A NOVEL FAST TWO STEP SUB-PIXEL MOTION ESTIMATION ALGORITHM IN HEVC

Wei Dai, Oscar C. Au, Chao Pang, Lin Sun, Ruobing Zou, and Sijin Li

Department of Electronic and Computer Engineering
HKUST, Clear Water Bay, Kowloon, Hong Kong
{weidai, eeau, pcece, lsunece, zouruobing, sliae}@ust.hk

ABSTRACT

Motion estimation (ME) is one of the most time consuming parts in video coding standard. As fast integer-pixel ME algorithm becoming more and more powerful, it is important to develop fast sub-pixel ME algorithm since the computational complexity of sub-pixel ME compared to integer-pixel ME has become relatively significant. In this paper, a novel fast sub-pixel ME algorithm is proposed. This algorithm first approximates the error surface of the sub-pixel position by a second order function and predicts the minimum point by minimizing the function at half-pixel accuracy. Then another second order approximation within a smaller area which is determined by the previous step is modeled to predict the best sub-pixel position. Experimental results show that the proposed method can reduce the sub-pixel search points significantly with negligible quality degradation.

Index Terms— block-based motion estimation, fast motion estimation, sub-pixel accuracy, video coding

1. INTRODUCTION

Motion estimation (ME) is one of the key elements in video coding standard which is dedicated to achieve high coding performance by reducing temporal redundancy. Developing fast algorithms for ME has been an essential and challenging issue. Conventional ME process usually contains two stages: integer-pixel search within a search range and sub-pixel search around the best integer-pixel position. The most simple and straightforward way to find the optimal position in both two stages is the full search (FS) algorithm. It checks every possible point in the search range and select the best point based on the rate-distortion (RD) performance. Although FS algorithm can reach the global minimum, the computational complexity is usually unaffordable.

In order to accelerate the ME process, a lot of fast ME algorithms have been proposed in the literature. Typically, these algorithms can be classified into two categories: fast integer-pixel ME algorithm and fast sub-pixel ME algorithm. Many fast ME algorithms belong to the first category, such as: three step search (TSS) [1], new three step search (NTSS) [2], PMVFAST [3] and so on. On the average, integer-pixel ME can be done by examining less than 10 points. But for traditional hierarchical sub-pixel ME, 16 points are needed for 1/4-pixel accuracy. Besides, the interpolation operation is required to get the value of sub-pixel positions, which means the computational complexity of sub-pixel ME becomes comparable to that of fast integer-pixel ME. So a fast sub-pixel ME algorithm is desirable to reduce computational complexity.

One way to speed up the sub-pixel ME process is to model the error surface in the locality of the best integer-pixel point. Jing-Fu Chang *et al.* [4] modeled the error surface as an unimodal and proposed a second order function with five parameters to approximate the error surface. The best sub-pixel position is obtained by minimizing the second order function followed with some refinement process. In [5], the error surface was approximated by a function with six parameters, and the best sub-pixel position was found through a simple 4-connected gradient descent search. Salih Dikbas *et al.* [6] introduced a function with nine parameters to model the error surface, and the best sub-pixel position was located by finding the minimum position of the function.

The rest of this paper is organized as follows. In Section 2, the existing sub-pixel ME models are briefly reviewed, and a novel two step fast sub-pixel ME algorithm is proposed. To test the performance of the proposed method, it is compared with other representative methods, and experimental results are shown in Section 3. Section 4 concludes the paper.

2. PROPOSED FAST SUB-PIXEL ME ALGORITHM

Several surface models have been used to approximate the error surface in the sub-pixel ME process, including the 9-term, 6-term and 5-term error models. Mathematically, they can be written as follows:

$$f_9(x,y) = Ax^2y^2 + Bx^2y + Cxy^2 + Dx^2 + Exy + Fy^2 + Gx + Hy + I \quad (1)$$

$$f_6(x,y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F \quad (2)$$

$$f_5(x,y) = Ax^2 + Bx + Cy^2 + Dy + E, \quad (3)$$

where parameters $A, B, \ldots, I$ are estimated by fitting the RD

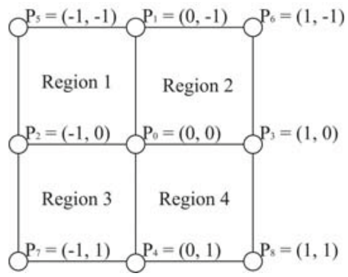

Fig. 1. Illustration of the best integer-pixel point P_0 with its 8 neighboring points and the region specification.

cost of integer-pixel points on the given models. The RD cost is defined as:

$$RDCost = SAD(m,n) + \lambda R(m,n), \tag{4}$$

where $SAD(m,n)$ is the SAD value with respect to the motion vector (m,n), $R(m,n)$ is the cost of the motion vector and λ is the *Lagrange multiplier*. The location of 9 integer-pixel points are given in Fig. 1.

In (1) (2) and (3), different number of parameters are used, and all the parameters can be obtained with only addition and bit shift operations [4]-[6]. It is obvious that the minimization of (1) and (2) is quite complicated and their minimum positions are not guaranteed to be within the $(-1,1) \times (-1,1)$ area. Therefore, some techniques are adopted to find the minimum position such as exhaustive search or gradient descent search [5]. But both of those methods need a large number of multiplications which are very time consuming. In contrast, (3) is very simple which only needs five points to model the error surface. Thus, the minimization is very easy and the location of the minimum point can be calculated as:

$$\begin{cases} x_{min} = -B/2A \\ y_{min} = -D/2C. \end{cases} \tag{5}$$

In order to get a better prediction of the optimal sub-pixel position, a fast sub-pixel ME algorithm based on six integer-pixel points is proposed. First, the error surface is modeled by a second order function with five parameters. Then, the 6^{th} point is selected based on the location of the minimum position in the first step and another five-parameter function is adopted to estimate the error surface within a smaller region. A new predicted position can be derived from the model in the second step. At last, the final motion vector is set as the point with minimum RD cost among the best integer-pixel point, best half-pixel point and best quarter-pixel point. The flowchart of the algorithm is illustrated in Fig. 2 and details are discussed below.

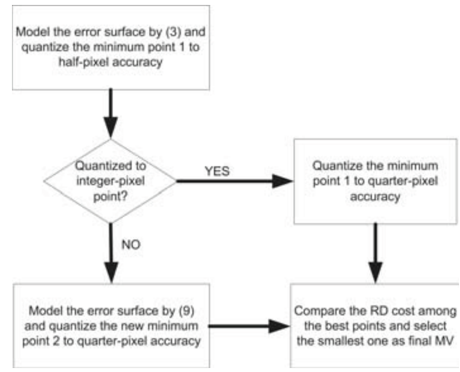

Fig. 2. Flowchart of the proposed algorithm.

2.1. First Approximation with Five Integer-Pixel Points

The first step of the algorithm is the same as [4] which uses five points P_0, P_1, P_2, P_3, P_4 to model the error surface:

$$\begin{cases} f_5'(P_0) = E_1 \\ f_5'(P_2) = A_1 - B_1 + E_1 \\ f_5'(P_3) = A_1 + B_1 + E_1 \\ f_5'(P_1) = C_1 - D_1 + E_1 \\ f_5'(P_4) = C_1 + D_1 + E_1, \end{cases} \tag{6}$$

where A_1, B_1, C_1, D_1, E_1 can be solved with only addition and bit shift operations [4]. The first predicted best sub-pixel position can be obtained by:

$$\begin{cases} x_1 = -B_1/2A_1 \\ y_1 = -D_1/2C_1. \end{cases} \tag{7}$$

Let $Quantize(x,y)$ be the operation of quantizing x to the nearest sub-pixel position with y accuracy ($y = 2$ means half-pixel accuracy, $y = 4$ means quarter-pixel accuracy). First, (x_1, y_1) is quantized to half-pixel accuracy:

$$\begin{cases} x_1^Q = Quantize(x_1, 2) \\ y_1^Q = Quantize(y_1, 2). \end{cases} \tag{8}$$

If (x_1, y_1) is quantized to the integer-pixel position, another quantization is applied to quantize (x_1, y_1) to quarter-pixel accuracy. Then the point with smaller RD cost between quantized point and best integer-pixel point will be regarded as the final motion vector and the algorithm will terminate immediately. Otherwise, following steps will be taken.

2.2. Second Approximation with Additional Integer-Pixel Point and Half-Pixel Point

In the second step of the proposed method, one more integer-pixel point is selected, and a second order approximation is performed within a smaller region. The 6^{th} integer-pixel

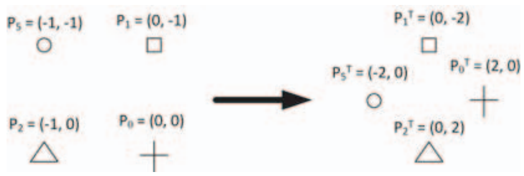

Fig. 3. Point mapping correspondences of region 1.

point is chosen based on the first predicted position (x_1, y_1). For example, if (x_1, y_1) falls in the region 1 in Fig. 1, it is very possible that the best sub-pixel position also falls in region 1. Then P_5 will be chosen as the 6^{th} point. The integer-pixel points P_0, P_1, P_2, P_5 and half-pixel point (x_1^Q, y_1^Q) are used to model the error surface in region 1. With additional information of P_5 and (x_1^Q, y_1^Q), a better approximation within region 1 will be achieved. Following derivations are all based on the assumption that (x_1, y_1) falls in region 1.

Because there are only five points available, only a function with five parameters can be determined:

$$\begin{aligned} f_5''(x,y) =& A_2(x+y)^2 + B_2(x+y) \\ &+ C_2(x-y)^2 + D_2(x-y) + E_2. \end{aligned} \tag{9}$$

Note that the contour of (9) is an ellipse whose axes align well with rotated x and y axes by $45°$. For simplicity, we rotate the $x-y$ plane and map the four integer-pixel positions to new positions which are shown in Fig. 3.

Mathematically, this transformation can be written as:

$$\begin{cases} x^T &= 2x + 2y + 2 \\ y^T &= -2x + 2y. \end{cases} \tag{10}$$

For each region, there are three possible best half-pixel points. In region 1, they are: $(-0.5, -0.5)$, $(0, -0.5)$ and $(-0.5, 0)$. These points will be mapped to the corresponding points P_{cor}: $(0, 0)$, $(1, -1)$ and $(1, 1)$ in the transformed space. The model (9) after transformation becomes:

$$f_5''^T(x^T, y^T) = A_2^T x^{T2} + B_2^T x^T + C_2^T y^{T2} + D_2^T y^T + E_2^T. \tag{11}$$

Substituting (x^T, y^T) with five points $P_0^T, P_1^T, P_2^T, P_5^T$ and P_{cor}:

$$\begin{cases} f_5''^T(P_{cor}) &= E_2^T, \quad \text{for} \quad P_{cor} = (0,0) \\ f_5''^T(P_{cor}) &= A_2^T + P_{cor}^X B_2^T + C_2^T + P_{cor}^Y D_2^T + E_2^T, \\ & \quad \text{for} \quad P_{cor} = (P_{cor}^X, P_{cor}^Y) \neq (0,0) \\ f_5''^T(P_5^T) &= 4A_2^T - 2B_2^T + E_2^T \\ f_5''^T(P_0^T) &= 4A_2^T + 2B_2^T + E_2^T \\ f_5''^T(P_1^T) &= 4C_2^T - 2D_2^T + E_2^T \\ f_5''^T(P_2^T) &= 4C_2^T + 2D_2^T + E_2^T. \end{cases} \tag{12}$$

These five coefficients can be solved by only addition and bits shift operations. Take $P_{cor} = (1, 1)$ as an example:

$$\begin{cases} B_2^T = & (f_5''^T(P_0) - f_5''^T(P_5))/4 \\ D_2^T = & (f_5''^T(P_2) - f_5''^T(P_1))/4 \\ E_2^T = & 2(f_5''^T(P_{cor}) - B_2^T - D_2^T) - (f_5''^T(P_5) + f_5''^T(P_0) \\ & +f_5''^T(P_1) + f_5''^T(P_2))/4 \\ A_2^T = & (f_5''^T(P_5) + f_5''^T(P_0))/8 - E_2^T/4 \\ C_2^T = & (f_5''^T(P_1) + f_5''^T(P_2))/8 - E_2^T/4. \end{cases} \tag{13}$$

The minimum position of (11) is:

$$\begin{cases} x_2^T &= -B_2^T/2A_2^T \\ y_2^T &= -D_2^T/2C_2^T. \end{cases} \tag{14}$$

The corresponding point in the original space can be obtained by the inverse transformation of (10):

$$\begin{cases} x_2 = (x_2^T - y_2^T - 2)/4 \\ y_2 = (x_2^T + y_2^T - 2)/4. \end{cases} \tag{15}$$

Also, quantization is applied to (x_2, y_2) in order to get the best quarter-pixel point:

$$\begin{cases} x_2^Q = Quantize(x_2, 4) \\ y_2^Q = Quantize(y_2, 4). \end{cases} \tag{16}$$

2.3. Compare and Get the Optimal Position

Finally, comparison is taken among the best integer-pixel position, best half-pixel position and best quarter-pixel position. The one with minimum RD cost will be chosen as the final motion vector.

3. EXPERIMENTAL RESULTS

The proposed algorithm has been implemented on the latest HEVC reference software HM3.0, and the encoder is set to be lowdelay-loco mode. To evaluate the performance of proposed algorithm, hierarchical search (HS) is chosen to be the anchor algorithm. Moreover, method in [4] (FPME) is selected as the representative algorithm of fast sub-pixel ME. The reason to choose FPME is that FPME and the proposed method are quite similar. The major difference is: FPME uses only 4 neighboring integer-pixel points and performs the refinement search iteratively after the prediction. However, our method uses 5 neighboring integer-pixel points and do the approximation twice without further refinement.

Four sequences of various resolution sizes are encoded with four QP values. ΔPSNR, total encoding time and average sub-pixel search points per partition (SP/PT) are measured and compared. Experimental results in Table. 1 show

Table 1. Comparison of the proposed method with FPME and Hierarchical Search method.

Sequence Name	Method	ΔPSNR (dB)	Total Encoding Time (s)	Reduced Time	SP/PT*
ParkScene (1080P)	Hierarchical Search	0	14027.95	0	16.00
	FPME	0	10147.28	27.66%	6.36
	Proposed	-0.02	7540.52	46.25%	0.81
Vidyo1 (720P)	Hierarchical Search	0	15267.41	0	16.00
	FPME	0	8348.19	45.32%	5.78
	Proposed	0	5867.93	61.57%	0.31
BasketballPass (WQVGA)	Hierarchical Search	0	2328.58	0	16.00
	FPME	-0.01	1020.94	56.16%	6.65
	Proposed	-0.04	760.13	67.36%	0.64
BasketballDrill (WQVGA)	Hierarchical Search	0	4726.58	0	16.00
	FPME	-0.01	3829.03	18.99%	6.36
	Proposed	-0.04	2784.83	41.08%	0.65

* Average sub-pixel search points per partition

that proposed method achieves significant encoding time reduction over the other two methods at the expense of negligible coding performance degradation. Moreover, SP/PT shows that proposed algorithm can greatly reduce the number of sub-pixel search points compared with conventional fast sub-pixel ME algorithms. Less than 1 point is searched by proposed algorithm while usually more than 6 points is needed for FPME. This means that our algorithm can predict the best sub-pixel position successfully with a high probability.

4. CONCLUSION

In this paper, a novel two step fast sub-pixel ME algorithm is presented. This algorithm makes use of 6 integer-pixel points to estimate the best sub-pixel position. The error surface of the sub-pixel position is first modeled within a large area and the best half-pixel point is obtained. Then, with the information of the best half-pixel point and one additional integer-pixel point, the error surface within a small region is further approximated and the best quarter-pixel point is obtained by minimizing the function. The one with minimum RD cost among best integer-pixel point, best half-pixel point and best quarter-pixel point will be chosen as the final motion vector. Experimental results show that the proposed algorithm greatly reduces the number of average search points while maintaining the coding performance of the conventional hierarchical search algorithm.

5. ACKNOWLEDGEMENT

This work has been supported in part by the Research Grants Council (GRF Project no. 610210) and the Hong Kong Applied Science and Technology Research Institute Project (ART/093CP).

6. REFERENCES

[1] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, ”Motion compensated interframe coding for video conferencing,” in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, Nov. 29CDec. 3 1981, pp. G5.3.1CG5.3.5.

[2] R. Li, B. Zeng, and M. Liou, ”A new three-step search algorithm for block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 438C442, Aug. 1994.

[3] A. M. Tourapis, O. C. Au, and M. L. Liou, ”Predictive Motion Vector Field Adaptive Search Technique (PMVFAST),” in *ISO/IEC JTC1/SC29/WG11 MPEG2000, Noordwijkerhout*, NL, March 2000.

[4] Jing-Fu Chang and Jin-Jang Leou, ”A quadratic prediction based fractional-pixel motion estimation algorithm for H.264,” in *Multimedia, Seventh IEEE International Symposium on*, 2005, pp. 8 pp.

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

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