IEICE Electronics Express, Vol.7, No.14, 982–987

One-step fractional motion estimation in H.264/AVC based on early predicted candidate

Nam Thang Ta[a)] **and Jun Rim Choi**

School of Electronics Engineering, Kyungpook National University

San-kyuk Dong, Daegu, 702–701, Korea

a) *thangta@ee.knu.ac.kr*

Abstract: The conventional two-step fractional motion estimation algorithm has been broadly adopted in the literature due to its high encoding performance. However it induces a huge computational complexity as well as a long latency. In this paper, we propose a fast fractional motion estimation algorithm. Based on high correlation between the motion vector of a block and its up-layer, as well as relationship of integer candidates, a one-step algorithm is proposed which not only reduces the computational complexity by eliminating unnecessary fractional-pels, but also saves hardware cost. Experimental results show that the proposed design can save 32% gate count, and reduce the latency by 39% compared with previous designs, while nearly maintaining the coding performance.

Keywords: h.264, fractional motion estimation, video signal processing

Classification: Integrated circuits

References

[1] ITU-T Recommendation and International Standard of Joint Video Specification. ITU-T Rec. H.264/ ISO/ IEC 14496-10 AVC, March 2005.

[2] Y. J. Wang, C. C. Cheng, and T. S. Chang, "A fast algorithm and its VLSI architecture for fractional motion estimation for H.264/MPEG-4 AVC coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 5, pp. 578–583, May 2007.

[3] T. C. Chen, Y. W. Huang, and L. G. Chen, "Fully utilized and reusable architecture for fractional motion estimation of H.264/AVC," *ICASSP*, vol. 5, pp. 9–12, May 2004.

[4] N. T. Ta, J. H. Kim, S. H. Kim, and J. R. Choi, "Fully parallel fractional motion estimation for H.264/AVC encoder," *ICIS 2009*, vol. 4, pp. 306–309, Nov. 2009.

[5] 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.

[6] Z. B. Chen and Y. He, "Prediction based directional refinement (PDR) algorithm for fractional pixel motion search strategy," *JVT-D069, 4th meeting: Klagenfurt*, Austria, 22-26 July 2002.

© IEICE 2010

DOI: 10.1587/elex.7.982
Received April 08, 2010
Accepted June 25, 2010
Published July 25, 2010

[7] Y. Song, Y. Ma, Z. Liu, T. Ikenaga, and S. Goto, "Hardware-oriented direction-based fast fractional motion estimation algorithm in H.264/AVC," *ICME 2008*, pp. 1009–1012, June 2008.
[8] JVT H.264/AVC Reference Software JM version 9.8.

1 Introduction

Motion estimation (ME) in the latest video standard H.264/AVC [1] is comprised of two parts: first integer motion estimation (IME) is used to find the integer motion vector (MV), then fractional motion estimation (FME) is performed to refine this MV to quarter pixel precision. To estimate motion, the sum of absolute difference (SAD) and the sum of absolute transformed difference (SATD) are used as the criterions of matching distortion in IME and FME respectively. The conventional FME approach needs a two-step interpolation for half-pel and quarter-pel refinements. Though this two-sequential-step brings high encoding performance, it introduces a long latency as well as huge computational load for searching seventeen fractional points for each of the forty-one MVs. Therefore, instead of IME, FME has recently become the bottleneck of the whole H.264/AVC encoding system. Some papers in literature focus on existing FME issues. Motivated by the high correlation of cost between neighboring fractional candidate positions in [2], the number of fractional candidates is reduced to eight or nine instead of seventeen candidates as reference software JM [8]. The reference [2] not only reduces the computation complexity by 50% compared with reference software, but also saves hardware cost. However, the issue of long latency has not been solved yet because of the two-sequential-step. To shorten the latency, a one-step algorithm is proposed in our previous work [4]. It is a straightforward design and is suitable for hardware with fixed twenty-one fractional-pel positions. Though this architecture can reduce the long latency of interpolation processing by half, it occupies quite a large amount of hardware area. To reduce the complexity of fractional processing, Center Biased Fractional Pel Search (CBFPS) was proposed in [5] and has also been adopted into the reference software. First this method predicts the fractional-pel MV of the current block using the median MV of neighboring blocks. Then it examines the zero sub-pel MV and the predicted fractional-pel MV. The one that yields the minimum matching error is chosen to be the start position. Finally, the approach in [5] employs a diamond search pattern to refine the fractional MV. The diamond search just stops with specified number of steps or minimum cost point located at center. However, for large blocks, the predicted MVs are not accurate enough. Thus, the H.264 reference software still uses full search for the large blocks including 8×16, 16×8 and 16×16 blocks. By examining the neighboring integer positions surrounding the best integer-pel position, the prediction based directional refinement algorithm for fractional pixel motion search strategy is proposed in [6]. A half-stop judgment rule is also adopted to stop the search process when the current minimum cost is be-

© IEICE 2010
DOI: 10.1587/elex.7.982
Received April 08, 2010
Accepted June 25, 2010
Published July 25, 2010

low a certain threshold. According to this approach, the computational load of fractional-pel motion estimation can be reduced from 17.4% to 34.7% of the original full fractional-pel search algorithm. However, this method only estimates four sub-SAD surrounding the best SAD integer and ignores the four other positions in diagonal directions. Moreover, this approach seems to be suitable for software C-model only, it can not be directly adapted to hardware implementation due to the requirement of four surrounding SAD integers. Same as the idea in [6], the approach in [7] early eliminates one half-pel position by estimating four surrounding sub-SAD, but it has two drawbacks. One is that the approach in [7] has to estimate four integer-pel positions twice, while saving only one half-pel position. The other is that all predicted directions of forty-one MVs are imposed by the predicted direction of a 16×16 block, which causes inaccurate prediction of fractional candidates.

Obviously, the long latency of FME mainly comes from the two-step procedure with seventeen search points in total. In this paper, we propose a one-step FME algorithm based on early predicted fractional points. The rest of the paper is organized as follows. Firstly, a one-step algorithm is proposed in section 2 with selection of predicted fractional motion vector and followed by video quality evaluation. The synthesis results and comparisons are presented in section 3. Finally, a conclusion will be given in section 4.

2 Proposed one-step algorithm

2.1 Selection of predicted fractional motion vector

As earlier analyzed, the predicted MV that is obtained from the median MV of the adjacent blocks on the left, top and top-right (or top-left) is not accurate enough in a large block. Therefore, it is not chosen as the prediction MV in our paper. In H.264, there are seven kinds of block size partitions from 16×16 to 4×4, and respectively seven modes [1]. It has been found that a block and its up-layer block are likely to have the same motion vector. Motivated from this, we use the MV of larger block as the prediction of the smaller block contained in it. For example, a 16×16 block is up-layer of the 16×8 and 8×16 blocks in it, while a 16×8 block is up-layer of the 8×8 block and so on. Thus the motion vector of the larger 16×16 block is used as the starting MV for the contained 16×8 and 8×16 blocks. Similarly, the MV of 16×8 block is used to predict the MVs of the contained smaller 8×8 blocks. Note that a 16×16 block does not have a larger block for use in up-layer prediction.

2.1.1 Processing 16×16 block

For prediction of the MV of a 16×16 block, we found that most distributions of fractional positions is biased into the region of the best SAD integer and the second best SAD integer positions. Thus the fractional-pel candidate can be predicted early by these best integer-pel positions, and the distribution of fractional candidate can be restricted.

© IEICE 2010
DOI: 10.1587/elex.7.982
Received April 08, 2010
Accepted June 25, 2010
Published July 25, 2010

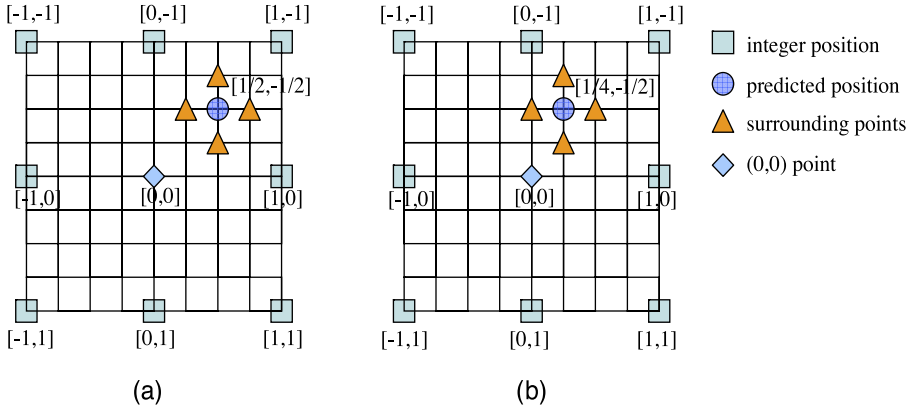

Fig. 1. (a) Prediction of 16×16 block. (b) Prediction of the rest of blocks.

In the proposed algorithm, we employ the best and second best SAD integer to predict the fractional point considered as starting MV of a 16×16 block. This method should also employ a threshold T to determine whether the half-pel position is biased to the center or not. For simplicity, the proposed algorithm is presented as follows:

(i) First half-pel position *h_pel* is determined by:

if ($\Delta SAD >$ *threshold_T*)

$h\_pel = h\_center;$

else

$h\_pel = h\_pred;$

where $\Delta SAD = |SAD\_2nd\_best - SAD\_best|$

h_center is the candidate at position $[0, 0]$ and *h_pred* is the half-pel position predicted by the second best integer as illustrated in Fig. 1 a. If the second best SAD integer is located at $[1, -1]$, *h_pred* derives the coordinate corresponding to the position at $[1/2, -1/2]$. If the second best SAD integer is located at $[1, 1]$, *h_pred* derives the position at $[1/2, 1/2]$ and so on.

In our tests, the *threshold_T*is set to 255.

(ii) After finding out *h_pel*, four diamond points (quarter positions) are searched around *h_pel*, and one plus-half-pel position is also examined. If *h_pel* is *h_center*, plus-half-pel will be *h_pred*, otherwise plus-half-pel will equal to *h_center*.

Note that the second best SAD integer-pels are estimated here among eight surrounding SAD integers. Two line of SAD buffers are employed in the IME part in order to estimate the second best SAD integer, and thus there is no duplication of SAD calculation. Two more comparators, one is a 8-inputs and the other is a 3-inputs, are used to find the best and second best integer SAD position. As a consequence, the predicted fractional candidate of a 16×16 block is available at the time of starting the FME task.

2.1.2 Processing the rest of blocks

In the proposed algorithm, the predicted MV of a block (*pred_mv*) is determined by the MV of its up-layer. First the predicted fractional motion vector

© IEICE 2010
DOI: 10.1587/elex.7.982
Received April 08, 2010
Accepted June 25, 2010
Published July 25, 2010

($frac\_pred\_mv$) is extracted by:
$frac\_pred\_mv = (pred\_mv - mv) \bmod 4$
where $pred\_mv$ is defined as the fractional unit, mv is the integer MV in the fractional unit. The result of the predicted fractional MV shows the fractional position as illustrated in Fig. 1 b. Then four diamond points are adopted around the $frac\_pred\_mv$ position. If the predicted fractional MV is not at the center $(0, 0)$, one additional point located at $(0, 0)$ is added. As a result, the proposed one-step algorithm only searches six fractional candidates in a unique step which reduces interpolation processing by half and the number of search points by 64.7%.

2.2 Video quality evaluation

The proposed FME algorithm is integrated into the reference software and tested with several standard test sequences. These video CIF (352×288) at 30 fps are chosen with different motion levels from slow to fast motion including *Flower garden*, *Highway*, *Coastguard*, *Mobile* and *Stefan*. The sequence type is IPPP and 90 frames for Stefan, 100 frames for the rest encoded with no rate-distortion optimization (RDO). Table I shows the average performance of the proposed algorithms with different QPs (16, 20, 24, 28, and 32). The average PSNR drop and increasing bit rate are 0.031 dB and 0.869% respectively, no more than 0.042 dB and 1.076% in the worst case. The video quality degradation is negligible while 64.7% of the computation complexity reduced.

Table I. Video quality loss compared with JM.

		One-step algorithm
Flower	PSNR(dB)	0.021
	Bitrate (%)	1.012
Highway	PSNR(dB)	0.023
	Bitrate (%)	0.928
Coast guard	PSNR(dB)	0.032
	Bitrate (%)	0.634
Mobile	PSNR(dB)	0.035
	Bitrate (%)	1.076
Stefan	PSNR(dB)	0.042
	Bitrate (%)	0.648
Average	PSNR(dB)	0.031
	Bitrate (%)	0.869

3 Experimental results

The proposed architecture has been implemented by Verilog and synthesized using the Chartard 0.18 um standard CMOS 1P5M technology. Table II shows the hardware comparison of the proposed FME architecture with previous designs. Our hardware design is slightly larger than that of [2], but it is two times faster. Compared with [3], our design can save 32% of gate

© IEICE 2010
DOI: 10.1587/elex.7.982
Received April 08, 2010
Accepted June 25, 2010
Published July 25, 2010

Table II. Hardware comparison.

	Technology	Gate count	Latency
[2]	0.18um	48.0 K	2000
[3]	0.18um	79.3 K	1648
Proposed	0.18um	54.1 K	998

count due to the smaller number of processing units. It also reduces the cycle count by 39% owing to the one-step proposed algorithm.

4 Conclusion

The one-step FME algorithm based on predicted fractional-pel position is proposed in this paper. The issues of FME are resolved by the proposed architecture with low cost, latency and computation complexity. The proposed FME architecture saves 32% of gate count, and reduces the latency by 39%, while almost maintaining the coding performance.

Acknowledgements

The authors would like to thank the System Semiconductor Promotion Center and BK21 project (Brain Korea 21) for supporting this research work.

© IEICE 2010
DOI: 10.1587/elex.7.982
Received April 08, 2010
Accepted June 25, 2010
Published July 25, 2010