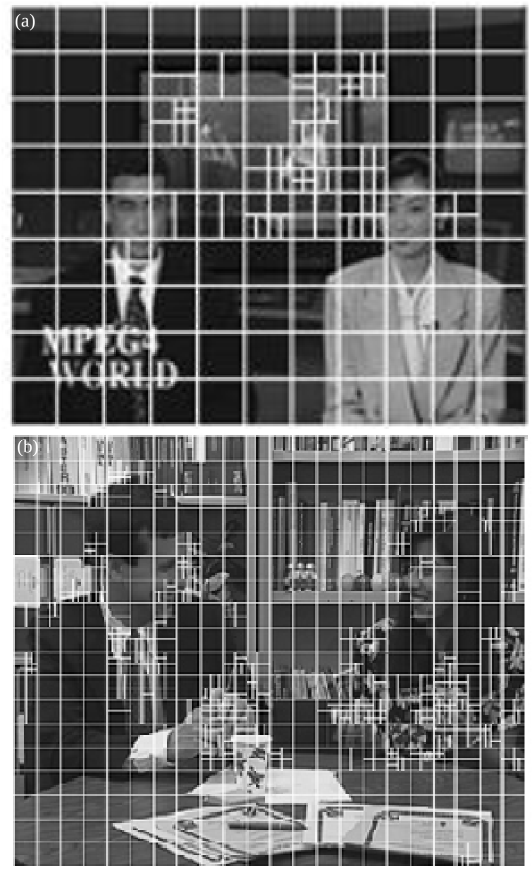

Fig. 2: Examples of block sizes chosen after inter mode search in H.264/AVC: a) News and b) Paris sequences

Hence, these regions are coded in small size blocks. Similarly, in Fig. 2b, the homogeneous regions are like suit of the man, table, hairs, etc. The background contains bookshelf having much more edge information but stationary in nature in the temporal domain. For this reason, these regions are coded in large block size. Hence, it is quite beneficial if the information about the homogeneity and the stationary nature are known for a MB before calculating all the RD-cost values for 7 different block shapes in the H.264/AVC. If this information is included in the H.264/AVC codec then it should perform faster than the conventional one.

There are relatively good amount of techniques are available using edge information for detecting homogeneity of a region. However, it is quite difficult to detect the stationary nature of a region inside a video frame because the parameterization of the moving objects inside a video frame is quite complicated. Motivating from this problem in this study a novel algorithm for fast inter-mode decision is proposed which is basically a combination between direct prediction technique and a novel motion complexity analysis scheme for a macroblock.

Literature review: There are many fast mode decision algorithms have been reported to reduce the computational complexity with marginal degradation in image quality. Choi *et al*. (2006) using Average Boundary Error (ABE) and Average Rate (AR) properties a skip decision method has been proposed which is one of the most popular algorithms for fast mode decision. This is basically a two stage algorithm. In the first stage, 16×16 block size mode checking is conducted for all possible early terminations. After that skip detection is performed for all intra modes by evaluating the average boundary errors.

Early detection of MB type is very important in fast mode decision to reduce the number of computation which causes by unnecessary candidate modes. However, several methods have been reported using some properties for grouping the candidate modes and then MBs are divided into large sub-macroblock types (Salgado and Nieto, 2006; Liu and Jia, 2009). The MB type is divided using mean of absolute difference (Jing and Chau, 2004). Moreover, depending upon the homogeneity, it is also possible to group large and sub-macroblock types (Yu *et al*., 2008). MBs are divided into 3 groups-low, medium and high complexity, based on macroblock detection level (Huang and Hu, 2009). The large and sub-macroblock grouping is also reported by Grecos and Yang (2007) for intra-mode decision using a Heuristic Method.

Another fast inter mode decision is based on Coded Block Pattern (CBP) criterion. The CBP is a syntax element in the H.264 macroblock layer that specifies which one out of six 8×8 blocks (for the baseline profile) may contain non-zero transform coefficient levels (Ostermann *et al*., 2004).

There is another popular and widely used property in fast mode decision present, based on information from the co-located macroblocks. The probability of having the same mode for current and co-located macroblock (s) is quite high. Using this property many algorithms have been developed (Kim and Kim, 2008). A detailed survey of fast inter-mode decision algorithms for H.264/AVC has been reported by Hilmi *et al*. (2010). The most important point is that there is a relatively low amount of attempts are taken for the motion complexity analysis of a macroblock.

MATERIALS AND METHODS

Generally most of the real world video sequences contain substantial amount of background and motionless object (s) which has a highly temporal correlation between successive frames. Hence, after calculating all of the coding modes, there are good amount of MBs are finally end up with SKIP mode or large-macroblock type as the best mode. The proposed method is a new technique to detect moving object (s) inside a macro block which can be divided into two groups. The first one is for SKIP mode where direct information from the co-located macro block is used. On the other hand, for the rest of the Large Macro Block types (LMB) (16×16, 16×8 and 8×16) motion complexity analysis scheme is used to detect the amount of motion inside a macroblock. If it is has less motion then the candidate mode will have the same type with the best mode at the co-located macroblock. Otherwise additional candidate mode (s) will be added. For small macroblock types (8×8, 8×4, 4×8 and 4×4) the motion complexity analysis scheme is not required because generally it contains fast moving object (s) with detailed information.

Motion complexity analysis: To apply this motion complexity analysis scheme, the very first job is to categorize the motion component inside a macroblock. That means it is required to distinguish between high, medium and low motion of macroblock inside a video sequence. There are some benchmark videos available which are representative of the different types of motion. For example akiyo, foreman and football are typical representatives of low, medium and high motion, respectively. If for these representative benchmark videos the Mean of Absolute Difference (MAD) values are calculated for each macroblock over the whole video sequence for a particular QP value then the calculated MAD values can be considered as random variables. Then, according to the central limit theorem, the distribution of these independent random variables tends towards a normal distribution which is shown in Fig. 3a. For this particular distribution the corresponding mean and standard deviation can be calculated and using these values the plotted Gaussian distribution is shown in Fig. 3b.

In the Fig. 3 only one benchmark video is considered (Foreman) for medium motion analysis. Hence, the next step is to analyze three kinds of motions (low, medium and high) together. Now, if all the above mentioned three representative benchmark video sequences are combined together then there should be 2 points of intersection, Thres Hold 1 (TH_1) and Thres Hold 2 (TH_2) as shown in Fig. 4. The TH_1 is the low Threshold and TH_2 can be considered as the high threshold.

The above discussed experiment is performed for a particular QP value. Now, the QP value is varied form 20-40 and corresponding TH_1 and TH_2 are calculated for

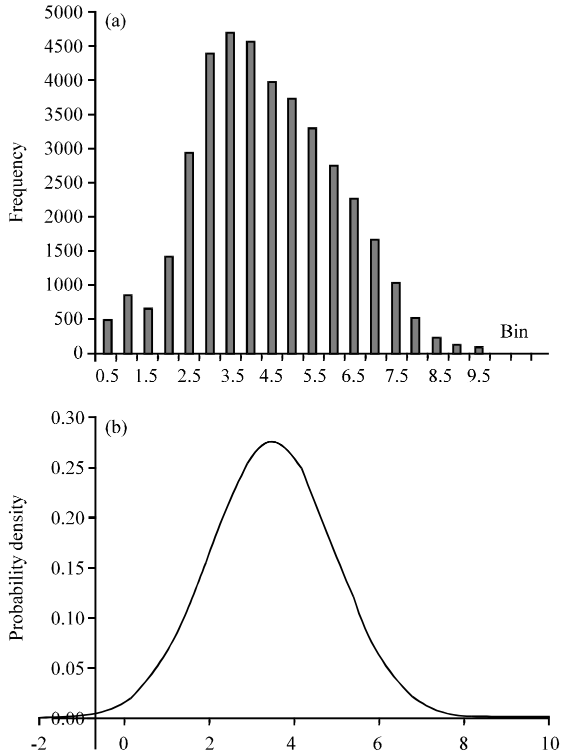

Fig. 3: Histogram of MAD value for the Foreman sequence at QP = 36

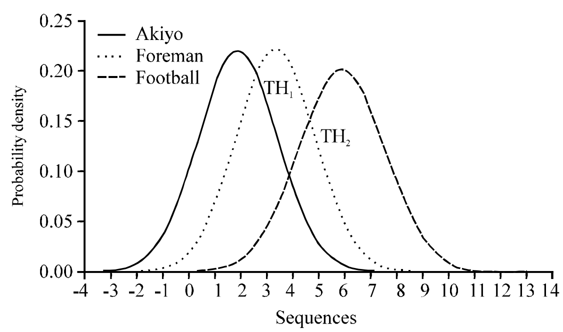

Fig. 4: Intersection of normal distribution from akiyo, foreman and football sequences

each QP. In Table 1 different threshold points for changing QP values are shown for the combined distributions of akiyo, foreman and football.

These values are plotted in a graph and the corresponding plot is shown in the Fig. 5. From this plot, it is found that TH_1 and TH_2 changes linearly with QP values. Hence, to model these thresholds for different QP values it is quite natural to go for a Linear Model. The corresponding Linear Model is given:

$$\begin{aligned} TH_1 &= 0.107QP \\ TH_2 &= 0.281QP\text{-}3.86 \end{aligned} \tag{1}$$

Table 1: Change in threshold points for different QP values

QP	TH_1	TH_2
0	1.44	1.76
24	1.75	2.54
28	2.10	3.46
32	2.60	4.64
36	3.26	5.98
40	3.58	7.38

Table 2: Probability of having the same mode of current MB and the top or left MB

Sequences	QP	Probability
Akiyo	20	0.62
	28	0.72
	36	0.83
Foreman	20	0.17
	28	0.18
	36	0.34
Football	20	0.23
	28	0.25
	36	0.25

MB correlation and error correction: The above discussed technique will speed up the process significantly but may cause degradation in image quality and bit rate increment. This degradation in performance will occur due to the possibility of moving of a MB in different direction with the co-located MB. To avoid this problem a technique is required to determine whether an additional search mode is needed or not.

Generally, if the motion of an object is increased between successive macroblocks, the RD cost is also increased and vice versa. The RD cost represents the motion characteristics of a macroblock. For example if RD cost of a MB is less than the RD cost of its co-located MB then it can be assumed that the current MB contains less detail of temporal information than the co-located MB. The chance to encode the current MB in the same mode as the co-located macroblock mode is quite high. On the other hand if the RD cost of the current MB is higher than the co-located MB an additional search mode is required for further investigation. Usually left and top MBs have strong correlation with the current MB. In Table 2, the probability of the current MB same mode with top or left MB is shown. Form Table 2, it is noticeable that the probability value is high for the sequence with less motion characteristic and also it depends upon the QP value.

Hence, the additional search mode is based on left and top macro block mode information which can be divided into two groups. For skip and 16×16 if left and top MBs are Large (LMB) then the additional search mode will from the LMB, otherwise full search mode. On the other hand for rest (16×8 and 8×16) if left and top MBs have the 16×8 or 8×16 mode the same decision will applied.

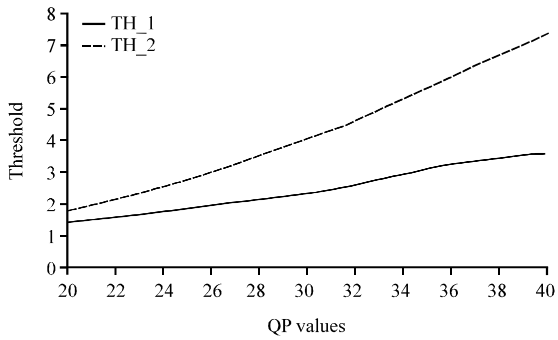

Fig. 5: Change in the threshold points for different QP values

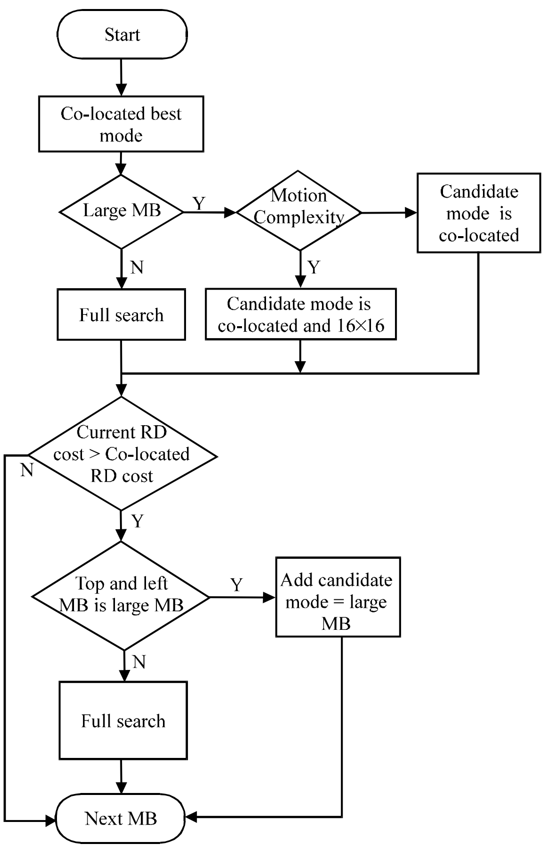

Fig. 6: Flow chart of the proposed algorithm

Overall procedure: The overall procedure can be divided into two sub-groups: initial candidate mode selection and error correction. In the first step, the initial candidate mode is selected based on the motion complexity analysis. After that using RD cost value the proper mode is selected. The overall procedure is shown in a flowchart in Fig. 6 and the corresponding pseudo code is given.

Initial candidate mode:

```
(case 0) if co-located MB: SKIP and motion vector 0
    set the initial MB current: SKIP
(case 1) if co-located MB: 16×16(mode 1)
  if MAD < TH1
      set the initial MB current: SKIP, 16×16
  else set the initial MB current: SKIP, 16×16, 16×8, 8×16
(case 2) if co-located MB: 16×8
   if TH1 <  MAD < TH2
     set the initial MB current: SKIP, 16×8
   else set the initial MB current: SKIP, 16×16, 16×8 (case 3) if co-located
  MB: 8×16
  if TH2 <  MAD
     set the initial MB current: SKIP, 8×16
  else set the initial MB current: SKIP, 16×16, 8×16
otherwise: full search
```

Error correction:

After computing the min. RD cost and mode from the initial candidate mode:

```
    if current best MB's RD cost > co-located MB's RD cost
      if current MB's best mode: SKIP, 16×16
        if top MB and left MB: large MB type
          set the additional mode: large MB type
          otherwise: full search
if current MB's best mode: 16×8
  if top MB and left MB: 16×8
     set the additional mode: large MB type
 otherwise: full search
if current MB's best mode: 8×16
 if top MB and left MB: 8×16
   set the additional mode: large MB type
 otherwise: full search
```

RESULTS AND DISCUSSION

To verify the performance of the proposed algorithm, the experiments are performed for a variety of video sequences with different motion characteristics. JM 11.0 reference software of the JVT (Joint Video Team) is used here as a reference code to evaluate the performance. Analysis are performed with encoding 100 frames with RD optimization enabled, QP = 20, 28, 36 and 40 for IPPP sequence types in main profile with a search range of MV = ±16 and the number of reference frames = 1.

Researchers have used the methods proposed by Kim and Kim (2008) and Zeng *et al.* (2009) for an objective comparison of the encoding performance. They are quite well-known as simple and fast algorithms for improving the speed of H.264/AVC video encoding system.

Analysis of RDO performance: Figure 7 shows the Rate-Distortion (RD) curves for some tested sequences. From the results, it can be obsurved that the proposed mode decision algorithm exhibited an RDO performance similar to the JM original encoder with the full inter mode search. Also, similar to the proposed method by Kim and Kim (2008) but with a little slower than the proposed algorithm for the encoding time. Although, the method proposed by Zeng *et al.* (2009) gave up to 73% of

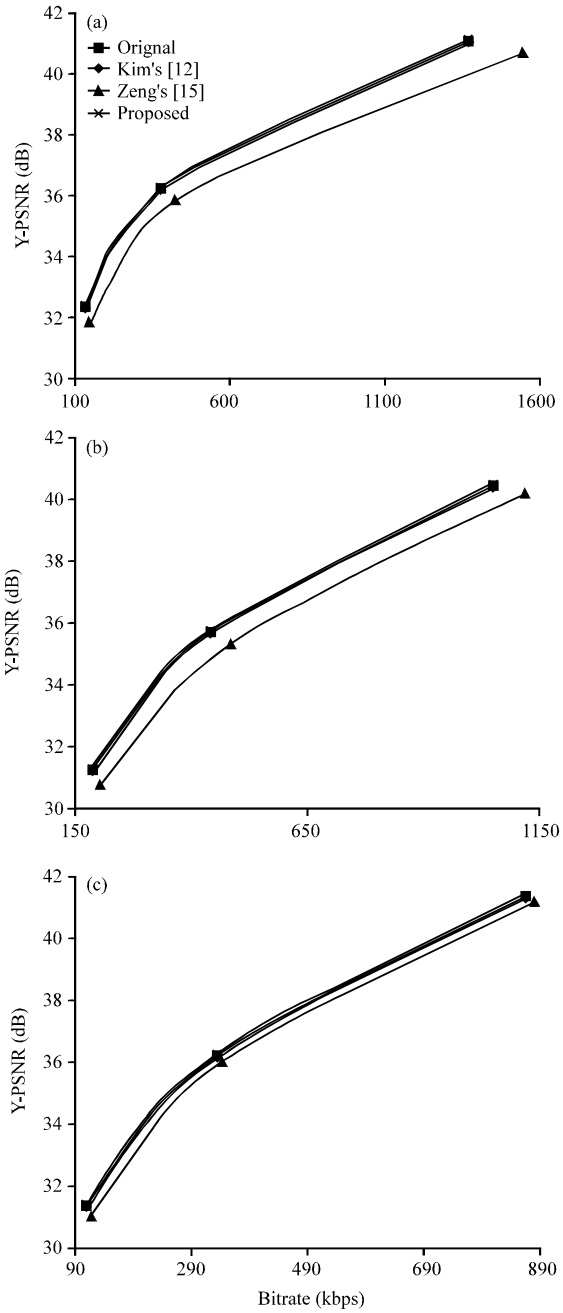

Fig. 7: The RD curves: a) Foreman; b) Paris and c) Stefan sequences

time-saving factor, it achieved somewhat degraded RDO performance in whole QP values. Especially, it shows very poor RD performance for the medium motion sequences such as the Foreman and Paris sequences. However, the suggested algorithm can provide a speed-up factor of 67-87% for all sequences with a negligible loss of the average PSNR and only a small bit increment.

Analysis of the overall performance: The comparison of the proposed algorithm by Kim and Kim (2008) and Zeng *et al*. (2009) methods is shown in Table 3. From this Table 3, it is very clear that the proposed algorithm gives less amount of bit rate (73% of bit rate improvement than the previous method) within very marginal PSNR degradation (0.01-0.02 dB). On the other hand for the speed up factor, the proposed method also gives also better result than the earlier methods. The method achieves 78.70% of time-saving on average value where the algorithms proposed by Kim and Kim (2008) and Zeng *et al*. (2009) give 73.19 and 71.492% of time saving factor on average respectively. In terms of PSNR, the proposed method gives so stable performance. For the Paris and Foreman sequences, the proposed algorithm achieved a quality loss of <0.15 (dB). Comparing to Kim and Kim (2008) and Zeng *et al*. (2009) methods, it is better performance. On the average PSNR performance, the method and the method proposed by Kim and Kim (2008) can provide a negligible loss of the average PSNR while the method proposed by Zeng *et al*. (2009) has large loss of PSNR (0.107 dB).

In case of the bit increment, the proposed method gave more reduced bit increment compared with other methods on the average value. For fast motion sequences like the Mobile and Stefan, there was large bit reduction or very small increment effect by the proposed algorithm. For the medium motion sequences such as the Foreman and Paris sequences somewhat large bit increment was observed using the proposed algorithms (Zeng *et al*., 2009). There was more stable performance of bit increment for these sequences (Kim and Kim, 2008). Also, the proposed method shows a little large bit increment at higher QP values (36 and 40) for some stationary sequences. This may be caused by a mismatch between a real threshold and the modeled threshold values because researchers made model for thresholds as QP changes, linearly.

For the time-saving factor, the proposed algorithm achieves 78.70% of time-saving on average value. With the medium motion sequences such as the Foreman and Flower sequences, the proposed algorithm speed-up about 9% compared by Kim and Kim (2008) and Zeng *et al*. (2009) Methods. When compared by Zeng *et al*. (2009), the suggested method can provide speed-up factor <12% with a minimal loss of image quality. For the stationary sequences like the Hall monitor, mother, daughter and akiyo, the method proposed by Zeng *et al*. (2009) gave more speed-up factor when comparing to the proposed algorithms (Kim and Kim, 2008). However, the method proposed by Zeng *et al*. (2009) caused large loss of quality (up to 0.814 dB) as researchers can shown from Table 3. The suggested

Table 3: Performance comparison with other methods (Kim and Kim, 2008; Zeng *et al*., 2009)

		Kim and Kim (2008)			Zeng *et al*. (2009)			Proposed		
Sequences	QP	PSNR	Bit rate	Time	PSNR	Bit rate	Time	PSNR	Bit rate	Time
Mobile	20	-0.040	0.015	-63.525	-0.017	0.369	-58.254	-0.067	0.373	-70.437
	28	-0.510	-0.473	-66.113	-0.018	0.761	-60.027	-0.083	-0.158	-72.511
	36	-0.066	-1.506	-70.746	-0.030	1.106	-59.729	-0.100	-1.451	-77.297
	40	-0.059	-0.742	-75.521	-0.065	0.913	-63.120	-0.078	-0.487	-80.906
Paris	20	-0.069	0.589	-73.909	-0.035	0.604	-61.810	-0.099	1.028	-77.865
	28	-0.057	1.004	-75.513	-0.050	1.080	-69.417	-0.073	1.138	-80.364
	36	-0.074	0.562	-77.113	-0.084	2.064	-78.181	-0.108	1.235	-82.727
	40	-0.104	0.300	-78.244	-0.138	2.216	-83.205	-0.112	0.251	-84.066
Mother and Daughter	20	-0.036	0.097	-72.618	-0.085	0.731	-71.235	-0.069	-0.186	-78.843
	28	-0.032	0.903	-77.195	-0.149	-0.253	-82.040	-0.052	0.972	-83.375
	36	-0.048	1.512	-79.101	-0.334	-1.890	-87.422	-0.014	1.601	-85.413
	40	-0.060	2.492	-80.075	-0.814	-3.927	-92.988	-0.099	1.537	-86.320
Hall monitor	20	-0.047	-0.853	-66.601	-0.032	0.252	-59.401	-0.089	-0.869	-72.300
	28	-0.059	-1.329	-77.989	-0.156	-4.550	-80.472	-0.087	-2.917	-83.793
	36	-0.104	1.549	-79.019	-0.032	-1.072	-91.183	-0.099	1.798	-86.535
	40	-0.090	1.537	-80.007	0.019	-0.032	-94.634	-0.163	2.230	-87.409
Container	20	-0.055	-0.204	-76.063	-0.026	0.029	-62.030	-0.087	-0.181	-81.993
	28	-0.054	0.584	-80.389	-0.060	-0.505	-74.442	-0.063	0.091	-85.689
	36	-0.080	0.316	-81.326	-0.080	-0.053	-86.973	-0.075	0.412	-86.601
	40	-0.067	0.000	-81.526	-0.151	-0.389	-91.953	-0.086	-0.597	-86.960
Stefan	20	-0.051	1.495	-68.877	-0.078	0.149	-63.797	-0.086	0.309	-75.122
	28	-0.074	2.874	-70.426	-0.103	0.728	-67.099	-0.097	0.089	-75.711
	36	-0.111	1.517	-74.144	-0.055	2.467	-69.528	-0.113	-0.303	-80.176
	40	-0.096	1.318	-77.363	-0.128	2.422	-73.241	-0.103	-0.013	-82.866
Akiyo	20	-0.038	0.178	-76.347	-0.055	0.289	-79.914	-0.044	0.397	-84.208
	28	-0.020	-0.499	-78.686	-0.094	1.057	-87.970	-0.017	-0.209	-86.414
	36	-0.039	-0.444	-80.292	0.218	-1.430	-91.188	-0.110	-0.897	-87.558
	40	-0.047	0.320	-80.573	-0.511	-2.107	-94.772	-0.056	0.067	-87.931
Foreman	20	-0.040	0.464	-64.487	-0.039	1.796	-61.323	-0.082	0.579	-71.877
	28	-0.049	0.515	-70.989	-0.098	3.628	-69.097	-0.092	1.073	-77.946
	36	-0.094	1.267	-76.049	-0.184	3.174	-76.738	-0.100	1.503	-82.347
	40	-0.136	1.450	-77.671	-0.458	1.537	-83.742	-0.150	1.328	-84.000
Football	20	-0.031	0.505	-64.383	-0.006	0.188	-60.104	-0.049	0.330	-67.367
	28	-0.027	1.150	-65.203	-0.004	0.630	-60.358	-0.038	0.539	-66.944
	36	-0.036	2.184	-69.641	-0.004	1.799	-62.096	-0.068	1.173	-74.327
	40	-0.072	3.085	-71.880	-0.013	2.374	-66.251	-0.101	1.709	-77.784
Average		-0.058	0.572	-74.100	-0.107	0.483	-73.383	-0.080	0.292	-79.972

method is faster than Kim and Kim (2008) for these stationary sequences even though a minimal loss of image quality. For sequences with fast motion component, the proposed algorithm based on the analysis of Motion Model can provide more time-saving factor of 10 and 6% where the algorithms proposed by Kim and Kim (2008) and Zeng *et al*. (2009), respectively.

CONCLUSION

In this study, researchers have proposed a novel scheme for fast inter-mode prediction techniques which can be applied for H.264/AVC. The proposed method is based on macroblock correlation and motion complexity analysis technique. An additional search process is used here to provide a better image quality and to reduce the bit rate. Experimental results show that the proposed algorithm gives superior result in terms of bit rate and speed-up factor with negligible loss of image quality.

ACKNOWLEDGEMENT

This research was supported by the National Research Foundation of Korea Grant funded by the Korean Government (MEST) (NRF-2010-0024786).

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.

Huang, H. and D. Hu, 2009. An efficient fast inter mode decision algorithm for H.264/AVC. Proceedings of the WRI International Conference on Communications and Mobile Computing, Volume 1, January 6-8, 2009, Yunnan, pp: 161-165.

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.

Liu, P. and K. Jia, 2009. A fast and novel intra and inter modes decision prediction algorithm for H.264/AVC based on the characteristics of macro-block. Proceedings of the 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, September 12-14, 2009, Kyoto, Japan, pp: 286-289.

Ostermann, J., J. Bormans, L. Peter, M. Detlev, N. Matthias and P. Fernando, *et al*., 2004. Video coding with H.264/AVC: Tools, performance and complexity. IEEE Circuits Syst. Magazine, 4: 7-28.

Richardson, I.E.G., 2003. H.264/MPEG-4 Part 10 white paper: Inter prediction April 2003. http://www4.rgu.ac.uk/files/h264_interpred.pdf.

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, October 8-11, 2006, Atlanta, GA., pp: 41-44.

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.

Yu, A.C.W., G.R. Martin and H. Park, 2008. Fast inter-mode selection in the H.264/AVC standard using a hierarchical decision process. IEEE Trans. Circuits Syst. Video Technol., 18: 186-195.

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.