Low Complexity Multiple Candidate Motion Estimation Based on Constrained One-bit Transforms

Changryoul Choi and Jechang Jeong
Dept. of Electronics and Communication Engineering Hanyang University
Seoul, Korea
e-mail : denebchoi@gmail.com & jjeong@ece.hanyang.ac.kr

***Abstract*— In this paper, we propose a low complexity multiple candidate motion estimation algorithm based on the constrained one-bit transform. We propose variations of constrained one-bit transform whose matching criteria are almost the same as the constrained one-bit transform. The motion estimation performances of the proposed variations are statistically similar to that of constrained one-bit transform in whole, but its local behaviors are very different. By adopting the multiple candidate search strategy into the typical constrained one-bit transform and its variation thereafter, we can efficiently determine two best motion vectors and enhance the overall motion estimation accuracy. Experimental results show that the proposed algorithm achieves peak-to-peak signal-to-noise ratio gains up to 0.66dB on average compared with the conventional constrained one-bit transform-based motion estimation with negligible complexity increase.**

Keywords-motion estimation; bit-wise matching; constrained one-bit transform

I. INTRODUCTION

Motion estimation (ME) is the key technique in video compression and has been widely used in many video applications such as video compression, video segmentation, and video tracking. ME is usually regarded as the computationally most intensive part, performing up to 50% computations of the encoding system [1]. The most popular technique for ME is block matching algorithm (BMA) which is deployed in many video compression standards [2][3] because of its simplicity and effectiveness. In BMA, a frame is partitioned into a number of rectangular blocks and a motion vector for that block is estimated within its search range in the reference frame by finding the closest block of pixels according to a certain matching criterion such as the sum of absolute differences (SAD) or the sum of squared differences (SSD). The full search block matching algorithm (FSBMA) can give optimal estimation of motion in terms of minimal matching error by checking all the candidates within the search range, but the prohibitively huge computational complexity makes it impractical for real-time video applications. Thus, many techniques have been proposed to reduce the high computational complexity of the FSBMA.

The techniques that exploit different matching criteria instead of the classical sum of absolute differences (SAD) such as one-bit transform (1BT), multiplication-free 1BT, two-bit transform (2BT), constrained one-bit transform (C1BT), and TGC-BPM were proposed to make the faster computation of the matching criteria using Boolean exclusive-OR (XOR) operations [5][6][7][8]. In [5], 1BT-based ME where the reference frames and the current frames are transformed into one-bit representations by comparing the original image frame against a bandpass filtered output was proposed. After this transform, the matching error criterion between two one-bit image frames, which is called the number of non-matching points of 1BT ($NNMP_{1BT}$) is given by

$$NNMP_{1BT}(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\{B^t(i,j) \oplus B^{t-1}(i+m, j+n)\} \qquad (1)$$

where $B^t(i, j)$ and $B^{t-1}(i, j)$ are the 1BT representations of the current and the previous image frames, respectively, $\oplus$ denotes the Boolean XOR operation, the motion block size is $N\times N$, and $-s \le m,\ n \le s$ is the search range [5].

To reduce the computational complexity of calculating the 1BTs, the multiplication-free filter was also proposed in [6]. Although the 1BT-based motion estimation accomplishes a reduction in arithmetic and hardware complexity, the reconstructed image is degraded due to bad motion vectors resulting from the reduced bit-depth (particularly for small block sizes) [7]. A 2BT-based ME was proposed to enhance the ME accuracy of the 1BT-based ME algorithms [7]. In the 2BT-based ME, the values of local mean μ, variance σ^2, and the approximate standard deviation σ_a are used to convert frames into two-bit representations. The 2BT-based ME uses the number of non-matching points ($NNMP_{2BT}$) as a matching criterion given as :

$$NNMP_{2BT}(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\{B_1^t(i,j) \oplus B_1^{t-1}(i+m, j+n)\} \\ \| \{B_2^t(i,j) \oplus B_2^{t-1}(i+m, j+n)\} \qquad (2)$$

where $B_{1,2}^t(i, j)$ and $B_{1,2}^{t-1}(i, j)$ are the 2BT representations of the current and the previous image frames, respectively, $\|$ denotes the Boolean OR operation, the motion block size is $N\times N$, and $-s \le m,\ n \le s$ is the search range. The variations of

 ISBN: 978-1-61208-195-3

the 2BT-based matching criterion to increase the dynamic range of the matching criterion were proposed in [9]. These variations outperform the typical 2BT-based ME.

In [8], a constraint mask bitplane was introduced to improve the performance of 1BT, which is called the C1BT. Although C1BT-based ME uses two bitplanes in matching criterion similar to 2BT, it is very simple to create the constraint mask bitplane in C1BT. Note that for 2BT, the computational complexity of transforming frames into two-bit representation is relatively high because it involves multiplication operations. And in general, C1BT-based ME provides slightly better ME performance compared to the 2BT based ME. In C1BT, image frames are filtered using the multiplication-free 1BT filter in [6]. Then, the filtered image frames are compared to the original pixel values as in 1BT and the corresponding constraint mask is calculated as follows:

$$CM(i,j) = \begin{cases} 1, & if\ \left| I(i,j) - I_F(i,j) \right| \geq D \\ 0, & otherwise \end{cases} \tag{3}$$

where I and I_F are original and filtered image frames, respectively and D is a threshold. The corresponding matching error criterion is as follows :

$$\begin{aligned} & CNNMP_{original}(m,n) \\ & = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \{CM^t(i,j) \,||\, CM^{t-1}(i+m, j+n)\} \\ & \qquad \bullet \{B^t(i,j) \oplus B^{t-1}(i+m, j+n)\} \end{aligned} \tag{4}$$

where $B^t(i, j)$ and $B^{t\text{-}1}(i, j)$ are the 1BT representations of the current and the previous image frames, respectively. $CM^t(i, j)$ and $CM^{t\text{-}1}(i, j)$ are the constraint mask of the current and the previous image frames, respectively. $||$, $\oplus$ and $\bullet$ denote the Boolean OR, XOR, and AND operation, respectively. And the motion block size is $N\times N$, and $-s \leq m,\ n \leq s$ is the search range [8].

In this paper, we propose a low complexity multiple candidate motion estimation algorithm based on the C1BT. By exploiting the almost identical operations in two different matching error criteria, we can efficiently determine two best motion vectors according to the respective matching criteria and can enhance the overall motion estimation accuracy. The rest of this paper is organized as follows. Section 2 presents our proposed multiple candidate ME algorithm. Experimental results and analyses are provided in Section 3. Finally, Section 4 provides conclusions.

II. Proposed Algorithm

To improve the overall ME performance of the C1BT-based ME, we adopt the strategy in [10] of multiple candidate ME exploiting the similar operations between two different matching criteria. However, the matching error criterion of C1BT cannot be effectively splitted as in [10] because of Boolean AND operation. Note that because of this AND operation the C1BT matching criterion does not satisfy the metric requirements. Therefore we tested several matching error criterion as in [9] to find some substitutes whose operations are very similar to the C1BT matching criterion and whose performance is somewhat different in sequence to sequence. Among the many variations of the matching error criteria, the following two matching criteria show the similar ME performance as the C1BT matching criterion, we call it as a C1BT-extension and a C1BT-hybrid.

$$\begin{aligned} & CNNMP_{extension}(m,n) \\ & = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \{CM^t(i,j) \oplus CM^{t-1}(i+m, j+n)\} \\ & \qquad + \{B^t(i,j) \oplus B^{t-1}(i+m, j+n)\} \end{aligned} \tag{5}$$

$$\begin{aligned} & CNNMP_{hybrid}(m,n) \\ & = CNNMP_{original}(m,n) + CNNMP_{extension}(m,n) \end{aligned} \tag{6}$$

Table 1 shows the average PSNR performance of the C1BT, C1BT-hybrid and C1BT-extension when the motion block size is 16×16 and the search range is ±16. Note that for C1BT, the best performance was achieved when D = 10, however for other two variations, the best performance was achieved when D = 30. Of the variations of C1BT, the average performance of the C1BT-hybrid is slightly better than that of the C1BT and C1BT-extension. And as we can see from the Table 1, the average performance varies from sequence to sequence. For example, for sequence of “hall”, C1BT outperforms C1BT-hybrid by 0.74dB on average and for the other sequences C1BT-hybrid always outperforms C1BT.

TABLE I. AVERAGE PSNR RESULTS OF C1BT, C1BT-EXTENSION AND C1BT-HYBRID

	C1BT (D = 10)	**C1BT-extension (D = 30)**	**C1BT-hybrid (D = 30)**
stefan	25.23	25.31	25.39
akiyo	42.54	42.51	42.57
mobile	23.64	23.76	23.8
hall	33.98	33.17	33.24
coastguard	29.24	29.36	29.42
container	38.25	38.26	38.28
table	28.07	28.17	28.33
flower	25.78	25.83	25.88
average	**30.84**	**30.80**	**30.86**

To exploit those uneven performance differences, we propose to use multiple candidate motion search based on

TABLE II. AVERAGE PSNR RESULTS OF ALGORITHMS WHEN THE MOTION BLOCK SIZE IS 16×16 (SEARCH RANGE = ±16)

	FSBMA	C1BT	AM2BT	DCMCW2BT	Proposed
stefan	25.75	25.23	25.53 (113.42)	25.50 (0.24)	25.53 (0.34)
akiyo	42.84	42.54	42.60 (1.59)	42.59 (0.02)	42.79 (0.03)
mobile	23.92	23.64	23.72 (182.15)	23.8 (0.2)	23.86 (0.16)
hall	34.34	33.98	33.56 (17.26)	33.91 (0.22)	34.21 (0.36)
coastguard	29.62	29.24	29.43 (45.8)	29.46 (0.2)	29.51 (0.64)
container	38.33	38.25	38.13 (2.78)	38.22 (0.02)	38.33 (0.07)
table	28.87	28.07	28.37 (56.56)	28.45 (0.3)	28.54 (0.64)
flower	26.03	25.78	25.91 (218.39)	25.95 (0.1)	25.94 (0.44)
average	**31.21**	**30.84**	**30.91 (37.53)**	**30.99 (0.08)**	**31.09 (0.16)**

the C1BT and C1BT-hybrid (MCC1BT). Note that the matching error criteria of these two ME algorithms share many identical operations. The proposed algorithm is as follows:

1) Calculate the matching error criteria as in (4) and (6).
2) Find two best motion vectors according to the respective matching criteria.
3) If two best motion vectors are the same, declare it as the best motion vector for the current block and go to 5).
4) Calculate SADs of the two best motion vectors and declare the motion vector with less SAD as the best motion vector for the current block.
5) Go to the next current block.

Note that the calculations of SADs are needed only when two best motion vectors are different, which is very rare as will be seen in the experimental results.

III. EXPERIMENTAL RESULTS

The performance of the proposed algorithm (D = 30) was compared with the C1BT, the AM2BT [4], DCMCW2BT [10] and FSBMA using the metric of SAD. The first 100 frames of 8 CIF (352 × 288) sequences are used as test sequences. All the searching processes were in spiral order.

Table 2 and 3 show the average PSNR comparison results when the motion block size is 16×16 and the search range is ±16 and when the motion block size is 8×8 and the search range is ±8, respectively. The average numbers of SAD calculations per motion block for AM2BT, DCMCW2BT and the proposed algorithm are also shown in the Tables. Note that the maximum number of calculations of SADs in one motion block is two for comparison.

From the Tables, we can see that the performance of the proposed algorithm outperforms the other algorithms. To be specific, the average PSNR of the proposed algorithm is better than that of the C1BT by 0.25dB, that of the AM2BT by 0.18dB, and that of the DCMCW2BT by 0.10dB when the motion block size is 16×16 and the search range is ±16. The SAD calculations of the proposed algorithm are needed about 1 out of 12 motion blocks on average. Compared with the motion block size is 16×16 and the search range is ±16. The gap between the proposed algorithm and the FSBMA is within 0.12dB. And for the computational complexity increase, we can see that the calculations of SADs are needed about 1 out of 12 (≈2/0.16) motion blocks on average which is very small. Compared the AM2BT, the ratio between the proposed algorithm and the AM2BT is about 1 over 235 in terms of the number of SAD calculations when the motion block size is 16×16 and the search range is ±16. Also when the motion block size is 8×8 and the search range is ±8, the average PSNR of the proposed algorithm is better than that of the C1BT by 0.66dB, that of the AM2BT by 0.37dB, and that of the DCMCW2BT by 0.22dB.

TABLE III. AVERAGE PSNR RESULTS OF ALGORITHMS WHEN THE MOTION BLOCK SIZE IS 8×8 (SEARCH RANGE = ±8)

	FSBMA	C1BT	AM2BT	DCMCW2BT	Proposed
stefan	26.74	25.58	26.30 (44.83)	26.32 (0.3)	26.33 (0.5)
akiyo	43.48	42.71	42.36 (1.67)	42.61 (0.04)	43.22 (0.1)
mobile	24.83	23.89	24.25 (63.44)	24.45 (0.28)	24.56 (0.36)
hall	35.87	34.79	34.57 (4.58)	35.02 (0.24)	35.47 (0.55)
coastguard	30.68	29.43	30.05 (16.21)	30.14 (0.36)	30.19 (0.81)
container	38.42	37.87	37.99 (2.42)	38.02 (0.04)	38.37 (0.12)
table	30.55	28.94	29.71 (16.22)	29.79 (0.28)	29.85 (0.71)
flower	27.45	26.82	27.12 (58.43)	27.19 (0.12)	27.26 (0.44)
average	**32.25**	**31.25**	**31.54 (25.98)**	**31.69 (0.21)**	**31.91 (0.45)**

IV. CONCLUSION AND FUTURE WORK

A low complexity C1BT-based multiple candidate motion estimation algorithm was proposed in this paper. By exploiting almost the identical operations in two different matching error criteria, we can efficiently determine two best motion vectors according to the respective matching criteria and can enhance the overall motion estimation accuracy. Experimental results show that the proposed algorithm achieves PSNR gains about 0.25dB and 0.66dB on average when the motion block size is 16×16 and 8×8, respectively compared with the conventional C1BT-based motion estimation without noticeable complexity increase. Note that the PSNR difference between the proposed algorithm and the FSBMA using the metric of SAD is only 0.12 dB on average, which is very small when the motion block size is 16×16. For future work, we plan to find an efficient local search algorithm to enhance the overall ME accuracy.

REFERENCES

[1] Z. He, C. Tsui, K. Chan, and M. Liou, “Low-power VLSI design for motion estimation using adaptive pixel truncation,” *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 10, no. 5, pp. 669-678, Aug. 2000.

[2] *Information Technology - Coding of Audio Visual Objects - Part 2: Visual*, JTC1/SC29/WG11, ISO/IEC 14496-2 (MPEG-4 Visual), 2002.

[3] Advanced Video Coding for Generic Audiovisual Services, *ITU-T Recommendation H.264*, May 2005.

[4] B. Demir and S. Erturk, “Block Motion Estimation Using Adaptive Modified Two-bit Transform,” *IET Image Process.*, pp. 215-222, 2007.

[5] B. Natarajan, V. Bhaskaran, and K. Konstantinides, “Low-Complexity Block-based Motion Estimation via One-Bit Transforms,” *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 7, no. 5, pp. 702-706, Aug. 1997.

[6] Sarp Erturk, “Multiplication-Free One-Bit Transform for Low-Complexity Block-Based Motion Estimation,” *IEEE Signal Processing Letters*, vol. 14, no. 2, 109-112, Feb. 2007.

[7] A. Erturk and S. Erturk, “Two-Bit Transform for Binary Block Motion Estimation,” *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 15, no. 7, pp. 938-946, July 2005.

[8] O. Urhan and S. Erturk, “Constrained One-Bit Transform for Low Complexity Block Motion Estimation,” *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 17, no. 4, pp. 478-482, Apr. 2007.

[9] Changryoul Choi and Jechang Jeong, “Enhanced Two-bit Transform Based Motion Estimation via Extension of Matching Criterion,” *IEEE Trans. Consumer Electron.,* vol. 56, no. 3, pp. 1883-1889, Aug. 2010.

[10] Changryoul Choi and Jechang Jeong, “Low Complexity Weighted Two-bit Transforms based Multiple Candidate Motion Estimation,” *IEEE Trans. Consumer Electron.,* Nov. 2011.