EFFICIENCY IMPROVEMENTS FOR A GEOMETRIC-PARTITION-BASED VIDEO CODER

Renan U. Ferreira[1], *Edson M. Hung*[1], *Ricardo L. de Queiroz*[1], *Debargha Mukherjee*[2]

[1]Universidade de Brasilia, Departamento de Engenharia Eletrica, Brasilia, Brazil
[2]Hewlett-Packard Laboratories, Palo Alto, USA
E-mails: (renan,mintsu,queiroz)@image.unb.br, debargha.mukherjee@hp.com

ABSTRACT

H.264/AVC has brought an important increase in coding efficiency in comparison to previous video coding standards. One of its features is the use of macroblock partitioning in a tree-based structure. The use of macroblock partitions based in arbitrary line segments, like wedge partitions, has been reported to increase coding gains. The main problem of these non-standard new partitions schemes is the increase in computational complexity. Thus, our work proposes improvements in this extension to the H.264/AVC standard. First, we present a motion vector prediction scheme based on directional partitions. Second, we present a method for complexity reduction based on the most frequent partitions. The results show that it is possible to still produce good coding gains with lower complexity than previous approaches.

Index Terms— Video coding, wedges, motion compensation, complexity reduction, motion vector prediction, H.264/AVC

1. INTRODUCTION

The tree-based structure for macroblock partitioning used in motion compensation in the H.264/AVC video coding standard [1] is an important advance aiming in improving coding efficiency by reducing residue energy [2]. However, this structure is limited to square and rectangular-shaped partitions. Previous works have shown that it is possible to improve the performance of this tree structure by partitioning blocks by different means than using vertical and horizontal directions [3]-[7].

Kondo et al. [3] proposed a method in which the motion compensation is carried for each of two sliced blocks from a macroblock. This slice division is done by a particular straight line segment with orientation given by two pixels on the macroblock's border. A similar partition method is proposed by Hung et al. [4] and Divorra et al. [5], with the difference that the line segment is expressed by a radius r from the macroblock's center and an orientation angle θ. Chen et al. [6] proposed an object-boundary-based motion partition which divides the macroblock into two regions defined by object boundaries, instead of line segments. In order to limit the exhaustive search incurred by the slicing method, Chen et al. [7] introduce a L-shaped block partitions, similar to tree pruning of the leafs in the quadtree segmentation. While the L-shaped segmentation is limited in the amount of partition possibilities, it has much lower complexity, compared to the other works.

The focus in this work is to improve geometric (or wedge or slice) partition-based motion compensation. Compared to the previous works, we propose modifications in the motion vector prediction, in the coding of parameters linked to this extra mode, and a method for reducing the computational complexity by limiting the number of partitions.

2. IMPROVEMENTS

2.1. Motion Vector Prediction Extension

In H.264/AVC, the macroblock may be divided by a horizontal or a vertical line so as to create the 16x8-pixel and 8x16-pixel partitions, respectively. Now, let us consider that the segmenting line can be rotated and translated in order to create the wedge-shaped partitions. Our idea is to compute the predicted Motion Vectors (MVp's) from the blocks that are adjacent to the wedge partition throughout a certain range of rotation angles. Because of that, our method treats the neighboring 4x4 blocks A, B and C of the current partition [1] differently from the prediction in H.264/AVC.

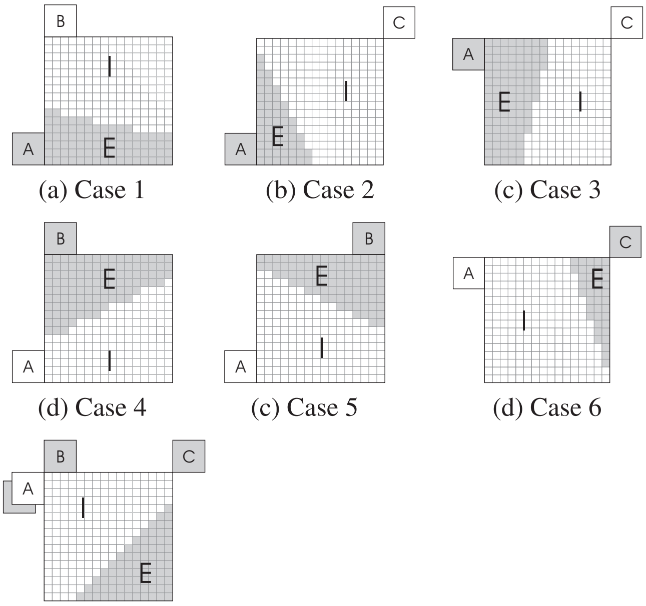
In our approach, the positions of these 4x4 blocks may vary accordingly to the angle parameter of the partition. When gathering the MVp from block B, this block can be either the leftmost or the rightmost 4x4 block at the bottom of the macroblock above the current one. Similarly, when the MVp comes from block A, this block can be either the topmost or the bottommost 4x4 block of the macroblock to the left of the current one. For block C, the prediction is the same as in H.264/AVC.

The wedge partitions may be defined as External (E) and Internal (I) partitions, according to the circle used to generate the segmenting line [4],[5]. Table 1 shows from which 4x4 block the MVp will be computed for a range of angle values in steps of 45°. Figure 1 shows examples for each case, where A, B and C are 4x4-pixel blocks.

We should consider the possibility that a neighbor macroblock was also coded in the wedge mode. In this situation,

Table 1: MV prediction rules.

Cases	Angle Range	Prediction for Partition I	Prediction for Partition E
Case 1	$0° \le \theta < 45°$	B (left)	A (bottom)
Case 2	$45° \le \theta < 90°$	C	A (bottom)
Case 3	$90° \le \theta < 135°$	C	A (top)
Case 4	$135° \le \theta < 180°$	A (bottom)	B (left)
Case 5	$180° \le \theta < 225°$	A (bottom)	B (right)
Case 6	$225° \le \theta < 270°$	A (top)	C
Case 7	$270° \le \theta < 360°$	A	Median



(a) Case 1 (b) Case 2 (c) Case 3

(d) Case 4 (c) Case 5 (d) Case 6

(d) Case 7

Fig. 1: Examples of MV prediction for each case.

the MVp chosen comes from the partition with most superposition with the 4x4 block. Furthermore, in the case the blocks A, B or C are unavailable, the replacement rules are the same as in H.264/AVC.

2.2. Complexity reduction

The main problem of using geometric partition in motion compensation is the large increase in computational complexity on the encoder side, since the encoder has to perform many tests to find the best-match partitions. In order to reduce the complexity, one needs to reduce the number of possible partitions. However, it is also important that this reduction does not incur in significant performance loss. One option is to use only a smaller subset of partitions that are most likely to be chosen. In order to find the most likely partition subset, we decided to use a training set of sequences and gathered their joint statistics. Figure 2 shows the joint histogram of radii and angles for the sequences "foreman", "mobile", "news" and "silent", all CIF size. These sequences were chosen for their different motion characteristics.

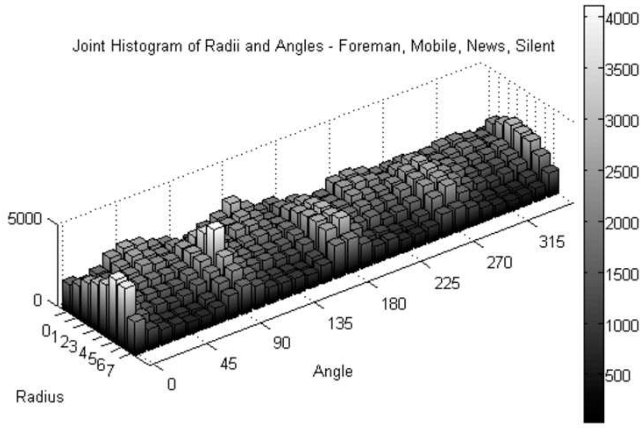


Fig. 2: Joint histogram of radii and angles, for a set of training sequences.

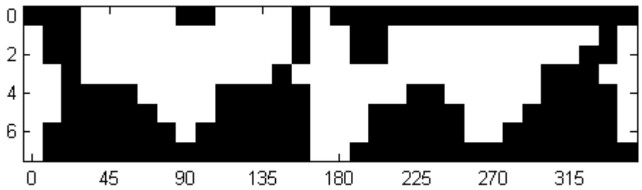


Fig. 3: 128 most likely partitions of the histogram in Fig. 2.

For this test histogram, we computed the 128 most likely partitions, which is shown in Fig. 3. With that information, we can reduce the number of tests and expect that this would lead to results close to those achieved by using all 256 partitions. A similar algorithm can be applied to fewer partitions.

2.3. Implementation

In order to use wedge partitions, it is necessary to add a new partition mode in H.264/AVC. We call it MODE_WEDGE and it was inserted right after MODE_16x16 and before MODE_16x8 (shown in Table 2), differently from that presented in [5]. Modes are entropy coded and smaller mode values are assumed to be more frequent. Because of that, we want to optimize encoding through proper placement of MODE_WEDGE in the list of modes. Here, we only used wedge partitions for 16x16-pixel blocks. Similar to other modes, the MODE_WEDGE is chosen depending on its RD cost.

The partition parameters, radius and angle, are sent to the decoder as side information. Their distributions are nearly uniform, leading to no effective compression through block coding. Because of that, for each macroblock coded in MODE_WEDGE, an overhead is generated. For convenience, we chose the number of partitions to be a power of 2, so that a fixed length code could be used to represent it. Thus, the

Table 2: MB modes with MODE_WEDGE.

Mode Value	MB Mode
1	MODE_16x16
2	MODE_WEDGE
3	MODE_16x8
4	MODE_8x16
5	MODE_8x8

size of the overhead, in bits (which is included in the RD cost function), determines the number of possible partitions.

Following the previous experiments [4],[5] concerning the partition parameters, we limited the maximum overhead size to 8 bits by using 3 bits for 8 possible radius values and 5 bits for 32 possible angle values. Thus, we have the radius $0 \leq r < 8$ varying in steps of $\Delta r = 1$ pixel, and angle $0° \leq \theta < 360°$ varying in steps of $\Delta\theta = 11.25°$. When fewer partitions are chosen, a fixed length code is used to code the most likely angle and radius pairs jointly.

3. EXPERIMENTAL RESULTS

Our implementation was done through modifications to the JSVM 9.2 reference H.264/AVC software [8]. In order to objectively verify our results, we compare PSNR curves, using the VCEG'S M33 recommendation [9]. Our changes in the code include the insertion of the MODE_WEDGE and the new MV prediction rules, along with motion estimation and compensation for the geometric partitions.

We implemented modifications for P frames only and used the IPPP... coding mode with 4 reference frames. The quantization parameter (QP) values used were recommended by VCEG [10], i.e. QPs 22, 27, 32 and 37 for I frames and QPs 23, 28, 33, 38 for P frames. Moreover, we used 32-pixels search range, full search for all modes except for the MODE_WEDGE (which uses fast search). Features such as quarter-pel precision, deblocking filter were turned on and we used the High Profile with UVLC entropy coder.

We first tested the position of insertion of MODE_WEDGE, which means the assignment of values of MODE_WEDGE = 2 and MODE_WEDGE = 4. We ran this test for 60 frames of CIF sequences "foreman", "mobile", "mother & daughter", "news" and "silent", using all 256 possible pairs of radius and angles. The setting MODE_WEDGE = 2 yields an average reduction in rate of 0.78% over the other case. Table 3 shows the gains for each sequence.

Table 3: Percentage comparative gains in rate of assignment values of MODE_WEDGE.

Sequence	Foreman	Mobile	Mother & Daughter	News	Silent
Gain (%)	0.844	0.973	0.728	0.566	0.766

Table 4: Average gains in Rate (%) and Distortion (PSNR increase in dB) over the H.264/AVC coder - Training set.

Sequence	Foreman	Mobile	News	Silent
Rate	7.67	7.65	5.88	6.39
PSNR	0.388	0.394	0.319	0.299

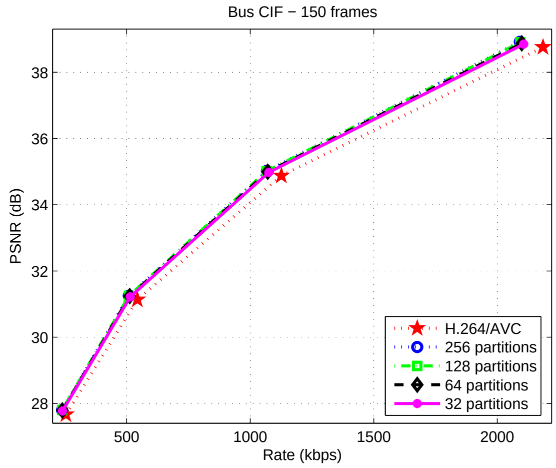
The distribution of the partition parameters is not very biased and for the 256 partitions, the parameter entropy was in average 7.87 bits. Hence, we opted for sending radius-angle information simply encoded with 8 bits.

Finally, we present the results relating to the complexity reduction. We ran these tests independently from the previous one and the sequences were separated in training and validation sets. As mentioned before (Section 2.2), the training set contains CIF sequences "foreman", "mobile", "news" and "silent". The validation set contains QCIF sequences "claire", "container" and "salesman", and CIF sequences "bus", "city", "football", "harbour", "mother & daughter". With the exception of sequence "bus", which has 150 frames, and "football", which has 260 frames, all other have 300 frames. Furthermore, we also tested 60 frames of 4CIF-sized sequence "crew". All sequences have 30 Hz frame-rate.

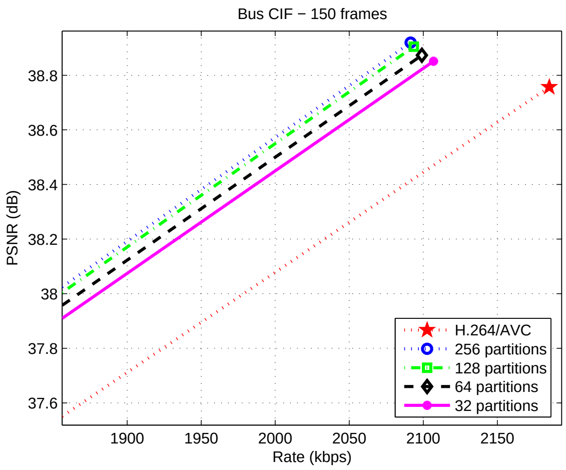
We tested 256, 128, 64 and 32 partitions yielding overheads of 8, 7, 6 and 5 bits, respectively. In order to encode the N partitions in $log_2(N)$ bits, we used a lookup table for each N. The gains for all the test sequences over the H.264/AVC coder, both in rate and distortion, for each number of partitions are shown in Tables 4 and 5. Table 4 shows the results for the training set with 256 partitions. Table 5 shows the re-

Table 5: Average gains in Rate (%) and Distortion (PSNR increase in dB) over the H.264/AVC coder - Validation set.

Sequence		Number of Partitions			
		256	128	64	32
Claire QCIF	Rate	7.56	7.27	6.52	6.44
	PSNR	0.414	0.397	0.354	0.349
Container QCIF	Rate	3.75	3.22	3.63	3.23
	PSNR	0.172	0.146	0.164	0.148
Salesman QCIF	Rate	7.06	6.46	6.08	5.18
	PSNR	0.380	0.347	0.325	0.277
Bus CIF	Rate	8.12	8.14	7.30	6.56
	PSNR	0.435	0.435	0.389	0.347
City CIF	Rate	7.46	7.33	6.79	6.10
	PSNR	0.348	0.341	0.315	0.281
Football CIF	Rate	4.37	4.10	3.64	3.10
	PSNR	0.244	0.228	0.203	0.172
Harbour CIF	Rate	10.81	10.78	10.39	9.79
	PSNR	0.514	0.511	0.492	0.460
Mother & Daughter CIF	Rate	4.69	4.75	4.74	3.88
	PSNR	0.207	0.210	0.209	0.170
Crew 4CIF	Rate	8.50	8.05	7.01	5.80
	PSNR	0.285	0.269	0.233	0.191



(a) Full curves.



(b) Zoom of (a).

Fig. 4: R-D curves of using wedge partitions in H.264/AVC with different number of partitions for sequence "bus".

sults for the sequences in the validation set. Figure 4 shows the R-D curves for sequence "bus" as an example.

4. CONCLUSIONS

This paper proposes some modifications in the use of geometric-based motion compensation for video coding. We have shown that the use of proper rules for the motion vector prediction, together with the better encoding of the partition mode within the H.264/AVC coder can further improve the coding efficiency of this technique. Unfortunately it is not possible to have a straightforward comparison of our results to those in [4], because of its distinct non-H.264/AVC implementation, and those in [5] because of the tests characteristics. However, a good comparison can be made with the results presented by Divorra et al. [11], which is shown in Table 6.

Moreover, we have analyzed the complexity reduction of this process by testing a restricted set of partitions. We used training sequences because it is still uncertain how these partitions may be chosen *a priori* for each sequence. Results have shown that the distribution of partitions is, in some way, similar across most sequences.

Table 6: Results comparison in Rate(%) gain over the H.264/AVC coder.

Sequence	Foreman	Mobile	Bus	Football
Results in [11]	6.12	3.84	4.29	2.27
Our approach	7.67	7.65	8.12	4.37

5. REFERENCES

[1] ITU-T, "ITU-T Recommendation H.264 : Advanced video coding for generic audiovisual services," ISO/IEC 14496-10, May 2003.

[2] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[3] S. Kondo and H. Sasai, "A motion compensation technique using slice blocks in hybrid video coding," *In Proc. of IEEE International Conf. on Image Processing*, Genova, Italy, pp.305-308, Sept. 2005.

[4] E. M. Hung, R. L. de Queiroz, and D. Mukherjee, "On macroblock partition for motion compensation," *In Proc of IEEE International Conf. on Image Processing*, Atlanta, USA, pp.1697-1700, Oct. 2006.

[5] Ò. Divorra, P. Yin, C. Dai, and X. Li, "Geometry-adaptive block partitioning for video coding," *In Proc. of IEEE International Conf. on Acoustics, Speech and Signal Processing*, Honolulu, USA, pp. 657-660, Apr. 2007.

[6] J. Chen, S.R. Lee, K-H. Lee, and W-J. Han, "Object boundary based motion partition for video coding," *In Proc of IEEE International Symposium on Circuits and Systems*, Lisboa, Portugal, Nov. 2007.

[7] S. Chen, Q. Sun, X. Wu, and L. Yu, "L-shaped segmentations in motion-compensated prediction of h.264," *In Proc of IEEE International Symposium on Circuits and Systems*, Seattle, USA, pp. 1620-1623, May 2008.

[8] ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, "JSVM 9.2 software," April 2006.

[9] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," Tech. Rep., ITU-T Q.6/SG16 VCEG, VCEG-M33, April 2001.

[10] T. K Tan, G. Sullivan, and T. Wedi, "Recommended simulation common conditions for coding efficiency experiments revision 1, ITU-T Q.6/SG16 VCEG, VCEG-AE010," Jan. 2007.

[11] Ò. Divorra, P. Yin, C. Gomila, "Geometry-adaptive Block Partioning, ITU-T Q.6/SG16 VCEG, VCEG-AF10," Apr. 2007.