

움직임 추정을 위한 새로운 블록 정합 알고리즘

정 수 목*

A New Block Matching Algorithm for Motion Estimation

Soo-Mok Jung*

■ Abstract ■

In this paper, an efficient block matching algorithm which is based on the Block Sum Pyramid Algorithm (BSPA) is presented. The cost of BSPA[1] was reduced in the proposed algorithm by using l_2 norm and partial distortion elimination technique. Motion estimation accuracy of the proposed algorithm is equal to that of BSPA. The efficiency of the proposed algorithm was verified by experimental results.

Keyword : Block Sum Pyramid Algorithm, Motion Estimation, Motion Vector, l_2 Norm

1. Introduction

In image sequence coding, the correlation between consecutive frames can be reduced by the motion estimation and motion compensation technique[2]. Motion estimation plays an important role in reducing the bit rates for transmission or storage of video signals. The accuracy and efficiency of motion estimation affects the efficiency of temporal redundancy removal.

Motion estimation methods are classified into two classes of block matching algorithms (BMA)[2][3] and pel-recursive algorithms (PRA)[4]. Owing to their implementation simplicity, block matching algorithms have been widely adopted by various video coding standards such as CCITT H.261[5], ITU-T H.263[6], and MPEG [7]. In BMA, the current image frame is partitioned into fixed-size rectangular blocks. The motion vector for each block is estimated by

* 삼육대학교 컴퓨터학과

finding the best matching block of pixels within the search window in the previous frame according to matching criteria.

Although Full Search algorithm (FSA) finds the optimal motion vector by searching exhaustively for the best matching block within the search window, its high computation cost limits its practical applications. To reduce computation cost of FSA, many fast block matching algorithms such as three step search[8], 2-D log search[3], orthogonal search[9], cross search[10], one-dimensional full search[11], variation of three-step search[12, 13], unrestricted center-biased diamond search[14] etc. have been developed. As described in[15], these algorithms rely on the assumption that the motion compensated residual error surface is a convex function of the displacement motion vectors, but this assumption is rarely true[16]. Therefore, the best match obtained by these fast algorithms is basically a local optimum.

Without this convexity assumption, Successive Elimination Algorithm (SEA) proposed by Li and Salari[17] reduces the computation cost of the FSA. To reduce the computation cost of SEA, Block Sum Pyramid Algorithm (BSPA)[1] and Multilevel Successive Elimination Algorithm (MSEA)[18] were proposed. Our research team proposed Efficient Multilevel Successive Elimination Algorithms for Block Matching Motion Estimation (EMSEA)[19] to reduce the computation cost of MSEA. In New Block Matching Algorithm for Motion Estimation (NBMA), the l_2 norm[20, 21, 22] was used to reduce the computation cost of BSPA. Also, the partial distortion elimination (PDE) scheme of EMSEA was improved and then the improved scheme was applied to BSPA in NBMA. The motion

estimation accuracy of NBMA is identical to that of FSA and the computation cost of BSPA is reduced by using NBMA.

2. Block Sum Pyramid Algorithm

The SEA achieves the same estimation accuracy as the FSA while requiring less computation time. In SEA, the displacement vector of the corresponding block in the previous frame is used as the initial motion vector for the present template block[23]. The SEA uses the sum norm of a block as a feature to eliminate unnecessary search points. The sum norm of a block B of size $N \times N$ is defined as

$$S_B = \sum_{i=1}^N \sum_{j=1}^N |B(i, j)| \quad (1)$$

where $B(i, j)$ is the gray level of the (i, j) th pixel of block B . Let S_T be the sum norm of the template block T , S_X be the sum norm of a candidate matching block X , and $curr_MAD_{min}$ be the current minimal MAD during the search process. Let $MAD(T, X)$ be the MAD between T and X and is defined as

$$MAD(T, X) = \sum_{i=1}^N \sum_{j=1}^N |T(i, j) - X(i, j)| \quad (2)$$

where $T(i, j)$ and $X(i, j)$ represent the gray values of the (i, j) th pixels of T and X . The authors had shown that the following inequality is true in [17]:

$$\begin{aligned} MAD(T, X) &= \|X - Y\|_1 \geq \|X\|_1 - \|Y\|_1 \\ &= \sum_{i=1}^N \sum_{j=1}^N |S_T - S_X| \end{aligned} \quad (3)$$

Based on the above inequality, the SEA

discards each candidate matching block X with $|S_T - S_X| \geq \text{curr_MAD}_{\min}$, which can save a lot of search time. Block Sum Pyramid Algorithm can eliminate those impossible matching blocks by exploiting the sum pyramid structure of a block. An image pyramid is a hierarchical data structure originally developed for image coding [24]. Assume that each block is of size $N \times N$ with $N = 2^n$. Then, for each block X , a pyramid of X can be defined as a sequence of blocks $\{X^0, \dots, X^{m-1}, X^m, X^{m+1}, \dots, X^n\}$ with X^{m-1} having size $2^{m-1} \times 2^{m-1}$ and being a reduced-resolution version of X^m as shown in [Fig. 1] Note that X^0 has only one pixel. A pyramid data structure can be formed by successively operating over 2×2 neighboring pixels on the higher levels. That is, the value of a pixel $X^{m-1}(i, j)$ on level $m-1$ can be obtained from the values of the corresponding 2×2 neighboring pixels $X^m(2i-1, 2j-1)$, $X^m(2i-1, 2j)$, $X^m(2i, 2j-1)$, and $X^m(2i, 2j)$ on level m as shown in equation(4).

$$X^{m-1}(i, j) = X^m(2i-1, 2j-1) + X^m(2i-1, 2j) \\ + X^m(2i, 2j-1) + X^m(2i, 2j) \quad (4)$$

For two blocks X and Y , let $\text{MAD}^m(X, Y)$ be $\text{MAD}(X^m, Y^m)$, i.e.,

$$\text{MAD}^m(X, Y) \\ = \sum_{j=1}^{2^m} \sum_{h=1}^{2^m} |X^m(j, h) - Y^m(j, h)| \quad (5)$$

where $X^m(j, h)$ and $Y^m(j, h)$ represent the values of the (j, h) th pixels on X^m and Y^m respectively. Thus, on the top level, $\text{MAD}^0(X, Y) = |S_X - S_Y|$. From the above definition, the following theorem holds.

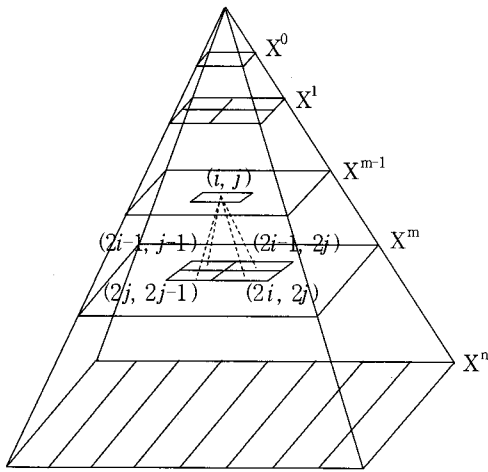
$$\text{MAD}(X, Y) \geq \text{MAD}^{n-1}(X, Y) \geq \text{MAD}^{n-2}(X, Y) \\ \geq \dots \geq \text{MAD}^0(X, Y) \quad (6)$$

Block Sum Pyramid Algorithm uses the above theorem (6). The Block Sum Pyramid Algorithm first constructs the sum pyramid of every block that corresponds to a search position in the previous frame. To search for the best matching block of a template block T , the sum pyramid of T is established. Then, the MAD between T and the block with displacement vector $(0, 0)$ is evaluated, and this value is considered as the current minimum MAD (symbolized as curr_MAD_{\min}). For any other search block X , the algorithm first checks the MAD on the top level, $\text{MAD}^0(T, X)$. If $\text{MAD}^0(T, X)$ is greater than curr_MAD_{\min} , this block can be eliminated from being considered as the best matching block. Otherwise, the MAD on the first level is checked.

If $\text{MAD}^1(T, X)$ is greater than curr_MAD_{\min} , for the same reason as above, this block can be eliminated. If it is not, the second level is tested. The process is repeated until this block is eliminated or the bottom level is reached. If the bottom level is reached, then $\text{MAD}(T, X)$ is calculated and checked. If $\text{MAD}(T, X) < \text{curr_MAD}_{\min}$, the current minimum distortion curr_MAD_{\min} is replaced with $\text{MAD}(T, X)$. Block Sum Pyramid Algorithm can eliminate many search blocks without evaluating their MADs. Assume that the size of the image frame is $W \times H$. For each level of the pyramid, calculation of the sum of 2×2 neighboring pixels requires $3(W-1)(H-1)$ additions. However, using the idea for fast calculation of the sum norm developed in [17], the complexity can be reduced to be $(2W-1)(H-1)$ additions for each level. If the block size is 16×16 , i.e., $N=16$, the overhead for constructing the sum pyramid is $4(2W-1)(H-1)$.

Since there are $(W/N)(H/N)$ template blocks in an image frame, the computation overhead for each template block is

$$\frac{4(2W-1)(H-1)}{[(W/N)(H/N)]} = N^2(8-4/W-8/H+4/WH) \approx 8N^2 \quad (7)$$



[Fig. 1] A pyramid data structure

BSPA procedure is as follows :

- step 1 : select an initial search block within the search window in the previous frame.
- step 2 : calculate the motion vector (MV) and the MAD at the selected search block. these MV and MAD become the current temporary motion vector and the current minimum MAD respectively ($temp_MV = MV$, $curr_MAD_{min} = MAD$)
- step 3 : select another search block among the rest of the search blocks
- step 4.0 : calculate the $MAD^0(T, X)$ at the selected search block. if ($curr_MAD_{min} \leq MAD^0(T, X)$) goto step 6
- step 4.1 : calculate the $MAD^1(T, X)$ at the selected search block. if ($curr_MAD_{min}$

$\leq MAD^1(T, X)$) goto step 6

.

step 4.(n-1) : calculate the $MAD^{(n-1)}(T, X)$ at the selected search block.

if ($curr_MAD_{min} \leq MAD^{(n-1)}(T, X)$) goto step 6

step 4.n : calculate the MAD at the selected search block.

if ($curr_MAD_{min} \leq MAD$) goto step 6

step 5 : $curr_MAD_{min} = MAD$

calculate the motion vector at the selected search block. This motion vector becomes the current temporary motion vector ($temp_MV = MV$)

step 6 : if (all the search blocks in the search window are not tested?) goto step3

step 7 : the optimum motion vector= $temp_MV$, the global minimum MAD = $curr_MAD_{min}$

3. A New Block Matching Algorithm for Motion Estimation

It is possible to rewrite (3) using the l_2 norm measurement. However, such approach requires the square root operation. An alternative bounds the l_2 norm by l_1 norm using the mathematical inequality[21, 22].

$$\|X - Y(i, j)\|_1 \leq \sqrt{N} \|X - Y(i, j)\|_2 \quad (8)$$

where N is the dimensionality of blocks X and Y . Combining this bound with the bound on the l_1 norm given in (3) results in

$$|\|X\|_1 - \|Y(i, j)\|_1| \leq \sqrt{N} \|X - Y(i, j)\|_2 \quad (9)$$

Squaring the both sides removes the square root operation

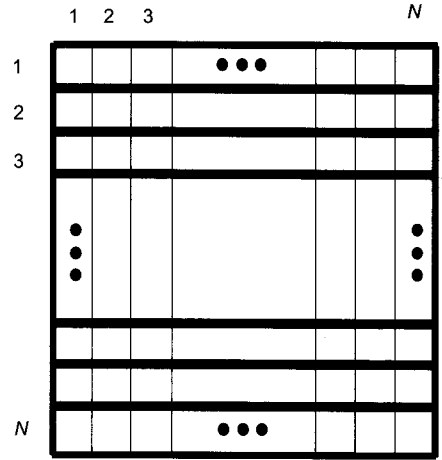
$$\frac{(\|X\|_1 - \|Y(i, j)\|_1)^2}{N} \leq (\|X - Y(i, j)\|_2)^2 \quad (10)$$

Again assume that vector (m, n) is the current best guess vector for the optimal displacement vector and that vector (i, j) is a possible candidate. If (i, j) is a better candidate than (m, n) then $(\|X\|_1 - \|Y(i, j)\|_1)^2/N \leq (\|X - Y(m, n)\|_2)^2$. Thus, if this condition is true, the norm squared at (i, j) must be explicitly calculated. However, if this condition is not true, then (i, j) cannot be a better candidate than (m, n) and the norm at (i, j) does not need to be calculated. This l_2 measurement was applied to SEA to reduce the computations cost of SEA[20].

Also, partial distortion elimination (PDE) technique can be applied to BSPA. The BSPA speeds up the process of finding the best motion vector by eliminating impossible candidate vectors before their MADs are computed. Partial distortion elimination (PDE)[25] can be used in with BSPA to reduce the computation further for these vectors where the norm $\|T(i, j) - X(i, j)\|$ must actually be computed. We observe that since all of the terms in the equation (2) are positive, if at any point the partially evaluated sum exceeds the current minimum MAD($curr_MAD_{min}$), that candidate block X cannot be the best matching block and the remainder of the sum does not need to be calculated. While it is not efficient to test the partial sum against the $curr_MAD_{min}$ every additional term is added, a reasonable compromise is to perform the test after N times additions when block size is $N \times N$

as shown in [fig. 2]. So, the maximum number of PDE test is N .

In BSPA, MAD is calculated all the $N \times N$ pixels and then the calculated MAD is compared with $curr_MAD_{min}$



[Fig. 2] PDE test in MAD calculation

4. Experimental Results of New Block Matching Algorithm for Motion Estimation

The standard QCIF(176×144) video sequences in the frame work of H.263 such as “grandma.qcif”, “suzie.qcif”, “claire.qcif” were used in the experiment and we tested the first 100 frames of the video sequences. The block(Y component of the macro-block in H.263) size was 16×16 pixels ($N=16$). The size of the search window was 31×31 pixels ($M=15$) and only integer values for the motion vectors were considered. The frame rate was fixed at 30 fps. In encoder implementations an offset is subtracted from $d_1(0, 0)$ to bias motion decisions in favor of nonmoving background blocks. Here we simply set the offset to zero for simplicity. Experimental re-

〈Table 1〉 The computations of NBMA

Algorithm	Test sequence	Avg. # of m.e./frame	Avg. # of rows/m.e.	Overhead (in rows)	Total (in rows)	Computations reduction
BSPA	grandma	350.7	16.00	29,749.3	35,360.5	
	suzie	607.1	16.00	27,620.5	37,334.1	
	claire	228.3	16.00	17,920.8	21,573.6	
NBMA	grandma	350.7	13.12	29,481.6	34,082.7	3.6%
	suzie	607.1	14.25	27,371.9	36,023.1	3.5%
	claire	228.3	13.89	17,759.5	20,930.6	3.0%

sults are shown in <table 1> In <table 1>, m.e. means matching evaluation that require MAD calculation, avg. # of rows means the number of calculated rows in the MAD calculation before partial distortion elimination. Overhead(in rows) is the sum of all the computations except MAD calculation. "in rows" means that the computations are represented in order of 1 row MAD computations.

It is important to notice that with the BSPA, the efficiency of the procedure depends on the order in which the candidate motion vectors are searched, and that the most promising candidates should be tested first. This eliminates the maximum number of candidates. In our experiment, we used spiral search.

The NBMA which incorporates the BSPA with l_2 norm and PDE technique can reduce the values of "Avg. # of rows/m.e." and "overhead(in rows)". By using PDE technique, the partially evaluated MAD is greater than or equal to the current minimum MAD($curr_MAD_{min}$) then the point cannot be the optimum motion vector and the remainder of the sum does not need to be calculated. So, the "Avg. # of m.e./frame" can be reduced. But, l_2 norm and PDE technique does not effect on the number of matching evaluation. So, "Avg. # of m.e./frame" does not change. The proposed algorithm (NBMA) can reduces the

computations of BSPA by 3.6%, 3.5%, 3.0% for grandma.qcif, suzie.qcif, and claire.qcif respectively.

5. Conclusions

A New Block Matching Algorithm based on BSPA has been proposed to reduce the computations of BSPA. The l_2 norm was used to reduce the computation cost of BSPA. Also, the partial distortion elimination scheme of EMSEA was improved and then the improved scheme was applied to BSPA in NBMA.

The NBMA can find the global optimum solution and outperforms BSPA and can reduce the computations of block matching calculation of BSPA 3.6% maximally. The NBMA is a very efficient solution for video coding applications that require both very low bit-rates and good coding quality.

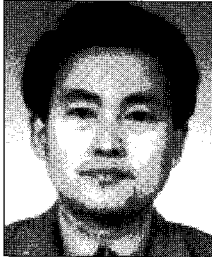
References

- [1] C. H. Lee and L. H. Chen, "A Fast Motion Estimation Algorithm Based on the Block Sum Pyramid Algorithm," *IEEE Trans. Image Processing*, Vol.6, No.11(Nov. 1997), pp.1587-1591.
- [2] H. G. Musmann, P. Pirsh and H. J. Gilbert,

- "Advances in Picture Coding," *Proc. IEEE*, Vol.73(Apr. 1985), pp.523-548.
- [3] J. R. Jain and A. K. Jain, "Displacement Measurement and Its Application in Inter-frame Image Coding," *IEEE Trans. Commun.*, Vol.COMM-29(Dec. 1981), pp.1799-1808.
- [4] A. N. Netravali and J. D. Robbins, "Motion Compensated Television Coding : Part I," *Bell Syst. Tech. J.*, Vol.58(Mar. 1979), pp. 631-670.
- [5] CCITT Standard H.261, *Video Codec for Audiovisual Services at px64 kbit/s*, ITU, 1990.
- [6] ITU-T DRAFT Standard H.263, *Video Coding for Narrow Telecommunication Channel at(below) 64kbit/s*, ITU, Apr. 1995.
- [7] ISO-IEC JTC1/SC2/WG11, *Preliminary Text for MPEG Video Coding Standard*, ISO, Aug. 1990.
- [8] T. Koga, K. Iinuma, Y. Iijima and T. Ishiguro, "Motion Compensated Interframe Coding for Video Conferencing," *Proc. Nat. Telecommunications Conf.*, (Nov. 1981), pp.G5.3.1-G.5.3.5.
- [9] A. Puri, H.-M. Hang and D. L. Schilling, "An Efficient Block Matching Algorithm for Motion Compensated Coding," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, 1987, pp.25.4.1-25.4.4.
- [10] M. Ghanbari, "The Cross-search Algorithm for Motion Estimation," *IEEE Trans. Commun.*, Vol.38, No.7(July 1990), pp.950-953.
- [11] M. J. Chen, L. G. Chen and T. D. Chiueh, "One-dimensional Full Search Motion Estimation Algorithm for Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 4(Oct. 1994), pp.504-509.
- [12] H. M. Jong, L. G. Chen and T. D. Chiueh, "Accuracy Improvement and Cost Reduction of 3-step Search Block Matching Algorithm for Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.4(Feb. 1994), pp. 88-91.
- [13] 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(Aug. 1994), pp. 438-442.
- [14] J. Y. Tham, S. Ranganath, M. Ranganath and A. Ali Kassim, "A Novel Unrestricted Center-biased Diamond Search Algorithm for Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.8, No.4 (Aug. 1998), pp.369-377.
- [15] B. Liu and A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.3, No.2(Apr. 1993), pp.148-157.
- [16] K. H. K. Chow and M. L. Liou, "Genetic Motion Search Algorithm for Video Compression," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.3(Dec. 1993), pp.440-446.
- [17] W. Li and E. Salari, "Successive Elimination Algorithm for Motion Estimation," *IEEE Trans. Image Processing*, Vol.4, No. 1(Jan. 1995), pp.105-107.
- [18] X. Q. Gao, C. J. Duanmu and C. R. Zou, "A Multilevel Successive Elimination Algorithm for Block Matching Motion Estimation," *IEEE Trans. Image Processing*, Vol. 9, No.3 (Mar. 2000), pp.501-504.
- [19] S. M. Jung, S. C. Shin, H. Baik and M. S. Park, "Efficient Multilevel Successive Elimination Algorithms for Block Matching Motion Estimation," *IEE Vision and Image*

- Signal Processing*, vol.149, No.2(Apr. 2002), pp.73-84.
- [20] H.S. Wang and R. M. Mersreau, "Fast Algorithms for the Estimation of Motion Vectors," *IEEE Trans. Image Processing*, Vol. 8, No.3(Mar. 1999), pp.435-438.
- [21] T. M. Apostol. *Mathematical Analysis*, Reading, MA : Addison-Wesley, 1975.
- [22] H. S. Wang and R. M. Mersereau, "Fast Algorithms for the Estimation of Motion Vectors," *IEEE Trans. Image Processing*, Vol. 8, No.3(Mar. 1999), pp.435-438.
- [23] Y. Q. Zhang and S. Zafar, "Predictive Block-matching Motion Estimation for TV Coding-Part II : Interframe Prediction," *IEEE Trans. Broadcast.*, Vol.37(Sep. 1991), pp.102-105.
- [24] P. J. Burt and E. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. Commun.*, Vol.COMM-31(Apr. 1993), pp.525-540.
- [25] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*, Boston, MA : Kluwer, 1991.

◆ 저 자 소 개 ◆



정 수 목 (jungsm@syu.ac.kr)

경북대학교 전자공학과에서 학사, 경북대학교 대학원 전자공학과에서 석사, 고려대학교 대학원 컴퓨터학과에서 박사학위를 취득하고, 삼육대학교 컴퓨터학과 부교수로 재직 중이다. 현재 멀티미디어, 영상처리를 연구 중이다. 주요 관심분야는 컴퓨터 아키텍처, 컴퓨터보안, 컴퓨터네트워크 등이다