Improved Fractional Pixel Motion Estimation Algorithm for H.264/AVC

Yanfei Shen Feng Dai Yongdong Zhang Jintao Li

(Institute of Computing Technology of Chinese Academy of Sciences)
Email: {syf,fdai,zhyd,jtli}@ict.ac.cn

Abstract: In H.264/AVC fractional pixel motion compensation is adopted in order to achieve more accurate motion description and higher compression efficiency. It, however, remarkably increased computation load in Motion Estimation (ME) process for video coder. In this paper an improved fractional pixel ME algorithm is proposed to reduce its computation load, which is based on the fact that the distribution of the fractional Motion Vector (MV) is centered on the best integer pixel. Experiment results show that our proposed algorithm can reduce 75% computation load at most and can also keep rate distortion performance as compared with full fractional ME method used in H.264 video coder.

Keywords: fractional pixel motion estimation video coding fast motion estimation algorithm H.264

1. Introduction

H.264/AVC[1] is the newest video coding standard of ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). The main goals of the H.264/AVC standardization effort have been enhanced compression performance and provision of a "network-friendly" video representation. It has achieved a significant improvement in rate-distortion (RD) efficiency — providing, typically, a factor of two in bit-rate savings when compared with existing standards such as MPEG-2 Video. However, many enhanced specifics introduced to the H.264 considerably affect the efficiency, performance and complexity of the motion estimation process. Rather than exploiting a fixed block size (16×16 or 8×8), H.264 supports a more flexible M×N partition of block sizes, varying among 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4. To achieve the highest coding efficiency, complexity mode decision techniques based on rate-distortion optimization require checking all possible inter-modes with motion estimation (ME). The long-term multi-frame motion-compensated prediction is a mandatory powerful tool supported by H.264. Thus it is unavoidable to increase the amount of calculations from the motion estimation.

In H.264, the ME process is divided into two steps: integer pixel ME and fractional pixel ME. Generally integer pixel ME takes most of the computational cost of the whole ME. However, with the development of fast integer ME algorithm, the computational cost of integer pixel ME has been greatly reduced [2][3][5][6][7]. For instance, the fast algorithm based on merge and split procedures decreases the average number of integer pixel search points to 6. However the conventional Full Fractional Pixel Search (FFPS) at quarter pixel accuracy needs to check 16 sub-pixel search points. Therefore the computational cost of fractional pixel ME becomes comparable to that of integer pixel ME, and even much higher in some cases. How to speed up the fractional pixel ME tends to the key issue in simplifying the whole ME process.

To reduce the complexity of fractional pixel motion estimation, several fast algorithms have been proposed. Their common idea is to simplify the search pattern[8][9][10]. Based on the assumption that the error surface is monotonic, two search patterns are proposed in to refine the motion vectors surrounding the point with only a minimal error. Similarly, based on the observation that the cost function is a smooth convex function in the prediction area, a Paraboloid Prediction based Fractional Pixel Search (PPFPS) algorithm [9] examines three half-pixel positions surrounding the best integer point and three quarter-pixel positions between the best and sub-optimal half-pixel positions only. However, both methods involve the costs of examining the neighboring integer positions surrounding the best integer pixel position. As a result, they can only work with the integer pixel motion estimation algorithms that satisfy the monotonic requirement. The Center Biased Fractional Pixel Search (CBFPS) [2] solves this problem well by checking the predicted position directly and refining the motion vector surrounding it. However, it suffers from falling into a local minimum in early stages because the predicted motion vectors are not accurate enough. In addition, in order to gain the half-pixel value, a 6 tap interpolation filter is used in H.264/AVC video coding standard, quarter-pixel value is calculated by half-pixel value. The interpolation process also consumes much computation load and memory resources for video encoder and it is very difficult to produce the fraction pixel value in real time in motion search.

In this paper an improved fractional pixel ME algorithm is proposed to reduce its computation load, which is based on the fact that the distribution of the fraction Motion Vector is centered on the best integer pixel, 80% computational cost of FFPS can be reduced at most in some cases. Moreover, our method does not require the costs of examining the neighboring integer pixel points, which allows it to work with any integer pixel ME algorithm.

The rest of the paper is organized as follows. Section 2 describes the fractional pixel ME methods proposed in H.264. Section 3 presents the characteristics of fractional

0-7803-9737-1/06/$20.00 ©2006 IEEE

pixel MV distribution. Our proposed fast fractional pixel ME algorithm is described in detail in section 4. Finally, complexity analysis and simulation results are shown in section 5.

2. Fractional Pixel ME Algorithms Proposed in H.264

In H.264/AVC, the fractional pixel search algorithms are constrained within the area bounded by eight neighboring integer pixel positions around the best position. A typical full fractional pixel search algorithm is hierarchical fractional pixel search algorithm, as provided in the JM test model. Fig.1 shows a typical process of the conventional FFPS algorithm. Firstly, its examines eight half pixel positions surrounding the best integer pixel position and then obtains the best half pixel resolution motion vector, then it checks eight quarter pixel positions surrounding the best half pixel position and obtains the best quarter pixel resolution MV.

A fast half pixel motion estimation algorithm, Center Biased Fractional Pixel Search (CBFPS) has been adopted by H.264 and integrated into the reference software. Firstly it predicts the fractional pixel MV of the current block using the median MV of the neighboring blocks. Then it examines the zero fractional pixel MV and the predicted MV. The one possessing the minimum matching error is chosen to be the start position. Finally the algorithm employs a diamond search method with a distance of quarter pixel to refine the fractional pixel MV. However the predicted MVs are not accurate enough in some cases, which results in a relatively large degradation in performance. Experimental results shows that compared with full search, this algorithm can reduce the computation load by 33%.

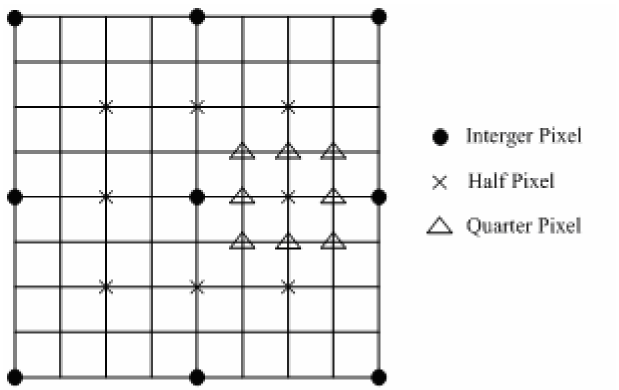

Fig.1 A typical process of full fractional pixel search

3. Characteristics of Fractional Pixel MV Distribution

In H.264/AVC, the fractional pixel search algorithms are constrained within the best integer pixel. The best quarter pixel accuracy MVs are obtained through above hierarchical fractional pixel search algorithm adopted in H.264/AVC reference software and 16 fractional search points need to be checked.

The best MVs are represented in quarter pixel accuracy, which includes the information of predicted integer pixel motion vector and predicted fractional pixel motion vector. We can extract the predicted fractional pixel motion vector information by using the following formula:

$$fractionalMV = bestMV \% \beta \qquad [1]$$

where the $bestMV$ is the best motion vectors in fractional pixel accuracy obtained by FFPS. $\%$ is the mode operation, β is fractional motion vector accuracy and $\beta = 4$ for quarter pixel case.

The characteristics of fractional pixel MV distribution are shown in Fig.2. We can see that most fractional pixel motion vector is the same as the integer motion vector, which is also demonstrated in table.1, especially for low motion video test sequences, such as Akiyo and Container, where above 80% difference between integer pixel accuracy MV and quarter pixel accuracy MV is less than one quarter pixel unit.

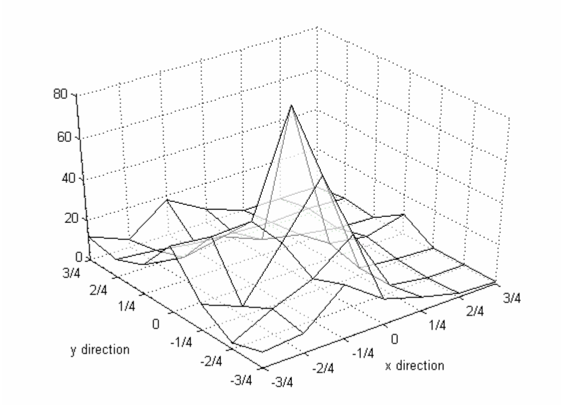

Fig.2 Characteristics of fractional pixel MV distributed

Table.1 The probability where the difference between integer pixel accuracy MV and quarter pixel accuracy MV is less than one quarter pixel unit

sequence	Foreman	Mobile	Paris	Akiyo	Container
	37%	24%	60%	92%	88%

4. Proposed Fast Fractional Pixel Motion Estimation Algorithm

For some successful integer pixel motion estimation algorithm, they are all based on the assumption that the matching error surface inside the search window is unimodal, as shown in Fig.3 [2], so that the matching error decreases monotonically as the search points moves closer to the global optimum. Our proposed fast fractional pixel search algorithm for H.264/AVC is also based on the assumption of unimodal for the fractional pixel error surface.

Based on the above characteristics of fractional pixel MV distribution, in our proposed algorithm, the candidate search points nearby the best integer pixel position are searched firstly in horizontal in quarter pixel unit step. If the best matching position is located at the middle place of three horizontal candidate search points or the search range is larger than one integer pixel unit, then the horizontal ME is stopped and the position which has the minimal Sum of Absolute Difference (SAD) is selected as the center of the vertical ME. Secondly, the candidate search points nearby the best horizontal quarter pixel positions are searched vertically in quarter pixel unit step. If the best matching position is located at the middle place of three vertical candidate search points or the search range is larger than one integer pixel unit, the position which has the minimal SAD is selected as the best motion vector. Our proposed fast fractional pixel ME search pattern is shown in Fig.4.

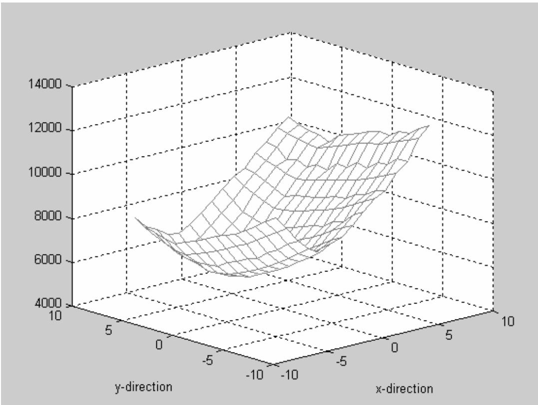

Fig.3 Error surface of fractional pixel motion estimation

Based on the above observations and analysis, our proposed fast fractional pixel ME algorithm can be split two separate steps in implementation: one is the horizontal search and the followed is the vertical search. They are outlined as follows:

Horizontal Search

Start
  Check the best integer pixel position and two neighbor horizontal quarter pixel position
  If the integer pixel position is the best matching position
    Stop horizontal search and set the integer pixel position as the center of vertical search.
  Else if the best matching position is located at the left of the integer pixel position
    Continue to check the left quarter pixel position until the best matching position is located at the middle place of three horizontal candidate search points or the search range is larger than one integer pixel unit, then Stop horizontal search and set the best matching pixel position as the center of vertical search.
  Else if the best matching position is located at the right of the integer pixel position
    Continue to check the right quarter pixel position until the best matching position is located at the middle place of three horizontal candidate search points or the search range is larger than one integer pixel unit, then Stop horizontal search and set the best matching pixel position as the center of vertical search.
End

The process of vertical search is the same as horizontal search except that horizontal search direction are replaced by vertical search direction. If the best matching position is the integer pixel position, only four neighbor quarter pixel positions need to be checked and about 75% computation load are saved, as shown in Fig.4. Complexity analysis will be detailed in the next section.

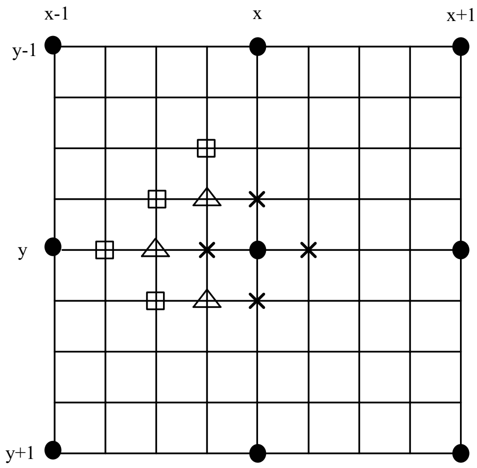

Fig.4 Fast fractional pixel ME search pattern

5 Simulation

The proposed algorithms are evaluated on the framework of H.264/AVC JM73 encoder [11]. We set the motion search range to 16, and set the number of reference frames to 1. The RD optimization and the Hadamard transform are enabled, the CABAC (context-based adaptive binary arithmetic coding) entropy coding option is selected in our experiments. The GOP structure is IPPP, QP parameter is set 30 because other QPs provided similar results.

We choose six test sequences with motion activities varying from small to large. They are Foreman, Paris, Akiyo, Bus, Claire, and Mobile. All sequences are in CIF format (4:2:0) and encoded at 30 frames per second. The number of encode frames are 200. All frames except for the first one are encoded as P-frames.

A. Performance analysis

Table.2 illustrates the coding performance of Full Fractional Pixel Search (FFPS) algorithm, our Proposed Fractional Pixel Search (PFPS) algorithm, the Center Biased Fractional Pixel Search (CBFPS) and Half Pixel Motion Vector Resolution (HPMVR) respectively for different test sequences. From the simulation results, we can see that our PFPS algorithm can achieve good results. Compared with the FFPS, our algorithm can save 75% computation load at most and the search points is reduced from 16 to 4 points, while the PSNR degradation is less than 0.05dB on average, at the same time, the bit rates remain almost unchanged. Compared with the Center Biased Fractional Pixel Search algorithm adopted by H.264 JM reference software, our proposed PFPS algorithm has minor performance degradation, but CBFPS can save only 33% computational load while our proposed can save 75% computation load at most especially for small motion video sequences.

Table.2 Comparison of different methods

Sequence	Method	QP = 30	
		PSNR(dB)	Bitrate (kbps)
Foreman	FFPS	35.33	301.66
	PFPS	35.31	312.28
	CBFPS	35.32	301.76
	HPMVR	35.10	344.7
Paris	FFPS	33.78	356.35
	PFPS	33.76	363.42
	CBFPS	33.75	355.91
	HPMVR	33.66	398.25
Akiyo	FFPS	38.18	55.33
	PFPS	38.16	56.90
	CBFPS	38.16	55.13
	HPMVR	38.06	66.72
Bus	FFPS	33.13	823.08
	PFPS	33.10	854.82
	CBFPS	33.12	842.08
	HPMVR	32.83	1055.73
Claire	FFPS	40.10	57.29
	PFPS	40.03	59.94
	CBFPS	40.01	58.57
	HPMVR	39.84	68.04
Mobile	FFPS	32.02	1265.04
	PFPS	31.99	1305.86
	CBFPS	32.01	1264.89
	HPMVR	31.89	1718.88

B. Complexity analysis

In Fig.5, the minimum checking points required in our proposed fast fractional motion estimation algorithm are illustrated. We can see that the candidate nearby the integer pixel position require fewer checking points. This can save more computation load for small motion video sequences where approximate 80% fractional pixel accuracy motion vector is the same as the integer pixel.

6 Conclusion

In this paper, we propose a fast fractional pixel motion estimation algorithm including a novel search pattern. Our PFPS algorithm can reduce 75% computation load at most and the search points is reduced from 16 to 4 points as compared with conventional FFPS, this greatly accelerates the speed of fractional pixel motion search for real time video coder. Simulation results show that the performance degradation is negligible and the proposed algorithm can be easily integrated into existing video coding systems to enhance the performance of real time video coder.

Acknowledgements

This research was supported by the National Nature Science Foundation of china (60473002 and 60302028).

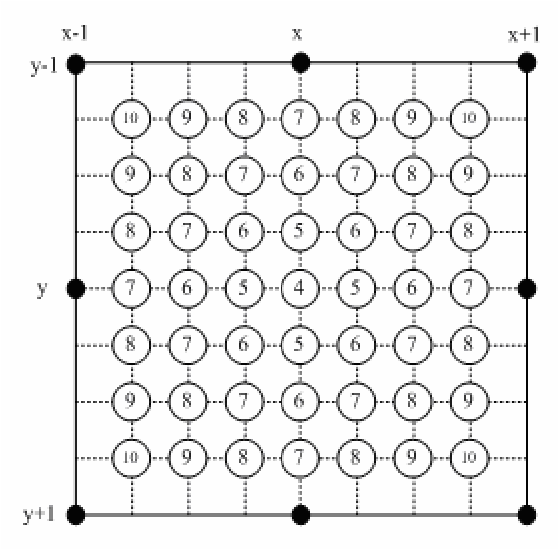

Fig.5 minimum checking points required in our proposed fast fractional motion estimation algorithm

Reference

[1] T.Wiegand et al., "Overview of the H.264/AVC video coding standard,"IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–576,Jul. 2003.

[2] Z.B. Chen, P. Zhou, and Y. He, "Fast integer pel and fractional pel motion estimation for JVT," JVT-F017, 6th meeting: Awaji, Japan, 5-13 Dec. 2002

[3] T. Koga, K. Iinuma, A. Hirano, Y. Lijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proceedings Nat. Telecommunications Conf. 81, New Orleans, LA, pp.G5.3.1-G5.3.5, Nov. 1981.

[4] J.Jain, A.Jain, "Displacement measurement and its application in interframe image coding," IEEE Transactions on Communications, vol.COM-29, pp.1799-1806, Dec. 1981.

[5] Ce Zhu, Xiao Lin, and Lap-Pui Chau, "Hexagon-Based Search Patten for Fast Block Motion Estimation", IEEE Trans. on CSVT, pp.349-355, Vol.12, No.5, May, 2002.

[6] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast blokc-matching motion estimation," IEEE Trans. on Image Processing, vol. vol.9, no. no.2, pp. pp. 287–290, Feb. 2000.

[7] S. Zhu and K.-K. Ma, "A New Diamond Search Algorithm for Fast Block Matching Motion Estimation," *in Proc. Int. Conf. Inform., Commun.*, Signal Process., pp.292–296, Singapore, Sept.9–12, 1997.

[8] Libo Yang, Keman Yu, Jiang Li, and Shipeng Li, "Prediction-based Directional Fractional Pixel Motion Estimation for H.264 Video Coding", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, USA, 18-23 March, 2005

[9] P. Yin, H.Y. C. Tourapis, A.M. Tourapis and J. Boyce, "Fast mode decision and motion estimation for JVT/H.264," in Proc. of ICIP 2003, pp.853-856, Sep.2003

[10] Cheng Du, Yun He, and Junli Zheng, "PPHPS: A Parabolic Prediction-Based, Fast Half-Pixel Search Algorithm for Very Low Bit-Rate Moving-Picture Coding", IEEE Trans. on CSVT, pp.514-518, Vol.13, No.6, June, 2003.

[11] JVT reference software JM73 http://bs.hhi.de/~suehring/tml/download/jm73.zip