Modified one-bit transform method with dynamic search range

Sojeong Lim
Jungwoo Kim
Sungwook Yu

Modified one-bit transform method with dynamic search range

Sojeong Lim
Jungwoo Kim
Sungwook Yu
Chung-Ang University
School of Electrical and Electronics Engineering
221 HeukSeok-Dong DongJak-Gu Seoul 156-756, Korea
E-mail: sungwook@cau.ac.kr

Abstract. *Here, a new low-complexity block motion estimation method is presented. The proposed method significantly reduces the computational complexity by checking only one of two search areas in the conventional hybrid method when the two search centers are close to each other. It also improves the performance by efficiently making use of the motion vector information of the co-located macroblock. The proposed techniques can be easily applied to many hybrid motion estimation methods. © The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: 10.1117/1.JEI.22.2 .023001]*

1 Introduction

Motion estimation is an essential part of most video coding standards. Although the full search (FS) method gives the best performance, it incurs enormous computational complexity. Thus, many low-complexity motion estimation algorithms have been proposed, including the one-bit transform (1BT), two-bit transform (2BT), and constrained 1BT (C1BT) methods.[1–3] These low-complexity methods greatly reduce the complexity, but they lead to very low peak signal-to-noise ratio (PSNR) values. In order to solve this problem, many hybrid methods have been proposed that combine the low-complexity methods with the FS method. For example, the modified 1BT (M1BT) method combines the 1BT method with the FS method.[4] The modified 2BT (M2BT) and modified C1BT (MC1BT) methods also combine low-complexity methods with the FS method.[5,6] This paper proposes an efficient search scheme that not only reduces the complexity, but also improves the performance of these conventional hybrid methods.

2 Conventional Hybrid Methods

The FS method is based on the sum of absolute differences (SAD) measure, which is defined as follows:

$$\text{SAD}(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |I(i, j) - I_{\text{ref}}(i + m, j + n)|, \quad (1)$$

where $I(\cdot)$ and $I_{\text{ref}}(\cdot)$ represent the current and reference images, respectively, N is the size of a macroblock, and (m, n) represents the motion vector. The 1BT method has been proposed to reduce the high computational complexity of the FS method.[1] It uses a filter kernel to filter the original image $I(i, j)$ and to obtain a filtered image $I_F(i, j)$.[7,8] The filtered image is compared with the original image to obtain a binary image $B(i, j)$ as follows:

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

Instead of using the SAD measure, the 1BT method uses the number of non-matching points (NNMP) measure, which is defined as follows:

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

where $B(\cdot)$ and $B_{\text{ref}}(\cdot)$ represent the current and reference binary images, respectively, $\bigoplus$ represents the 1-bit XOR operation, and s determines the search range. The 2BT and C1BT methods are also based on 1-bit operations although they use slightly different NNMP measures.[2,3] Although the 1BT, 2BT, and C1BT methods are all low-complexity solutions, they lead to relatively very low PSNR values when compared with the FS method.

Thus, Ref. 4 proposed a hybrid method, called the M1BT algorithm, which combines the low-complexity 1BT algorithm with the high-accuracy FS algorithm. In the first stage of the M1BT method, two motion vectors, mv_1 and mv_2, are selected, which have the lowest and second lowest NNMP values. Then, the SAD of the motion vector mv_1 is computed. If it is less than a threshold value T_1, mv_1 is selected as the final motion vector. Otherwise, the SAD of the second motion vector (mv_2) is computed. If it is smaller than T_1, mv_2 is selected as the final motion vector. If not, the search process goes on to the second stage.

The second stage consists of two steps, as shown in Fig. 1. In the first step of the second stage, the SAD values of four additional points (in addition to the center point) are computed, and then the point with the minimum SAD is selected

Fig. 1 Two-step search in the M1BT method.

Table 1 Euclidean distance between mv_1 and mv_2 in M1BT.

ED(mv_1, mv_2)	1	2	3	4	5	6+
Probability	75.1%	8.9%	2.4%	1.5%	1.2%	10.9%

proposed method will check only the search points around mv_X, where mv_X represents either mv_1 or mv_2, whichever has a smaller SAD value (it should be noted that the SAD of mv_2 may be smaller than that of mv_1, even though the NNMP of mv_1 is always smaller than that of mv_2). This new scheme will significantly decrease the number of search points at the expense of slight PSNR degradation, and the simulation results will be given in the next section.

The M1BT method performs well when either mv_1 or mv_2 is located near mv_{FS}, where mv_{FS} represents the optimal motion vector determined by the FS method. This is usually the case, but sometimes, both ED(mv_1, mv_{FS}) and ED(mv_2, mv_{FS}) can become quite large. On the other hand, all of the search points in the M1BT method are located within very small areas around mv_1 and mv_2. To be more precise, for every search point X around mv_1, ED(mv_1, X) is always less than or equal to 4, as can be seen in Fig. 1. The same is true for mv_2 and every search point around mv_2. Thus, when both ED(mv_1, mv_{FS}) and ED(mv_2, mv_{FS}) are quite large, ED($mv_{M1BT} mv_{FS}$) will also be quite large, leading to significant PSNR degradation, where mv_{M1BT} represents the final motion vector determined by the M1BT method.

After analyzing diverse simulation results, we found that ED(mv_1, mv_{FS}) and ED(mv_2, mv_{FS}) are quite large when the area around the current macroblock contains high motion. Thus, the proposed method will adaptively determine d_1 and d_2 (in Fig. 1), based on the degree of motion (it should be mentioned that d_1 and d_2 are fixed to 2 and 1, respectively, in the original M1BT method). In order to account for the degree of motion, we use the information of the co-located macroblock (i.e., the macroblock at the same position in the previous frame) as follows:

$$d_1 = \text{median}[\text{LB}, k_1(|mv_x| + |mv_y|), \text{UB}], \quad (5)$$

where (mv_x, mv_y) represents the motion vector of the co-located macroblock, k_1 is a scaling constant, and LB and UB represent the lower and upper bounds for d_1. The values of k_1, LB, and UB in Eq. (5) have been determined to be 6, 3, and 24, respectively, based on extensive simulation results. After d_1 is decided, we will determine d_2 (i.e., the ED between search points) in such a way that all of the search points are located evenly within the search area. Thus, we determine d_2 as follows:

$$d_2 = \text{round}(d_1/3). \quad (6)$$

It should be emphasized that the proposed dynamic search range method does not increase the number of search points even if d_1 and d_2 are larger than the original values. Instead, it just checks a different set of search points. In other words, the proposed method skips some of the search points around the search center, but checks some additional search points that are far from the search center (when d_1 and d_2 are large). Finally, it should be noted that there are some frames that do not have motion vector information for the co-located
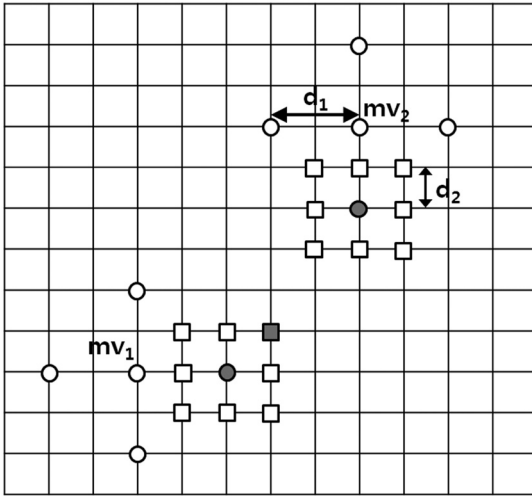
as the center point for the second step. In the second step, the SAD values of eight additional points (i.e., the square-shaped points in Fig. 1) are computed. The two steps are applied for mv_1 and mv_2, and the point with the minimum SAD is selected as the final motion vector.

The M1BT method leads to a good tradeoff between performance and complexity, and as a result, several similar approaches have been proposed. For example, the 2BT and C1BT methods have been combined with the FS method to yield the M2BT[5] and MC1BT[6] methods, respectively.

3 Proposed Method

The SAD computation in Eq. (1) requires relatively complex full-bit operations, whereas the NNMP computation in Eq. (3) uses only simple 1-bit operations. In order to reduce the complexity of the SAD operation, it is possible to use some subsampled SAD (SSAD) measures as follows:[4]

$$D_1 = \sum_{(i \bmod 2) = (j \bmod 2)} \sum |I(i,j) - I_{\text{ref}}(i+m, j+n)|$$

$$D_2 = \sum_{(i \bmod 2) = 0, (j \bmod 2) = 0} \sum |I(i,j) - I_{\text{ref}}(i+m, j+n)|. \quad (4)$$

The SSAD computation, however, is still much more complex than the NNMP computation. Thus, it is important to reduce the number of search points that require SAD (or SSAD) computations.

It is possible to use only one of mv_1 or mv_2 in order to reduce the number of search points, but this will lead to considerable performance degradation. However, after analyzing extensive simulation results, we found that mv_1 and mv_2 are, in most cases, located very close to each other. Table 1 shows the probability distribution of Euclidean distance (ED) between mv_1 and mv_2 for various video sequences. As can be seen from the table, ED(mv_1, mv_2) is very small in most cases.

Here, it is important to note that the search areas around mv_1 and mv_2 mostly overlap when ED(mv_1, mv_2) is very small. Thus, the proposed method will search only one search area without checking the other one if ED(mv_1, mv_2) is less than or equal to 4. To be more precise, the

Table 2 Comparison with the 1BT and M1BT methods.

Video sequence (format)	1BT PSNR (dB)	M1BT PSNR (dB)	M1BT # of search points	DM1BT$_1$ PSNR (dB)	DM1BT$_1$ # of search points	DM1BT$_1$ Hit Rate	DM1BT$_2$ PSNR (dB)	DM1BT$_2$ # of search points	DM1BT PSNR (dB)	DM1BT # of search points	DM1BT Hit rate
Foreman (CIF)	30.68	30.89	704,695 (100%)	30.89	445,860 (63.3%)	99.2%	31.34	692,244 (98.2%)	31.33	439,067 (62.3%)	98.3%
Bus (CIF)	23.78	24.14	815,624 (100%)	24.13	513,091 (62.9%)	98.8%	24.52	789,266 (96.8%)	24.51	497,513 (61.0%)	97.3%
News (CIF)	35.40	36.02	281,999 (100%)	36.01	209,170 (74.2%)	99.8%	36.22	281,908 (100.0%)	36.19	209,120 (74.2%)	99.5%
Football (CIF)	23.45	24.30	1,349,431 (100%)	24.28	814,486 (60.4%)	96.9%	25.15	1,313,936 (97.4%)	25.09	794,950 (58.9%)	92.7%
Tempete (CIF)	26.14	26.47	1,324,429 (100%)	26.46	760,006 (57.4%)	99.3%	26.59	1,318,707 (99.6%)	26.58	756,909 (57.1%)	98.4%
Carphone (QCIF)	29.94	30.45	203,708 (100%)	30.44	132,248 (64.9%)	99.3%	30.76	195,403 (95.9%)	30.73	127,398 (62.5%)	98.8%

macroblocks. For example, the first two frames in a video sequence with an IPPP... structure are such cases. For frames like this, $d_1 = 9$ and $d_2 = 3$ will be used, which are values that have also been determined based on extensive simulation results. We will call the proposed method the dynamic M1BT (DM1BT) method since the number of search points and the search area are dynamically determined.

4 Comparison with the Conventional Methods

Table 2 compares the proposed DM1BT method with the conventional 1BT and M1BT methods in terms of the PSNR and the total number of search points that require SSAD computation. Both the macroblock size N and the search range s were set to 16, and the D_2 SSAD measure in Eq. (4) was used in the simulation. As mentioned in the previous section, the DM1BT method is based on two techniques. First, it reduces the complexity by checking only one of two neighboring search areas. Second, it improves

the performance by adaptively deciding d_1 and d_2. In order to examine the effect of each technique, Table 2 shows three kinds of results for the proposed method, where the DM1BT$_1$ method uses only the first technique, the DM1BT$_2$ method uses only the second technique, and the DM1BT method uses both techniques.

First of all, we can see that the DM1BT$_1$ method uses a much smaller number of search points than M1BT. As mentioned, this is because the DM1BT$_1$ method checks only one of the two search areas when the distance between mv_1 and mv_2 is small. On average, the DM1BT$_1$ method reduces the number of search points by 36.1%. On the other hand, the PSNR degradation is negligible. This is because the final motion vectors of DM1BT$_1$ are, in most cases, the same as those of M1BT. The hit rate in Table 2 represents the probability that the final motion vectors of DM1BT$_1$ and M1BT are the same. As can be seen, the hit rates of the DM1BT$_1$ method for various test sequences are quite high. Table 2 also shows that the DM1BT$_2$ method significantly improves the

Table 3 Comparison with the 2BT and M2BT methods.

Video sequence (format)	2BT PSNR (dB)	M2BT PSNR (dB)	M2BT # of search points	DM2BT$_1$ PSNR (dB)	DM2BT$_1$ # of search points	DM2BT$_1$ Hit rate	DM2BT$_2$ PSNR (dB)	DM2BT$_2$ # of search points	DM2BT PSNR (dB)	DM2BT # of search points	DM2BT Hit rate
Foreman (CIF)	30.71	31.34	720,754 (100%)	31.32	453,415 (62.9%)	99.1%	31.39	706,998 (98.1%)	31.36	445,867 (61.9%)	98.3%
Bus (CIF)	24.35	24.52	833,291 (100%)	24.52	523,829 (62.9%)	98.6%	24.66	807,013 (96.8%)	24.64	508,556 (61.0%)	96.9%
News (CIF)	35.88	36.35	283,032 (100%)	36.35	209,693 (74.1%)	99.7%	36.38	282,950 (100.0%)	36.35	209,653 (74.1%)	99.5%
Football (CIF)	24.30	24.96	1,353,316 (100%)	24.94	813,979 (60.1%)	96.7%	25.25	1,317,692 (97.4%)	25.19	794,374 (58.7%)	93.6%
Tempete (CIF)	26.41	26.62	1,306,869 (100%)	26.62	745,594 (57.1%)	99.4%	26.66	1,301,947 (99.6%)	26.64	743,050 (56.9%)	98.8%
Carphone (QCIF)	30.51	30.90	199,361 (100%)	30.90	128,967 (64.7%)	97.8%	30.97	192,367 (96.5%)	30.96	124,952 (62.7%)	98.7%

Table 4 Comparison with the C1BT and MC1BT methods.

Video Sequence (Format)	C1BT PSNR (dB)	MC1BT PSNR (dB)	MC1BT # of Search points	DMC1BT$_1$ PSNR (dB)	DMC1BT$_1$ # of Search points	DMC1BT$_1$ Hit rate	DMC1BT$_2$ PSNR (dB)	DMC1BT$_2$ # of Search points	DMC1BT PSNR (dB)	DMC1BT # of Search points	DMC1BT Hit rate
Foreman (CIF)	31.02	31.40	699,790 (100%)	31.39	442,384 (63.2%)	99.3%	31.62	687,784 (98.3%)	31.60	435,762 (62.3%)	98.4%
Bus (CIF)	24.35	24.48	807,258 (100%)	24.47	508,174 (63.0%)	99.1%	24.69	781,731 (96.8%)	24.67	492,953 (61.1%)	97.7%
News (CIF)	36.09	36.43	275,743 (100%)	36.42	206,088 (74.7%)	99.8%	36.49	275,658 (100.0%)	36.48	206,042 (74.7%)	99.6%
Football (CIF)	24.02	24.61	1,350,827 (100%)	24.60	823,101 (60.9%)	97.1%	25.22	1,317,920 (97.6%)	25.17	804,675 (59.6%)	93.7%
Tempete (CIF)	26.47	26.63	1,301,353 (100%)	26.63	744,666 (57.2%)	99.5%	26.67	1,296,774 (99.6%)	26.67	742,301 (57.0%)	98.9%
Carphone (QCIF)	30.22	30.61	196,581 (100%)	30.61	127,794 (65.0%)	99.4%	30.95	188,765 (96.0%)	30.94	123,171 (62.7%)	99.0%

PSNR performance of the M1BT method. As explained in the previous section, this is because the DM1BT$_2$ method adaptively increases the search range when a video sequence contains high motion. It should be noted that the DM1BT$_2$ method uses a slightly smaller number of search points than the M1BT method. This is because some of the candidate search points in the DM1BT$_2$ method go beyond the boundary of the frame when d_1 and d_2 are large.

As expected, we can see that the DM1BT method, which is based on both techniques, not only decreases the number of search points, but also improves the PSNR performance. On average, the DM1BT method increases the PSNR of M1BT by 0.36dB and reduces the number of search points by 37.3%. It can also be seen that the hit rates of DM1BT (as compared with DM1BT$_2$) are very high although they are slightly smaller than the hit rates of DM1BT$_1$ (as compared with M1BT).

Finally, it should be mentioned that the proposed search scheme can be easily applied to other hybrid search methods, such as the M2BT[5] and MC1BT[6] methods, which also use two search centers and fixed search ranges. Table 3 compares the proposed DM2BT method with the conventional 2BT and M2BT methods, whereas Table 4 compares the proposed DMC1BT method with the conventional C1BT and MC1BT methods. As can be seen, the DM2BT and DMC1BT methods also efficiently enhance the conventional M2BT and MC1BT methods in terms of both PSNR performance and computational complexity.

5 Conclusions

We proposed a new low-complexity block motion estimation method. Using the fact that the two search centers are closely located in most cases, the proposed method significantly reduces the number of search points and hence the overall complexity. It also improves the PSNR performance by using an adaptive search scheme based on the motion vector information of the co-located macroblock. The proposed search scheme can be easily applied to many hybrid motion estimation methods.

References

1. B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-complexity block-based motion estimation via one-bit transform," *IEEE Trans. Circ. Syst. Video Tech.* **7**(4), 702–706 (1997).
2. A. Ertürk and S. Ertürk, "Two-bit transform for binary block motion estimation," *IEEE Trans. Circ. Syst. Video Tech.* **15**(7), 938–946 (2005).
3. O. Urhan, "Constrained one-bit transform for low complexity block motion estimation," *IEEE Trans. Circ. Syst. Video Tech.* **17**(4), 478–482 (2007).
4. P. H. W. Wong and O. C. Au, "Modified one-bit transform for motion estimation," *IEEE Trans. Circ. Syst. Video Tech.* **9**(7), 1020–1024 (1999).
5. B. Demir and S. Ertürk, "Block motion estimation using modified two-bit transform," *Lect. Notes Comput. Sci.* **4263**, 522–531 (2006).
6. H. Y. Oh et al., "Modified constrained one-bit transform based fast block motion estimation," *IEEE Trans. Consum. Electron.* **53**(3), 1093–1097 (2007).
7. O. Urhan, "Constrained one-bit transform-based motion estimation using predictive hexagonal pattern," *J. Electron. Imag.* **16**(3), 033019 (2007).
8. S. Ertürk, "Multiplication-free one-bit transform for low-complexity block-based motion estimation," *IEEE Signal Proc. Lett.* **14**(2), 109–112 (2007).

Sojeong Lim received her BS degree in electrical and electronics engineering from Chung-Ang University, Seoul, Korea in 2012. She is currently working toward an MS degree in electrical and electronics engineering from Chung-Ang University. Her research interests include video compression algorithms for H.264/AVC and HEVC.

Jungwoo Kim received his BS degree in electrical and electronics engineering from Chung-Ang University, Seoul, Korea in 2012. He is currently working toward an MS degree in electrical and electronics engineering from Chung-Ang University. His research interests include video compression algorithms for H.264/AVC and HEVC.

Sungwook Yu received his BS degree in electrical engineering from Seoul National University, Seoul, Korea, in 1992. He received his MS and PhD degrees in electrical and computer engineering from the University of Texas at Austin, in 1996 and 2000, respectively. From 1999 to 2000, he worked at SiLogiX in Austin, TX, and from 2000 to 2004, he worked at Intel Corp. in Austin, TX. He worked for a year at Samsung Semiconductor Inc. in Korea before joining the faculty of electrical and electronics engineering department in Chung-Ang University, Seoul, Korea in 2005. His research interests are in the area of ASIC design for image and video processing applications.