Optical Engineering

SPIEDigitalLibrary.org/oe

Fast mode decision for the H.264/AVC video coding standard based on frequency domain motion estimation

Abdelrahman Abdelazim
Stephen J. Mein
Martin R. Varley
Djamel Ait-Boudaoud

SPIE

Fast mode decision for the H.264/AVC video coding standard based on frequency domain motion estimation

Abdelrahman Abdelazim, Stephen J. Mein, Martin R. Varley, and Djamel Ait-Boudaoud
University of Central Lancashire, Preston, Lancashire,
PR1–2HE United Kingdom
E-mail: AAbdelazim@uclan.ac.uk

Abstract. The H.264 video coding standard achieves high performance compression and image quality at the expense of increased encoding complexity. Consequently, several fast mode decision and motion estimation techniques have been developed to reduce the computational cost. These approaches successfully reduce the computational time by reducing the image quality and/or increasing the bitrate. In this paper we propose a novel fast mode decision and motion estimation technique. The algorithm utilizes preprocessing frequency domain motion estimation in order to accurately predict the best mode and the search range. Experimental results show that the proposed algorithm significantly reduces the motion estimation time by up to 97%, while maintaining similar rate distortion performance when compared to the Joint Model software. *© 2011 Society of Photo-Optical Instrumentation Engineers (SPIE).* [DOI: 10.1117/1.3597609]

Subject terms: motion estimation; phase correlation; frequency domain; mode decision; H.264 advanced video coding.

Paper 101102LPRR received Dec. 28, 2010; revised manuscript received May 4, 2011; accepted for publication May 17, 2011; published online Jul. 6, 2011.

1 Introduction

The H.264 advanced video coding (AVC) standard[1] is the newest standard from the ITU-T video coding experts group and the ISO/IEC moving pictures experts group. Its main advantages are the great variety of applications in which it can be used and its versatile design. This standard has shown significant rate distortion (RD) improvements, as compared to previous standards for video compression.

Although the standard has shown significant RD improvements, it has also increased the overall encoding complexity due to the very refined motion estimation (ME) and mode decision processes where variable block size ME is employed. In H.264, there are seven different block sizes that can be used in intermode coding (16×16 = mode 1, 16×8 = mode 2, 8×16 = mode 3, 8×8 = mode 4, 8×4 = mode 5, 4×8 = mode 6, and 4×4 = mode 7). In addition, the SKIP mode (mode 0), direct mode, and two intramodes (INTRA_4 and INTRA_16) are also supported. To achieve the highest coding efficiency, the encoder tries all the possible modes and selects the best one which minimizes the RD cost.

However, this method is not computationally efficient, and consequently limits the use of H.264 encoders in real-time applications. Therefore, algorithms which can reduce computational complexity of H.264 encoding without compromising coding efficiency are very desirable for real-time implementation of H.264 encoders.

Several fast mode decisions[2–6] have been proposed in the literature. This section provides a review of some existing fast intermode decision techniques and their limitations.

In Ref. 2 the mean of absolute difference between the current and the co-located block in the reference frames have been used to predict the modes. This scheme achieved up to 48% computational cost reduction. A similar algorithm has been proposed in Ref. 3, where the sum of the absolute difference value of the current MB is calculated and compared to a threshold. Based on the comparison results, the modes are selected adaptively.

In Ref. 4, the motion vector information has been used to predict the modes and the scheme utilizes the spatial property of the motion vector to predict the modes efficiently.

Another fast intermode decision algorithm based on temporal correlation of modes in P slices was proposed in Ref. 5. A time reduction of 57% on average was claimed, with a bitrate increment of 0.07% and a loss of 0.05 dB, as compared to the standard. However, if the local temporal information is unreliable, for example, when the scene changes, the RD performance will be degraded because of mode misprediction.

A recently developed algorithm was proposed in Ref. 6. This scheme achieves up to 63% time savings when compared to the standard reference software. However, the algorithm is based on heuristic analysis obtained from a set of video sequences which can lead to a significant RD degradation if the algorithm is used to encode sequences with different characteristics. Furthermore, the spatial correlations between MBs have been exploited and this correlation is unreliable for sequences with a complex background.

From the information above, it can be seen that fast intermode decision algorithms can achieve time savings in the range of 40% to 65% with some RD performance degradations. It also can be noticed that all the fast intermode decision schemes are based on spatial domain ME information.

Recently, there has been a lot of interest in motion estimation techniques operating in the frequency domain. These are commonly based on the principle of cyclic correlation and offer well-documented advantages in terms of computational efficiency due to the employment of fast algorithms. One of the best-known methods in this class is phase correlation,[7] which has become one of the ME methods of choice for a wide range of professional studio and broadcasting applications.[8] In addition to computational efficiency, phase correlation offers key advantages in terms of its strong response to edges and salient picture features, its immunity to illumination changes and moving shadows, and its ability to measure large displacements. Several attempts[9,10] have been proposed to adapt the phase correlation to the standard. In Ref. 9, the authors proposed an adaptive block size phase correlation ME, which has been compared to the full search block matching (FSBM) algorithm.[11] The comparison results indicated a significant increase in the bitrate. Furthermore, block sizes up to 32×32 were used to estimate the motion which increases the computational complexity. In Ref. 10, the authors used the phase correlation to predict the ME block size by generating a binary matrix, and then selected the

0091-3286/2011/$25.00 © 2011 SPIE

mode from the binary matrix. Although the authors claimed a 50% reduction in the ME time, the algorithm showed significant RD performance degradation for slow video sequences.

In this paper, we propose a novel fast mode decision algorithm. In addition to saving up to 97% of the ME time for similar RD performances, our algorithm differs from the above-mentioned algorithms as it preprocesses the macroblock in the frequency domain using 16×16 phase correlation, and based on these results, we directly predict the mode and the search range.

The rest of the paper is organized as follows. Section 2 describes the proposed mode decision algorithm. Section 3 contains a comprehensive list of experiments and a discussion. Section 4 concludes the paper.

2 Proposed Scheme

In video compression, knowledge of motion helps to exploit similarity between adjacent and nearby frames in the sequence, and remove the temporal redundancy between neighboring frames in addition to the spatial and spectral redundancies.[12] The phase correlation method measures the movement between the two fields directly from their phases. The basic principles are described below.

Assuming a translational shift between the two frames:

$$s_t(x, y) = s_{t+1}(x + \Delta x, y + \Delta y). \tag{1}$$

Their two-dimensional (2D) Fourier transforms are:

$$S_t(f_1, f_2) = S_{t+1}(f_1, f_2)\exp[2j\pi(f_1\Delta x + f_2\Delta y)]. \tag{2}$$

Therefore, the shift in the spatial-domain is reflected as a phase change in the spectral domain. The cross-correlation between the two frames is:

$$C_{t,t+1}(f_1, f_2) = S_{t+1}(f_1, f_2) \cdot S_t(f_1, f_2). \tag{3}$$

The normalized cross-power spectrum is:

$$R_{t,t+1}(f_1, f_2) = \frac{S_{t+1}(f_1, f_2) \cdot S_t^*(f_1, f_2)}{|S_{t+1}(f_1, f_2) \cdot S_t^*(f_1, f_2)|}. \tag{4}$$

From Eqs. (2) and (4), we have:

$$R_{t,t+1}(f_1, f_2) = \exp[-2j\pi(f_1\Delta x + f_2\Delta y)]. \tag{5}$$

The 2D inverse transform is given by:

$$c_{t,t+1}(x_1, y_1) = \delta(x_1 - \Delta x, y_1 - \Delta y). \tag{6}$$

The displacement can be found by using the location of the pulse in Eq. (6). The maximum correlation is achieved when the two images are identical [value = 1 at (0, 0)]. Our observation on the phase correlation results for different images extracted from different video sequences revealed that if the correlation between the macroblock and its prediction is greater than or equal to 0.8; 92% of the time the macroblock contains objects that have a minimum size of 16×8 or 8×16 and the motion vector has a maximum value of 8 in any direction. On the other hand, when the correlation is less than 0.8, this indicates that the contents of the macroblock are either large objects with large movements or a number of small objects with various movements.

Using the above insights, we developed the following algorithm: if the correlation value is equal to 1, then we choose the SKIP mode as the best mode. Otherwise, if the correlation value is greater than or equal to 0.8, we limit the mode selection process to modes {0, 1, 2, and 3}. Additionally, we limit the search range to 8. Finally, if the correlation value is less than 0.8, we enable all the modes and ME is performed using the defined search range. The proposed algorithm is shown in flowchart form in Fig. 1.

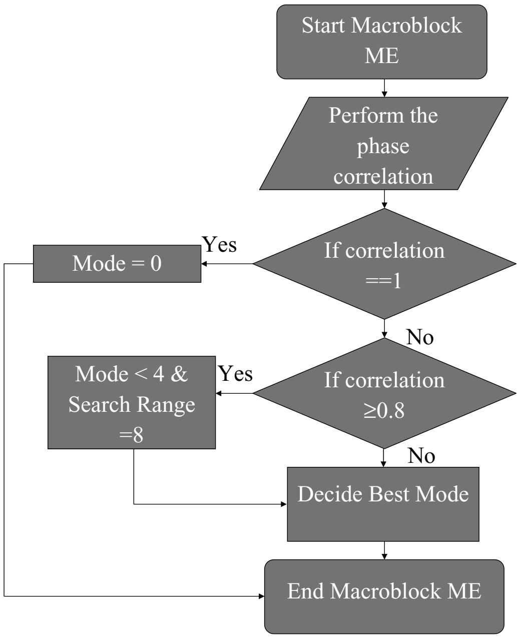

Fig. 1 The proposed algorithm.

3 Experimental Results

To assess the proposed algorithm, a comprehensive set of experiments for eight kinds of video sequences with different motion characteristics was performed.

The chosen search range was 32 pixels for the full ME. The configuration file for the encoder had the following settings: RD optimization ON, IPPP structure, CABAC coding, and the number of reference slices was 1.

In these experiments, the source code for the H.264 Reference Software Version JM14.2 (Ref. 11) was used. Four sizes, QCIF (176×144), CIF (352×288), (640×480), and (1024×768) were used in an Intel Core 2 CPU 6420 @ 2.13 GHz with 2.0 GB RAM. The Intel VTune performance analyzer was used to measure the number of machine cycles differences, reflecting the total encoding time.

Table 1 shows the percentage cycle savings, the percentage search point savings, the Bjontegaard Delta bit rate (BDBR) percentage differences, and the Bjontegaard delta peak signal-to-noise ratio (BDPSNR) differences (in decibels)[13] between the JM software and the proposed new algorithm, and between the proposed algorithm and the algorithm proposed in Ref. 13. In the first comparison, Table 1 shows that the BDBR differences are in the range of 0.2 to 1.3, while the BDPSNR differences are in the range of −0.08 to −0.01. The minus signs denote PSNR degradation and bitrate savings, respectively. This clearly shows that the proposed algorithm has very similar RD performance to H.264/AVC reference software. Furthermore, ME time savings up to 97% and percentage cycle savings up to 67% are

Table 1 Comparison on BDPSNR and BDBR cycle differences and ME time saving between the proposed algorithm and JM software and the algorithm proposed in Ref. 3.

		Against the JM software				Against the algorithm proposed in Ref. 3		
Sequence	Size	BDPSNR (dB)	BDBR (%)	Cycles Saving (%)	ME Time Saving (%)	BDPSNR (dB)	BDBR (%)	ME Time Saving (%)
Akiyo	QCIF	−0.08	+1.3	66.98	97.02	0.03	−0.4	48.12
	CIF	−0.04	+0.96	57.36	86.49	0.04	0.24	42.65
Foreman	QCIF	−0.05	+1.22	36.43	45.17	0.06	−0.7	22.53
	CIF	−0.04	+1.14	35.74	44.68	0.02	−0.05	21.45
Tempete	QCIF	−0.01	+0.61	39.75	44.31	0.04	0.03	24.09
	CIF	−0.05	+0.76	40.07	46.8	0.01	−0.45	26.76
Silent	QCIF	−0.01	+0.36	60.53	80.9	0.03	0.51	50.32
	CIF	−0.03	+0.77	52.69	75.02	0.06	0.25	47.22
Stefan	QCIF	−0.03	+0.3	30.54	36.53	0.05	0.61	19.66
	CIF	−0.04	+0.5	29.39	35.81	0.06	−0.32	18.55
Mobile	QCIF	−0.02	+0.2	26.06	34.88	0.07	0.01	22.49
	CIF	−0.05	+0.6	27.4	32.74	0.01	0.83	20.87
Rena	640×480	−0.05	+0.8	42.5	64.5	−0.03	0.9	29.8
Uli	1024×768	−0.03	+0.6	39.6	51.8	0.05	−0.04	26.72
Average		−0.04	+0.7	41.8	55.4	0.04	0.1	30.08

observed. It also can be seen that the reduction in the CPU cycles depends on the characteristics of the image sequences. For a slow image sequence with a simple background, the reduction is much more significant than for fast image sequences or sequences with a more complex background. The reason for this is that in slow video sequences, the number of big block sizes increases significantly.

The second comparison in Table 1 indicates that the proposed algorithm consistently outperforms a recently proposed approach[3] in all aspects; an average of 30% encoding time savings, 0.04 dB PSNR improvement, and 0.1% total bit rate reduction.

Moreover, when comparing the results to the results in Ref. 10, in addition to the significant time reduction gain (40%), the RD performance is maintained similar to the JM software for the various sequences, while in Ref. 10, the performances have been degraded rather significantly for some of the sequences.

4 Conclusion

The H.264/AVC increases memory bandwidth and spends a significant amount of processing time for the motion estimation process in order to determine the optimal motion vector. As a means of increasing the coding efficiency, in this paper, we proposed a fast mode decision and a motion estimation scheme with rate distortion performance similar to the standard. Our technique can reduce up to 97% of the ME time (67% in CPU cycles), resulting in significant time/cycle savings as compared to H.264/AVC. It is very relevant to low complexity video coding systems.

References

1. T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 560–576 (2003).
2. X. Jing and L. P. Chau, “Fast approach for H.264 inter-mode decision,” *Electron. Lett.* **40**(17), 1050–1052 (2004).
3. J. Bu, S. Lou, Ch. Chen, and J. Zhu, “A predictive block-size mode selection for inter frame in H.264,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol 2, pp. 917–920, IEEE, Toulouse, France (2006).
4. L. Shen, Z. Liu, Z. Zhang, and X. Shi, “Fast inter mode decision using spatial property of motion field,” *IEEE Trans. Multimedia* **10**(6), 1208–1214 (2008).
5. B. G. Kim, “Novel inter-mode decision algorithm based on macroblock tracking for the p-slice in H.264/AVC video coding,” *IEEE Trans. Circuits Syst. Video Technol.* **18**(2), 273–279 (2008).
6. H. Zeng, C. Cai, and K. K. Ma, “Fast mode decision for H.264/AVC based on macroblock motion activity,” *IEEE Trans. Circuits Syst. Video Technol.*, **19**(4), 491–499 (2009).
7. J. J. Pearson, D. C. Hines, S. Goldman, and C. D. Kuglin, “Video rate image correlation processor,” *Proc. SPIE* **119**, 197–205 (1977).
8. G. A. Thomas, “Television motion measurement for DATV and other applications,” *BBC Res. Dept. Rep.* (1987).
9. Y. Ismail, M. Shaaban, and M. Bayoumi, “An adaptive block size phase correlation motion estimation using adaptive early search termination technique,” *IEEE International Symposium on Circuits and Systems*, pp. 3423–3426, IEEE, New Orleans, LA (2007).
10. M. Paul and G. Sorwar, “An efficient video coding using phase-matched error from phase correlation information,” *IEEE 10th Workshop on Multimedia Signal Processing*, pp. 378–382, IEEE, Cairns, Australia (2008).
11. Source code link: http://iphome.hhi.de/suehring/tml/download/old_jm/jm14.2.zip.
12. C. Stiller and J. Konrad, “Estimating motion in image sequences,” *IEEE Signal Processing Magazine*, **15**(4), 70–91 (1999).
13. G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” *Doc. VCEG-M33* wftp3.itu.int/av-arch/video-site/0104_Aus/VCEG-M33.doc (2001).