LETTER
Fast Intra Prediction Mode Decision for H.264/AVC

Do QUAN[†a)], *Student Member and* Yo-Sung HO[†], *Member*

SUMMARY In this letter, we present a simple but efficient intra prediction mode decision for H.264/AVC. Based on our investigation, the DC mode appears to be the superior prediction mode among the various candidates. We propose an intra-mode decision algorithm where the DC mode is chosen as a candidate for the best prediction mode. By experimental results, on average, the proposed algorithm significantly saves 81.905% of the entire encoding time compared to the H.264 reference software; besides, it reduces negligible peak signal-to-noise ratio (PSNR) values and slightly increases bitrates.
key words: H.264/AVC, video coding, intra-mode decision

1. Introduction

H.264 is a powerful video coding standard in terms of PSNR value and visual quality. One of the coding approaches uses variable block sizes of macroblock (MB) modes. Although H.264 provides a rate-distortion optimization (RDO) technique to find the best mode, in order to check all combinations of modes for each MB, it still consumes much time and remains complex due to the heavy computational load. These are the obstacles to implementing an encoder for real-time applications, such as video telephony and video conferencing.

For intra prediction, H.264 uses three different block sizes: intra-16×16, intra-8×8, and intra-4×4. Among these modes, intra-4×4 has nine different modes, such as the DC mode, horizontal mode, vertical mode, and diagonal modes with a variety of angles for intra-luma prediction. Intra-8×8 using 8×8 transform has nine modes which are similar to intra-4×4 modes. Lastly, intra-16×16 has four modes including the DC mode, horizontal mode, vertical mode, and diagonal mode. For intra-chroma prediction, intra-8×8 has four modes but different orders compared with intra-16×16.

In order to decide the best mode, H.264 uses the RDO method, which is based on the Lagrangian cost function to minimize the mode cost [1]. Figure 1 summarizes the mode decision algorithm in H.264.

There have been a lot of mode decision algorithms. In these algorithms, the authors tried to reduce the complexity while keeping the acceptable visual quality. The following algorithms are related to our work in this letter. First, Tsai et al. [2] used an intensity gradient technique to make an
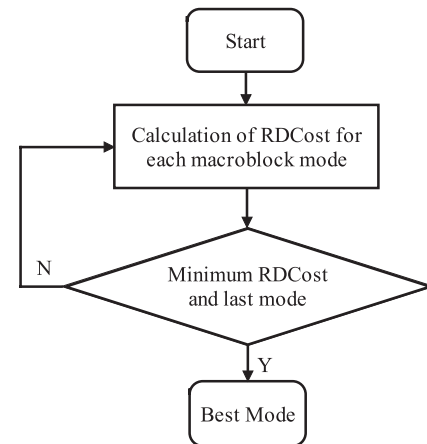
Fig. 1 Mode decision algorithm in H.264.

efficient intra prediction. Pan et al. [3] proposed a fast mode decision algorithm for intra-frame coding using local edge information of each macroblock. La et al. [4] used the dominant edge direction to make fast mode decision. A simple algorithm based on direction detection of the edge inside the block was developed by Lin et al. [5]. Furthermore, Wang et al. [6] proposed a fast algorithm based on the dominant edge strength.

In this letter, we propose a simple but efficient intra prediction mode decision algorithm. Instead of checking all combinations of modes for each MB, we directly check the best mode which is observed from our research. We expect to get significantly reduced coding time, acceptable increase in bitrates and decrease in PSNR values from our experiment. Finally, with the same test sequences and the initial conditions, we compare the results of our proposed algorithms with those of the above-mentioned algorithms.

2. Proposed Algorithm

First of all, we determine the intra-chroma modes. In our proposed algorithm, four intra-8×8 modes for intra-chroma prediction are calculated by the sum of absolute Hardamard transform differences (SATD). The best mode is the minimum of SATD.

For luma-prediction, we use fidelity range extensions (FRExt) as an option of our algorithm and determine the best mode of intra-8×8 among nine modes. By our research on intra-4×4 and intra-16×16, we investigate that the DC mode is the superior prediction mode among the various candi-
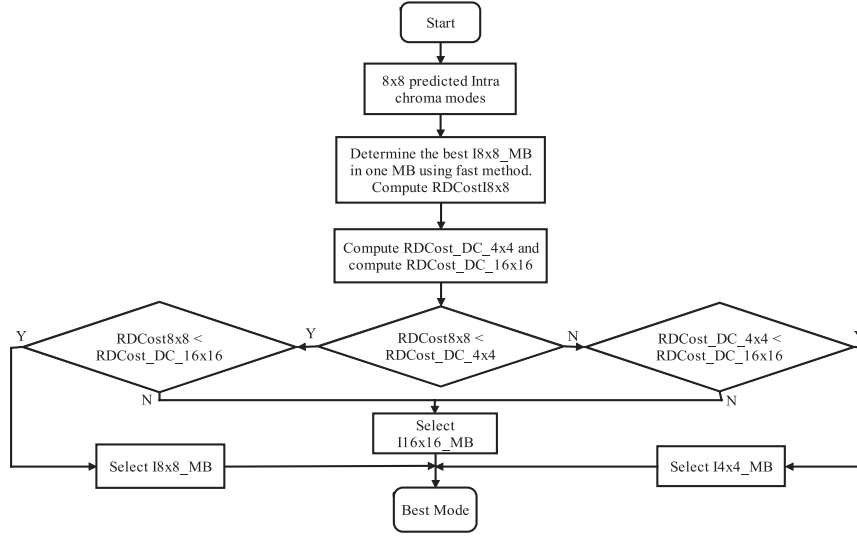
Fig. 3 Flowchart of fast intra-mode decision algorithm.
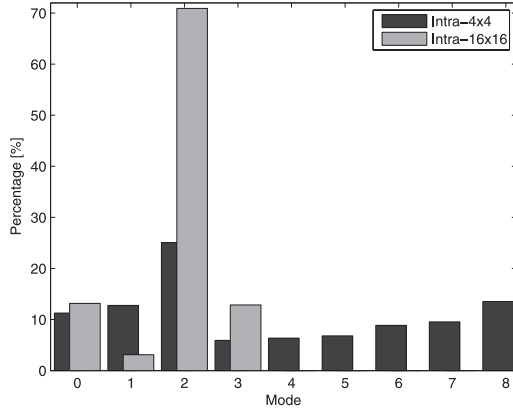
Fig. 2 Mode selection ratio for "Tempete" sequence.

Table 1 Encoding parameters for fast intra-mode decision algorithm.

Parameter	Settings
RDO mode	Fast high complexity mode
GOP structure	I I I …
Hadamard transform	Used
Quantization parameters	28, 32, 36, and 40
Number of frames	100

- *Step 4*: Find the minimum value of RDCost8×8, RDCost_DCmode_4×4 and RDCost_DCmode_16×16 to get the optimal RDO mode.

3. Experimental Results

We implement our proposed algorithm on the reference software JM11.0 [9]. In addition, we examine various CIF and QCIF sequences, adopted as the test sequences, i.e., Coastguard, Silent, Table tennis, Tempete, and Suzie, in MPEG standard. The system platform is Intel Pentium(R) 4, a processor of speed 3.6 GHz, 2.00 GB of DDR RAM, and Microsoft Windows XP for the simulations. Table 1 shows the simulation conditions.

In order to compare our proposed algorithm and JM11.0, we use PSNR difference, percentage bitrate difference, and run-time percentage difference. These can be calculated by using the following formulas:

$$\Delta PSNRY = PSNRY_{proposed} - PSNRY_{JM11.0} \, [dB].$$

$$\Delta Bitrate = \frac{Bitrate_{proposed} - Bitrate_{JM11.0}}{Bitrate_{JM11.0}} \times 100\% [\%].$$

$$\Delta Time = \frac{Time_{proposed} - Time_{JM11.0}}{Time_{JM11.0}} \times 100\% [\%].$$

dates. Figure 2 shows an example of this investigation for "Tempete" test sequence. Other test sequences have similar results. Therefore, the DC mode is chosen as the candidate for the best prediction mode which is strongly agreed for 4×4 intra prediction mode in [7], and both 16×16 luma prediction and 8×8 intra-chroma prediction in [8]. This way chooses the DC mode as the best prediction mode for intra-4×4 and intra-16×16 blocks. We compare the RDCost of intra-4×4, intra-8×8, and intra-16×16 to determine the best mode as the final decision of current macroblock. We repeat all processes as mentioned above for the next macroblock.

Figure 3 shows the flowchart of the proposed intra-mode decision algorithm. The procedure of the algorithm can be summarized as follows:

- *Step 1*: Determine the best mode among four intra-8×8 intra-chroma modes for intra-chroma prediction.
- *Step 2*: Determine the best mode of intra-8×8_MB. Compute RDCost8×8.
- *Step 3*: Compute RDCost_DCmode_4×4 and RDCost_DCmode_16×16.

Table 2 Comparison of coding performance and complexity reduction with original JM11.0 for all QCIF sequences.

Test sequences	$\Delta PSNRY$ [dB]	$\Delta Bitrate$ [%]	$\Delta Time$ [%]
Coastguard	-0.053	+4.999	-85.059
Silent	-0.073	+4.862	-80.916
Table tennis	-0.140	+7.353	-81.019
Tempete	-0.085	+2.805	-81.377
Suzie	-0.125	+3.784	-81.153
AVERAGE	**-0.095**	**+4.761**	**-81.905**
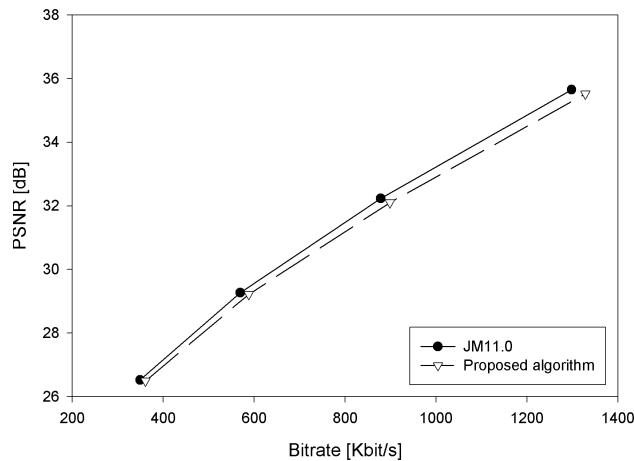
Fig. 4 Without significant rate-distortion degradation of "Tempete" sequence for intra-mode decision.

Fig. 5 Time saving of "Tempete" sequence for intra-mode decision.

Table 3 Comparison of coding performance and complexity reduction with other works.

Sequences	Method	$\Delta PSNRY$ [dB]	$\Delta Bitrate$ [%]	$\Delta Time$ [%]
Silent (QCIF)	Ref. [2]	-0.186	+3.753	-74.72
	Ref. [3]	-0.183	+3.540	-65.170
	Ref. [4]	-0.122	+5.441	-71.21
	Ref. [5]	-0.055	+2.661	-61.77
		-0.039	+1.913	-57.67
	Ref. [6]	-0.013	+2.998	-59.45
	Proposed	**-0.073**	**+4.862**	**-80.916**
Coastguard (QCIF)	Ref. [2]	-0.151	+2.932	-74.91
	Ref. [3]	-0.106	+2.361	-55.026
	Ref. [4]	-0.088	+5.405	-72.29
	Ref. [5]	-0.068	+0.86	-63.9
		-0.047	+0.631	-59.62
	Ref. [6]	-0.036	+2.83	-61.49
	Proposed	**-0.053**	**+4.999**	**-85.059**
Tempete (CIF)	Ref. [2]	-0.162	+2.609	-75.72
	Ref. [3]	-0.229	+3.514	-57.697
	Ref. [4]	-0.185	+4.019	-72.75
	Ref. [5]	-0.107	+1.671	-63.55
		-0.076	+1.206	-60.08
	Ref. [6]	-0.019	+2.257	-61.01
	Proposed	**-0.177**	**+2.991**	**-82.874**

Table 2 shows the simulation results for the fast intra-mode decision method. In the table, the positive numbers indicate increases, and the negative numbers show decreases. All experimental results of the proposed algorithm are relative to those of the reference software JM11.0.

From Table 2, the experimental results show that the proposed fast intra-mode decision achieves 81.905% time savings on average while keeping a negligible loss in PSNR values and slightly increments in bitrates. The results of time savings are significant while the PSNR value and bitrate are little decreasing 0.095 dB and acceptable increasing 4.761% on average, respectively.

In order to illustrate a negligible loss in PSNR values, an increment in bitrates, and a significant time saving, we show the performance of "Tempete" test sequence. Other test sequences have similar improvements.

Figure 4 shows the rate-distortion (RD) curves of "Tempete" sequence. Using this sequence, the PSNR values of our proposed algorithm are 26.5 dB, 29.22 dB, 32.1 dB, and 35.52 dB while those of the original reference software JM11.0 are 26.53 dB, 29.27 dB, 32.23 dB, and 35.65 dB for QP values as 40, 36, 32, and 28, respectively. We can easily see that two PSNR ranges are overlapped. It means that the RD performance of the proposed algorithm is almost simi-lar to that of the reference software JM11.0. Therefore, the proposed algorithm has a negligible loss in PSNR values.

Figure 5 shows the time saving of "Tempete" sequence. For this sequence, the running time of our proposed algorithm lies between 14.2 seconds and 21.3 seconds while that of the original reference software JM11.0 lies between 72.7

seconds and 120 seconds depending on the QP values. Figure 5 indicates that the proposed algorithm outperforms the original H.264 in the reference software JM11.0 in terms of time saving.

Table 3 compares the results of our proposed algorithm and those of the other algorithms. The results in the table are the average results of four QP values, as shown in Table 1.

From Table 3, the time saving of our proposed algorithm lies between 80.916% and 85.059% while those of other methods lie between 55.026% and 75.72%. It means our method reduces time much better than the other methods can.

As shown in Table 3, the results of our proposed algorithm and the other algorithms still keep a negligible loss in PSNR values. In other words, these algorithms have similar rate-distortion degradation. Besides, the PSNR values of our proposed algorithm are similar to those of the algorithms in [3] and [4] and are slightly increment in bitrates, compared to the rest of the other algorithms. Therefore, we can say that our proposed algorithm outperforms the other algorithms, especially in terms of time saving.

4. Conclusion

We proposed a fast intra-mode decision algorithm based on our investigation that the DC mode is the superior prediction mode among all such modes. The proposed algorithm is simple but efficient. Using this algorithm, we achieved 81.905% time savings on average without significant rate-distortion degradation.

References

[1] K.-P. Lim, G. Sullivan, and T. Wiegand, "Text description of joint model reference encoding methods and decoding concealment methods," JVT-N046, 14th JVT Meeting, HongKong, China, Jan. 2005.

[2] A.-C. Tsai, A. Paul, J.-C. Wang, and J.-F. Wang, "Intensity gradient technique for efficient intra-prediction in H.264/AVC," IEEE Trans. Circuits Syst. Video Technol., vol.18, no.5, pp.694–698, May 2008.

[3] 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 Syst. Video Technol., vol.15, no.7, pp.813–822, July 2005.

[4] B. La, M. Eom, and Y. Choe, "Fast mode decision for intra prediction in H.264/AVC encoder," IEEE Intl. Conf. of Image Processing, pp.321–324, Sept. 2007.

[5] W.-G. Lin, A.-C. Tsai, J.-F. Wang, and J.-F. Yang, "A simple direction detection algorithm for fast H.264 intra prediction," Proc. IEEE TENCON, pp.1–4, Oct. 2007.

[6] J.-F. Wang, J.-C. Wang, J.-T. Chen, A.-C. Tsai, and A. Paul, "A novel fast algorithm for intra mode decision in H.264/AVC encoders," Intl. Symposium on Circuits and Systems, pp.3498–3501, Sept. 2006.

[7] Y.-D. Zhang, F. Dai, and S.-X. Lin, "Fast 4x4 intra-prediction mode selection for H.264," IEEE Intl. Conf. on Multimedia and Expo, pp.1151–1154, June 2004.

[8] J.S. Park and H.J. Song, "Selective intra prediction mode decision for H.264/AVC encoders," Trans. Eng., Computing and Technol., vol.13, pp.51–55, May 2006.

[9] JVT H.264/AVC reference software, version JM11.0, available online at http://iphome.hhi.de/suehring/tml/download