TWO-STAGE MODE SELECTION OF H.264/AVC VIDEO ENCODING WITH RATE DISTORTION OPTIMIZATION

Win-Bin Huang, Yi-Li Lin**, Hung-Wei Cheng*, Alvin W.Y. Su***, and Yau-Hwang Kuo**

Dept. of CSIE, National Cheng-Kung University, Tainan, Taiwan, ROC
*{huangwb, chenghw, kuoyh}@cad.csie.ncku.edu.tw
** yllin@csie.ncku.edu.tw, *** alvinsu@mail.ncku.edu.tw

ABSTRACT

In this paper a two-stage mode selection (TSMS) algorithm is presented to speed up the H.264/AVC video encoding process with rate distortion optimization (RDO). However, lots of additional computing power is required and this makes the realization of H.264/RDO in a resource-limited system very difficult. The proposed TSMS employs a two-stage decision process: the first stage is to predict some probable encoding modes according to the information when one encodes the preceding macroblocks and video frames. The second stage refines the decision with techniques based on Baye's probability rule and Back-Propagation neural network (BPN). According to the experiment results, over 50% of the computation time is reduced with very slight loss in peak signal-to-noise ratio (PSNR) and a slightly increment in bit rate when TSMS is applied. The TSMS is even faster than the encoding program with running RDO part. All programs are based on the H.264/AVC standard reference software (JM 9.2).

1. INTRODUCTION

THE newest video coding standard H.264/AVC [1] proposed jointly by ITU-T and MPEG provides outstanding compression performance. There are many new features recommended by the new standard, including variable block size motion estimation, quarter-pel motion estimation, complicated intra prediction, in-loop deblocking filter and content-adaptive binary arithmetic coding (CABAC) [16]. Besides, a rate distortion optimization (RDO) algorithm [14] used in H.263 is also recommended in H.264/AVC. RDO is applied to the mode decision part of the encoder to select the most probable mode for blocks. The cost function used in the mode decision has been modified to contain both distortion and actual compressed data size. RDO effectively improves both the compression ratio and the video quality of H.264/AVC. However, calculating the new cost function requires additional information, and many extra operations have to be added to the already complicate encoding process. For small systems such as mobile devices and DSP-based computing engines, embedding a H.264/RDO encoder becomes a very difficult work.

Many papers were proposed to reduce the computation complexity of H.264/AVC with RDO. They can be roughly classified into two categories for computation reduction in either intra prediction or inter prediction. For example, [2]-[6] and [9] are designed for intra prediction part and [7]-[8] and [11] are proposed for inter prediction part.

In this paper, a novel algorithm called the two-stages mode selection (TSMS) is proposed to achieve similar performance and requires less computation time compared to JM9.2. The proposed algorithm selects the most probable mode in 2 stages. In the first stage, a rough mode selection chooses the candidates based on the recorded information when coding the preceding macroblocks and video frames. The second stage chooses the most probable modes from those provided in the previous stage by using either a Baye's probability model or the back-propagation neural network (BPN) [13]. The decision model parameters are generated by analyzing many video sequences. Simulation results of six video sequences are presented in this paper. The results show that the TSMS can reduce at least 50% of the computation time with similar quality compared to H.264/AVC standard with RDO. It is even faster than the encoder without RDO with similar video quality. It is noted that all programs are modified from JM9.2. Most parts of the programs are identical to the original reference software.

The rest of this paper is organized as follows. H.264/AVC is briefly reviewed in chapter 2. The proposed algorithm is presented in chapter 3. The experiment result is shown in chapter 4 followed by a conclusion in chapter 5.

2. CODING FLOW OF H.264/AVC

2.1. Forward Coding Path

The general coding flow of H.264 can be found in [15]. Video frames are encoded macroblock by macroblock in H.264. For I frames, intra prediction is applied to each macroblock based on the un-filtered previously encoded blocks. As to P frames, various coding modes [15] can be used, but only the most suitable one will be applied. Usually, mode decision is done by obtaining the mode has minimum Sum of Absolute Difference (SAD). Once the mode is chosen, the predicted data is used to subtract from the current macroblock to produce a residual macroblock which is transformed, quantized and entropy-coded to generate the final bitstream. The coded macroblock has to be reconstructed. After the current frame is coded, in-loop filter [12] is applied and the result becomes a future reference when encoding other video data.

2.2. H.264 with RDO

H.264/AVC recommends the rate distortion optimization (RDO) method proposed in [14] to improve the compression efficiency. The basic idea of RDO is to find the optimal balance between resultant data size and image quality. By observing the coding result, the data sizes of using different coding modes may be different even though the respective distortions are identical. Similarly, it is also possible that the data sizes are the same but the distortions are different. The reason is quite complicated because the number of bits, required to encode information such as prediction modes, motion vectors, the index of the referred frame and so on, has great influence on the coding efficiency. Thus, it is not good enough to determine the coding modes by using the distortion measure (SAD or SSD) only. To find the match of distortion and the corresponding bit rate, H.264/RDO uses a new cost function called the R-D cost. EQ.1 shows the cost function.

$$\text{RD-cost} = \text{Distortion} + (\text{Lagrange Multiplier}) \times (\text{Bits}) \quad (1)$$

Lagrange Multiplier is a popular technique to optimize a certain function subject to a number of constraints. In EQ.1, the function to be minimized is the distortion measure and the constraint function is the number of bits used to encode the image block. Since it includes the actual number of bits required to encode the image block 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 modes before the actual cost can be derived. One can imagine that the increased amount of computation is large. In our experience, at least 30% of increase is necessary compared to the original H.264 encoder with RDO if JM9.2 is used.

It is noted that *Lagrange Multiplier* parameters are pre-computed for different quantization parameters (QP). The Distortion measure is the sum of squared distance (SSD) instead of SAD. After the RD-costs of all available coding modes are calculated, the mode with minimum RD-cost is chosen. The coding flow with RDO is shown in Fig 1.

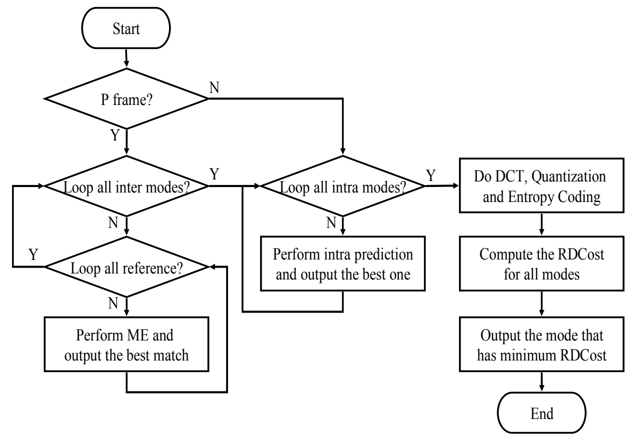

Fig. 1. Coding flow of H.264/AVC with RDO.

3. THE PROPOSED ALGORITHM

In stead of completing the computation for all the possible prediction modes, it is more practical to perform the operations only on part of inter/intra modes.The proposed algorithm uses a two-stage decision process to find the more probable modes. In the proposed algorithm, the coding modes are divided into four types: Big, Small, Skip and Intra as shown in Table I. Each type may contain one or more modes. The proposed algorithm tries to choose only part of the types for the decision making process to reduce the computation. The detail coding flow is shown in Fig.2.

TABLE I
Mode Classification

Type	No. of Mode	Mode
Big	3	Inter 16x16, 16x8 and 8x16
Small	4	Inter 8x8, 8x4, 4x8 and 4x4
Skip	1	skip
Intra	2	Intra 4x4, intra 16x16

3.1. The First Stage

In this stage, the available information is from the encoding process for the previously-coded video frame as well as the macroblocks of the current frame. The types of 9 macroblocks from the same spatial position and the adjacent positions of the previous frame, and 4 adjacent macroblocks of the current frame are used to predict the possible type for the current macroblock. The candidate type, denoted by **PreTp1**, is selected according to the type that appears most frequently in the types of the above 13 macroblocks. The statistics performed on many video sequences shows that 84% hit rate is achieved in this stage if **PreTp1** is also the optimal type. Furthermore, bits can be saved without sacrificing the quality too much if skip type is the most probable type. Since the computation to check the skip type is not much, the skip type is always checked in this stage.

3.2. The Second Stage

In this stage, a refined decision is performed based on the result of the previous stage for it is possible that the type

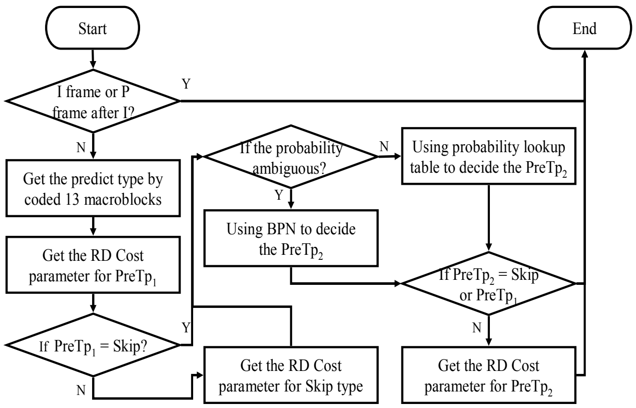

Fig.2. The coding flow of the proposed algorithm.

chosen in the previous stage may not be the optimal one. Let the probability of $\mathbf{PreTp_1}$ hitting the optimal type be P_1. Then, P_2, P_3 and P_4 are the probabilities that $\mathbf{PreTp_1}$ didn't hit optimal type and the optimal type may be one of type-2, type-3 or type-4, respectively. Thus, the type selection problem can be transformed into a state transition problem with different transition probabilities among the states. Each of the type is considered as a state. The probability staying in the $\mathbf{PreTp_1}$ is P_1. The transition probabilities to the other 3 states given the current state are P_2, P_3 and P_4, respectively. This is formulated as,

$$\textit{Transition Probability} = P_t(x \mid s). \qquad (2)$$

In EQ.2, s represents the current state and t represents the destination states. The transition probability is a function of SSD of the current state. To be specific, $P_t(x|s)$ is defined as the probability it will transit into type t when the average SSD equals to x given that $\mathbf{PreTp_1}$ is s. Because the continuous variable x represents the transition probability precisely, the probability table size may be too large. Therefore, we divide the range of x into 6 regions. EQ.2 is rewritten into EQ.3.

$$\textit{Transition Probability} = P_t(x_R \mid s) \qquad (3)$$

To select the most probable type is to pick out the type with the highest transition probability. The second stage prediction type is denoted as $\mathbf{PreTp_2}$. Since the original type selection problem has been transformed into a state transition problem, the only thing left is to build the transition probability table. We use 5 different video sequences, Akiyo, Foreman, Coastguard, Mobile and Carphone, to build it. These sequences are encoded using the original JM9.2 encoder with RDO and the information obtained in the encoding process is used to build the table.

In some situations, two transition probabilities of a given state s are close to each other and it is hard to make the decision. When the difference between two states is smaller than 0.1, the transition is ambiguous in this paper. Therefore, another decision making algorithm is necessary. In this paper, a back propagation neural network (BPN) is involved. In such cases, BPN replaces the previous method and decides the type of $\mathbf{PreTp_2}$.

Thus, the problem is changed into a classification problem. BPN tries to classify the final type using the information obtained when encoding the currently macroblock. After deciding **PreTp1** in the first stage, RD Costs of different modes are obtained. Thus, information about SSD, MV bits and coded macroblodck bits are also available. These are used as inputs of the BPN. Similarly, the BPN off-line training employs the information when the 5 sequences are encoded with JM9.2. The proposed BPN has two hidden layers.

4. EXPERIMENT RESULT

Several video sequences are used to test the proposed algorithm. JM 9.2 is the base of all the testing software encoders. The motion estimation algorithm has been modified from full search to diamond search (DS) [10]. To compare the performance between the proposed and the original encoder w/wo RDO, three statistics are collected. They are bit rate, PSNR and the compression time. No other program modification is made to the JM9.2 except the mode decision part to maintain the result as fair as possible. The encoder configured as follows. Baseline profile encoder is used. GOP size is 30. The number of reference frames is 10. Frame rate is 30 fps. Rate control is turned off. 300 frames of 12 QCIF sequences are tested. Some of them are of slow motion and some of them are of fast motion. 7 different QPs, that are 4, 12, 20, 28, 36, 44 and 51, are used for all sequences and coding schemes. Among the 12 sequences, 5 of them are used to generate the parameters of the probability model and the BPN. 7 of them are not used to generate the parameters. Thus, the experiments are divided into two parts: inside testing and outside testing.

4.1. Inside Testing

Akiyo, carphone, coastguard, foreman and mobile are used in this part. The proposed algorithm outperforms JM9.2 w/o RDO in terms of compression time and coding efficiency. Compared to JM9.2 with RDO, the PSNR loss is within 0.5 dB and the increase in bit rate is around 2%. However, compression time reduction is greatly reduced. TABLE II shows the results of three sequences with QP=28 noly. More results will be reported elsewhere. For slow moving sequences, the time saving is up to 89% with little PSNR loss. The improvement is not as significant for fast moving sequences. It is noted that different degrees of improvements for other QPs are achieved.

4.2. Outside Testing

Seven sequences are used in this part. They are container, hall objects, mother and daughter, news, salesman, silent and stefan. Similarly, TABLE III shows only the results of three sequences with QP equals to 28.

Compared to JM9.2 with RDO, the reductions in compression time are about 84%, 80% and 50%, respectively. The proposed algorithm also outperforms the original encoder without RDO in every aspect for all 7 sequences. The experiment results show that the proposed algorithm works well for outside testing sequences.

TABLE II
EXPERIMENT RESULT OF INSIDE TESTING SEQUENCES (QP=28)

Sequence		PSNR Y(dB)	Bit Rte(kb/s)	Total Time(s)
Akiyo	RDO	38.59	39.48	561.36
	No RDO	38.25	42.67	401.44
	TSMS	38.31	38.55	61.15
Foreman	RDO	35.95	151.48	719.43
	No RDO	35.72	160.10	544.23
	TSMS	35.77	154.83	265.39
Mobile	RDO	33.30	418.90	708.83
	No RDO	33.16	439.52	493.34
	TSMS	33.25	421.45	370.33

TABLE III
EXPERIMENT RESULT OF OUTSIDE TESTING SEQUENCES (QP=28)

Sequence		PSNR Y(dB)	Bit Rte(kb/s)	Total Time(s)
Container	RDO	36.27	54.82	619.31
	No RDO	36.03	64.81	447.98
	TSMS	36.05	54.21	89.83
Mother And Daughter	RDO	37.62	53.59	590.04
	No RDO	37.37	57.65	431.71
	TSMS	37.31	52.36	113.22
Stefan	RDO	34.06	605.60	758.81
	No RDO	33.90	623.85	493.34
	TSMS	33.95	615.55	373.73

*Diamond Search [10] is applied in all compression schemes.

5. CONCLUSION AND FUTURE WORK

The proposed TSMS algorithm successfully reduces the compression time of H.264/AVC with RDO greatly. The slight loss in video quality is almost not perceivable to human eyes and the bit rate increase is not much. Compared to the original encoder without RDO, the proposed algorithm is still better even for fast moving sequences.

The proposed algorithm, however, may have some further modifications to achieve better performance. For example, TSMS has to examine all the modes in a certain type before the decision is made. It is possible to reduce the number of modes to examine to decide the final mode. Second, BPN used in this paper is an algorithm to achieve average performance based on a quadratic cost function. Other networks such as Radial Bases Function (RBF) network and fuzzy neuron network may be applied to the algorithm too. There are still methods other than mode decision to be combined to the proposed algorithm to reduce the computation time of the H.264/RDO encoder.

6. 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, May.

[2] B. Meng, O. C. Au, C. W. Wong, and H. K. Lam, "Efficient Intra-Prediction Algorithm in H.264," in *Proc. IEEE ICIP 2003*, Barcelona, Spain, September 2003.

[3] B. Meng, O. C. Au, C. W. Wong, and H. K. Lam, "Efficient Intra-Prediction Mode Selection for 4x4 Blocks in H.264," in *Proc. IEEE ICME 2003*, Baltimore, Maryland July 2003.

[4] B. Meng, and O. C. Au, "Fast Intra-Prediction Mode Selection for 4x4 Blocks in H.264," in *Proc. ICASSP 2003*, Hong-Kong, April 2003.

[5] C. S. Kim, H. H. Shih and C. C. Kuo, "Feature-Based Intra-Prediction Mode Decision for H.264," *IEEE Proceedings of International Conference Image Processing 2004,* Singapore, October 2004.

[6] C. S. Kim, Q. Li, and C. C. J. Kuo, "Fast Intra-Prediction Model Selection for H.264 Codec," *SPIE International Symposium ITCOM 2003*, Orlando, Florida, July, 2003.

[7] C. Y. Chang, C. H. Pan, and H. Chen, "Fast Mode Decision for P-Frames in H.264," *Picture Coding Symposium,* San Francisco, December 2004.

[8] D. Wu, S. Wu, K. P. Lim, F. Pan, Z. G. Li, and, X. Lin, "Block Inter Mode Decision for Fast Encoding of H.264," in *Proc. IEEE ICASSP 2004*, Montreal, Quebec, Canada.

[9] F. Pan, X. Lin, R. Susanto, K. P. Lim, Z. G. Li, G. N. Feng, D. J. Wu, and S. Wu, "Fast Mode Decision for Intra prediction," *JVT document: JVT-G013*.

[10] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation," *IEEE Transactions on Circuits and System for Video Coding Technology*, vol. 8, No. 4, August 1998.

[11] K. P. Lim, S. Wu, D. J. Wu, S. Rahardja, X. Lin, F. Pan, Z. G. Li, "Fast Inter Mode Selection," *JVT document: JVT-I020*.

[12] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive Deblocking Filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, No. 7, July 2003.

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

[14] 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 Transactions on Circuits and System for Video Technology*, vol. 6, April. 1996.

[15] Wiegand, T.; Sullivan, G.J.; Bjntegaard, G.; Luthra, A., "Overview of the H.264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on,* Vol 13, Issue 7, July 2003

[16] Marpe, D.; Schwarz, H.; Wiegand, T., "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *Circuits and Systems for Video Technology, IEEE Transactions on* Vol 13, Issue 7, July 2003