Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **2**, March **2015**

Speed up Video Coders with Only Examining the Predicted Modes in a Coding Block

Win-Bin Huang
Department of Information Management,
Peking University
No.5 Yiheyuan Road Haidian District, Beijing,

Mong-Fong Horng
Department of Electronic Engineering
National Kaohsiung University of Applied Sciences
No.415, Jiangong Rd., Sanmin Dist., Kaohsiung City 807, Taiwan

Received March, 2014; revised September, 2014

Abstract. *In a rate-distortion optimized (RDO) video coder, the best mode with the minimal Lagrange cost is selected after calculating the rate-distortion cost which simultaneously considers the distortion and the coding bits of all possible modes. The proposed method, called two-staged mode decision (TSMD), employs a two-staged decision process: the first stage is to predict some probable encoding modes according to the information when one encodes the preceding macro-blocks and video frames. The second stage refines the decision with either the pre-computed look-up-table or the pre-constructed Back Propagation Neural (BPN) network. According to our experiment results, over 50% of the computation time is reduced with a small loss in Peak Signal-to-Noise ratio (PSNR) and a slight bit-rate is incremented which are not easily sensed by human eyes.*

Keyword: Video coding, Mode decision, Look-up-table, Rate distortion optimization, Back propagation neural network.

1. **Introduction.** The famous video coding standards H.264/AVC [1] and H.265/HEVC [33] proposed jointly by ITU-T and MPEG provide outstanding compression performance. There are many new features recommended by H.264/AVC standard, including motion estimation based on image blocks of various sizes, quarter-pixel motion estimation, complicated intra prediction, in-loop de-blocking filter and content-adaptive binary arithmetic coding (CABAC) [2]. Besides, a rate distortion optimization (RDO) algorithm [3] used in H.263 is also recommended in H.264/AVC as an option. The complexity of the mode prediction and block partitions during rate distortion optimization is also particularly influenced in the performance of HEVC coding scheme [34]. The H.264/AVC with rate-distortion optimization (H.264/RDO) is applied to the mode decision part of the encoder to select the most encoding probable modes for the sub-blocks of a macro-block (MB). RDO effectively improves both the compression ratio and the video quality of H.264/AVC with the rate-distortion (R-D) cost J generally expressed by

$$J = D + \lambda \cdot R \tag{1}$$

where λ is the Lagrange multiplier, D is the reconstruction distortion, and R is the number of coded bits of a MB [24] [26]. In a H.264/RDO coder, luma samples of a MB may be divided into sixteen 4×4 blocks when performing intra prediction. There are 9 modes for each 4×4 luma block. One can also use an entire luma MB for the prediction. For chroma part, a 8×8 block is used exclusively. There are four modes for a MB luma block, and four modes for a 8×8 chroma block, respectively. Therefore, the number of combinations for luma and chroma components for intra prediction a MB is 4×(9×16+4). That means one has to calculate 592 R-D costs before the actual coding modes are determined [4]. For inter prediction, there are seven block types (16×16, 16×8, $8times16$, 8×8, 8×4, 4×8, 4×4). If five reference frames are used, totally 4×(1+2+2+4+8+8+16)×5 motion estimations have to be completed before one can decide which mode gives the best RD-cost. Therefore all of the Lagrange costs used in equation 1 for all possible block partitions should be completed for the modes in a H.264/RDO coder, and the actual computation of the distortion and the bit consumption required for all candidate modes is quite large. For small systems such as mobile devices and DSP-based computing engines, the realization of standard RDO video coders becomes a very difficult work.

Many papers were proposed to reduce the computation complexity of a H.264/RDO coder and H.265/HEVC coder. For example, [4-7] are designed for intra prediction part, [8-10] are proposed for inter prediction part and [11-20], [30-32] are proposed for macro-block mode decision. [34-37] are also proposed for inter prediction in H.265/HEVC. The proposed method is designed for macro-block mode decision. In this paper, a mode decision algorithm called two-staged mode decision (TSMD) is proposed to speed up the encoding process with little sacrifice in quality. TSMD disables the useless modes of a coding block and only examines the modes which might be the best mode used in H.264/RDO coder. All comparisons were based on JM18.6 reference software and all programs are modified from JM18.6 only when it is necessary [29]. At first, to determine the coding mode at a MB, TSMD looks at the MB's neighborhood in its current and the last frames, moreover, the mode which is mostly used in the neighborhood is assigned to the MB. The second stage in TSMD chooses the most probable mode from the modes picked in the previous stage with either a pre-computed look-up-table or a pre-constructed back-propagation neural (BPN) network [21]. The look-up-table, which records the most probable mode of a MB according to the Baye's probability model when given a candidate mode in the first stage, makes the mode decision faster. Moreover, the BPN which is pre-constructed with the available information of a MB increases the precision of the mode decision. Since TSMD is a supervisor learning algorithm, we use sample video sequences for training a look-up-table and a BPN. The training phase is simple, and should be done for the H.264/RDO coding method at different quantization parameter (QP). All the parameters in the table and the BPN are generated by analyzing the training video sequences. The simulation results of eleven QCIF and six CIF sequences show that, in most cases, TSMD reduces at least 50% of the computation time with similar quality when compared to H.264/RDO of JM18.6. Moreover, six high resolution video sequences, like 1280x720 and full-HD, are adopted in our experiments with good performance by using the proposed algorithm.

The rest of this paper is organized as follows. Our observations of a H.264/RDO coder and the motivation of this paper are briefly explained in Section II. The proposed TSMD algorithm is presented in Section III. Experiment results and comparisons with other algorithms are shown in Section IV followed by the conclusion in Section V.

2. **Observations and Motivation.** The filtered frame is referred in the proceeding encoding process. H.264/AVC recommends RDO as an option proposed in [3] to improve

the compression efficiency. Thus, H.264/RDO introduces the R-D optimization formulated in Equation 1 to minimize the R-D cost function subject to the distortion and the number of bits used to encode the image blocks a number of constraints. Because the actual number of bits required to encode the image block is involved in its cost function, it is inevitable that one has to complete all the encoding process including DCT-like integer transform, quantization and so on for each possible mode before the actual cost can be derived. After the RD-costs of all possible coding modes are calculated, the mode with minimum RD-cost is chosen. According to the block size and the coding type of coding modes, in this paper, all possible coding modes of a MB are named as Skip, Inter 16x16, Inter 16x8, Inter 8x16, Inter 8x8, Inter 8x4, Inter 4x8, Inter 4x4, Intra 4x4 and Intra 16x16. To display the performance of a video coder, moreover, three statistics, bit rate, PSNR and computation time, are collected. No other program modifications are made to the JM18.6 except the mode decision part to maintain the result as fair as possible. The configuration used in the section is set as follows:

1. Baseline profile encoder is used.
2. GOP size is 30.
3. The number of reference frames is 10.
4. Frame rate is 30 fps.
5. Rate control and Hadamard transform is turned off.
6. Fixed QP.
7. Search range is ±16 pels.
8. Full search is used.
9. The QCIF “Foreman” sequence with 300 frames is used.

The programs run on a Pentium IV-3.0 GHz with 1 GB memory and Microsoft Window-XP sp2 system. In the paper, changes of PSNR, Bit-rate and Encoding-Time with respect to JM18.6 coder are calculated by using the Equation 2 4, respectively.

$$\Delta\mathrm{PSNR} = \mathrm{PSNR}_{\mathrm{JM(RDO)}} - \mathrm{PSNR}_{\mathrm{Compared}} \tag{2}$$

$$\Delta\mathrm{BitRate} = \frac{(\mathrm{BitRate}_{\mathrm{JM(RDO)}} - \mathrm{BitRate}_{\mathrm{Compared}})}{\mathrm{BitRate}_{\mathrm{JM(RDO)}}} \times 100\% \tag{3}$$

$$\Delta\mathrm{Time} = \frac{(\mathrm{Time}_{\mathrm{JM(RDO)}} - \mathrm{Time}_{\mathrm{Compared}})}{\mathrm{Time}_{\mathrm{JM(RDO)}}} \times 100\% \tag{4}$$

In order to predict whether a mode should be disable or not, two definitions in the paper are required: the first is to partition the coding modes into four sets according to block size and characteristics of coding modes. The four mode sets are Skip, Intra, Inter_Big, and Inter_Small, and each set may contain one or more inclusive modes, as shown in Table 1.

Table 1. Mode Groups.

Sets	No. of Sets	Inclusive Modes
Skit	1	Skip
Intra	2	Intra 16x16, Intra 4x4
Inter_Big	3	Inter 16x16, 16x8 and 8x16
Inter_Small	4	Inter 8x8, 8x4, 4x8 and 4x4

The other definition is to collect the modes of the thirteen MBs which have been processed, as shown in Fig. 1. The locations of the MBs are all the current MBs neighborhood which has been processed in the current frame and nine MBs in the previous frame. Moreover, the location of the center one of those MBs in the previous frame is the same as that of the current one.

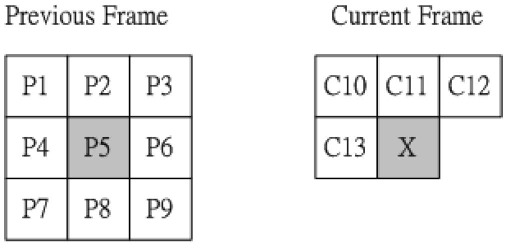

Figure 1. Thirteen reference MBs. The X is the current MB.

Furthermore, the mode set that appears most frequently in the above thirteen MBs is selected as the candidate mode set of the current MB. Once the current MBs mode set is selected, the inclusive modes in the candidate mode set are only enabled and all the encoding processes of each inclusive mode are completed before the actual cost can be derived. After the RD-costs of all the inclusive modes are calculated, the mode with minimum RD-cost is chosen. Since the MBs of the first Inter frame in a group of pictures (GOP) have no coding mode which can be referred, they are encoded by the original H.264 with RDO algorithm. Besides the Intra frame is also encoded by the original algorithm. Therefore, we gathered the encoding mode set of all MBs in some sequences at QP=24 except for the two frames of all GOPs. In the next section, therefore, a two-staged mode selection (TSMD) algorithm is proposed to improve image quality of the above prediction method; meanwhile, its execution time is also quite small.

3. **Two-Staged Mode Decision Algorithm (TSMD).**

For reducing the computation, a two-staged mode decision (TSMD) algorithm is proposed to disable some useless modes and only examine the probable modes which might be the best one used in H.264/RDO coders.

3.1. **The TSMD's first stage.**

At first, TSMD also collects the thirteen MBs' modes mentioned in last section. The mode set that appears most frequently in the thirteen MBs is selected as the candidate mode set, $PreSet_1$. This predicted mode set $PreSet_1$ is one of the four coding mode sets, Skip, Intra, Inter_Big, and Inter_Small. Then, the RD-costs of all the inclusive modes in PreSet1 are calculated, the mode with minimum RD-cost is chosen. In the stage, moreover, the "Skip" set is always checked because bits can be saved without sacrificing the quality too much if "Skip" is the most probable set and the computation to check the "Skip" set is little. However, Table 4 already reveals that PreSet1 may be not the optimal one, so the second stage is required. The coding flow of the TSMD's first-stage is shown in Fig. 2.

3.2. **The TSMD's second stage.**

Briefly, the coding flow of the second stage is shown in Fig. 3. The second predicted mode set, $PreSet_2$, is first obtained by the pre-computed look-up-table method. $PreSet_2$ is selected by the pre-constructed BPN networks whenever necessary. Then, the RD cost parameters of all modes in $PreSet_2$ are calculated. The results are compared with those obtained in the first stage. The modes has minimal RD cost is selected as an optimal mode. The detail description of the TSMD's second

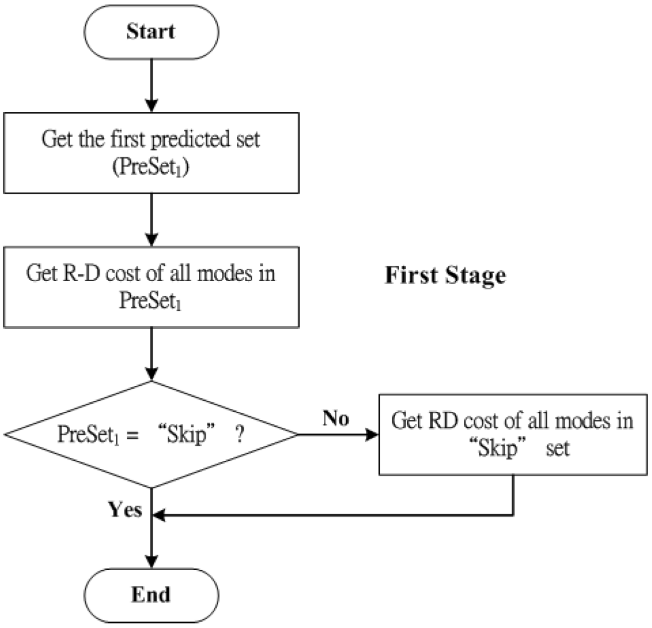

FIGURE 2. The coding flow of the TSMD's first stage.

stage is shown as follows. Moreover, complete computer simulations can be seen in section IV.

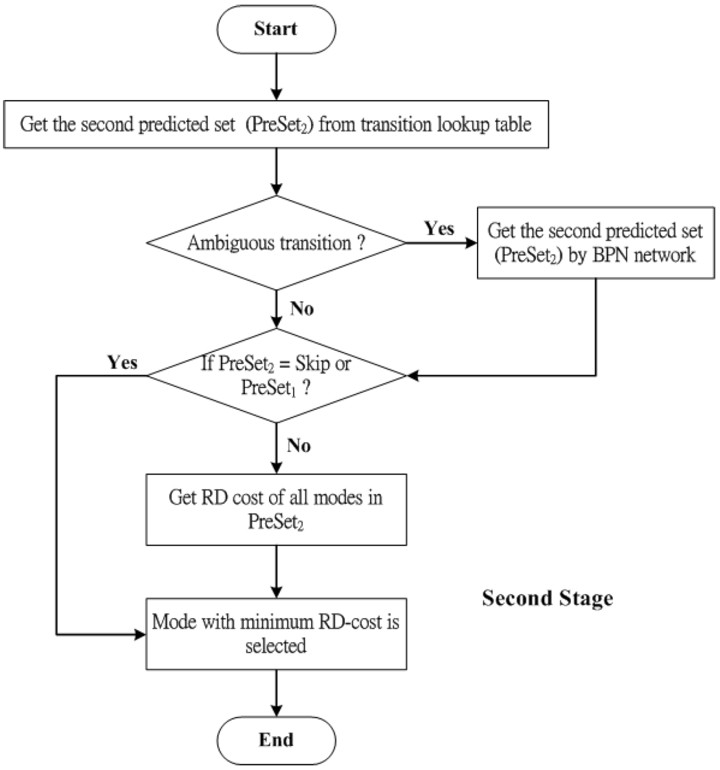

FIGURE 3. The coding flow of the second stage of TSMD algorithm.

3.3. **Overall TSMD algorithm.** The simple TSMD algorithm is summarized in the following pseudo-code.

```
For (all frames)
{
 If (I frame or the first Inter frame of GOP)
     Encode with H.264/RDO method.
 Else
     { // Stage 1
     Gather 13 referenced MBs' coding modes.
     Assign $PreSet_1$ = the mode set which appears most frequently in 13 MBs.
     Calculate R-D cost of all modes in $PreSet_1$.
     If ($PreSet_1 \neq$ "Skip")
         Calculate R-D cost of all modes in "Skip" mode set
     Assign the Best mode = the mode having the minimal R-D cost in Skip and $PreSet_1$.
       // Stage 2
     Get the SSD of the best mode selected in stage 1.
     Obtain the region of the SSD and the current QP.
     Assign $PreSet_2$ = LUT($PreSet_1$, $R_{SSD}$, QP).
     If ($PreSet_1$ = "Ambiguous Transition")
         Obtain the bit consumption used in texture coding and MV coding
         when the mode selected in stage 1 is encoded.
         $PreSet_2 = BPN_{PreSet_1}$(SSD, QP, $Bits_{texture}$, $Bits_{MV}$ ).
     If ($PreSet_2 \neq$ "Skip" or $PreSet_1$)
         Calculate R-D cost of all modes in $PreSet_2$.
     Select the best mode = the mode having the minimal R-D cost in Skip, $PreSet_1$ and $PreSet_2$.
     }
}
```

4. **Experiment Results.** 300 frames of eleven QCIF and six high-resolution video sequences are used. Both slow motion and fast motion video sequences are included. JM18.6 is the base of all the testing software encoders. No other program modification is made to the JM18.6 except the mode decision part to maintain the result as fair as possible.

The experiments are divided into two parts: inside testing and outside testing. Five QCIF video sequences, "Akiyo", "Foreman", "Carphone", "Coastguard" and "Mobile", are used to construct the look-up-table and all the parameters of four BPN networks. Experiment results of the training sequences and the other sequences are shown in inside testing and in outside testing, respectively. Tables 2 show related information when encoding various sequences with QP=24.

4.1. **Inside Testing.** Tables 3 and 4 show that Foreman sequence is encoded at QP = 18, 24 and 30, when Hadamard transform is turned on and fast motion estimation (FME) is used, respectively. In the tables, the PSNR and bit-rate performance of TSMD is middle high among the three methods, but its reduction in executing time is also quite significant. TSMD is to enable and examine the predicted modes, and no other changes of the encoding algorithm of an MB are taken place. According to the experiments shown in tables, the improvements of the TSMD's encoding efficiency are similar whether or not Hadamard transform and FME are used.

Table 2. My caption

Sequence	Type	PSNR Y(dB)	ΔPSNR Y(dB)	Bit Rate (kb/s)	ΔBit Rate (%)	Total Time(s)	ΔTime (%)
Foreman QCIF	RDO Only	38.85	0	232.71	0	3092.65	0
	No RDO	38.65	0.2	243.89	-4.8	2449.82	20.8
	TSMD	38.74	0.11	238.66	-2.55	1450.14	53.11
AKiyo QCIF	RDO Only	41.352	0	64.65	0	1286.40	0
	No RDO	40.987	0.365	70.18	-8.54	1129.09	12.22
	TSMD	41.045	0.307	65.12	-0.72	169.22	86.84
Carphone QCIF	RDO Only	39.77	0	257.45	0	2150.72	0
	No RDO	39.5	0.27	270.05	-4.89	1957.64	8.97
	TSMD	39.585	0.185	259.39	-0.75	1135.13	50.22
Coast-guard QCIF	RDO Only	37.29	0	416.72	0	3127.03	0
	No RDO	37.141	0.149	428.12	-2.37	2907.90	7.01
	TSMD	37.221	0.069	427.58	-2.60	1294.74	58.59
Mobile QCIF	RDO Only	36.752	0	748.10	0	2754.73	0
	No RDO	36.587	0.165	769.28	-2.83	2510.31	8.87
	TSMD	36.689	0.063	760.10	-1.60	1377.20	50.01

Table 3. Executing results of Foreman QCIF sequence with 300 frames at different QPs when Hadamard transform is turned on.

QP	Type	PSNR Y (dB)	ΔPSNR Y (dB)	Bit Rate (kb/s)	ΔBit Rate (%)	Total Time (s)	ΔTime (%)
18	RDO	43.60	0	524.37	0	2955.73	0
	No RDO	43.54	0.053	547.79	-4.46	2743.34	7.18
	TSMD	43.38	0.222	544.48	-3.83	1367.56	53.73
24	RDO	38.98	0	234.08	0	2777.78	0
	No RDO	38.97	0.0118	244.51	-4.45	2572.31	7.39
	TSMD	38.862	0.118	239.12	-2.15	1330.92	52.08
30	RDO	34.86	0	110.46	0	2556.75	0
	No RDO	34.85	0.015	115.64	-4.68	2361.92	7.62
	TSMD	34.65	0.208	113.05	-2.34	925.75	63.79

4.2. **Outside Testing.** “Container”, “Hall_Objects”, “Mother_and_Daughter”, “News”, “Silent” and “Stefan”, are used in this part. Both QCIF and CIF formats of these sequences are included. All configurations of the encoder are the same as the list mentioned in the above section. Again, the outside testing results at QP=24 are shown in Table 5, and six QCIF/CIF video sequences are processed by JM18.6 with RDO, JM18.6 without RDO, TSMD, respectively. Therefore, it is observe that the proposed TSMD also outperforms JM18.6 without RDO in terms of both encoding time and compression ratio in most cases. Compared to JM18.6 with RDO, the PSNR loss is also within 0.4 dB and the bit rate increase is within 10% for the worst case. In some cases, the bit-rates are even less than those produced by JM18.6 with RDO, of course, with sacrifice of quality. The reduction in encoding time is also quite significant for both slow and fast motion sequences, such as Stefan and Container, respectively.

Table 4. Executing results of Foreman QCIF sequence with 300 frames at different QPs when FME is used.

QP	Type	PSNR Y (dB)	ΔPSNR Y (dB)	Bit Rate (kb/s)	ΔBit Rate (%)	Total Time (s)	ΔTime (%)
18	RDO	43.47	0	525.91	0	839.64	0
	No RDO	43.30	0.174	544.93	-3.61	653.10	22.21
	TSMD	43.30	0.174	543.78	-3.39	371.06	55.80
24	RDO	38.84	0	232.73	0	811.92	0
	No RDO	38.64	0.193	242.26	-4.09	653.75	19.48
	TSMD	38.73	0.0102	238.38	-2.43	395.62	51.27
30	RDO	34.69	0	108.79	0	788.73	0
	No RDO	34.44	0.251	112.90	-3.77	650.68	17.50
	TSMD	34.5	0.19	110.81	-1.85	392.45	50.24

Table 5. Experiment results of the QCIF outside testing at QP=24.

Sequence	Type	PSNR Y(dB)	ΔPSNR Y(dB)	Bit Rate (kb/s)	ΔBit Rate (%)	Total Time(s)	ΔTime (%)
Container QCIF	RDO Only	38.791	0	100.06	0	1926.89	0
	No RDO	38.606	0.185	115.26	-15.1	1726.68	10.38
	TSMD	38.692	0.099	110.35	-0.28	695.86	63.88
Hall-Objects QCIF	RDO Only	40.106	0	113.90	0	1401.18	0
	No RDO	39.82	0.286	132.08	-15.96	1215.81	13.22
	TSMD	39.907	0.199	110.51	2.97	340.71	75.68
Mother and Daughter QCIF	RDO Only	40.361	0	91.89	0	1656.79	0
	No RDO	40.066	0.295	99.44	-8.21	1480.81	10.62
	TSMD	40.215	0.146	92.68	-0.86	581.046	64.92
News QCIF	RDO Only	39.896	0	144.73	0	1614.90	0
	No RDO	39.567	0.329	154.60	-6.82	1435.62	11.10
	TSMD	39.744	0.152	147.61	-1.98	491.094	69.58
Silent QCIF	RDO Only	39.013	0	164.23	0	1822.46	0
	No RDO	38.575	0.438	172.34	-4.93	1645.20	9.72
	TSMD	38.907	0.106	167.80	-2.17	771.81	57.65
Stefan QCIF	RDO Only	37.493	0	958.69	0	3392.03	0
	No RDO	37.286	0.207	988.37	-3.09	3179.81	6.25
	TSMD	37.329	0.164	977.88	-2.01	1324.01	60.96

Furthermore, in Table 6, the proposed speed-ups are compared with other algorithms in different sequences. It is noted that the reduction in encoding time is quite significant when TSMD is applied. Therefore, it is more practical to implement a H.264 coder in small systems such as mobile devices and DSP-based computing engines. Furthermore, it is our future work to decrease the losses in coding efficiency under the proposed framework.

TABLE 6. Comparison of four fast mode decision algorithms with different sequences.

Algorithms		TSMD	Kim's [30]	Zeng's [31]	Hilmi's [32]
Akiyo	DPSNR (dB)	-0.01	-0.04	-0.11	-0.05
	DBR (%)	0.25	-0.36	0.58	-0.08
	ΔTime (%)	88.11	78.56	87.12	86.20
Container	DPSNR (dB)	-0.068	-0.05	-0.06	-0.07
	DBR (%)	0.701	0.34	0.09	0.23
	ΔTime (%)	81.73	80.12	74.12	85.41
Stefan	DPSNR (dB)	-0.10	-0.09	-0.10	-0.10
	DBR (%)	0.50	2.11	0.98	-0.12
	ΔTime (%)	77.84	70.64	66.99	76.08
Foreman	DPSNR (dB)	-0.145	-0.06	-0.12	-0.10
	DBR (%)	1.548	0.71	3.37	0.96
	ΔTime (%)	78.33	71.01	68.84	77.83

5. **Conclusion.**

Because the rate-distortion optimized H.264/AVC coders has to complete the process, which includes transform, quantization and entropy coding, of all modes of a coding block, the best mode is picked up according to the minimum Lagrange cost function. Though the H.264/RDO coders achieve a good balance between bit rate and image distortion, lots of additional computing power is demanded. The TSMD proposed uses the Baye's probability model and BPN to predict the modes which might be the best one used in H.264/RDO. Besides, the best mode having the minimal Lagrange cost is chosen when only the modes predicted in TSMD are completed. TSMD successfully reduces the computation complexity of H.264/AVC with RDO. The loss in video quality is slight and the bit rate increase is not much. Compared to the original encoder without RDO, TSMD is still better even for fast moving sequences, such as Foreman and Stefan sequences. Furthermore, TSMD has good performance in high resolution video sequence, such as 1280x720 and 1920x1080, in our experiment. The proposed algorithm, however, may have some further modifications to achieve better performance. For example, TSMD has to examine all the modes in a certain mode set before the decision is made. It is possible to reduce the number of modes to examine to decide the final mode. There are still methods other than mode decision to combine with the proposed algorithm to reduce the computation time of the H.264/RDO encoder.

REFERENCES

[1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 — ISO/IEC 14496-10 AVC), *ISO/IEC JTC1 /SC29/WG11*, 2003.

[2] D. Marpe, H. Schwarz, and T. Wiegand, Context-based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard, *IEEE Trans. Circuits Sys. Video Technol.*, vol 13, pp. 620-636, July 2003.

[3] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, Rate-Distortion Optimized Mode Selection for Very Low Bit Rate Video Coding and Emerging H.263 Standard, *IEEE Trans. Circuits Sys. Video Technol.*, vol. 6, pp. 182-190, 1996.

[4] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu and S. Wu, Fast Mode decision Algorithm for Intraprediction in H.264/AVC Video Coding, *IEEE Trans. Circuits Sys. Video Technol.*, vol. 15, pp. 813-822, 2005.

[5] C. H. Tseng, H. M. Wang, and J. F. Yang, Improved and Fast Algorithms for Intra 4x4 Mode Decision in H.264/AVC, *Proc. IEEE Int. Conf. Circuits and Systems*, vol. 3, pp. 2128-2131, 2005.

[6] Y. K. Lin and T. S. Chang, Fast Block Type Decision algorithm for Intra Prediction in H.264 FRExt, *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 585-588, 2005.

[7] J. U. Rehman and Y. Zhang, Fast Intra Prediction Mode Decision Using Parallel Processing, *Proc. IEEE Int. Conf. Machine Learning and Cybernetics*, vol. 8, pp. 5094-5098, 2005.

[8] 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 Trans. Circuits Sys. Video Technol.*, vol. 15, pp. 953-958, 2005..

[9] C. H. Kuo, M. Shen, and C.-C. J. Kuo, Fast Inter-Prediction Mode Decision and Motion Search for H.264, *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 1, pp. 663-666, 2004.

[10] Q. Dai, D. Zhu, and R. Ding, Fast Mode Decision for Inter Prediction in H.264, *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 119-122, 2005.

[11] H. Zhu, C. K. Wu, Y. L. Wang, and Y. Fang, Fast Mode Decision for H.264/AVC based on Macroblock Correlation, *Proc. IEEE Int. Conf. Advanced Information Networking and Applications*, vol. 1, pp. 775-780, 2005.

[12] E. arsura, L. D. Vecchio, R. Lancini, and L. Nisti, Fast Macroblock Intra and Inter Modes Selection for H.264/AVC, *Proc. IEEE Int. Conf. Multimedia and Expo*, pp. 378-381, 2005.

[13] D. Zhang, Y. Shen, S. Lin, and Y. Zhang, Fast Inter Frame Encoding Based on modes Pre-decision in H.264, *Proc. IEEE Int. Conf. Multimedia and Expo*, pp. 530-533, 2005.

[14] X. J. Zhu and S. A. Zhu, Fast Mode Decision and Reduction of Reference Fames for H.264 Encoder, *Proc. IEEE Int. Conf. Control and Automation*, vol. 2, pp. 1040-1043, 2005.

[15] T. Y. Kuo and C. H. Chan, Fast Macroblock Partition Prediction for H.264/AVC, *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 1, pp. 675-678, 2004.

[16] K. H. Han and Y. L. Lee, Fast Macroblock Mode Decision in H.264, *Proc. IEEE Int. Conf. TENCON*, vol. 1, pp. 347-350, 2004.

[17] Z. Fan and Z. Xudong, Fast Macroblock Mode Decision in H.264, *Proc. IEEE Int. Conf. Signal Processing*, vol. 2, pp. 1179-1182, 2004.

[18] Y. H. Kim, J. W. Yoo, S. W. Lee, J. Shin, J. Paik and H. K. Jung, Adaptive Mode Decision for H.264 Encoder, *Electronics Letters*, vol. 40, pp. 1172-1173, 2004.

[19] J. Lee and B. Jeon, Fast Mode Decision for H.264, *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 2, pp. 1131-1134, 2004.

[20] K. P. Lim, S. Wu, D. J. Wu, S. Rahardja, X. Lin, F. Pan, and Z. G. Li, Fast INTER mode selection, *JVT of MPEG and VCEG*, Doc. JVT-J033, 2003.

[21] S. Haykin, Neural Networks, *A Comprehensive Foundation*, 2nd Ed. Prentice Hall, NJ, 1999, ISBN: 0-13-273350-1.

[22] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, Overview of the H.264/AVC video coding standard, *IEEE Trans. Circuits Sys. Video Technol.*, vol. 13, pp. 560-576, 2003.

[23] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, Adaptive Deblocking Filter, *IEEE Trans. Circuits Sys. Video Technol.*, vol. 13, pp. 614-619, 2003.

[24] G. J. Sullivan and T. Wiegand, Rate-distortion optimization for video compression, *IEEE Signal Process. Mag.*, vol. 15, no. 11, pp. 74V90, 1998.

[25] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, Rate-constrained coder control and comparison of video coding standards, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688V703, 2003.

[26] G. J. Sullivan and R. L. Baker, Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks, *Proc. IEEE Int. Conf. Global Telecomm*, pp. 85V90, 1991.

[27] G. J. Sullivan and G. Bjontegaard, Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-resolution Progressive-scan Source Material, ITU-T VCEG, Doc. VCEG-N81, 2001.

[28] G. Bjontegaard, Calculation of Average PSNR Difference between RDcurves, ITU-T VCEG, Doc. VCEG-M33, 2001.

[29] Joint Model (JM), H.264/AVC Reference Software. [Online].
Available: http://iphome.hhi.de/suehring/tml/download.

[30] Huanqiang Zeng, Canhui Cai and Kai-Kuang Ma, Fast Mode Decision for H.264/AVC Based on Macroblock Motion Activity, *IEEE Trans. On Circuits And Systems For Video Technology*, vol. 19, no. 4, 2009.

[31] Jong-Ho Kim and Byung-Gyu Kim, Fast Block mode decision algorithm in H.264/AVC video coding, J.Vis.Communication Image.R 19, pp 175-183, 2008.

[32] Badrul Hilmi, Kalyan Goswami, Jong-Hyuk Lee, and Byung-Gyu Kim, Fast Inter-mode Decision Algorithm for H.264/AVC Using Macroblock Correlation and Motion Complexity Analysis' *Proc. IEEE Int. Conf. Consumer Electronics*, pp. 90-91, 2012.

[33] ITU-T Recommendation H.265, High efficiency video coding, International Telecommunication Union, 2013.

[34] I. K. Kim, J. Min, T. Lee, W. J. Han and J. Park, Block partitioning structure in the HEVC Standard, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1697-1706, 2012.

[35] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, Comparison of the coding efficiency of video coding standards Vincluding High Efficiency Video Coding (HEVC), *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669-1684, 2012.

[36] Y. Yuan, I. K. Kim, X. Zheng, L. Liu, X. Cao, S. Lee, M. S. Cheon, T. Lee, Y. He, and J. H. Park, Quadtree based non-square block structure for inter frame coding in HEVC, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1707-1719, 2012.

[37] F. Bossen, B. Bross, K. Suhring, and D. Flynn, HEVC complexity and implementation analysis, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685-1696, 2012.