2019 IEEE 4th International Conference on Signal and Image Processing

Fast Sub-pixel Prediction Based on Error Surface Fitting for HEVC

Lei Cheng[1], Wei Yan[1*], Guoqing Xiang[2] , Xiao Liu[1] , Yunyao Yan[2]
[1]School of Software and Microelectronics, [2]National Engineering Laboratory of Video Technology,
Peking University
Beijing 100871, P.R. China
e-mail: leicheng@pku.edu.cn, yanwei@ss.pku.edu.cn*, gqxiang@jdl.ac.cn, 1601210409@pku.edu.cn,
yunyao_yan@163.com

***Abstract*—The sub-pixel motion estimation is a very important module for improving the motion estimation accuracy in the HEVC video codec standard, which is highly computationally intensive and very time consuming. With the maturity of fast integer-pixel motion estimation algorithm, the time cost of sub-pixel motion estimation becomes more and more significant, so it is very important to develop fast sub-pixel motion estimation algorithm. In this paper, an innovative fast algorithm is proposed for the sub-pixel motion estimation. The algorithm uses the 6-parameter quadratic function to fit the residual surface function, and the matching residuals of the optimal integer pixel prediction point obtained by the whole pixel motion estimation and its adjacent 8 integer pixel points to determine the coefficient of the function, and then the extreme point of the function is obtained, the types of extremum points are discussed, and finally different 1/4 pixel search strategies are applied to the extreme points of different classes. The experimental results show that the algorithm reduces the average number of sub-pixel points to 5 points, greatly reduces the sub-pixel prediction time and computational complexity, while the image coding quality is basically unchanged.**

Keywords-HEVC; motion estimation; sub-pixel prediction; residual surface fitting;

I. INTRODUCTION

HEVC is the latest video coding standard developed by JCT-VC. With many new technologies, HEVC can save more than 50% bit rate under the same video quality as H.264/AVC, but at the same time its computational complexity has doubled. H.265/HEVC coding standard has been widely used in mobile video terminals, Internet video and other applications. Therefore, the efficient implementation of HEVC codec has great practical significance and economic benefits. Motion estimation is a process of obtaining motion vectors, where motion vectors (MV) refer to the displacement of the coordinates of the current block to be coded and the best matching block within a given search range in the reference video frame. With the motion estimation technique, only the motion vector and residual data (the difference between the current block to be coded and the predicted block) need to be transmitted, thereby greatly improving the coding efficiency. Motion estimation usually involves a two-step search: first, the optimal integer pixel MV is found in a given search window by integer pixel motion estimation; then, the fractional pixel motion estimation is used to search sub-pixel points around the optimal integer pixel point.

Motion estimation is also the most computationally intensive and time-consuming module in video coding, so the research on fast motion estimation algorithm has been a hot topic in the field of video coding. Some fast motion estimation algorithms are adopted in HEVC reference software, such as the new three-step search method, four-step search method and UMHexagonS algorithm for integer pixel motion estimation.The algorithm for the sub-pixel motion estimation is classic hierarchical search method: first, traverse the optimal integer pixel and its surrounding 1/2 pixel points to get the optimal 1/2 pixel point, and then traverse the 1/2 optimal pixel and 8 surrounding 1/4 pixel points to get the optimal 1/4 pixel point. Fast integer pixel motion estimation has reduced the number of search points to less than 10. In contrast, the fractional-pixel motion uses a cascaded two-step search method to search at least 16 points. If a sub-pixel interpolation calculation is added, then the complexity of fractional-pixel motion estimation (FME) is significantly higher than the integer pixel motion estimation (IME). [1] gives the ratio of the occupation time of each module of HEVC encoder, in which FME module accounts for 45.9%, while IME module only accounts for 18.3%.

The reason why the above FME algorithm complexity is so high can be summarized into two points: one is the interpolation generation process of the sub-pixel values, and the other is the calculation and comparison process of the RDCost between the matching blocks. The direction of an optimized FME algorithm is to try to reduce the number of search points. In this paper, we propose a fast sub-pixel motion estimation algorithm that only performs 1/4-precision pixel search. The algorithm uses the 6-parameter quadratic function to fit the residual surface function, and the matching residuals of the optimal integer pixel prediction point obtained by the whole pixel motion estimation and its adjacent 8 integer pixel points to determine the coefficient of the function, and then the extreme point of the function is obtained, the types of extremum points are discussed, and finally different 1/4 pixel search strategies are applied to the extreme points of different classes.

The organization of this paper is as follows: the second section reviews the existing FME fast algorithm research. The third section details the proposed innovative fast algorithm. The fourth section gives a summary analysis of the experimental results of the algorithm. The fifth section summarizes the paper.

978-1-7281-3660-8/19/$31.00 ©2019 IEEE

II. RELATED LITERATURE

J. W. Suh et al. [2] describes a 9-parameter quadric surface function, a 6-parameter quadric surface function and a 5-parameter quadric surface function. The quadric surface function is used to predict errors around the integer pixel accuracy MV, and the motion estimation accuracy is extended directly from the integer pixel accuracy to the sub-pixel accuracy. There is no interpolation process, which reduces the computational complexity.

P. R. Hill et al. [3] and S. Dikbas et al. [4] use the quadric surface functions with six parameters proposed in [2]. The difference is that they do not use matrix to solve function coefficients as in [2], but use simple integer pixel SAD values to solve function coefficients, which reduces the computational complexity.

Jing-Fu Chang et al. [5], Z. Wei et al. [6] and W. Dai et al. [7] both use the 5-parameter quadric surface function to fit the error surface function of the sub-pixel position, and combine the specific search strategy to reduce the number of search points. Y. Li et al. [8] takes the 6-parameter quadric surface function to fit the error surface function of the sub-pixel position, to reduce the number of search points.

W. Lin et al. [9], K. Ng et al. [10] and L. Wang et al. [11] model the SAD curve as two one-dimensional linear functions which is a simplification of quadric surface function in [2], then combines directional search strategy to carry out sub-pixel search.

In [12], firstly, the information of MV and MV of IME in AMVP is used to judge whether the current encoding region is static. FME is skipped directly in the static region, and 1/2 pixel and 1/4 pixel motion searches are carried out in the dynamic region using the direction adaptive search algorithm.

E. Badry et al. [13] fits a simplified 3-parameter quadric surface function by the matching error of 25 integer pixel positions around the best integer point. The time is saved a lot with large performance loss.

III. THE PROPOSED ALGORITHM

A. Quadric Surface Function Model Description

In HEVC motion estimation, *RDCost* is used as a measure of matching residuals:

$$RDCost = SAD(x,y) + \lambda R(\Delta x, \Delta y) \tag{1}$$

where SAD is the sum of absolute difference between the original block and the predictive block with respect to the motion vector (x,y), $R(\Delta x, \Delta y)$ is the number of bits to code the MVD and λ is the Lagrange multiplier and is introduced to balance the importance between SAD and $R(\Delta x, \Delta y)$. SATD is used to replace SAD in FME. For simplification, we use SAD instead of RD_Cost as the matching residual function.

$$SAD(x,y) = \sum_{i=0}^{W} \sum_{j=0}^{H} |O(i,j) - P(i-x, j-y)| \tag{2}$$

where W and H represent the width and height of the current encoding block, $O(x,y)$ represents the pixel value at position (i,j) in the current frame, $P(i-x, j-y)$ is the pixel value at position $(i-x, j-y)$ in the reference frame.

It is very meaningful to use function model to model SAD function. In [2], three function models of 9- parameter quadric surface function, 6-parameter quadric surface function and 5-parameter quadric surface function are given respectively.

$$S_9(x,y) = ax^2y^2 + bx^2y + cxy^2 + dx^2 + exy + fy^2 + gx + hy + i \tag{3}$$

$$S_6(x,y) = ax^2 + bxy + cy^2 + dx + ey + f \tag{4}$$

$$S_5(x,y) = ax^2 + bx + cy^2 + dy + e \tag{5}$$

where a,b,c,d,e,f,h,i are all unknown coefficients to be determined.

9-parameter model uses cubic terms to improve fitting accuracy, but its computational complexity is the highest; for 5-parameter model, its computational complexity is the lowest, but the fitting accuracy provided is the worst; 6-parameter model is the trade-off between the former two models [8]. In this paper, we use a 6-parameter quadric surface function to fit the SAD function, and use the matching error of the best integer pixel point and its adjacent positions to solve the model coefficient.

B. Solution of Coefficient and Extreme Point of 6-parameter Quadric Surface Function

Both [2] and [8] use the same method of solving the least squares solution of the overdetermined equation to obtain the coefficients of the 6-parameter quadric surface function. This method is not only complicated in calculation, but also unable to obtain good fitting accuracy. Here we present a simple method for solving coefficients by substituting integer pixels. Fig. 1 shows all the pixels to be used in Part B.

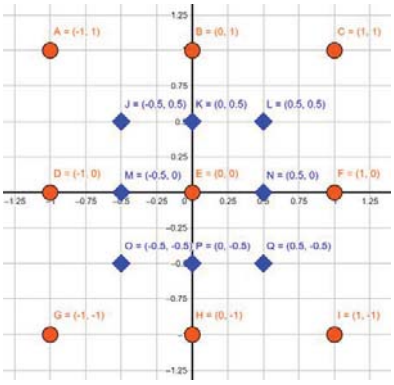

Figure 1. Integer pixel points and Sub-pixel points (Solid circulars : Integer pixel points; Solid diamond: Sub-pixel points).

1) Calculating coefficients a,c,d,e,f by substituting SAD values of integral pixel B,D,E,F,H

$$\begin{aligned} S(0,0) &= f \\ S(0,-1) &= c - e + f \\ S(0,1) &= c + e + f \\ S(-1,0) &= a - d + f \\ S(1,0) &= a + d + f \end{aligned} \tag{6}$$

By equation (6), we can get:

$$\begin{aligned} a &= \frac{S(-1,0) + S(1,0)}{2} - S(0,0) \\ c &= \frac{S(0,-1) + S(0,1)}{2} - S(0,0) \\ d &= \frac{S(1,0) - S(-1,0)}{2} \\ e &= \frac{S(0,1) - S(0,-1)}{2} \\ f &= S(0,0) \end{aligned} \tag{7}$$

2) Calculating coefficient b by substituting SAD values of integral pixel A,C,G,I:

$$\begin{aligned} S(-1,-1) &= a + b + c - d - e + f \\ S(-1,1) &= a - b + c - d + e + f \\ S(1,-1) &= a - b + c + d - e + f \\ S(1,1) &= a + b + c + d + e + f \end{aligned} \tag{8}$$

By equation (8), we can get:

$$b = \frac{S(-1,-1) + S(1,1) - S(-1,1) - S(1,-1)}{4} \tag{9}$$

3) Derivation and get the extreme point:

$$\begin{aligned} A &= \frac{\partial S}{\partial^2 x} = 2a \\ B &= \frac{\partial S}{\partial x \partial y} = b \\ C &= \frac{\partial S}{\partial^2 y} = 2c \\ H &= AC - B^2 = 4ac - b^2 \end{aligned} \tag{10}$$

$$\begin{aligned} x_{\min} &= \frac{be - 2cd}{H} \\ y_{\min} &= \frac{bd - 2ae}{H} \end{aligned} \tag{11}$$

The obtained extremum point is $(x_{\min}, y_{\min})$ quantized to 1/4 pixel accuracy [5], and the result is set as (x_{quan}, y_{quan}).

C. Classification of Extremum Points and Corresponding Search Strategies

According to the sufficient condition of extreme value of binary function in higher mathematics, we can see that:

- When H>0 and A<0, S(x,y) takes the maximum value at (x_{quan}, y_{quan});
- When H>0 and A>0, S(x,y) takes a minimum value at (x_{quan}, y_{quan});
- When H<0, (x_{quan}, y_{quan}) is the saddle point, and S(x, y) does not have an extreme value;
- When H=0, (x_{quan}, y_{quan}) does not exist, we let (x_{quan}, y_{quan})=(0,0).

We adopt different search strategies for different classification of extreme points above:

a) Maximum point: Search for three points in the same quadrant as the quadrant of the maximum point and three points in the opposite quadrant from the quadrant of the maximum point. Let A be the maximum point, then the points marked by the blue triangle in Figure 2 are the search points.

b) Minimum point: Search for 4 horizontal and vertical points centered on this point. Let B point be a minimum point, then the 4 points marked by the red triangle in Figure 2 are the search points.

c) Saddle Point: Search for three points near (0,0) direction with the point symmetrical to (0,0) as the center. Let C be the saddle point in the graph, and its symmetrical point with (0,0) is C'. All the search points are four orange points.As shown in Figure 3.

d) Extremum points do not exist: Search for the nearest four 1/4 pixels in horizontal and vertical direction centered on (0,0) points. Let O be the origin, then the 4 points marked by the purple triangle in the figure are the search points. As shown in Figure 3.

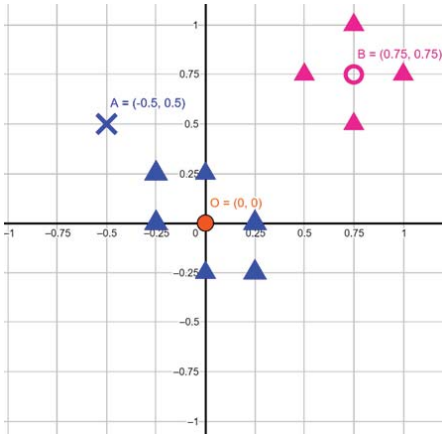

Figure 2. Illustration of search points corresponding to the maximum point and minimum point.

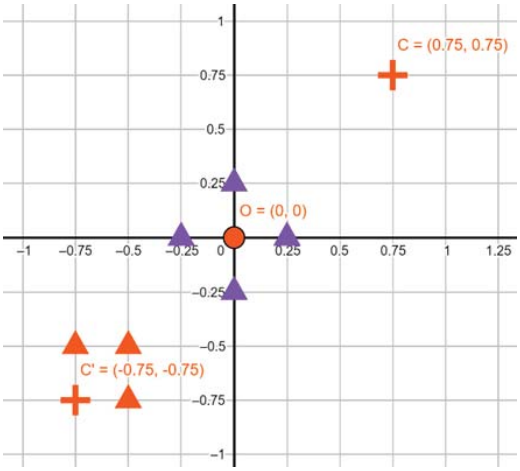

Figure 3. Illustration of search points corresponding to the saddle point and the absence of extreme point.

IV. Experimental Results

The algorithm has been implemented in the HEVC reference software HM16.0. Detailed configuration is shown in TABLE I. In order to evaluate the performance of the algorithm, hierarchical search (HS) is chosen as the anchoring algorithm. In addition, the methods in [8] and [12] are selected as representative algorithms of fast sub-pixel ME and compared with our proposed algorithms. Proposed III algorithm in [8] is selected for implementation. We select 12 sequences of five resolutions (from ClassA to ClassE) for testing.

To evaluate the coding performance, BD-Rate is measured against the anchor. Time Saving (TS) of FME process is calculated compared with the original FME in HM16.0 as follows:

$$TS = \frac{AnchorT - ProposedT}{AnchorT} \times 100\% \tag{12}$$

where $AnchorT$ and $ProposedT$ denotes FME time consumptions of anchor and proposed models, respectively.

TABLE I. Test Conditions

TEST CONDITIONS	
Max. CU size	64x64
Max. CU depth	4
QuantizationParameter(QP)	22,27,32,37
Search Range	[-64,64]
Number of frames to be encoded	100
configuration	Encoder_lowdelay_main

TABLE II. Coding Performance of Different Algorithms

Resolution	Sequences	Proposed algorithm vs. Anchor		[12] vs. Anchor		[8] vs. Anchor	
		BD-Rate	TS(%)	BD-Rate	TS(%)	BD-Rate	TS(%)
Class A	Traffic	1.40%	51.43	0.80%	22.13	1.80%	47.24
	PeopleOnStreet	1.00%	52.53	0.60%	22.97	1.60%	48.57
Class B	BasketballDrive	0.70%	55.78	0.30%	21.42	0.50%	46.6
	Cactus	0.30%	55.82	1.00%	17.66	0.20%	48.12
	BQTerrace	2.10%	54.91	1.60%	19.78	2.90%	47.54
Class C	BQMall	0.50%	58.56	0.90%	22.87	0.40%	48.28
	BasketballDrill	0.80%	57.12	0.80%	23.65	1.30%	46.32
Class D	BasketballPass	2.20%	57.77	1.60%	21.67	2.00%	49.05
	BlowingBubbles	1.80%	56.03	0.40%	21.05	1.10%	47.42
Class E	FourPeople	0.30%	56.59	0.60%	19.54	0.60%	48.03
	Johnny	1.10%	56.37	0.80%	22.52	1.40%	47.29
	KristenAndSara	1.00%	56.98	0.30%	23.59	1.30%	47.49
AVG		**1.10%**	**55.82**	**0.81%**	**21.57**	**1.26%**	**47.66**

From the experimental data in Table II, we can see that the performance loss of the proposed algorithm is only 1.1% compared with that of the reference software algorithm, and the time savings are as high as 55.82%. Compared with the proposed algorithm, the algorithm in [8] not only increases the number of search points by 4 on average, saves 47.66%

of the time and 8% less than the latter, but also results in a performance loss of 1.26% higher than the latter. The reason of performance loss in [8] is to default the extreme point to the minimum point directly, and the search is centered on the point, thus causing a large error. Compared with the algorithm in [12], the number of search points in the proposed algorithm is three points less on average, and the search process in the proposed algorithm is non-hierarchical search. On the contrary, the latter is hierarchical search., that is, three 1/2 pixels are searched first, and then four 1/4 pixels are searched. The latter algorithm involves complex threshold updating operations, so the time saving of the latter is only 21.57%, which is 0.38 times of the former, while the performance loss is 0.81%, which is 2.4 times of the former. The proposed algorithm greatly saves coding time with almost no performance loss.

V. CONCLUSION

In this paper, a fast sub-pixel motion estimation algorithm with only 1/4 precision pixel search is proposed. This algorithm uses quadric surface function to fit matching residuals, and combines with specific search strategy, which greatly accelerates the speed of sub-pixel motion estimation, and is very easy to carry out hardware parallel search. Compared with other fast algorithms, the proposed algorithm has reasonable prediction quality (only 1.10% performance loss) and great time saving (55.82% time saving).

ACKNOWLEDGMENT

This work is partially supported by National Science and Technology Major Project under contract No.2016YFC0801001 and National Engineering Laboratory of Video Technology.

REFERENCES

[1] D. Lim, Y. Choi, H. Lee and S. Chae, "A fast fractional motion estimation algorithm for high efficiency video coding," 2016 International Conference on Electronics, Information, and Communications (ICEIC), Da Nang, 2016, pp. 1-4.

[2] J. W. Suh and Jechang Jeong, "Fast sub-pixel motion estimation techniques having lower computational complexity," in IEEE Transactions on Consumer Electronics, vol. 50, no. 3, pp. 968-973, Aug. 2004.

[3] P. R. Hill, T. K. Chiew, D. R. Bull and C. N. Canagarajah, "Interpolation Free Subpixel Accuracy Motion Estimation," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 12, pp. 1519-1526, Dec. 2006.

[4] S. Dikbas, T. Arici and Y. Altunbasak, "Fast Motion Estimation With Interpolation-Free Sub-Sample Accuracy," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 7, pp. 1047-1051, July 2010.

[5] Jing-Fu Chang and Jin-Jang Leou, "A quadratic prediction based fractional-pixel motion estimation algorithm for H.264," Seventh IEEE International Symposium on Multimedia (ISM'05), Irvine, CA, 2005, pp.1074-1089.

[6] Z. Wei and Z. Xin, "A fast hierarchical 1/4-pel fractional pixel motion estimation algorithm of H.264/AVC video coding," 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), Melbourne, VIC, 2013, pp. 891-895.

[7] W. Dai, O. C. Au, S. Li, L. Sun and R. Zou, "Fast sub-pixel motion estimation with simplified modeling in HEVC," 2012 IEEE International Symposium on Circuits and Systems, Seoul, 2012, pp. 1560-1563.

[8] Y. Li, Z. Liu, X. Ji and D. Wang, "HEVC fast FME algorithm using IME RD-costs based error surface fitting scheme," 2016 Visual Communications and Image Processing (VCIP), Chengdu, 2016, pp. 1-4.

[9] W. Lin, K. Panusopone, D. M. Baylon, M. Sun, Z. Chen and H. Li, "A Fast Sub-Pixel Motion Estimation Algorithm for H.264/AVC Video Coding," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 2, pp. 237-242, Feb. 2011.

[10] K. Ng, L. Po, S. Li, K. Wong and L. Wang, "A Linear Prediction Based Fractional-Pixel Motion Estimation Algorithm," 2010 4th International Conference on Multimedia and Ubiquitous Engineering, Cebu, 2010, pp. 1-6.

[11] L. Wang and W. Tai, "A Fast Linear-Prediction Fractional-Pixel Search Algorithm," 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Crans-Montana, 2016, pp. 943-948.

[12] Jiaying Yan, Yonghong Tian, Yaowei Wang, Siwei Dong and Tiejun Huang, "A fast skip and direction adaptive search algorithm for Sub-Pixel Motion Estimation on HEVC," 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Hong Kong, 2017, pp. 157-162.

[13] E. Badry, A. Shalaby and M. S. Sayed, "Fast fractional-pixel motion estimation using Lagrangian-based error surface interpolation," 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, QC, 2017, pp. 151-155.