Weighted Constrained One-Bit Transform Method for Low-Complexity Block Motion Estimation

Youngkyoung Choi, Hyungwook Kim, Sojeong Lim, and Sungwook Yu

This letter proposes a new low-complexity motion estimation method. The proposed method classifies various nonmatching pixel pairs into several categories and assigns an appropriate weight for each category in the matching stage. As a result, it can significantly improve performance compared to that of the conventional methods by adding only one 1-bit addition and two Boolean operations per pixel.

Keywords: Motion estimation, one-bit transform, two-bit transform, weighted transform.

I. Introduction

Motion estimation is an essential part of most video coding standards, such as H.264/AVC [1]. Although the full search (FS) method gives the best performance, it incurs enormous computational complexity. Thus, it is not suitable for low-complexity and low-power applications. As a result, the one-bit transform (1BT) method was proposed. The 1BT method greatly reduces the complexity [2], but it leads to very low peak signal-to-noise ratio (PSNR) values compared with those of the FS method.

To solve the problem of low PSNR, the two-bit transform (2BT) method [3] and the constrained 1BT (C1BT) method [4] were proposed. The 2BT method makes use of local means and variances to improve the motion estimation accuracy. The C1BT method uses the so-called constraint mask to exclude erroneous nonmatching points. Both the 2BT and C1BT methods use different matching criteria from the ones used in the original 1BT method.

Although both methods improve the performance of the 1BT method, they use the same weight for all kinds of nonmatching pixel pairs. This letter proposes a new method, which applies different weights in an efficient way to improve the performance significantly. The remainder of this letter is organized as follows. Section II briefly explains the conventional 1BT, C1BT, and 2BT methods, and section III explains the proposed method. Section IV compares the proposed method with the conventional methods in terms of PSNR performance and computational complexity. Finally, our conclusions are given in section V.

II. Conventional Methods

The 1BT method was proposed to reduce the high complexity of the FS method [2]. It uses a kernel K to filter the original image $I(i, j)$ and obtain a filtered image $I_F(i, j)$. The kernel used in the original 1BT is

$$K(i,j)=\begin{cases}1/25 & \text{if } \ i,j\in\{0,4,8,12,16\},\\ 0 & \text{otherwise.}\end{cases} \tag{1}$$

Some other kinds of kernels have also been proposed that avoid the multiplication operations that are required in the original kernel [5]. The filtered image is compared with the original image to obtain a binary image $B(m, n)$:

$$B(i,j)=\begin{cases}1 & \text{if } \ I(i,j)\ge I_F(i,j),\\ 0 & \text{otherwise.}\end{cases} \tag{2}$$

Instead of using the sum of absolute differences (SAD) measure, the 1BT method uses the number of nonmatching points (NNMP) measure, which is defined as

Manuscript received Mar. 5, 2012; revised June 11, 2012; accepted June 22, 2012.

This research was supported by the Chung-Ang University Research Scholarship Grants in 2010.

Youngkyoung Choi (phone: +82 2 820 5740, choimuyul@naver.com), Hyungwook Kim (ahinsharr@gmail.com), Sojeong Lim (lim.sojeong@gmail.com), and Sungwook Yu (corresponding author, sungwook@cau.ac.kr) are with the Electrical and Electronics Engineering Department, Chung-Ang University, Seoul, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.12.0212.0105

Table 1. Pixel classification method in C1BT method.

Group	Range	$B(i, j)$	$CM(i, j)$
G_1	$I(i, j) \geq I_F(i, j)+D$	1	1
G_2	$I_F(i, j) \leq I(i, j) < I_F(i, j)+D$	1	0
G_3	$I_F(i, j)-D < I(i, j) < I_F(i, j)$	0	0
G_4	$I(i, j) \leq I_F(i, j)-D$	0	1

$$NNMP(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\{B(i,j) \oplus B_{\text{ref}}(i+m, j+n)\}, \quad -s \leq m,\ n \leq s, \qquad (3)$$

where $B(\cdot)$ and $B_{\text{ref}}(\cdot)$ represent the current and reference binary images, respectively, $\oplus$ represents the XOR operation, (m, n) represents the motion vector, and s determines the search range. Although the 1BT method greatly reduces the complexity, it leads to very low PSNR values compared with those of the FS method.

As a result, the 2BT and C1BT methods were proposed to improve the performance of the 1BT method [2], [3]. Although there are some differences, the 2BT and C1BT methods are similar to each other in that both methods convert the original 8-bit images into 2-bit images. For example, the C1BT method classifies 8-bit pixels into four different groups and defines the constraint mask $CM(i, j)$ according to the method shown in Table 1. In this table, D is a parameter, which is determined experimentally. The 2BT method requires a similar transformation stage, although it uses the block mean μ and standard deviation σ instead of $I_F(i, j)$ and D.

III. Proposed Method

The proposed method also classifies 8-bit pixels into four different groups. We will use $I_F(i, j)$ and D although the block mean μ and standard deviation σ may also be used. Also, we will use the multiplication-free kernel described in [5]. The proposed method, however, uses different matching criteria from the ones used in the C1BT and 2BT methods. The matching criteria in the proposed and conventional methods can be easily explained by comparing Tables 2, 3, and 4. For convenience, we will use (G_i, G_j) to represent the case where the pixels in the current and reference binary images are in the groups G_i and G_j, respectively.

In Tables 2 and 3, we can first see that the 1BT, C1BT, and 2BT methods use very similar criteria. For example, the case (G_1, G_1) is regarded as a match in all three of the methods. On the other hand, the case (G_1, G_4) is regarded as a nonmatch in all the methods. However, some cases are regarded as

Table 2. Matching criteria in 1BT/C1BT methods.

		Reference image			
		G_1	G_2	G_3	G_4
Current image	G_1	O	O	X	X
	G_2	O	O	X/O	X
	G_3	X	X/O	O	O
	G_4	X	X	O	O

Table 3. Matching criteria in 2BT method.

		Reference image			
		G_1	G_2	G_3	G_4
Current image	G_1	O	X	X	X
	G_2	X	O	X	X
	G_3	X	X	O	X
	G_4	X	X	X	O

Table 4. Matching criteria in proposed method.

		Reference image			
		G_1	G_2	G_3	G_4
Current image	G_1	O	Z_1	X_1	X_2
	G_2	Z_1	O	Z_2	X_1
	G_3	X_1	Z_2	O	Z_1
	G_4	X_2	X_1	Z_1	O

nonmatches in some methods, but as matches in the other methods. For example, the case (G_1, G_2) is regarded as a nonmatch only in the 2BT method and the case (G_2, G_3) is regarded as a match only in the C1BT method.

Table 4 shows the matching criteria of the proposed method. As shown, the proposed method categorizes the 16 cases into three groups.

Group 1: the matching cases marked with O
Group 2: the nonmatching cases marked with X (both X_1 and X_2)
Group 3: the nonmatching cases marked with Z (both Z_1 and Z_2)

The problem in the conventional methods is the fact that they use the same weight for both the Z and X cases in the NNMP computation, even though Z cases are clearer matches than X cases. Thus, the proposed method uses the weighted NNMP (WNNMP), which assigns different weights for different nonmatching cases as follows:

$$WNNMP(m,n) = w_1 \times NNMP_Z(m,n) + w_2 \times NNMP_X(m,n), \qquad (4)$$

where $NNMP_Z(m, n)$ and $NNMP_X(m, n)$ represent the number of points that belong to the groups Z and X, respectively. Since the Z cases are closer to the matching cases O, w_1 should be smaller than w_2. We will call the proposed method the weighted C1BT (WC1BT) method since it uses the same pixel classification process.

Although (4) can be used in its present form, the Boolean expression required in $NNMP_Z(m, n)$ is very complicated since the cases in Group Z are not grouped together. Thus, we modify (4) to obtain

$$\begin{aligned} WNNMP(m,n) &= w_1 \times NNMP_{Z \text{ or } X}(m,n) \\ &\quad + (w_2 - w_1) \times NNMP_X(m,n), \end{aligned} \tag{5}$$

where the $NNMP_{Z \text{ or } X}(m, n)$ and $NNMP_X(m, n)$, after some simplification processes, are given as

$$\begin{aligned} &NNMP_{Z \text{ or } X}(m,n) \\ &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \left\{ \begin{aligned} &[B(i,j) \oplus B_{\text{ref}}(i+m, j+n)] \\ &\| [CM(i,j) \oplus CM_{\text{ref}}(i+m, j+n)] \end{aligned} \right\}, \end{aligned} \tag{6}$$

$$\begin{aligned} &NNMP_X(m,n) \\ &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \left\{ \begin{aligned} &[B(i,j) \oplus B_{\text{ref}}(i+m, j+n)] \\ &\bullet [CM(i,j) \,\|\, CM_{\text{ref}}(i+m, j+n)] \end{aligned} \right\}. \end{aligned} \tag{7}$$

It should be mentioned that (6) and (7) require non-Boolean 1-bit addition operations, which can be efficiently implemented in several ways, including the look-up table method proposed in [2]. The cases Z and X may be subcategorized even further. In other words, we can assign a different weight for each of the cases Z_1, Z_2, X_1, and X_2 as follows:

$$\begin{aligned} &WNNMP(m,n) \\ &= w_{11} \times NNMP_{Z1}(m,n) + w_{12} \times NNMP_{Z2}(m,n) \\ &\quad + w_{21} \times NNMP_{X1}(m,n) + w_{22} \times NNMP_{X2}(m,n). \end{aligned} \tag{8}$$

However, this modification results in little PSNR changes in our simulation. Thus, considering the two extra NNMP calculations required in (8), the subcategorization does not seem to be a good tradeoff between the performance and complexity. A more detailed evaluation of the proposed method in terms of the performance and complexity will be given in the next section.

IV. Comparison with Conventional Methods

Table 5 compares several motion estimation methods in terms of PSNR performance. It is assumed that the macroblock size N is 16 and the search range s is also 16. In the proposed WC1BT method, the ratio of w_2 to w_1 is important, rather than the exact values of w_1 and w_2. In general, the WC1BT shows the best performance when w_2/w_1=5, as can be seen in Table 5. Thus, we may use w_1=1 and w_2=5 in (5). The selection of D (in Table 1) is also important. In general, the WC1BT method shows the best results when D=25, as can be seen in Table 5, although the optimal D values are different for different video sequences. The selection of D is also important in the C1BT method, and the optimal value for D was 10 in [4]. On average, the WC1BT (D=25) method increases the PSNR of the 2BT

Table 5. Average PSNR values (dB) of various motion estimation methods.

	Football (288×352) (260 frames)	News (288×352) (300 frames)	Bus (288×352) (150 frames)	Coastguard (288×352) (300 frames)	Foreman (288×352) (300 frames)	Silent (288×352) (300 frames)
FS (SAD)	25.67	36.90	25.02	30.48	32.12	35.97
1BT [2]	23.51	33.63	23.74	29.91	30.48	34.40
2BT [3]	24.30	35.88	24.35	29.93	30.71	34.74
C1BT [4] (D=10)	24.02	36.09	24.35	30.04	31.02	34.85
WC1BT (w_2/w_1=3, D=25)	24.37	36.46	24.62	30.27	31.39	35.01
WC1BT (w_2/w_1=5, D=25)	24.35	36.49	24.62	30.30	31.38	35.03
WC1BT (w_2/w_1=7, D=25)	24.31	36.43	24.61	30.30	31.35	34.98
WC1BT (w_2/w_1=5, D=15)	24.36	36.42	24.51	30.19	31.43	35.09
WC1BT (w_2/w_1=5, D=20)	24.39	36.46	24.58	30.25	31.43	35.08
WC1BT (w_2/w_1=5, D=25)	24.35	36.49	24.62	30.30	31.38	35.03
WC1BT (w_2/w_1=5, D=30)	24.30	36.46	24.65	30.33	31.33	34.94
WC1BT (w_2/w_1=5, D=35)	24.22	36.44	24.66	30.34	31.29	34.85
$\text{WC1BT}_{\text{adaptive}}$ (w_2/w_1=5)	24.39	36.46	24.69	30.36	31.40	35.17

Table 6. Number of operations (per pixel) of various motion estimation methods.

	Transform						Matching				Memory
	Addition (pp)	Multiplication (pp)	Shift (pp)	Subtraction (pp)	Comparison (pp)	Boolean operation (pp)	Boolean operation (pp)	1-bit addition (pp)	Addition (pp)	Shift (pp)	Bit (pp)
1BT [2]	25	1	-	-	1	-	1	1	-	-	1
MF1BT [5]	16	-	1	-	1	-	1	1	-	-	1
2BT [3]	2.8125	1.0625	-	0.03125	3	1	3	1	-	-	2
C1BT [4]	16	-	1	1	2	-	3	1	-	-	2
WC1BT	16	-	1	1	2	-	5	2	0.004	0.004	2

by 0.38 dB and that of the C1BT (D=10) by 0.30 dB, as can be seen in Table 5.

It should be mentioned that the D value can be set in an adaptive way for additional PSNR improvement. For example, [6] uses the standard deviation σ to determine the threshold value adaptively. The $\text{WC1BT}_{\text{adaptive}}$ method in the last line of Table 5 represents the WC1BT method with an adaptive threshold technique. Since the optimal D values of the C1BT and the WC1BT are different, we slightly modify the threshold equation in [6] and use D=2.5×average(σ/2, 10). As shown in Table 5, this combination leads to an additional 0.05 dB increase in the PSNR value at the expense of additional complexity required for the computation of the standard deviation.

Table 6 compares various motion estimation methods in terms of their respective necessary number of operations. It is important to note that most of the values in Table 6 have been taken from [4]. When compared with the 2BT and C1BT methods, the WC1BT method requires only two more Boolean operations and one more 1-bit addition per pixel, ignoring minor overhead in the matching stage. Both Boolean operations and 1-bit additions are relatively simple operations when compared with full-bit additions, subtractions, and comparison operations, which are required in the transformation stage. Thus, it can be seen that the complexity increase is negligible.

V. Conclusion

In this letter, we proposed a new low-complexity motion estimation method. By classifying various nonmatching pixel pairs into several categories and assigning an appropriate weight for each category, the proposed method can increase the PSNR of the 2BT and C1BT methods by 0.38 dB and 0.30 dB, respectively. We also minimized the complexity overhead by efficiently reorganizing the WNNMP equation to obtain simple Boolean expressions. Thus, the proposed method can provide a useful solution for various low-complexity and low-power applications.

References

[1] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)," JVT-G050, Mar. 2003.

[2] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-Complexity Block-Based Motion Estimation via One-Bit Transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 4, Aug. 1997, pp. 702-706.

[3] A. Ertürk and S. Ertürk, "Two-Bit Transform for Binary Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, July 2005, pp. 938-946.

[4] O. Urhan, "Constrained One-Bit Transform for Low Complexity Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, Apr. 2007, pp. 478-482.

[5] S. Ertürk, "Multiplication-Free One-Bit Transform for Low-Complexity Block-Based Motion Estimation," *IEEE Signal Process. Lett.*, vol. 14, no. 2, Feb. 2007, pp. 109-112.

[6] F.K. Chen et al., "Dynamically Constrained One-Bit Transform for Motion Vector Estimation," *Proc. IEEE Int. Conf. Inf. Assurance Security*, Aug. 2009, pp. 375-378.