Vol.2011-AVM-73 No.8
2011/7/15

Fractional-pixel Position Prediction Techniques Based on Bézier Curve for H.264/AVC Encoder

Chang-Uk Jeong[†] and Hiroshi Watanabe[††]

The high computation load of fractional-pixel motion estimation (FME) cannot be ignored although it improves visibly the PSNR performance after the integer-pixel motion estimation (IME) process. Most of conventional FME methods must include the interpolation procedure to form the fractional-pixel points from the integer-pixel information. However, the interpolating requires a considerable memory usage and any amount of processing time. In this paper, therefore, we propose interpolation-free fractional-pixel position prediction techniques based on Bézier curve and post the results of the simulation which show a remarkable performance improvement compared to the existing FME methods in terms of PSNR.

1. Introduction

Nowadays, the mobile network technologies including 3G and 4G wireless standards have been alarmingly many developments. Nevertheless, the transmission of a large amount of multimedia data such as high definition video contents increases dramatically traffic on both wireless and wire-based networks. Video compression standards, which are 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 such as variable block-size motion estimation and motion compensation, multiple motion vectors per macroblock, multiple reference frames, and quarter-pixel precision motion estimation and compensation, context-adaptive binary arithmetic coding (CABAC) for entropy coding, and so on. Although the new features allow it to encode video data more effectively than conventional ways, the increased computational complexity requires sufficient CPU power to perform real-time video encoding, particularly, in mobile applications. Recently, high efficiency video coding (HEVC) standard is under joint development by ISO/IEC MPEG and ITU-T VCEG. HEVC is being drafted to achieve much more 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 generally can be divided into three units, a temporal redundancy eliminator, a spatial redundancy eliminator, and an entropy encoder. The temporal redundancy eliminator estimates and extracts the motion of an object using close correlation between the adjacent video frames while the information about stationary objects is eliminated. As motion estimation (ME) is core to the temporal model, it occupies more than 80% of the total encoding time [3]. Thus, many researches on fast ME have been performed to reduce the high complexity of the ME module.

Motion estimation methods can be classified into pel-recursive algorithm (PRA) and block matching algorithm (BMA) according to the elementary unit, i.e. pixel or block, in motion estimation. BMA, in particular, has been adopted in many video compression standards due to low computational cost with robustness to errors. The three-step search (3SS) [3], the new three-step search (N3SS) [4], the four-step search (4SS) [5], the diamond search (DS) [6], the hexagon-based search (HEXBS) [7], the efficient three-step search algorithm (E3SS) [8], and the unsymmetrical-cross multi-hexagon-grid search (UMHexagonS) [9] are fast block-based

[†] Graduate School of Global Information and Telecommunication Studies, Waseda University
[††] Graduate School of Global Information and Telecommunication Studies, Waseda University

ME algorithms which were developed to effectively reduce the computational load of the ME module. The mean squared error (MSE), the mean absolute error (MAE), and the sum of absolute error (SAE) are widely used block distortion measures to search for the best matched block within a search range in the candidate area. The motion estimation module in the H.264/AVC reference model software JM [10] uses the sum of absolute transformed differences (SA(T)D). SA(T)D enhances the precision of the distortion measure by transforming the residual data but increases computational complexity.

Indeed, the position of the motion of an object in video sequence can be described as fractional-pixel as well as integer-pixel of the sampling grid distance. Fractional-pixel motion estimation (FME) improves visibly the image quality, whereas it needs the expense of higher computational complexity. The runtime of the FME module is over 30% of the total encoding time [3]. The conventional full fractional-pixel search in H.264/AVC determines the best half-pixel motion vector (MV) from eight points spaced half-pixel apart around the best integer-pixel position, followed by the quarter-pixel MV in the same way. However, the conventional method 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) [9], the fast sub-pixel motion estimation techniques having lower computational complexity [11], the quadratic prediction-based FME (QPFPS) [12], the prediction-based directional fractional-pixel search [13], the novel fractional-pixel search using motion prediction and fast search pattern [14], the low-complexity algorithm for fractional-pixel motion estimation [15], and the fast Motion Estimation with interpolation-Free sub-sample accuracy [16] have been developed to reduce the computation of the FME module. In this paper, fractional-pixel position prediction techniques based on Bézier curve are proposed. Our proposed methods focus on performing FME without using the interpolation process. In the experimental results, the performances of FME algorithms are evaluated in terms of PSNR and bitrate.

2. Surface Modeling

2.1 Regular Parabolic Models for FME

As described in Equations 1-3, the three equations are used to model parabolas with matching error surface, respectively. The parabolic functions can create a simple and intuitive error surface model. We note that the models have perfectly bilateral symmetry by a plane or an axis. The prediction degradation of the functions may be unavoidable because real-world FME error surfaces are not actually in perfect symmetry. Therefore, some of them need the interpolation-based refinement process additionally.

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

$$F_u(u) = c_1u^2 + c_2u + c_3,\ u = x\ or\ y \tag{2}$$

$$F(x,y) = c_1x^2y^2 + c_2x^2y + c_3xy^2 + c_4xy + c_5x^2 + c_6x + c_7y^2 + c_8y + c_9 \tag{3}$$

2.2 Free-form Surface Modeling

We have considered free-form surface modeling to replace the regular parabolic models. Free-form surface modeling is used to describe the skin of a 3D geometric element. They do not have rigid radial dimensions, unlike regular surfaces. Free-form splines include the following kinds of methods: Cardinal, Hermite, Bézier, and nonuniform rational B-spline (NURBS). Cardinal spline is a sequence of individual curves joined to form a larger curve. Hermite spline uses 2 points and 2 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, 3 control points required. For such a reason, we introduce the Bézier-based methods for FME. It will subsequently be explained particularly. On the other hand, NURBS has been the most popular spline method. 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 4.

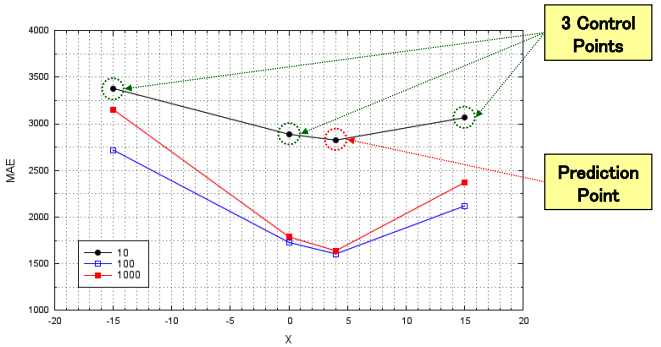


Figure 1 Average FME error surfaces simplified for 10, 100, and 1000 error surfaces.

2.3 Bézier curve

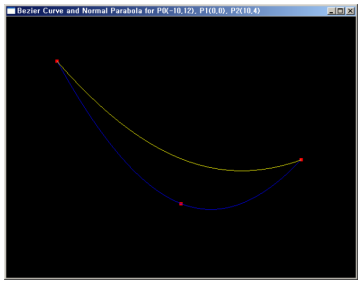
As shown in the simplified FME error surfaces (Figure 1), three control points can be used to predict the best fractional-pixel position. Particularly, quadratic Bézier curve is in accordance with this case. In this paper, we propose quadratic Bézier curve-based methods to model FME prediction surfaces. Bézier curve algorithm can be explained as the following mathematic Equations 4 and 5. Equation 4 shows generalization of Bézier curve.

$$
\begin{aligned}
&p(t) = \sum_{i=0}^{n} p_i J_{n,i}(t), \quad (0 \le t \le 1) \\
&J_{n,i}(t) = {}_nC_i t^i (1-t)^{n-i} \\
&{}_nC_i = \frac{n!}{i!(n-i)!}
\end{aligned}
\tag{4}
$$

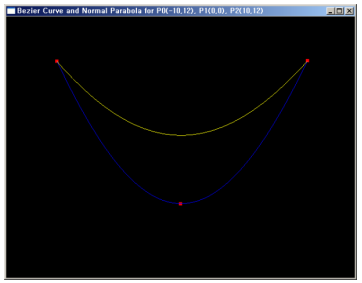
where n denotes degree of Bézier curve, p_0, p_1, … , p_n are control points, and ${}_nC_i$ is binomial coefficient. Quadratic Bézier curve formula using three points is described in Equation 5. Equation 5 will be used mainly in our proposed methods.

$$
\begin{aligned}
p(t) &= \sum_{i=0}^{2} p_i J_{2,i}(t) \\
&= p_0 C_0 t^0 (1-t)^2 + p_1 C_1 t^1 (1-t)^1 + p_2 C_2 t^0 (1-t)^2 \\
&= p_0 (1-t)^2 + p_1 2t(1-t) + p_2 t^2
\end{aligned}
\tag{5}
$$

Figure 2 shows examples of quadratic Bézier curve and normal parabola. Although a quadratic Bézier curve is a parabolic segment, it is different from normal parabola. Actually, quadratic Bézier curve does not pass all control points. The lowest position on the curve will be determined as the best fractional-pixel MV.



(a) Curves for (-10,12), (0,0), (10,4)



(b) Curves for (-10,12), (0,0), (10,12)

Figure 2 Examples of Bézier curve (yellow) and normal parabola (blue).

3. Proposed Bézier Curve-based FME

3.1 Proposed Method 1

In this paper, the four FME methods based on quadratic Bézier curve are proposed. The First method is to implement the quadratic Bézier curve formula as it is. In our methods, the formula denoted in Equation 5 is used directly. If "t" is increased by 0.125, as described in Table 1, all t values match the fractional-pixels at ¼-pel resolution. That is, in this case, the quantization process may not be required. To obtain the best x and y prediction positions, the maximum computation is as follows.

$$
[p_x(t)\ code] \times 9 + [p_y(t)\ code] \times 9 = [p(t)\ code] \times 18 \tag{6}
$$

To improve the curve precision, the gap of t can also be adjusted. However, the computational cost will be increased according to the precision.

Table 1 t in [0,1] and fractional-pixels at ¼-pixel MV resolution (±4 is not used for FME.).

t	0.000	0.125	0.250	0.375	0.500	0.625	0.750	0.875	1.000
Sub-pixel	-4	-3	-2	-1	0	1	2	3	4

3.2 Proposed Method 2

The Second method applies the differential operation on the quadratic Bézier curve formula. The differential operation on Equation 5 can be performed as follows.

$$p(t)' = 2t(p_0 - 2p_1 + p_2) - 2(p_0 - p_1) \tag{7}$$

Let $p(t)'$ be 0. It is possible to obtain "t" to predict the minimum error cost, as shown below.

$$\begin{aligned} &p(t)' = 2t(p_0 - 2p_1 + p_2) - 2(p_0 - p_1) = 0 \\ &t = (p_0 - p_1) \div (p_0 - 2p_1 + p_2) \end{aligned} \tag{8}$$

As described in Equation 9, we can find the best fractional-pixel position by substituting the above "t".

$$\begin{aligned} &if\ (p_0, p_1, p_2) = (H_1, C, H_2) = (V_1, C, V_2) = (-1,0,1), \\ &then\ p_{x\,or\,y}(t) = 2t - 1 \end{aligned} \tag{9}$$

3.3 Proposed Method 3

The third method is performed by predicting p_1' in order to pass all control points in the formula. p_1' should be predicted so that p_1 can exist on the curve. As $p_1 = p(0.5)$, p_1' can be computed as shown below.

$$\begin{aligned} &p_1 = p(\frac{1}{2}) = p_0(1-\frac{1}{2})^2 + 2p_1'\frac{1}{2}(1-\frac{1}{2}) + p_2(\frac{1}{2})^2 \\ &p_1' = \frac{1}{2}(4p_1 - p_0 - p_2) \end{aligned} \tag{10}$$

Then the curve can pass all points (p_0, p_1, p_2) as p_1 is replaced to p_1', as shown below.

$$t = (p_0 - p_1') \div (p_0 - 2p_1' + p_2) \tag{11}$$

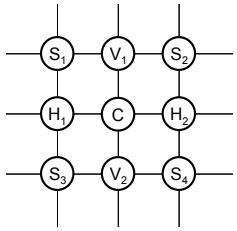


Figure 3 The nine integer-pixel search points.

In the final step, the simple code of Equation 12 is carried out to find the best fractional-pixel position. As a result, the value is the same with that of the quadratic prediction function (Equation 1). However, the third Bézier curve-based method has significance because it is applied to the forth method.

$$p_{x\,or\,y}(t) = 2t - 1 = (p_0 - p_2) \div (2p_0 - 4p_1 + 2p_2) \tag{12}$$

3.4 Proposed Method 4

Before the forth method is introduced, some simple simulations are shown in Table 2. Table 2 (a) shows the average on the error costs of the five integer-pixel points for Claire sequence. Table 2 (b) is for Stefan sequence. In the Claire case, the shape of the curve is almost symmetrical. (Refer to Figure 2 (b).) On the other hand, x-axis integer positions for Stefan look asymmetrical. (Refer to Figure 2 (a).) Based on this simulation concept, we propose the fourth Bézier curve-based method.

Table 2 Integer-pixel motion estimation (IME) error costs.

(a) IME error costs for "Claire" sequence

(x, y)	-1	0	1
-1		126.818	
0	138.030	108.275	137.138
1		126.042	

(b) IME error costs for "Stefan" sequence

(x, y)	-1	0	1
-1		311.905	
0	285.172	238.091	294.118
1		310.384	

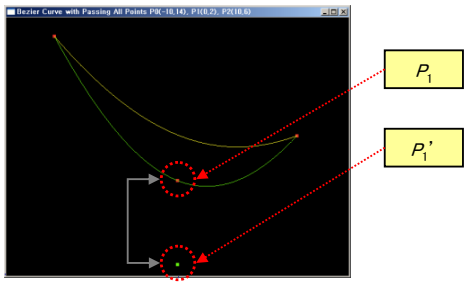


Figure 4 The determination of the shape and depth of the Bézier curve by controlling p_1'.

We know that the shape and depth of the Bézier curve can be decided by controlling p_1', as illustrated in Figure 4. To create some thresholds, we suppose as follows: The higher the ratio of H_1 to H_2 (or V_1 to V_2), a threshold can be created so that the bigger the gap between p_1 and p_1'. The ratio of $H_1 + H_2$ to a double C is also related to creating another threshold. The thresholds used in our forth method are described in the pseudo code of Table 3. D is using the third Bézier curve-based method. Each threshold can be created based on the former assumption. Finally, the position of p_1 can be determined by applying threshold TH.

Table 3 The creation of thresholds used in the proposed method 4.

□...

$D = (0.5 \times (4 \times C - H_1 - H_2)) - C$

$THRESHOLD\_1(T1) = (H_1 \div H_2) - 1$ or $(H_2 \div H_1) - 1$
$THRESHOLD\_2(T2) = (H_1 + H_2) \div (2 \times C)$
$THRESHOLD\_3(TH) = if\ (1.5 > T2)\ then\ T1;\ else\ T2 - 1$

$P_1 = C + (D \times TH)$

...

4. Experimental Results

The simulations have been performed based on JM software version 12.4. Our test is conducted with the default settings of search range=16, quantization parameter=28, entropy coding=CAVLC, and main profile. Rate distortion optimization is turned on. UMHexagonS is used for IME. To assess only the prediction of them without the interpolation process, the small diamond search process used in QPFPS is skipped. As shown in the PSNR Table 4, our proposed methods are a little bit better than QPFPS. In the proposed method 1 (t += 0.010), the PSNR performance for Football sequence is noticeable. The performances of method 3 are the same with QPFPS. On the other hand, the bit rate performance is a little bit dropped.

The PSNR performance for Claire sequence is impressive, as shown in Table 6. The proposed method 4 is superior to CBFPS without using the interpolation process. However, it is not suitable for all the cases. We need more research for the other cases.

Table 4 △PSNR (dB) performance comparison at quarter-pixel MV resolution.

FME	Sequence			
	Claire (QCIF)	Football (CIF)	News (CIF)	Table (CIF)
QPFPS	39.676	37.563	38.500	36.225
M_1 (t+=0.125)	39.701	37.565	38.527	36.226
M_1 (t+=0.010)	39.701	37.577	38.511	36.227
METHOD_2	39.701	37.560	38.522	36.227
METHOD_3	39.676	37.563	38.500	36.225

Table 5 △Bitrate (bps) performance comparison at quarter-pixel MV resolution.

FME	Sequence			
	Claire (QCIF)	Football (CIF)	News (CIF)	Table (CIF)
QPFPS	34181	1772098	241286	903982
M_1 (t+=0.125)	34135	1777802	242633	903053
M_1 (t+=0.010)	34135	1779106	242100	904092
METHOD_2	34135	1779343	242534	903686
METHOD_3	34181	1772098	241286	903982

Table 6 Performances of the proposed method 4 for “Claire” sequence.

FME	PSNR	Bitrate	Interpolation	Complexity
CBFPS	39.716	33324	Used	High
QPFPS	39.676	34181	Free	Very low
METHOD_4	39.741	34351	Free	Very low

5. Conclusion

We have proposed the FME methods based on quadratic Bézier curve. The simulation results show that quadratic Bézier-based methods are a little bit better than the quadratic-based. As shown in the simulation results, particularly, the Bézier curve-based method using the thresholds has higher efficiency compared to the existing FME methods, especially for a sequence. The PSNR performance is superior to that of the interpolation-based method.

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 ITU-T and ISO/IEC JTC1, Geneva, JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-G050r1 (2003).
3) 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, USA, pp. G5.3.1-G5.3.5 (1981).
4) R. Li, B. Zeng, and M. L. Liou, “A new three-step search algorithm for block motion estimation,” IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 438-442 (1994).
5) L. M. Po and W. C. Ma, “A novel four-step search algorithm for fast block motion estimation,” IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 313-317 (1996).
6) 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).
7) 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).
8) X. Jing and L. P. Chau, “An efficient three-step search algorithm for block motion estimation,” IEEE Trans. Multimedia., vol. 6, no. 3, pp. 435-438 (2004).
9) 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).
10) JVT H.264/AVC Reference Software Joint Model (JM), http://iphome.hhi.de/suehring/tml/
11) J. W. Suh and J. Jeong, “Fast sub-pixel motion estimation techniques having lower computational complexity,” IEEE Trans. on Consumer Electronic, vol. 50, pp. 968-973 (2004).
12) 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).
13) L. Yang, K. Yu, J. Li, and 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).
14) J. S. Kim, K. W. Lee, and 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).
15) M. Sayed, W. Badawy, and G. Jullien, “Low-complexity algorithm for fractional-pixel motion estimation,” in Proc. 16th IEEE Int. Conf. on Image Processing., vol., no., pp.1565-1568 (2009).
16) S. Dikbas, T. Arici, and Y. Altunbasak, “Fast motion estimation with interpolation-free sub-sample accuracy,” IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 7, pp. 1047-1051 (2010).