

# A Vertically Symmetrical Linear Model-Based Fractional-Pixel Motion Estimation Algorithm for H.264/AVC Encoder

Chang-Uk Jeong<sup>†</sup> and Hiroshi Watanabe<sup>††</sup>

Integer-pixel motion estimation (IME) runtime in the H.264 encoder takes up over 50% of the total runtime. In which, many IME algorithms have been developed up to now. On the other hand, fractional-pixel motion estimation (FME) part also occupies about 40% of the total motion estimation runtime. After performing the FME process, 1-3 (dB) PSNR improvement can be achieved. The FME part, therefore, has been becoming an important research topic for video coding applications. Conventional FME algorithms such as the full fractional-pixel search require some search points as well as high memory usage due to the interpolation process. In this paper, a vertically symmetrical linear model-based FME algorithm including a grouping strategy is proposed to reduce the computational complexity and improve the performances without regard to the interpolation process. The simulation results show the proposed algorithm can produce slightly better performances compared with the existing prediction method.

## 1. Introduction

H.264/AVC [1], which is a recently published video coding standard, was developed and has been updated by Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. The H.264 encoder, similar to that of the existing video standards such as ISO/IEC MPEG-1, MPEG-2, MPEG-4, ITU-T H.261, H.263 [2], is composed of a temporal model, a spatial model, and an entropy encoder. The runtime of the temporal model, which is named motion estimation (ME) part to eliminate the temporal redundancy between adjacent frames, occupies more than 80% of the whole runtime of the H.264 encoder [3]. Furthermore, due to various advanced encoding technologies such as variable block-size ME, multiple reference frames, and quarter-pixel ME and compensation, the computational complexity was increased dramatically. Although the advanced encoding technologies have significantly improved the quality performance of the H.264 standard, the high complexity imposes restrictions on real time video applications, especially for mobile devices.

The motion estimator in the H.264 encoder conducts the integer-pixel ME (IME), followed by the fractional-pixel ME (FME). Particularly, the runtime of IME is about half of the total encoding time [3]. In which, many fast IME algorithms including the diamond search [4], the hexagon-based search [5], the unsymmetrical-cross multi-hexagon-grid search (UMHexagonS) [6] have been developed up to now to reduce integer-pixel search points. The proposed IME algorithms have different search strategies to satisfy the accuracy of ME and the search speed, respectively.

On the other hand, the FME part also takes up over 40% of the total ME runtime [3]. After performing the FME process, about 1-3 (dB) PSNR improvement can be expected. The FME part, therefore, has been recently regarded as an important research topic in H.264/AVC. The conventional full fractional-pixel search always requires a fixed number of search points and consumes a large amount of memory due to the interpolation process for fractional-pixel search. The center biased fractional-pixel search [6], the prediction-based directional fractional-pixel search [7], the novel fractional-pixel search using motion prediction and fast search pattern [8], and so on are attempts to decrease fractional-pixel search points. Most of FME algorithms, however, perform the interpolation process before finding the best fractional-pixel MV. Thus, in this paper, a linear model-based fractional-pixel motion estimation algorithm for H.264/AVC is proposed to efficiently reduce the computational complexity and maintain the quality performance without the interpolation process.

---

<sup>†</sup> Graduate School of Global Information and Telecommunication Studies, Waseda University

<sup>††</sup> Graduate School of Global Information and Telecommunication Studies, Waseda University

## 2. Previous Prediction-Based FME Algorithm

The quadratic prediction-based FME (QPFPS) [9] was proposed to reduce the computational load at quarter-pixel motion vector (MV) resolution. A degenerate quadratic prediction function used in this algorithm affects the determination of the best quantized predicted MV at quarter-pixel MV resolution, which is shown below.

$$F(x, y) = Ax^2 + Bx + Cy^2 + Dy + E \quad (1)$$

As described in Equation 2, the differential operation can be executed on the quadratic prediction function with respect to  $x$  and  $y$  to assume the minimum error cost  $F(x, y)$ , which can be obtained by being substituted with  $x_p$  and  $y_p$ .

$$\begin{cases} 2Ax_p + B = 0 \\ 2Cy_p + D = 0 \end{cases} \quad (2)$$

In the final step, the predicted position is located at the center of the small diamond search pattern, and then the small diamond search algorithm will be carried out to refine the best fractional-pixel search point. In this step case, however, the interpolation process may be needed to create a fractional-pixel search area.

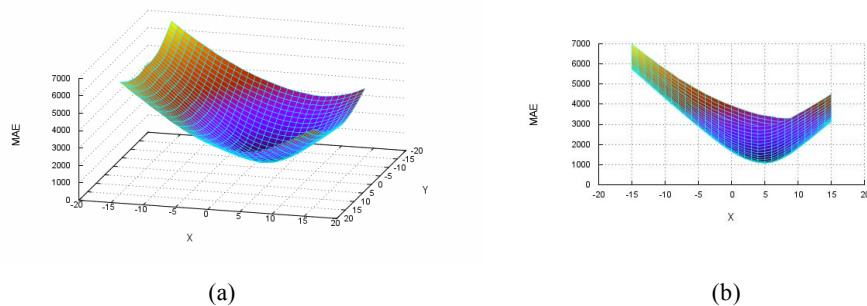


Figure 1 (a) Error surface of FME at 1/16-pixel MV resolution, (b) Another angle of (a).

## 3. Error Surface of FME

We have simulated and analyzed the error surface of FME directly. Figure 1 (a) shows the FME error surface for CCIR601 sequence “Garden” at 1/16-pixel MV resolution. Unlike that of IME, the FME error surface is clearly unimodal because the sub-pixels were generated by the bilinear interpolation using the existing integer-pixels. Actually, most of FME algorithms are influenced by a kind of interpolation. Therefore, the approach to the FME part should be different from IME. Particularly, we take notice of Figure 1 (b). The shape looks like parabolic definitely. It means we can apply parabolic models including some quadratic functions for FME. The graph also shows it has vertical symmetry with respect to  $x=5$ .

Furthermore, the vertically symmetrical characteristic of the FME error surface has been observed closely, as illustrated in Figure 2. 1000 FME error surfaces for CCIR601 sequence “Stefan” are gathered sequentially by performing full fractional-pixel search at 1/16-pixel MV resolution and their best MV is (4, 2). The shape average for 10, 100, and 1000 FME error surfaces is computed, respectively, and then only the three minimum MAE values corresponding to the  $x$ -coordinates -15, 4, and 15 for each error surface are extracted and used to analyze the general characteristics. As a result, Figure 2 shows that the average FME error surfaces simplified represent almost obviously vertical symmetry with respect to  $x=4$ .

From understanding Figure 1 and 2, a linear model graph can be drawn, as shown in Figure 3. Line  $F(x)$  adjoins point  $(-1, F(-1))$  and  $(0, F(0))$ . Here, we can introduce symmetry assumption for  $F(x)$ . Line  $G(x)$ , which has the negative slope value of  $F(x)$ , is passing point  $(1, G(1))$ . That is, the basic principle of the proposed FME algorithm is to find the location where the two lines intersect.

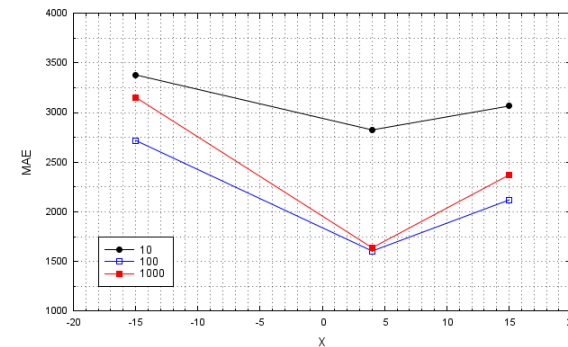


Figure 2 Average FME error surfaces simplified for 10, 100, and 1000 error surfaces.

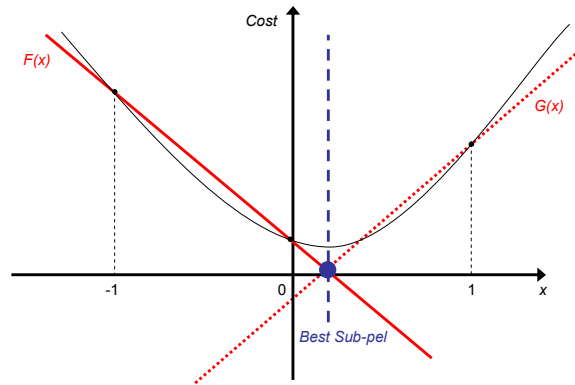


Figure 3 The basic concept of the proposed FME algorithm.

#### 4. Proposed FME Algorithm

The proposed linear model-based fractional-pixel motion estimation algorithm reuses the matching error cost of the nine integer-pixel search points as shown in Figure 4. Basically, the linear prediction function applied in the proposed algorithm is described as follows.

$$\begin{cases} F(x) = (-H_1 + C)x + C \\ G(x) = (-H_1 + C)(1-x) + H_2 \end{cases} \quad (3)$$

In function  $F(x)$ , for example,  $-H_1 + C$  is the slope value of the linear equation for a group  $(H_1, C, H_2)$ . Function  $G(x)$  represents a symmetrical linear function with the negative slope value corresponding to  $F(x)$ . The intersection point  $x$  between  $F(x)$  and  $G(x)$  can be shown below.

$$x = \frac{(H_2 - H_1)}{2(C - H_1)} \quad (4)$$

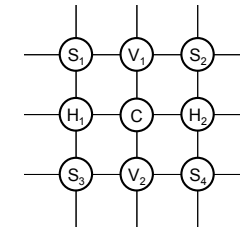


Figure 4 The nine integer-pixel search points reused in the proposed FME algorithm.

In addition to the linear-based prediction, we propose a grouping strategy to enhance the accuracy. The nine integer-pixel search points are grouped according to the close proximity. As listed in Table 1, they are divided into three groups, the horizontal, the vertical, and the both horizontal and vertical, of three points. For instance,  $(S_1, V_1, S_2)$ ,  $(V_1, C, V_2)$ , and  $(S_1, C, S_4)$  are the respective members of the above mentioned groups. Each group is used to determine the best  $x$ ,  $y$ ,  $x$  or  $y$  position, respectively. All the groups are calculated to predict the matching error cost of FME, and then the  $x$ - and  $y$ -coordinate of the location which produces the minimum prediction cost are regarded and selected as the best fractional-pixel MV. Before transmitting the best fractional-pixel MV, the quantization step suitable for quarter- or higher pixel MV resolution is required. We use the quantization process explained in [9] at quarter-pixel MV resolution. As shown in Table 2, the best fractional-pixel MV can also be quantized at 1/8-pixel MV resolution.

Table 1 Horizontal, vertical, and both groups using the nine integer-pixel search points.

Group	Member	Coordinate
Horizontal	$(S_1, V_1, S_2)$ , $(S_1, V_1, H_2)$ , $(H_1, V_1, S_2)$ , $(S_1, C, S_2)$ , $(H_1, C, H_2)$ , $(H_1, C, S_2)$ , $(H_1, C, S_4)$ , $(S_1, C, H_2)$ , $(S_3, C, H_2)$ , $(H_1, V_1, H_2)$ , $(H_1, V_2, H_2)$ , $(S_3, V_2, S_4)$ , $(S_3, V_2, H_2)$ , $(H_1, V_2, S_4)$ , $(S_3, C, S_4)$	$x$
Vertical	$(V_1, C, V_2)$ , $(V_1, C, S_3)$ , $(V_1, C, S_4)$ , $(S_1, H_1, S_3)$ , $(S_1, H_1, V_2)$ , $(S_2, H_2, S_4)$ , $(S_2, H_2, V_2)$ , $(V_1, H_1, S_3)$ , $(V_1, H_1, V_2)$ , $(V_1, H_2, S_4)$ , $(V_1, H_2, V_2)$ , $(S_1, C, V_2)$ , $(S_1, C, S_3)$ , $(S_2, C, V_2)$ , $(S_2, C, S_4)$	$y$
Horizontal and vertical	$(S_1, C, S_4)$ , $(S_2, C, S_3)$	$x$ or $y$

Table 2 Quantization operation for FME at 1/8-pixel MV resolution

Integer-converted $x$ or $y$	Quantized $x$ or $y$	Predicted $p = x$ or $y$
-7	-0.875	$p < -0.8125$
-6	-0.750	$-0.8125 \leq p < -0.6875$
-5	-0.625	$-0.6875 \leq p < -0.5625$
-4	-0.500	$-0.5625 \leq p < -0.4375$
-3	-0.375	$-0.4375 \leq p < -0.3125$
-2	-0.250	$-0.3125 \leq p < -0.1875$
-1	-0.125	$-0.1875 \leq p < -0.0625$
0	0.000	$-0.0625 \leq p \leq 0.0625$
1	0.125	$0.0625 < p \leq 0.1875$
2	0.250	$0.1875 < p \leq 0.3125$
3	0.375	$0.3125 < p \leq 0.4375$
4	0.500	$0.4375 < p \leq 0.5625$
5	0.625	$0.5625 < p \leq 0.6875$
6	0.750	$0.6875 < p \leq 0.8125$
7	0.875	$p > 0.8125$

## 5. Experimental Results

The proposed algorithm has been evaluated based on the H.264/AVC reference software JM12.4 [10] and KTA2.7 [11]. In JM12.4, the simulation is conducted with the default settings of search range=16, quantization parameter=28, rate distortion optimization=off, entropy coding= CAVLC, and baseline profile. In KTA2.7, which includes advanced coding efficiency tools such as adaptive interpolation filters, ME with 1/8-pixel MV resolution, motion vector competition, adaptive quantization matrix selection, and so on, the simulation is conducted with the default settings of search range=16, quantization parameter=40, rate distortion optimization=on, entropy coding=CABAC, and main profile. After carrying out UMHexagonS for IME, the FME module is applied. Quarter- and 1/8-pixel ME are performed on JM12.4 and KTA2.7, respectively. The four sequences - QCIF sequence “Claire” and CIF sequences “Football”, “Mobile”, and “News” are used for quarter-pixel ME. The other four sequences - QCIF sequence “Salesman” and CIF sequences “Husky” and “Stefan”, and HDTV720p sequence “City” are used for 1/8-pixel ME. They include a variety of motion contents and activities, respectively. 100 frames are encoded for each sequence. We compare our algorithm with QPFPS algorithm. To assess only the prediction of them without the interpolation process, the small diamond search process used in QPFPS is skipped.

From Tables 3 and 5, we can see that the PSNR performance of the proposed algorithm shows a little better improvement compared with the quadratic prediction-based method. As shown in Tables 4 and 6, the bit rate is also lower than the quadratics’. Particularly, as described in Figure 5, the rate distortion curve for CIF sequence “Football” shows that our proposed algorithm is efficient slightly better compared with the existing prediction method. In this test, note that the ME time is not considered because all of them do not use any search points, which computational complexity is much higher than that of prediction-based algorithms, at fractional-pixel MV resolution.

Table 3  $\Delta$ PSNR (dB) performance comparison at quarter-pixel MV resolution.

FME	Sequence			
	Claire (QCIF)	Football (CIF)	Mobile (CIF)	News (CIF)
QPFPS	39.631	36.171	33.792	37.981
Proposed	39.649	36.175	33.794	37.982

Table 4  $\Delta$ Bit rate (bps) performance comparison at quarter-pixel MV resolution.

FME	Sequence			
	Claire (QCIF)	Football (CIF)	Mobile (CIF)	News (CIF)
QPFPS	34802	1599065	2062130	232426
Proposed	34627	1579102	2059097	231466

Table 5  $\Delta$ PSNR (dB) performance comparison at 1/8-pixel MV resolution.

FME	Sequence			
	Salesman (QCIF)	Husky (CIF)	Stefan (CIF)	City (HDTV)
QPFPS	27.455	23.033	26.730	28.244
Proposed	27.487	23.038	26.748	28.247

Table 6  $\Delta$ Bit rate (bps) performance comparison at 1/8-pixel MV resolution.

FME	Sequence			
	Salesman (QCIF)	Husky (CIF)	Stefan (CIF)	City (HDTV)
QPFPS	10915	855072	181999	274032
Proposed	10927	854767	181714	273811

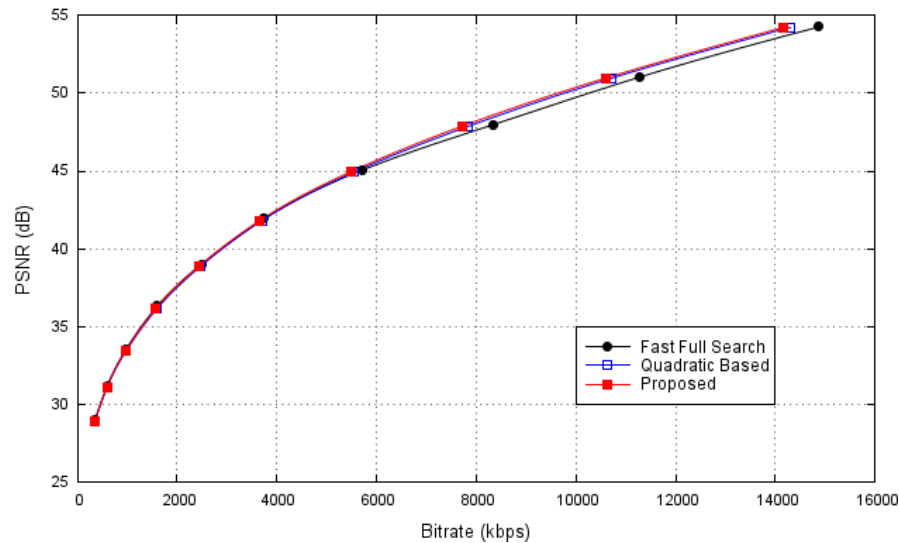


Figure 5 Rate distortion curves for CIF sequence "Football".

## 6. Conclusion

In this paper, we proposed a vertically symmetrical linear model-based fractional-pixel motion estimation algorithm which is based on a simple linear equation reusing the matching error cost of integer-pixel points and expanded by a grouping strategy. Particularly, the proposed FME algorithm was designed to implement in the H.264 encoder without respect to the interpolation process. The results of the experiment show that our algorithm can bring about a slightly better performance in terms of PSNR compared with a quadratic prediction-based algorithm, whereas the bit rate is decreased.

## References

- 1) "Draft ITU-T Rec. and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264-ISO/IEC 14 496-10 AVC)," Joint Video Team (JVT) of ITU-T and ISO/IEC JTC1, Geneva, JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-G050r1 (2003).
- 2) K. R. Rao and J. J. Hwang, Techniques and Standards for Image, Video and Audio Coding. Englewood Cliffs, NJ: Prentice Hall (1996).
- 3) T. Koga, K. Inuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proc. Nat. Telecommun. Conf., New Orleans, LA, USA, pp. G5.3.1-G5.3.5 (1981).
- 4) S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," IEEE Trans. Image Process., vol. 9, no. 2, pp. 287-290, (2000).
- 5) C. Zhu, X. Lin, and L. P. Chau, "Hexagon-based search pattern for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 12, pp. 349-355, (2002).
- 6) Z. Chen, P. Zhou, and Y. He, "Fast integer pel and fractional pel motion estimation for JVT," JVT-F017, 6th meeting: Awaji, Japan, 5-13 (2002).
- 7) L. Yang, K. Yu, J. Li, S. Li, "Prediction-based directional fractional pixel motion estimation for H.264 video coding," in Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing., vol. 2, pp. II/901-II/904 (2005).
- 8) J. S. Kim, K. W. Lee, M. H. Sunwoo, "Novel fractional pixel motion estimation algorithm using motion prediction and fast search pattern," in Proc. of IEEE Int. Conf. on Multimedia and Expo., pp. 821-824, (2008).
- 9) J. F. Chang and J. J. Leou, "A quadratic prediction based fractional-pixel motion estimation algorithm for H.264," in Proc. Seventh IEEE Int. Symp. on Multimedia., pp. 491-498, (2005).
- 10) JVT H.264/AVC Reference Software Joint Model (JM), Download: <http://iphome.hhi.de/suehring/tml/>
- 11) VCEG H.264/AVC Reference Software Key Technical Area (KTA), Download: <http://iphome.hhi.de/suehring/tml/>