FIXED-POINT REALIZATION OF AFFINE MOTION COMPENSATION IN HEVC

Chihiro TSUTAKE　　*Toshiyuki YOSHIDA*

Dept. of Information Science,
Graduate School of Eng., Univ. of Fukui,
Fukui 910-8507, Japan

Abstract – Affine motion estimation/compensation (AME/AMC) is known as one of promising techniques for inter prediction coding, which can efficiently remove temporal redundancy in video sequences. In our previous work, we have proposed two 3-parameter affine motion models and their block-matching-based efficient search technique for HEVC, and have demonstrated its superiority over conventional techniques. Although the AME/AMC in the technique has been implemented in floating-point arithmetic, fixed-point arithmetic is preferable in an actual hardware realization. This paper thus investigates the optimal fractional bit lengths for the AME and AMC to realize them in fixed-point arithmetic. They have been determined by considering both of the coding and computation efficiencies in the experiments given in this paper.

Keywords – video coding, H.265/HEVC, affine motion estimation, affine motion compensation, fixed-point arithmetic

I. INTRODUCTION

Current video coding standards employ a motion compensation (MC) technique, many of which are realized based on the translational motion model

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \tag{1}$$

where $[x, y]^t$ and $[x', y']^t$ represent the coordinates on the target and reference frames, respectively, and $[v_x, v_y]^t$ is a motion vector (MV). This model is sufficient in relatively small coding blocks because a wide range of motion, e.g., zoom and rotation, can be approximated as translational motion.

Recently, much effort has been made for affine motion compensation (AMC) to improve the motion model (1) [1, 2]. Compared with the translational model comprising an MV alone, the affine motion model requires more parameters to be estimated, resulting in much larger computation cost. In Refs. [1, 2], the authors have introduced fast affine motion estimation (AME) techniques in HEVC using least-square-based minimization methods. Although the techniques are much faster than an exhaustive block matching (BM) technique, hardware implementation of such a technique requires modules totally different from conventional BM-based ones.

In our pervious work [3], we have proposed a BM-based fast AME technique, and demonstrated its superiority compared with the least-square-based ones. In the techniques, the coding efficiency (BD-Rate [%]) can be improved about −9.74% over the original HEVC test model (HM). As reviewed in Sect. II, the AMC in Ref. [3] is realized by a coordinate calculation followed by a pixel interpolation, both of which were calculated in floating-point arithmetic. In an actual hardware implementation, however, since floating-point arithmetic requires more hardware cost, fixed-point-based implementation is preferable. We thus investigate the optimal fractional bit lengths for the coordinate calculation and pixel interpolation in our AME/AMC technique to realize them in fixed-point arithmetic.

The rest of this paper is organized as follows. Sect. II briefly reviews our previous work for the AME/AMC. Sect. III makes some preparations for our discussion. Sect. IV gives experimental results and determines the optimal fractional bit lengths. Finally, Sect. V concludes this paper.

II. OUR PREVIOUS WORK [3]

In Refs. [1, 2], the 4-parameter affine motion model

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad (2)$$

has been employed, where a and b represent affine parameters. Although the model can compensate for both of zoom and rotation simultaneously, it requires large computation cost in a BM-based parameter estimation because a nest of search loops is necessary: the search loops of $[v_x, v_y]^t$ should be nested in the ones for a and b.

To relax the ME complexity, Ref. [3] has introduced the following two 3-parameter affine motion models

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1+s & 0 \\ 0 & 1+s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \quad (3)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & -r \\ r & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \quad (4)$$

where s and r represent the zoom and pseudo small rotation parameters, respectively. Although our models cannot compensate for zoom and rotation simultaneously, they are advantageous in terms of the rate-distortion optimization compared with the 4 parameter model (2), whose details are illustrated in Ref. [3].

The impact of the AMC efficiency depends on the number of search points N and quantization step size Δ in practice. We have found $N = 16$ and $\Delta = 1/256$ for s and r as the optimal in Ref. [3] based on several experiments, which quantizes s and r into

$$\begin{cases} s_i = i \cdot \Delta & (i = -8, \cdots, -2, -1, 1, 2, \cdots, 8) \\ r_i = i \cdot \Delta & (i = -8, \cdots, -2, -1, 1, 2, \cdots, 8) \end{cases} \quad (5)$$

for the quantization index i.

As the transformed coordinate $[x', y']^t$ leads to a real number in general, a pixel interpolation technique is necessary in both of the AME and AMC. Although an upsampling of the entire reference frame is efficient from the viewpoint of the computation cost, such a technique requires large amount of frame memories. To avoid such a difficulty, in Ref. [3], we have introduced a bilinear interpolation technique applied to a factor of 4 interpolation. In more detail, the pixel values on the coordinate $[x', y']$ are calculated by using its surrounding pixels by the 4-time DCT interpolation filter (DCTIF) standardized in HEVC.

Since a brute-force parameter estimation approach, referred to as "full search", requires much computation cost compared with a least-square-based one, we have introduced a simple BM-based fast AME technique, referred to as "iterative greedy BM-based search" in Ref. [3]. Our approach has been compared with a least-square-based one [2] in Ref. [3], and we have concluded that our approach outperforms Ref. [2] in terms of both of the coding efficiency and computation cost for 10 test sequences including translation/zoom/rotation, where the average BD-Rate [%] and ΔT [%] have been obtained as -9.74% and 20.79%, respectively.

III. PREPARATIONS

Although additional hardware modules for the AME are drastically relaxed in Ref. [3], floating-point arithmetic, requiring more hardware cost compared with fixed-point one, is employed in Ref. [3]. We thus investigate the optimal fractional bit lengths for the AME/AMC to realize them in fixed-point arithmetic in this paper. This section makes some preparations in terms of floating-point-based calculation in Ref. [3] to be realized in fixed-point arithmetic, and elaborates our fixed-point realization strategy.

A. Floating-point arithmetic in Ref. [3]

a. Coordinate calculations (F_1)

Since s and r in Eqs. (3) and (4) are real values, x' and y' lead to real ones in general. However, since we have quantized in s and r as in Eq. (5), a substitution of Eq. (5) into Eqs. (3) and (4) leads to

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x + x \cdot (i \cdot \Delta) + v_x \\ y + y \cdot (i \cdot \Delta) + v_y \end{bmatrix}, \quad (6)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x - y \cdot (i \cdot \Delta) + v_x \\ y + x \cdot (i \cdot \Delta) + v_y \end{bmatrix}. \quad (7)$$

Note that x and y are integer values and the fractional part of $i \cdot \Delta$ can be represent in 8 bits as in Eq.(5). As the fractional bit length of $i \cdot \Delta$ is large enough compared with quarter-pel-precision v_x or v_y, we can conclude that the necessary bit lengths for the fractional part of x' and y' are equal to that of $i \cdot \Delta$, which can be restricted less than 8 bits. The arithmetic for Eqs. (6) and (7) is referred to as "F_1" in this paper.

b. DCTIF (F_2)

In the HEVC standard, fractional part of the 4-time DCTIF coefficients are quantized in 6 bits for the luminance component. Since, however, the AME/AMC

Table 1: Test sequences utilized in this paper. "Z", "R", and "T" in the third column denote Zoom, Rotation, and Translation, respectively.

Test Seq.	Resolution	Dominant Motion
1. BQSquare	416×240	Z + T
2. Cactus	1920×1080	R + T
3. Drone1	1920×1080	Z + R + T
4. Drone2	1920×1080	Z + R + T
5. FungusZoom	416×240	Z

scheme has been introduced in Ref. [3], the fractional bit length for the DCTIF should be reconsidered, which is referred to as "F_2" in this paper[1].

c. Pixel interpolation (F_3)

To obtain the pixel value on the coordinate $[x', y']^t$, Ref. [3] applies the bilinear interpolation

$$\begin{aligned} f(x', y') &= f(x_0, y_0) \cdot (1 - \delta_x) \cdot (1 - \delta_y) \\ &+ f(x_0 + 1, y_0) \cdot \delta_x \cdot (1 - \delta_y) \\ &+ f(x_0, y_0 + 1) \cdot (1 - \delta_x) \cdot \delta_y \\ &+ f(x_0 + 1, y_0 + 1) \cdot \delta_x \cdot \delta_y \end{aligned} \quad (8)$$

for both in the AME and AMC, where f is the 4-time-upsampled reference pixel value by the DCTIF, x_0 and y_0 represent the integer part of x' and y', respectively, and δ_x and δ_y represent the fractional part of x' and y', respectively. Each term in Eq. (8) comprises a pixel value and fractional parts of x and y directions, to be calculated in a fixed-point arithmetic with a required fractional bit length, which is referred to as "F_3".

B. Fixed-point realization

Since the arithmetics F_1, F_2, and F_3 are necessary in both of the AMC and AME, we investigate the required fractional bit lengths for them. Let n_1^{AMC}, n_2^{AMC}, and n_3^{AMC} denote the the fractional bit lengths for F_1, F_2, and F_3 in the AMC, respectively, while n_1^{AME}, n_2^{AME}, and n_3^{AME} for the AME.

As the fractional bit length for x' and y' is restricted into n_1^{AMC} bits, the factors δ_x, δ_y, $(1-\delta_x)$, and $(1-\delta_y)$ in Eq. (8) are represented in $(n_1^{\mathrm{AMC}}-2)$ bits; δ_x and δ_y are the fractional part of $4x'$ and $4y'$, respectively because the bilinear interpolation (8) is applied to pixels

[1] We optimize the fractional bit length of DCTIF for luminance component only.

Table 2: The BD-Rates varying n_1^{AMC}, n_2^{AMC}, and n_3^{AMC} with $(n_1^{\mathrm{AME}}, n_2^{\mathrm{AME}}, n_3^{\mathrm{AME}}) = (\infty, \infty, \infty)$ fixed. The bold face indicate the optimal fractional bit lengths for each step.

n_1^{AMC} [bit]	n_2^{AMC} [bit]	n_3^{AMC} [bit]	BD-Rate [%]
7	∞	∞	-6.79
8	$\boldsymbol{\infty}$	$\boldsymbol{\infty}$	**-9.20**
8	5	∞	-8.99
8	**6**	$\boldsymbol{\infty}$	**-9.08**
8	7	∞	-9.09
8	8	∞	-9.05
8	6	6	-6.15
8	6	7	-7.51
8	6	8	-8.34
8	**6**	**9**	**-8.89**
8	6	10	-9.02

Table 3: The BD-Rates varying n_1^{AME}, n_2^{AME}, and n_3^{AME} with $(n_1^{\mathrm{AMC}}, n_1^{\mathrm{AMC}}, n_3^{\mathrm{AMC}}) = (8, 6, 9)$ fixed. The bold face indicate the optimal fractional bit lengths for each step.

n_1^{AME} [bit]	n_2^{AME} [bit]	n_3^{AME} [bit]	BD-Rate [%]
7	∞	∞	-6.62
8	$\boldsymbol{\infty}$	$\boldsymbol{\infty}$	**-8.89**
8	5	∞	-8.80
8	**6**	$\boldsymbol{\infty}$	**-8.90**
8	7	∞	-8.94
8	8	∞	-8.95
8	6	6	-5.09
8	6	7	-6.80
8	6	8	-7.57
8	**6**	**9**	**-8.91**
8	6	10	-8.95

4-time interpolated by the DCTIF. Then, the cross-factors in Eq. (8), e.g., $\delta_x \cdot \delta_y$ and $(1-\delta_x) \cdot (1-\delta_y)$, are represented in $(2 \cdot n_1^{\mathrm{AMC}} - 4)$ bits. We can thus search the optimal n_2^{AMC} around $(2 \cdot n_1^{\mathrm{AMC}} - 4)$. This is also the case for n_2^{AME}.

Since the simultaneous determinations of n_k^{AMC} and n_k^{AME} $(k = 1, 2, 3)$ are infeasible because of their computation costs, we independently optimize them in the AMC in the order n_1^{AMC}, n_2^{AMC}, and n_3^{AMC} at first with $(n_1^{\mathrm{AME}}, n_2^{\mathrm{AME}}, n_3^{\mathrm{AME}}) = (\infty, \infty, \infty)$, i.e., F_1, F_2, and F_3 are calculated in floating-point arithmetic. In more detail, we first optimize n_1^{AMC} with $n_2^{\mathrm{AMC}} = n_3^{\mathrm{AMC}} = \infty$ fixed. n_2^{AMC} is then determined with optimal n_1^{AMC} in the previous step and $n_3^{\mathrm{AMC}} = \infty$ fixed, and n_3^{AMC} is finally optimized with n_1^{AMC} and n_2^{AMC} fixed to the optimal values. The fractional bit lengths in the AME, i.e., n_1^{AME}, n_2^{AME}, and n_3^{AME}, will be optimized by the same way with n_1^{AMC}, n_2^{AMC}, and n_3^{AMC} fixed to the optimal values.

IV. EXPERIMENTAL RESULTS

To determine the necessary and sufficient fractional bit lengths defined in the previous section, F_1, F_2 and F_3 are realized in fixed-point arithmetic in the encoder/decoder in Ref. [3]. In this experiment, the 5 test sequences including translation/zoom/rotation listed in Table 1 are encoded with n_k^{AMC} and n_k^{AME} ($k = 1, 2, 3$) varied, and the average BD-Rate over the 5 test sequences is evaluated. Since the average BD-Rate is about -9.2% by the floating-point encoder in Ref. [3], the smallest value for n_k^{AMC} and n_k^{AME} ($k = 1, 2, 3$) is selected as the optimal with its BD-Rate as dose to -9.2% as possible.

A. Results in AMC

Table 2 shows the BD-Rates with n_1^{AMC}, n_2^{AMC}, and n_3^{AMC} varied. We first varied n_1^{AMC} with $n_2^{\mathrm{AMC}} = n_3^{\mathrm{AMC}} = \infty$ as mentioned in Sect. III.B. As can be seen in Table 2, $(n_1^{\mathrm{AMC}}, n_2^{\mathrm{AMC}}, n_3^{\mathrm{AMC}}) = (7, \infty, \infty)$, i.e., $i \cdot \Delta$ is quantized in 7 bits, drastically degrades the BD-Rate while $(n_1^{\mathrm{AMC}}, n_2^{\mathrm{AMC}}, n_3^{\mathrm{AMC}}) = (8, \infty, \infty)$ keeps the original BD-Rate. Therefore, we conclude that $n_1^{\mathrm{AMC}} = 8$ is the optimal, which can be easily predicted because s and r in Eqs. (3) and (4) are quantized in 8 bits in Eq. (5).

We then varied n_2^{AMC} in the range from 5 to 8 with $n_1^{\mathrm{AMC}} = 8$ and $n_3^{\mathrm{AMC}} = \infty$ fixed. From Table 2, since the BD-Rate almost saturates in $(n_1^{\mathrm{AMC}}, n_2^{\mathrm{AMC}}, n_3^{\mathrm{AMC}}) = (8, 6, \infty)$, we thus determined $n_2^{\mathrm{AMC}} = 6$ as the optimal, which is same as in the HEVC standard.

We finally evaluate n_3^{AMC} with $n_1^{\mathrm{AMC}} = 8$ and $n_2^{\mathrm{AMC}} = 6$ fixed. A variation of n_3^{AMC} as in Table 2 gives the result that $(n_1^{\mathrm{AMC}}, n_2^{\mathrm{AMC}}, n_3^{\mathrm{AMC}}) = (8, 6, 9)$ gives almost the same BD-Rate with $(n_1^{\mathrm{AMC}}, n_2^{\mathrm{AMC}}, n_3^{\mathrm{AMC}}) = (8, 6, \infty)$. Therefore, $n_3^{\mathrm{AMC}} = 9$ is selected as the optimal.

B. Results in AME

In this section, we determine the optimal bit lengths of n_1^{AME}, n_2^{AME}, and n_3^{AME} under the condition $(n_1^{\mathrm{AMC}}, n_2^{\mathrm{AMC}}, n_3^{\mathrm{AMC}}) = (8, 6, 9)$ optimized in the AMC. The same strategy in Sect. IV.A was applied to them, whose results are listed in Table 3.

From Table 3, $(n_1^{\mathrm{AME}}, n_2^{\mathrm{AME}}, n_3^{\mathrm{AME}}) = (8, 6, 9)$ keeps the BD-Rate for $(n_1^{\mathrm{AME}}, n_2^{\mathrm{AME}}, n_3^{\mathrm{AME}}) = (\infty, \infty, \infty)$, and we conclude that $(n_1^{\mathrm{AME}}, n_2^{\mathrm{AME}}, n_3^{\mathrm{AME}}) = (8, 6, 9)$ is the optimal in the AME, which is exactly same as in the AMC.

V. CONCLUSIONS

This paper has investigated the necessary and sufficient fractional bit lengths for the AME and AMC in Ref. [3] to be realized in fixed-point arithmetic. Since the AME and AMC comprise three kinds of calculations, i.e., the coordinate calculation, DCT-based 4-time interpolation, and bilinear interpolation, the necessary and sufficient fractional bit lengths have been determined for the three calculations. Based on the experiments for encoding 5 sequences with varying the fractional bit lengths for the three calculations, we have conclude that fractional bit lengths 8, 6, and 9 bits are necessary and sufficient for the coordinate calculation, DCT-based interpolation filter, and bilinear interpolation, respectively.

REFERENCES

[1] C. Heithausen and H. Vorwerk, "Motion compensation with higher order motion model for HEVC," IEEE International Conference on Acoustic Speech and Signal Processing, pp. 1438–1442, April 2015.

[2] C. Heithausen, M. Meyer, M. Blaser, and J. -R. Ohm, "Improved higher order motion compensation in HEVC with block-to-block translational shift compensation," IEEE International Conference on Acoustic Speech and Signal Processing, pp. 2008–2012, Sptember 2016.

[3] C. Tsutake and T. Yoshida, "Block-Matching-Based Implementation of Affine Motion Estimation for HEVC," IEICE Trans. on Information, submitted.