ON THE PERFORMANCE IMPROVEMENT OF H.264 THROUGH FOREGROUND AND BACKGROUND ANALYSES

Zhe-Kuan Lin, Horng-Horng Lin, Yu-Hsin Chen, and Jen-Hui Chuang

Dept. of Computer and Information Science,
National Chiao-Tung University, Hsinchu 300, Taiwan

ABSTRACT

A more efficient coding scheme for H.264 by heuristically assign macroblock partition types for video foreground and background coding is proposed. High visual quality of foreground regions are retained while low bit-rate background coding is achieved. More importantly, the encoding time is reduced significantly owing to the elimination of the exhausted searches over all partition types during the RD optimization.

1. INTRODUCTION

One of the main features of emerging H.264 video compression standard is its adoption of a rate-distortion (RD) optimization concept to find a balance between video coding bit-rates and the quality. For this purpose, H.264 provides more prediction options, e.g., several search modes in intra-predictions and various block partitions in inter-predictions, for coding macroblocks. However, the implementation of the RD optimization needs to evaluate all possible coding types for each macroblock intra- and inter- prediction, and find the best type satisfying the given RD criterion. As a result, the encoding process of H.264 is extremely time consuming.

In this paper, a study of possible improvements of H.264 coding performance, by referencing video foreground and background information, is reported. Some heuristics selections of macroblock size partitions for inter-predictions are proposed to effectively maintain the visual quality of compressed foreground regions and to reduce coding sizes of background areas in a video frame. Moreover, the encoding time can then be greatly reduced since exhausted searches of all macroblock partitions are no longer needed.

1.1. Related work

Video content analysis has been an important topic in the field of multimedia research. Many video layer decomposition methods, e.g., [2], [3], [9], [10], [14], are proposed to segment video frames into object regions and to cluster similar regions of consecutive frames as grouped layers. The most generic video layer decomposition is to divide a video sequence to two-layered foreground and background representations. By referencing this kind of layer information and applying different coding schemes to different layers, a video sequence can be compressed in a more efficient way.

Many coding schemes are also proposed for content-based video compressions. In [7], a procedure for selecting MPEG-4 coding models is proposed to dynamically compress different segmentation regions of videos. In [1], Eisert et al. use high level 3-D object models to synthesize video frames and propose an improved hybrid scheme to combine traditional waveform coding and 3-D model-based coding. Lu et al. [5] decompose a video sequence into *sprites* and apply a directional spatial prediction to reduce the coding bit-rates of background sprites. Though these methods are not based on H.264, all their results indicate higher coding efficiencies can be obtained by referencing high level content information.

Referring to H.264 researches, adaptive block-size transformations are presented in [13] to extend the uses of variable block sizes in the motion prediction to the transformation coding. With these flexibilities, higher coding performances, namely rate savings and quality increases, can be achieved. In [6], based on simplifying the cost function computations, an efficient intra-prediction mode selection method is used to avoid full block mode searches as well as to maintain similar PSNR and bit-rates. Also for reducing the computational costs of H.264 intra-prediction mode selections, Kim et al. [4] apply a multi-stage sequential decision process to filter out probable modes and adopt a simplified RD optimization method to determine the final choice. In [8], [11], bottom-up merging designs are suggested to lessen the burden of searching all possible macroblock partitions for H.264 inter-predictions.

1.2. Our approach

Through investigating the effects of using various macroblock sizes on the compression quality and bit-rates, some heuris-

---

This work was supported by MediaTech Research Center at Chiao-Tung University.

0-7803-9332-5/05/$20.00 ©2005 IEEE

tics for selections of inter-prediction macroblock partitions in H.264 are proposed in this paper to effectively and efficiently encode video foreground and background regions. By employing the proposed heuristic assignments of block partitions, the PSNR of compressed foreground regions are increased, comparing to the standard RD optimization method. Also, the compression sizes of background areas in P- and B- frames are lowered down due to fewer bits are required to encode the video background. More importantly, the coding time can thus be reduced significantly. The proposed method is shown to be simple and effective, and the derived video files completely agree the H.264 decoding specifications.

The organization of the paper is as follows. In Section 2, investigations into the effects of using various macroblock sizes on the coding performances are presented. Based on the investigation results, heuristic partition assignments for the foreground and background macroblocks are suggested. Experimental results based on the proposed heuristics are presented in Section 3. Finally, conclusions and discussions of future works are given in Section 4.

2. MACROBLOCK PARTITION ANALYSIS

H.264 is different from the previous MPEG standards in many ways. One of them is the tree-structured block sizes for macroblock partitions used in inter-prediction motion compensations. The various block sizes provide high flexibility for optimizing rates and distortions in the coding of P- and B- frames. During inter-prediction, a macroblock may be split into 16×16, 8×16, 16×8, and 8×8 blocks. Moreover, a 8×8 block can be further divided into 4×8, 8×4, and 4×4 sub-blocks. Besides, there are additional 16×16 *skip* type for P-frame and *direct* type for B-frame to efficiently encode pictures with regions of very small motion. These partitions are particularly efficient in coding diverse shape boundaries.

2.1. Partition Type Decision

The choices of macroblock partitions would affect the required bits for coding motion compensated residues. Conventional, for selecting a suitable partition, an RD optimization criterion defined by

$$D(B, \hat{B}|Q) + \lambda \cdot R(B, \hat{B}|Q), \qquad (1)$$

where Q represents the given quantization parameters, B stands for a macroblock, $\hat{B}$ denotes its reconstruction after decoding, and D and R are functions measuring the block distortion and the needed coding bits, is proposed in [12]. The scalar λ of (1) is set to control a balance between compression distortions and rates. The best macroblock partition can be computed by evaluating partitions of all possible sizes for a macroblock and finding the choice with the minimal cost with respect to (1). It results in costly search in deciding a macroblock partition.

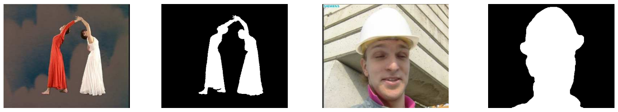

(a) Img: Dancer (b) Msk: Dancer (c) Img: Foreman (d) Msk: Foreman

Fig. 1. Example of image frames and their foreground/background masks used in our experiments.

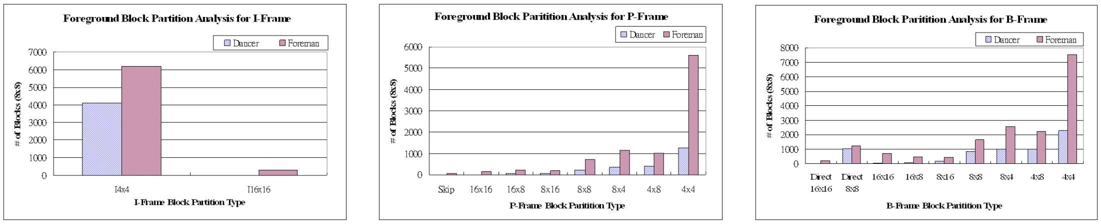
The main idea of the proposed approach is that the decision process of macroblock partitions can be significantly simplified if the foreground and background segmentations of a video sequence are obtained. In this case, we can deliberately conduct a high quality foreground layer encoding while adopt a rate-saving background compression. Owing to this rather specific objective, it becomes possible to heuristically assign appropriate macroblock partitions to encode the foreground and background macroblocks in the inter-prediction process, without resorting to the expensive RD optimization.

2.2. Partition Type Distribution

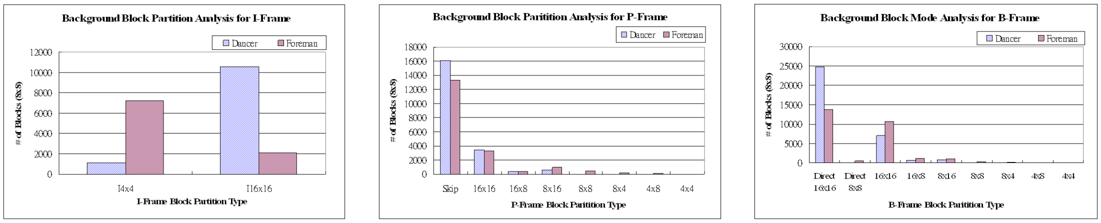
Our investigation of the effects of using various macroblock partition types is based on the modifications of the RD optimization. To achieve a quality-oriented coding, a small λ value is set in (1) for foreground macroblocks. On the contrary, a large λ value is used to find suitable partition sizes for background blocks. Statistics of the adopted partition types are then plotted to show the kind of macroblock size partition that is mostly used in either the high quality foreground coding or the low bit-rate background compression.

Two benchmark video sequences, Dancer and Foreman, are used in our study of partition types. As shown in Fig. 1, the foreground and background masks for image frames are manually marked. A JM7.3 encoder, which is a public C model of H.264, is adopted and its RD optimization part is modified to allow coding foreground and background macroblocks with different λ values. Listed below are the detailed settings for our study.

- Each macroblock having at least a foreground pixel is marked as foreground.
- 59 image frames of each video sequence are encoded in the order of IBPBPBI. . ..
- The default quantization parameter is set to 28.
- The default λ parameters are $\lambda_0 = 27.41$ for I- and P- frames and $\lambda_0 = 73.11$ for B-frames.
- $\lambda_F = 0.01\lambda_0$ is set for the encoding of foreground macroblocks while $\lambda_B = 6\lambda_0$ is for the background.



(a) Foreground: I-Frame (b) Foreground: P-Frame (c) Foreground: B-Frame



(d) Background: I-Frame (e) Background: P-Frame (f) Background: B-Frame

Fig. 2. The statistics of partition types used to encode foreground and background macroblocks with $\lambda_F = 0.01\lambda_0$ and $\lambda_B = 6\lambda_0$, respectively.

We call this method the optimal-λ RD optimization for its use of different λ settings to optimally encode video foreground and background regions. With the above settings, six histograms of macroblock partition types are plotted in Fig. 2. As depicted in Figs. 2 (a) and (d), the block size of I4×4 are used most to generate high quality intra-coded macroblocks for I-frames. On the other hand, the amounts of I16×16 sized macroblocks are increased in coding the video background. The observations suggest us to use I4×4 for foreground and I16×16 for background to speed up the computation in coding I-frames.

Similarly, according to Figs. 2 (b) and (c), the partition size of 4×4 is of the largest portion among all the partition types. It is thus considered to be the most suitable type for encoding foreground blocks to achieve better quality. Moreover, from Figs. 2 (e) and (f), we can see there are peaks at the skip type of P-frame and the Direct 16×16 type of B-frame. However, these two types are not suitable for coding videos with large motions, due to their lack of precise macroblock motion estimation. Hence we heuristically choose the size 16×16, in addition to the skip type of P-frame and the Direct 16×16 type of B-frame, to yield low bit-rate background compressions for the inter-prediction motion compensation process. Together with using different λ_F and λ_B, the proposed foreground and background coding scheme can achieve very high level of efficiency since there are at most two candidates that need to be taken into consideration during the search of proper block partition types.

3. EXPERIMENTAL RESULT

In our experiments, three methods, the standard RD optimization, the optimal-λ RD optimization and the proposed scheme, are compared for foreground PSNRs and background data sizes. As depicted in Fig. 3, the results show that heuristic partition type assignments for macroblock inter-predictions are feasible as the foreground coding quality and the background rate-saving can both be achieved for coding most P- and B- frames. As for I-frames, though the size of I4×4 gives better coding quality for the foreground, the adoption of partition size I16×16 is somehow not decisive for background coding size reduction. Even the optimal-λ method can only lowered the coding sizes by a very small amount. It implies that, for intra-predictions, block size settings are not a major factor to compression sizes.

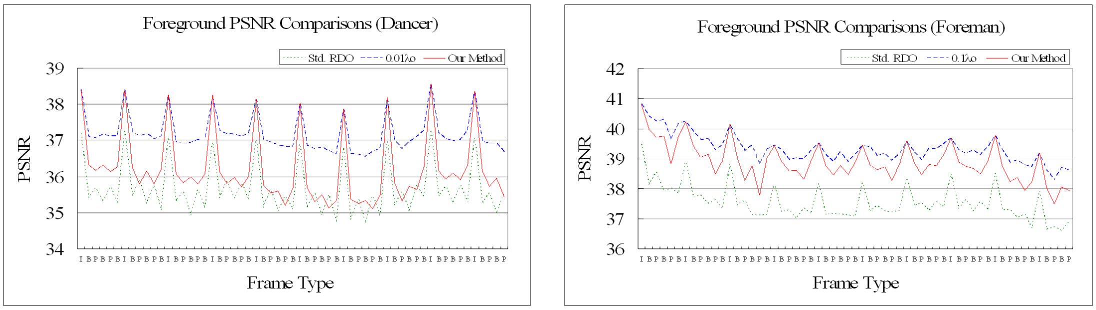
Table 1. The resulting comparisons of Dancer.

Dancer	Time (Sec)	Ave. PSNR	Size (Bit)
Std. RDO	483	35.60	192451
Opt. RDO	492	37.16	149843
Our Method	173	36.12	175836

Table 2. The resulting comparisons of Foreman.

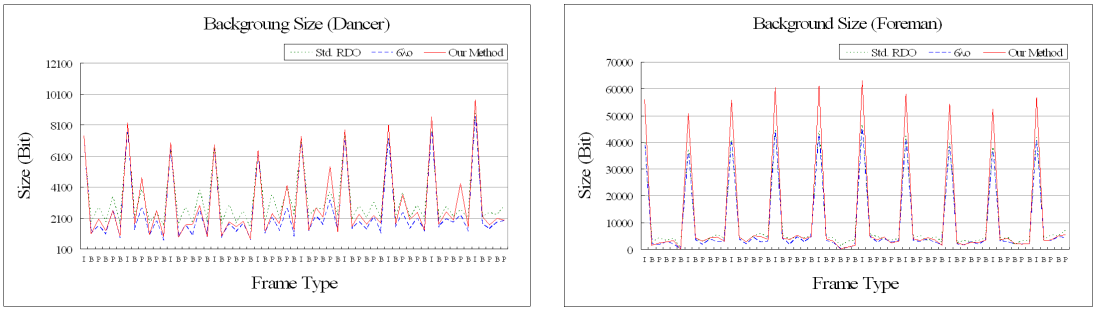
Foreman	Time (Sec)	Ave. PSNR	Size (Bit)
Std. RDO	496	37.52	608598
Opt. RDO	579	39.34	538662
Our Method	187	38.87	732055

The entire coding time, average foreground PSNRs and total background bit sizes are summarized in Table 1 and Table 2. Except for the larger background coding size in the foreman sequence (due to its higher coding bit-rates of I-frames), the other results match our expectations. They also support our idea that heuristically choosing candidate macroblock partition types for the foreground and background coding can be effective and efficient.



(a) Foreground PSNR: Dancer

(b) Foreground PSNR: Foreman



(c) Background Size: Dancer

(e) Background Size: Foreman

Fig. 3. Experimental comparisons of foreground PSNRs and background sizes are illustrated with respect to the method of the standard RDO, of the optimal-λ RDO, and of the proposed method.

4. CONCLUSION

Investigations of foreground and background coding with different λ parameters in RD optimization have been reported. Heuristic assignments of macroblock partition types are proposed and simulated. Our results show, once a more specific compression objective is given, exhausted searches for best macroblock partitions may not be required and the coding performance can still be retained. The time complexity of H.264 encoding can therefore be largely reduced. Because our study is mainly focused on determining inter-prediction block sizes, combining other intra-prediction mode decision methods to enhance content-based codings is the next step of our future work.

5. REFERENCES

[1] P. Eisert, T. Wiegand, and B. Girod, "Model-Aided Coding: A New Approach to Incorporate Facial Animation into Motion-Compensated Video Coding," *IEEE Trans. CSVT*, vol. 10, no. 3, pp. 344–358, Apr. 2000.

[2] B.J. Frey, N. Jojic, and A. Kannan, "Learning Appearance and Transparency Manifolds of Occluded Objects in Layers," *Proc. CVPR*, vol. 1, pp. 45–52, Madison, WI, 2003.

[3] Q. Ke and T. Kanade, "A Subspace Approach to Layer Extraction," *Proc. CVPR*, vol. 1, pp. 255–262, Kauai, Hawaii, 2001.

[4] C. Kim, H.-H. Shih, and C.-C. Kuo, "Feature-Based Intra-Prediction Mode Decision for H.264," *Proc. IEEE ICIP*, Singapore, 2004.

[5] Y. Lu, W. Gao, and F. Wu, "Efficient Background Video Coding with Static Sprite Generation and Arbitrary-Shape Spatial Prediction Techniques," *IEEE Trans. CSVT*, vol. 13, no. 5, pp. 394–405, May 2003.

[6] B. Meng, O.C. Au, C.-W. Wong, and H.-K. Lam, "Efficient Intra-Prediction Model Selection for 4×4 Blocks in H.264," *Proc. IEEE ICME*, vol. 3, pp. 521–524, Baltimore, Maryland, 2003.

[7] E. Reusens, R. Castagno, C. LeBuhan, L. Piron, T. Ebrahimi, and M. Kunt, "Dynamic Video Coding: An Overview," *Proc. IEEE ICIP*, vol. 2, pp. 377–380, Lausanne, Switzerland, 1996.

[8] I. Rhee, G.R. Martin, S. Muthukrishnan, and R.A. Packwood, "Quadtree-Structured Variable-Size Block-Matching Motion Estimation with Minimal Error," *IEEE Trans. CSVT*, vol. 10, no. 1, pp. 42–50, Feb. 2000.

[9] J. Shi and J. Malik, "Motion Segmentation and Tracking Using Normalized Cuts," *Proc. Sixth IEEE ICCV*, pp. 1154–1160, Bombay, India, 1998.

[10] Y. Tsaig and A. Averbuch, "A Region-Based MRF Model for Unsupervised Segmentation of Moving Objects in Image Sequences," *Proc. CVPR*, vol. 1, pp. I889–I896, Kauai, Hawaii, 2001.

[11] Y.-K. Tu, J.-F. Yang, Y.-N. Shen, and M.-T. Sun, "Fast Variable-Size Block Motion Estimation Using Merging Procedure with An Adaptive Threshold," *Proc. IEEE ICME*, vol. 2, pp. 789–792, Baltimore, Maryland, 2003.

[12] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. CSVT*, vol. 13, no. 7, pp. 560–576, July 2003.

[13] M. Wien, "Variable Block-Size Transformation for H.264/AVC," *IEEE Trans. CSVT*, vol. 13, no. 7, pp. 604–613, July 2003.

[14] J. Wills, S. Agarwal, and S. Belongie, "What Went Where," *Proc. CVPR*, vol. 1, pp. 37–44, Madison, WI, 2003.