Interpolation-Free Fractional Pixel Motion Estimation Based on Data Trend Approximation

Chang-Uk Jeong and Hiroshi Watanabe
Graduate School of Global Information and Telecommunication Studies
Waseda University
1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo 169-0051, Japan

Abstract—**High computational complexity of a fractional pixel motion estimation (FME) module can in no way be negligible although the module improves visual quality after the integer pixel motion estimation (IME) process. Most of the conventional FME methods must include the interpolation procedure to form fractional pixel search points from the information of the integer pixel matching error costs. The interpolation, however, requires a considerable memory usage and certain amount of processing time. In this paper, therefore, we propose interpolation-free fractional pixel motion estimation techniques by using the data trend approximation and the error cost scaling. The results of our simulation show that the proposed methods produce similar or better performances compared with the existing FME methods, whereas they do not need any additional search points.**

I. Introduction

Nowadays, the mobile network technologies including 3G and 4G wireless standards have been showing rapid progress. Nevertheless, the transmission of a large amount of multimedia data such as high definition video content increases traffic dramatically on both wireless and wire-based networks. Video compression standards, such as ISO/IEC MPEG-1, MPEG-2, MPEG-4, ITU-T H.261, H.263 [1], and H.264/AVC [2] by ISO/IEC MPEG and ITU-T VCEG, keep evolving accordingly. H.264/AVC is a state of the art video compression standard for encoding and decoding video data using many advanced features. Although the new features allow it to encode video data more effectively than the conventional ways, increased computational complexity requires sufficient CPU power to perform real-time video encoding, especially in mobile applications. Recently, high efficiency video coding (HEVC) standard has been jointly developed by ISO/IEC MPEG and ITU-T VCEG. HEVC is being drafted to achieve much higher video coding efficiency compared to H.264/AVC by reducing bitrates by half with similar image quality, but is expected to be increased in computational complexity.

The video encoder is generally divided into three units, a temporal redundancy eliminator, a spatial redundancy eliminator, and an entropy encoder. As motion estimation (ME) is core to the temporal model, it occupies more than 89% of the total encoding time [3]. Thus, many researches on fast ME have been conducted to reduce the high computational complexity of the ME module.

Motion estimation methods can be classified into pixel-recursive algorithm and block matching algorithm according to the elementary unit, i.e. pixel or block, in motion estimation. Block matching algorithm, in particular, has been adopted in many video compression standards due to low computational cost with robustness to errors. The diamond search (DS) [4], the hexagon-based search (HEXBS) [5], the efficient three-step search (E3SS) [6], and the unsymmetrical-cross multi-hexagon-grid search (UMHexagonS) [7] are fast block-based ME algorithms, which were developed to effectively reduce the computational complexity of the integer pixel ME (IME) module.

Fractional pixel motion estimation (FME) improves the image quality visibly, whereas it needs the expense of higher computational complexity. The runtime of the FME module is over 37% of the total encoding time [3]. The conventional full fractional pixel search (FFPS) in H.264/AVC is wasteful and inefficient because a fixed number of search points are used all the time. In addition, the interpolation process must be performed to create fractional pixel search area, which requires a large amount of memory and high computational complexity. The center biased fractional pixel search (CBFPS) [7], the fast sub-pixel motion estimation techniques having lower computational complexity [8], the quadratic prediction-based FME (QPFPS) [9], and the fast motion estimation with interpolation-free sub-sample accuracy [10] have been developed to reduce the computational cost of the FME module. In this paper, interpolation-free fractional pixel motion estimation techniques based on the data trend approximation and the error cost scaling are proposed. Our proposed techniques focus on performing FME without using the interpolation process. In the experimental results, the performances of our algorithm will be evaluated in terms of PSNR and bitrates.

II. Mathematical Models

A. Parabolic Models to Calculate Matching Errors

$$F(x,y)=c_1x^2y^2+c_2x^2y+c_3xy^2+c_4xy+c_5x^2+c_6x+c_7y^2+c_8y+c_9 \quad (1)$$

$$F(x,y)=c_1x^2+c_2xy+c_3y^2+c_4x+c_5y+c_6 \quad (2)$$

$$F(x,y)=c_1x^2+c_2x+c_3y^2+c_4y+c_5 \quad (3)$$

As described in Equations 1 [8][10], 2 [8], and 3 [8][9], the three equations are used to model the matching error $F(x,y)$ at fractional pixel resolution, respectively. Particularly, the two parabolic models in Equations 1 and 2 definitely require the nine matching error costs at integer pixel resolution, as shown in Figure 1, to determine the nine coefficients c_1-c_9 and the six coefficients c_1-c_6, respectively. In other words, if all the nine matching error costs are not provided by the IME process, the estimation can not be guaranteed. To fix this problem, each of [8] and [10] uses the full search (FS) and the eight neighbor search (ENS) [10] for IME, however, FS is very wasteful in computational complexity and the rectangular search pattern consisting of eight IME search points (matching errors) used in ENS is inefficient compared with the small diamond search pattern (SDSP) consisting of five IME search points, as shown in Figure 1. In addition, most of powerful IME algorithms including UMHexagonS in H.264/AVC terminate the search process using SDSP in the final search step. Accordingly, if we should use the FME module based on Equations 1 and 2, it will cause an increase in the total encoding time with some modifications of the fast IME module. The two models in Equations 1 and 2 are also unstable on the extension to the quarter pixel or less than one FME because some matching errors at the external half pixel locations can not be directly estimated by computing the nine coefficients.

Contrary to the former two mathematical models, the parabolic model in Equation 3 can be applied without any difficulty under the state-of-the-art IME techniques because it needs only the five IME matching errors, as described in Figure 1, to determine the five coefficients c_1-c_5. The parabolic model in Equation 3 can also be decomposed into two one-dimensional parabolic models, which approximate the horizontal and vertical matching errors separately, as described in the following equation:

$$F(p) = c_1 p^2 + c_2 p + c_3 \quad (p = x \text{ or } y) \tag{4}$$

As presented in [9], the minimum matching error cost $F(p)$ can easily be found by differentiating $F(p)$ with respect to x and y, respectively. When $dF/dp(p)=0$, the x and y coordinates are regarded as the best prediction position (x_b,y_b).

$$F'(p) = 2c_1 p + c_2 = 0, \quad p_b = \frac{-c_2}{2c_1} \tag{5}$$

The 1-D parabolic model uses only the three IME matching error costs, which correspond to (H_1,C,H_2) or (V_1,C,V_2), to compute the three coefficients c_1-c_3 as derived in Equation 6.

$$\begin{aligned} &c_1 = (I_1 + I_2 - 2C)/2 \quad (I_1, I_2 = H_1, H_2 \text{ or } V_1, V_2) \\ &c_2 = (-I_1 + I_2)/2 \\ &c_3 = C \end{aligned} \tag{6}$$

The best predicted x_b and y_b coordinates are estimated independently of each other. Here, however, the 1-D parabolic model based prediction has a serious fault as shown below:

$$C > I_2, \quad if\ p_b = \frac{-c_2}{2c_1} > \frac{1}{2} \tag{7}$$

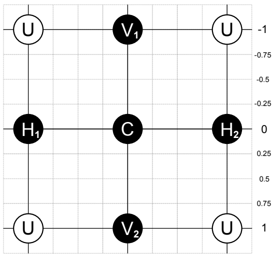
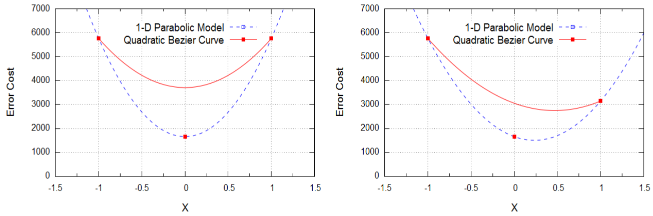


Fig. 1. The five known integer pixel search points (H_1,H_2,C,V_1,V_2) and the four unknown IME search points (U).

Equation 7 is an abnormal case and there is a contradiction because the matching error C at the local location (0,0) always returns the lowest error cost in IME. That is, the 1-D parabolic model based prediction alone is absolutely impossible to find the best prediction position at the location greater than 0.5 or less than -0.5. It will have a serious impact on the FME at quarter pixel resolution or less than one. Moreover, as H_1 or H_2 is equal to zero, the 1-D parabolic graph tends to be concave down rather than concave up. As an alternative solution, to enhance the reconstruction PSNR performance, QPFPS proposed by [9] adopted an interpolation based refinement procedure in its final search step, whereas it leads to an increase in computational complexity.

B. Surface Modeling to Approximate Data Trend

The above-mentioned parabolic models can be extended to higher order polynomial surface models in order to achieve more accurate prediction. However, higher order polynomial functions require more computational complexity and more IME matching error costs and often result in unwanted undulations. Thus, we have considered different forms of error surface modeling except the above-mentioned parabolic models. Free-form surface modeling is used to describe the skin of a 3-D geometric element. They do not have rigid radial dimensions, unlike the parabolic surface modeling. Free-form splines include the following kinds of methods: Cardinal, Hermite, Bézier, and non-uniform rational B-spline (NURBS). Cardinal spline is a sequence of individual curves joined to form a larger curve. Hermite spline uses two points and two tangents to model 2-D curve. Bézier spline is widely used to model smooth curves. Particularly, quadratic and cubic Bézier curves are the most typical. To model quadratic Bézier curve, only three control points are required. Most of fast IME algorithms such as UMHexagonS terminate the final search step by using SDSP, which is made up of the five search points as shown in Figure 1, as the smallest search pattern. Particularly, each of the IME search points (H_1,C,H_2) and (V_1,C,V_2), as shown in Figure 1, corresponds to the three control points of quadratic Bézier curve. A quadratic Bézier curve is also a parabolic segment but does not pass by all control points. Although the curve is not an interpolation between the control points, it can approximate the data trend. For such a reason, we introduce the quadratic Bézier curve based data trend approximation method for FME. On the other hand, NURBS is one of today's most popular splines. NURBS can be defined by degree, weighted control points, knot vector, and evaluation rules. To model a free-form curve based on NURBS, the number of control points must be greater than or equal to four. NURBS is also generalization of B-spline and Bézier spline.



(a) (b)

Fig. 2. Examples of 1-D parabolic model and quadratic Bézier curve. (a) The two curves plotted using the three matching errors located at (-1,5759), (0,1659), (1,5759). (b) The two curves at (-1,5759), (0,1659), (1,3146).

C. *Quadratic Bézier Curve*

In the 1-D parabolic model in equation 4, the three IME search points (H_1,C,H_2) or (V_1,C,V_2), as shown in Figure 1, are used to predict the best fractional pixel position at the horizontal or vertical location. As already mentioned above, quadratic Bézier curve is in accordance with this case because the three IME search points correspond to the three control points of quadratic Bézier curve. The quadratic Bézier curve algorithm can be explained as the following Equations 8 and 9. Equation 8 shows generalization of Bézier curve.

$$\begin{aligned} &p(t)=\sum_{i=0}^{n} p_i J_{n,i}(t), \quad (0 \le t \le 1) \\ &J_{n,i}(t) = {}_nC_i t^i (1-t)^{n-i} \\ &{}_nC_i = \frac{n!}{i!(n-i)!} \end{aligned} \tag{8}$$

where n denotes degree of Bézier curve, p_0, p_1, … , p_n are control points, and ${}_nC_i$ is binomial coefficient. While the parameter t moves from 0 to 1, the function $p(t)$ traces a curve. Let the matching error costs corresponding to the local coordinates (x_n,0) and (0,y_n) be X_n and Y_n. Considering the coordinates for 1-D surface modeling, the IME matching errors (H_1,C,H_2) and (V_1,C,V_2) represent {(x_0=-1,X_0), (x_1=0,X_1), (x_2=1,X_2)} and {(y_0=-1,Y_0), (y_1=0,Y_1), (y_2=1,Y_2)}, respectively. When the control point p_n=(x_n,X_n) or p_n=(y_n,Y_n), each of coordinates x_n (y_n) and X_n (Y_n) is entered separately. The quadratic Bézier curve formula using the three control points (p_0,p_1,p_2) is described in Equation 9. The following Equation 9 will be used as the core of our proposed techniques.

$$\begin{aligned} p(t) &= \sum_{i=0}^{2} p_i J_{2,i}(t) \\ &= p_0 C_0 t^0 (1-t)^2 + p_1 C_1 t^1 (1-t)^1 + p_2 C_2 t^0 (1-t)^2 \\ &= p_0 (1-t)^2 + p_1 2t(1-t) + p_2 t^2 \end{aligned} \tag{9}$$

Figure 2 shows examples of quadratic Bézier curve and 1-D parabolic model. As shown in the examples, quadratic Bézier curve does not pass by all the three control points but two points are always passed. The x or y coordinate with the lowest matching error cost will be regarded as the best prediction position x_b or y_b. The best prediction position found by using quadratic Bézier curve, however, tends to be biased more than that of the 1-D parabolic model in one direction, as described in Figure 2 (b). Thus, we also introduce a preprocessing solution to correct the one direction-biased position.

III. Proposed Bézier Curve Based FME

A. *Proposed Method 1*

In this paper, the three FME methods based on quadratic Bézier curve are proposed. The first method applies the differential operation on the quadratic Bézier curve formula. The differential operation on Equation 9 can be performed as follows:

$$p(t)' = 2t(p_0 - 2p_1 + p_2) - 2(p_0 - p_1) \tag{10}$$

Let $p(t)'$ be zero. When (p_0,p_1,p_2)=(X_0,X_1,X_2) or (Y_0,Y_1,Y_2), it is possible to obtain the optimum t_b to determine the minimum matching error cost, as shown below:

$$\begin{aligned} &p(t)' = 2t(p_0 - 2p_1 + p_2) - 2(p_0 - p_1) = 0 \\ &t_b = (p_0 - p_1)/(p_0 - 2p_1 + p_2) \end{aligned} \tag{11}$$

As described in Equation 12, when (p_0,p_1,p_2)=(x_0,x_1,x_2)=(y_0,y_1,y_2)=(-1,0,1), we can find the best fractional pixel prediction position $p(t_b)$ by substituting the above optimum t_b for t in Equation 9.

$$p(t_b) = p_0(1-t_b)^2 + p_1 2t_b(1-t_b) + p_2 t_b{}^2 = 2t_b - 1 \tag{12}$$

Table I shows the factional pixel motion vector (FMV) matching probability between the best FMV found by FFPS and the best prediction FMV found by the 1-D parabolic model based prediction (1-D_PM), and that found by the proposed first method (BÉZIER) at quarter pixel resolution. In the simulation, we suppose that the FMV matching performance of FFPS is always the best. The CIF sequence "Football" is used as test input image and it consists of 100 frames. UMHexagonS for IME is selected and it returns the five neighboring IME matching errors every time. The best fractional pixel prediction positions determined by the two mathematical model based algorithms are quantized by the quantization operations introduced by [9]. Abnormal cases, for instance, such as the IME matching error H_1 (H_2)=0 or unknown, are not allowed and the FME process for the macroblock in the exceptional case is skipped. If the x or y coordinate of the best FMV found by FFPS is equal to that found by a mathematical model based prediction, it counts the number of the matching FMVs by the position |P|. As shown in Table I, the 1-D parabolic model based prediction can produce more accurate FMVs compared with the quadratic Bézier curve based prediction. As mentioned above, however, the 1-D parabolic model based prediction can never find the best x_b or y_b located at |P|>0.5, whereas the quadratic Bézier curve based prediction can do that. That is, compared with the 1-D parabolic model based prediction, the quadratic Bézier curve based prediction provides higher robustness to large motions. We should take note of the fact that, in general, the macroblocks having large motions result in higher distortion than those having small motions.

TABLE I
The Factional Pixel Motion Vector Matching Probability (%)

Method	Position	[0,±0.75]	|P| = 0	|P| = 0.25	|P| = 0.5	|P| = 0.75
1-D_PM	x-coord.	28.673	50.725	23.179	15.063	00.000
1-D_PM	y-coord.	24.592	39.588	23.874	14.763	00.000
BÉZIER	x-coord.	22.465	36.714	13.949	14.273	24.193
BÉZIER	y-coord.	18.278	25.977	13.483	14.364	25.168

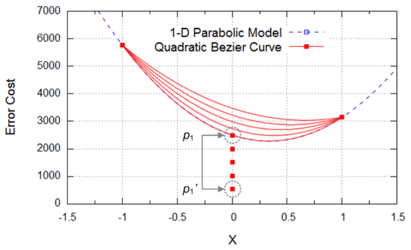


Fig. 3. The determination of the shape of Bézier curve by controlling p_1'.

B. Proposed Method 2

The second method is performed by predicting p_1' in order to pass by all the three control points including p_1. p_1' should be predicted so that p_1 can exist on Bézier curve. As p_1 is equal to $p(t=0.5)$, p_1' can be computed as shown below:

$$p_1 = p(\frac{1}{2}) = p_0(1-\frac{1}{2})^2 + 2p_1'\frac{1}{2}(1-\frac{1}{2}) + p_2(\frac{1}{2})^2$$
$$p_1' = \frac{1}{2}(4p_1 - p_0 - p_2) \qquad (13)$$

If p_1 in equation 11 is replaced by p_1', then the Bézier curve can pass by p_1 as well as p_0 and p_2, as shown below:

$$t_b = (p_0 - p_1')/(p_0 - 2p_1' + p_2) \qquad (14)$$

Finally, when $(p_0,p_1,p_2)=(X_0,X_1,X_2)$ or (Y_0,Y_1,Y_2), the formula in Equation 15 is carried out to determinate the best fractional pixel prediction position $p(t_b)$. The result of the prediction is the same with that of the 1-D parabolic model based prediction. The proposed second method, however, can easily be extended to the following third method.

$$p(t_b) = 2t_b - 1 = (p_0 - p_2)/(2p_0 - 4p_1 + 2p_2) \qquad (15)$$

C. Proposed Method 3: Determination of Adjusting Factors

As shown in Table II, each location (x,y) corresponds to the five IME search points (H_1,H_2,C,V_1,V_2). Table II (a) shows the average matching error costs on the five IME search points for the "Claire" sequence and Table II (b) is for the "Stefan" sequence. In the "Claire" case, on reference to Figure 2 (a), the prediction curves are almost symmetrical vertically. In the "Stefan" case, on the other hand, the horizontal search points (H_1,C,H_2) will form slightly uneven curves, on reference to Figure 2 (b). That is, the simulation implies that the FMVs for the "Claire" are more center-biased than those for the "Stefan". Furthermore, we can assume the following:

TABLE II
AVG. INTEGER PIXEL MOTION ESTIMATION MATCHING ERROR COSTS

(A) ERROR COSTS FOR "CLAIRE"

(x, y)	-1	0	1
-1		122.100	
0	133.955	103.513	133.174
1		121.288	

(B) ERROR COSTS FOR "STEFAN"

(x, y)	-1	0	1
-1		311.212	
0	283.471	236.197	294.131
1		308.656	

TABLE III
THE PSEUDO CODE FOR THE PROPOSED METHOD 3

```
$D = (0.5 \times (4 \times p_1 - p_0 - p_2)) - p_1$
$AF1 = if\ (p_0 > p_2)\ then\ (p_0 / p_2) - 1;\ else\ (p_2 / p_0) - 1$
$AF2 = (p_0 + p_2) / (2 \times p_1)$
$AF3 = if\ (1.5 > AF2)\ then\ AF1 \times 10;\ else\ AF2 - 1$
$p_1' = p_1 + (D \times AF3)$
```

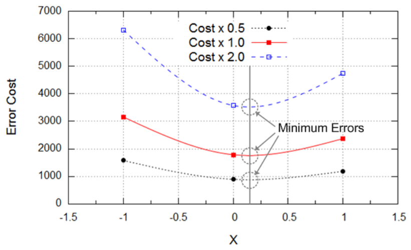


Fig. 4. Examples of the error cost scaling. The best prediction x coordinate of the original curve is almost the same with those of the two distorted curves scaled up and down by vertical.

1) First, in quadratic Bézier curve, the more similar the matching error H_1 (V_1) is to H_2 (V_2), the closer the best prediction position is to the center. The best prediction position will also be similar to that of 1-D parabolic model.

2) Second, in quadratic Bézier curve, the higher or lower H_1 is than H_2, the more the best prediction position is biased in one direction. That is, the best prediction position will be located farther away from that of 1-D parabolic model.

3) Third, in quadratic Bézier curve, the farther H_1 and H_2 are away from C, the closer the best prediction position is to the center. The best prediction position will also be similar to that of 1-D parabolic model.

4) Finally, in quadratic Bézier curve, the closer H_1 and H_2 are to C, the more the best prediction position is biased in one direction. That is, the best prediction position will be located farther away from that of 1-D parabolic model.

In the proposed second method, we know that the shape of quadratic Bézier curve can be determined by controlling the predicted control point p_1', as illustrated in Figure 3. The adjusting factors used in the proposed third method are described in the pseudo code of Table III. Each adjusting factor was composed based on the former assumptions. Let $(p_0,p_1,p_2)=(X_0,X_1,X_2)$ or (Y_0,Y_1,Y_2). In Table III, the variable D, which represents the original distance between p_1 and p_1', can be obtained by applying Equation 13 in the proposed second method. The adjusting factor $AF1$ is based on the assumption as follows: The higher the ratio of p_0 to p_2 is, the bigger the gap between p_1 and p_1' is. The ratio of p_0+p_2 to a double p_1 is for the adjusting factor $AF2$, which represents how far away they are. The critical value 1.5 in the second conditional sentence is used to determine the adjusting factor $AF3$. Through many experiments, the critical value was selected higher than $AF2$=1.29 computed by the matching errors (H_1,C,H_2) in Table II (a). The last line of the pseudo code shows that the position of p_1 is determined by applying D adjusted by $AF3$.

TABLE IV
THE PSEUDO CODE FOR THE ERROR COST SCALING

```
$if\ (p_0 < p_2)\ then\ Max = p_2;\ else\ Max = p_0;$
$if\ (Max < p_1)\ then\ Max = p_1;$
$if\ (Max > 130)\ then$
{
    $S\_Rate = 130 / Max;$
    $p_0 = p_0 \times S\_Rate;$
    $p_1 = p_1 \times S\_Rate;$
    $p_2 = p_2 \times S\_Rate;$
}
```

D. Proposed Method 3: Error Cost Scaling

As stated above, the proposed third method determines the position of p_1' applying the ratios computed by reusing the IME matching error costs. Here, we realize that p_1' is seriously affected by the matching error cost level. Particularly, the IME matching errors for the video images including large motions, such as "Football" and "Stefan" sequences, are very irregular and vigorous. We therefore need to scale the error cost level artificially.

As shown in Figure 4, there are three curves, including the two distorted ones minimized by 50% and enlarged by 200% in matching error cost. Due to this scaling, the error cost levels were changed obviously but the positions on *x*-axis are not. Although the IME matching errors are distorted by the scaling, the best prediction position can still be found without any difficulty. That is, the error cost scaling normalizes and distorts the irregular matching error costs, whereas the characteristic shape of the curve is maintained.

Based on this observation, the error cost scaling process is carried out as described in Table IV. First, it gets the highest matching error cost of the three IME matching errors $(p_0,p_1,p_2)=(X_0,X_1,X_2)$ or (Y_0,Y_1,Y_2). Second, if the maximum error cost is higher than the critical value 130, it changes the maximum error cost to the critical value and saves the scaling rate *S_Rate*. The critical value 130 was adjusted by carrying out a number of simulations on reference to the average IME matching errors in Table II (a). In the end of the error cost scaling process, it applies the scaling rate to the other matching error costs. After that, the prediction process of the proposed third method is started.

The procedure of the proposed third method can be summarized as follows:

Step 1) If the IME process for a macroblock is terminated and returns the five IME matching error costs, the error cost scaling of the proposed third method is performed.

Step 2) The adjusting factors for controlling p_1' are determined by the pseudo code described in Table III.

Step 3) The predicted p_1' is entered in Equation 14 and the optimum t_b is computed.

Step 4) The best prediction position is found by applying the optimum t_b in Equation 12. The best prediction position is quantized by the quantization process in [9].

IV. Experimental Results

The simulations have been performed based on the H.264/AVC reference software JM version 12.4 [11]. We have conducted them with the default settings of search range=16, quantization parameter (QP)=20, 24, 28, 32, and rate distortion optimized mode=1 under baseline profile. UMHexagonS [7] implemented in JM is used for fast IME. The performances of the proposed methods 1-3 (METHOD_1-3) are evaluated by comparing with those of CBFPS [7] and the 1-D parabolic model based prediction (1-D_PM). CBFPS was chosen as one of the most popular interpolation based FME algorithms. In the final step of the proposed methods and 1-D_PM, the best fractional pixel prediction position is quantized by the quantization operations at quarter pixel resolution, as presented by [9]. The six popular sequences used in the test are as follows: QCIF "Claire" and "Salesman", CIF "Football", "News", "Stefan", and "Table". Each sequence includes different types of motions such as small, middling, and large motion. The number of frames to be encoded is 100.

As listed in Tables V-VIII, the PSNR performance of METHOD_3 is better than that of 1-D_PM. Particularly, the average PSNR at QP=28 is considerably close to that of CBFPS. Considering the computational complexity, METHOD_3 has no use for fractional pixel search points formed by the interpolation, whereas CBFPS requires at least five FME search points per FME on average. The bitrates comparisons in Tables V-VIII show that METHOD_3 can produce competitive bitrates compared with 1-D_PM. On the other hand, the total PSNR drop of METHOD_1 with respect to CBFPS is lower compared with that of 1-D_PM but the bitrates are a little bit increased. METHOD_2 always shows the same results with 1-D_PM.

V. Conclusion

In this paper, the quadratic Bézier curve based fast FME was introduced to reduce the high computational complexity of the FME module. Although the parabolic models have generally been applied to speed up the FME module, according to an analysis conducted by us, each of them has some problems. The proposed techniques without using additional search points were developed to achieve similar quality performance with the interpolation based FME. The simulation results show that the PSNR performance of METHOD_3 is close to CBFPS, whereas the proposed method is interpolation-free. Moreover, the simplicity of the algorithm will make it suitable for hardware implementation, it can directly be implemented in the IME module, and easily be extended to 1/8 or 1/16 pixel ME.

References

[1] K. R. Rao and J. J Hwang, *Techniques and Standards for Image, Video and Audio Coding*, Englewood Cliffs, NJ: Prentice Hall, 1996.

[2] *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 ISO/IEC MPEG and ITU-T VCEG, JVT-G050r1, Geneva, Switzerland, Mar. 2003.

[3] Y. W. Huang, B. Y. Hsieh, S. Y. Chien, S. Y. Ma, and L. G. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," *IEEE Trans. on Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 507-522, Apr. 2006.

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

[5] C. Zhu, X. Lin, and L. P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. on Circuits Syst. Video Technol.*, vol. 12, pp. 349-355, May 2002.

[6] X. Jing and L. P. Chau, "An efficient three-step search algorithm for block motion estimation," *IEEE Trans. on Multimedia.*, vol. 6, no. 3, pp. 435-438, Jun. 2004.

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

[8] J. W. Suh and J. Jeong, "Fast sub-pixel motion estimation techniques having lower computational complexity," *IEEE Trans. on Consumer Electronics.*, vol. 50, pp. 968-973, Aug. 2004.

[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, Dec. 2005.

[10] S. Dikbas, T. Arici, and Y. Altunbasak, "Fast motion estimation with interpolation-free sub-sample accuracy," *IEEE Trans. on Circuits Syst. Video Technol.*, vol. 20, no. 7, pp. 1047-1051, Jul. 2010.

[11] JVT H.264/AVC Reference Software Joint Model (JM), http://iphome.hhi.de/suehring/tml/

TABLE V

Performance Comparison (QP=20)

Sequence	Method	PSNR (dB)	Bitrates (kbps)	Search points
Claire	CBFPS	45.713	109.838	4.904
	1-D_PM	45.674	116.882	0.000
	METHOD_1	45.676	116.561	0.000
	METHOD_2	45.674	116.882	0.000
	METHOD_3	45.672	116.225	0.000
Salesman	CBFPS	42.217	180.509	5.085
	1-D_PM	42.175	195.919	0.000
	METHOD_1	42.163	195.487	0.000
	METHOD_2	42.175	195.919	0.000
	METHOD_3	42.177	195.746	0.000
Football	CBFPS	43.595	4009.248	8.589
	1-D_PM	43.564	4103.251	0.000
	METHOD_1	43.575	4110.252	0.000
	METHOD_2	43.564	4103.251	0.000
	METHOD_3	43.564	4098.579	0.000
News	CBFPS	43.651	669.869	5.257
	1-D_PM	43.605	703.241	0.000
	METHOD_1	43.601	703.435	0.000
	METHOD_2	43.605	703.241	0.000
	METHOD_3	43.602	701.683	0.000
Stefan	CBFPS	43.299	4244.916	6.871
	1-D_PM	43.264	4325.671	0.000
	METHOD_1	43.259	4330.272	0.000
	METHOD_2	43.264	4325.671	0.000
	METHOD_3	43.263	4322.832	0.000
Table	CBFPS	42.699	2918.530	7.370
	1-D_PM	42.640	3045.521	0.000
	METHOD_1	42.647	3047.391	0.000
	METHOD_2	42.640	3045.521	0.000
	METHOD_3	42.642	3041.230	0.000

TABLE VI

Performance Comparison (QP=24)

Sequence	Method	PSNR (dB)	Bitrates (kbps)	Search points
Claire	CBFPS	42.715	61.224	4.795
	1-D_PM	42.732	64.267	0.000
	METHOD_1	42.712	64.154	0.000
	METHOD_2	42.732	64.267	0.000
	METHOD_3	42.733	63.845	0.000
Salesman	CBFPS	38.947	103.531	5.193
	1-D_PM	38.907	113.136	0.000
	METHOD_1	38.915	113.371	0.000
	METHOD_2	38.907	113.136	0.000
	METHOD_3	38.909	113.287	0.000
Football	CBFPS	40.464	2637.679	8.137
	1-D_PM	40.464	2711.268	0.000
	METHOD_1	40.456	2714.177	0.000
	METHOD_2	40.464	2711.268	0.000
	METHOD_3	40.455	2706.036	0.000
News	CBFPS	41.135	390.722	5.092
	1-D_PM	41.101	410.918	0.000
	METHOD_1	41.097	411.953	0.000
	METHOD_2	41.101	410.918	0.000
	METHOD_3	41.103	410.904	0.000
Stefan	CBFPS	39.844	2560.536	6.630
	1-D_PM	39.810	2631.269	0.000
	METHOD_1	39.813	2634.099	0.000
	METHOD_2	39.810	2631.269	0.000
	METHOD_3	39.814	2630.662	0.000
Table	CBFPS	39.253	1591.188	7.075
	1-D_PM	39.212	1667.450	0.000
	METHOD_1	39.213	1672.877	0.000
	METHOD_2	39.212	1667.450	0.000
	METHOD_3	39.214	1667.462	0.000

TABLE VII

Performance Comparison (QP=28)

Sequence	Method	PSNR (dB)	Bitrates (kbps)	Search points
Claire	CBFPS	39.716	33.324	4.660
	1-D_PM	39.676	34.181	0.000
	METHOD_1	39.701	34.135	0.000
	METHOD_2	39.676	34.181	0.000
	METHOD_3	39.741	34.351	0.000
Salesman	CBFPS	35.799	59.638	5.286
	1-D_PM	35.763	63.914	0.000
	METHOD_1	35.790	64.222	0.000
	METHOD_2	35.763	63.914	0.000
	METHOD_3	35.789	64.397	0.000
Football	CBFPS	37.576	1730.412	7.493
	1-D_PM	37.563	1772.098	0.000
	METHOD_1	37.560	1779.343	0.000
	METHOD_2	37.563	1772.098	0.000
	METHOD_3	37.573	1775.998	0.000
News	CBFPS	38.517	230.314	4.920
	1-D_PM	38.500	241.286	0.000
	METHOD_1	38.522	242.534	0.000
	METHOD_2	38.500	241.286	0.000
	METHOD_3	38.510	240.792	0.000
Stefan	CBFPS	36.452	1441.358	6.459
	1-D_PM	36.436	1502.765	0.000
	METHOD_1	36.439	1506.394	0.000
	METHOD_2	36.436	1502.765	0.000
	METHOD_3	36.443	1501.999	0.000
Table	CBFPS	36.250	861.838	6.591
	1-D_PM	36.225	903.982	0.000
	METHOD_1	36.227	903.686	0.000
	METHOD_2	36.225	903.982	0.000
	METHOD_3	36.235	902.755	0.000

TABLE VIII

Performance Comparison (QP=32)

Sequence	Method	PSNR (dB)	Bitrates (kbps)	Search points
Claire	CBFPS	36.753	18.552	4.535
	1-D_PM	36.749	18.607	0.000
	METHOD_1	36.750	18.727	0.000
	METHOD_2	36.749	18.607	0.000
	METHOD_3	36.785	18.895	0.000
Salesman	CBFPS	32.700	33.643	5.294
	1-D_PM	32.655	34.548	0.000
	METHOD_1	32.670	34.735	0.000
	METHOD_2	32.655	34.548	0.000
	METHOD_3	32.657	34.704	0.000
Football	CBFPS	34.581	1068.910	6.921
	1-D_PM	34.582	1097.966	0.000
	METHOD_1	34.578	1097.710	0.000
	METHOD_2	34.582	1097.966	0.000
	METHOD_3	34.581	1095.737	0.000
News	CBFPS	35.626	135.864	4.826
	1-D_PM	35.627	140.616	0.000
	METHOD_1	35.627	141.439	0.000
	METHOD_2	35.627	140.616	0.000
	METHOD_3	35.638	140.976	0.000
Stefan	CBFPS	32.796	671.957	6.372
	1-D_PM	32.787	715.740	0.000
	METHOD_1	32.803	716.966	0.000
	METHOD_2	32.787	715.740	0.000
	METHOD_3	32.799	715.795	0.000
Table	CBFPS	33.252	438.204	5.979
	1-D_PM	33.228	455.419	0.000
	METHOD_1	33.234	458.244	0.000
	METHOD_2	33.228	455.419	0.000
	METHOD_3	33.229	456.706	0.000