

# Extended Successive Elimination Algorithm for Fast Optimal Block Matching Motion Estimation

Changryoul Choi and Jechang Jeong

Dept. of Electronics and Communication Engineering Hanyang University  
Seoul, Korea

e-mail : denebchoi@gmail.com & jjeong@hanyang.ac.kr

**Abstract**—In this paper, we propose an extended successive elimination algorithm (SEA) for fast optimal block matching motion estimation (ME). By reinterpreting the typical sum of absolute differences measure, we can obtain additional decision criteria whether to discard the impossible candidate motion vectors. Experimental results show that the proposed algorithm reduces the computational complexity up to 19.85% on average comparing with the multilevel successive elimination algorithm. The proposed algorithm can be used with other SEA to improve the ME performance.

**Keywords**—*motion estimation; successive elimination algorithm; block matching*

## I. INTRODUCTION

Motion estimation (ME) has been widely used in many video applications ranging from video compression to video segmentation, video tracking, etc [1]. The block matching algorithm (BMA) for ME is the most popular and 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, e.g. the sum of absolute differences (SAD), the sum of squared differences (SSD), etc. The full search algorithm (FSA) can give the optimal estimation of the motion in terms of minimal matching error by checking all the candidates within the search range, but the huge computational complexity of the FSA makes it inadequate for the real-time applications. Thus, many fast but optimal algorithms which provide the same accuracy as the FSA are proposed including the fast searching and the fast matching algorithms in the literature [4-9].

The fast matching technique aims at reducing the whole calculations of the matching criterion for each candidate block by comparing only a subset of the pixels in the block. In lossy fast matching, it predicts the total matching criterion based on the statistical property of it or calculates the matching criterion of the sub-sampled patterns only [4-5]. In lossless fast matching, it is based on the simple idea that since the total distortion is monotonically increasing by progressively adding the partial distortion of the pixel-by-

pixel differences, a block comparison can be safely terminated if the accumulated sum of partial distortions becomes greater than the up-to-date minimum distortion [6].

The fast searching is the technique to reduce the number of searching points. In lossy fast searching, it exploits some general properties of the typical images. One of these properties is that since the real world video sequences usually vary slowly, their distribution of motion vector is center-biased. Also, many lossy fast searching algorithms assume that the matching error planes are convex to reduce the searching points [7]. In lossless fast searching, based on some mathematical inequalities, it calculates the lower bound of the matching criterion (typically the SAD) and safely skips the impossible candidate motion vectors [8-9]. In [8], the successive elimination algorithm (SEA) was proposed. The SEA estimates the SAD by calculating a block sum difference which is the lower bound of the true SAD. The estimated SAD is used as a decision criterion whether to eliminate the impossible candidate. Gao et al. extended the idea of SEA to multilevel successive elimination algorithm (MSEA) [9]. By splitting the block into small sub-blocks, closer estimations to the true SAD are given and the estimated SAD is used as decision criteria whether to discard the impossible candidate.

And there are some techniques that exploit different matching criteria instead of the classical SAD or SSD were also proposed to make the faster computation of the matching criteria using bit-wise operations [10].

In this paper, we propose an optimal fast searching algorithm which is an extension of the typical SEA. By reinterpreting the typical SAD measure, we can obtain additional decision criteria for pruning out bad motion vectors. The rest of this paper is organized as follows. Section II gives a review of the fast searching ME algorithms. Section III presents our proposed algorithm. Experimental results and analyses are provided in Section IV. Finally, Section V provides conclusions.

## II. PREVIOUS ALGORITHMS

The SAD between the current block (CB) and the reference block (RB) is usually used as a matching criterion for ME and is defined as follows:

$$SAD(x,y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |CB(i,j) - RB(i+x,j+y)| \quad (1)$$

where  $N \times N$  is the motion block size and  $(x,y)$  is the candidate motion vector within the search range.

#### A. Successive Elimination Algorithm

Let  $CB_0$  and  $RB_0$  be the sum norms of the CB and the RB which are defined as follows:

$$CB_0 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} CB(i,j) \quad (2)$$

$$RB_0 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} RB(i+x,j+y)$$

where  $(x,y)$  is the candidate motion vector within the search range. Note that the sum norms of the RB can be calculated efficiently over the whole image [8]. From the triangle inequality, we can easily derive the following inequality:

$$|CB_0 - RB_0| \leq SAD \quad (3)$$

This inequality (3) shows the key idea of the SEA. If the calculated sum norm of the RB of position  $(x,y)$  and the CB does not satisfy the inequality (3) (in this case, the SAD is replaced by the up-to-date minimum SAD in the searching process), this means that the candidate motion vector of position  $(x,y)$  is not the optimal motion vector and the calculation of SAD is unnecessary and skipped [8].

#### B. Multilevel Successive Elimination Algorithm

The MSEA extended the idea of SEA to a multilevel case [9]. By splitting the block into small sub-blocks, closer estimations to the true SAD are given and the estimated SAD is used as decision criteria whether to discard the impossible candidate. First, the block is partitioned into four sub-blocks of size  $N/2 \times N/2$ . Then each sub-block is partitioned into four sub-blocks of size  $N/4 \times N/4$ . This process can be repeated until the size of the sub-blocks becomes  $2 \times 2$ . The maximum level of such partition is  $L_{max} = \log_2 N - 1$  when the motion block size is  $N \times N$ . Let  $CB_l^{(k)}$  and  $RB_l^{(k)}$  be the sum norms of the  $k$ th sub-blocks at the  $l$ th level in the CB and the RB, respectively. Based on (3), we can obtain

$$\sum_{k=0}^{N_l-1} |CB_l^{(k)} - RB_l^{(k)}| \leq \sum_{k=0}^{N_{l+1}-1} |CB_{l+1}^{(k)} - RB_{l+1}^{(k)}| \leq SAD \quad (4)$$

where  $N_l$  is the number of sub-blocks at the  $l$ th level. From (4), we can attain monotonically increasing SAD estimation

values as the level increases. Therefore, more and more impossible candidates can be eliminated earlier as the level increases [9].

### III. PROPOSED ALGORITHM

Let  $a_i$  and  $b_i$  be the sequences of length  $N \times N$ . As in (3), we can summarize the SEA as follows:

$$SAD = \sum |a_i - b_i| \geq \left| \sum (a_i - b_i) \right| = \left| \sum a_i - \sum b_i \right| \quad (5)$$

And we can think of the absolute operation as follows:

$$\sum |a_i - b_i| = \sum \text{sign}(a_i - b_i) \times (a_i - b_i) \quad (6)$$

In case of SEA, the equality holds only when either of the following is satisfied:

$$\begin{aligned} \text{sign}(a_i - b_i) &= 1, \forall i \\ \text{sign}(a_i - b_i) &= -1, \forall i \end{aligned} \quad (7)$$

which is rare the case. Using (5) and (6), we can think of the following inequality:

$$\begin{aligned} \sum |a_i - b_i| &= \sum \text{sign}(a_i - b_i) \times (a_i - b_i) \\ &\geq \left| \sum \alpha_i \times (a_i - b_i) \right| \end{aligned} \quad (8)$$

where  $\alpha_i$  takes on either 1 or -1. Therefore, to estimate the SAD values more precisely, the sequence  $\alpha_i$  must be almost the same as the signs of  $(a_i - b_i)$ s or totally inverted signs of them. Due to summations of the smaller sub-blocks and absolute operations in (4), the MSEA can be considered as the process of estimating the true SAD by forcing the signs of difference sequence into the true signs of it more and more as the level increases. In this case, the difference of the sum norms in each level plays as a one and only candidate for estimating signs of the sequences.

The basic idea of the proposed algorithm is that we can enhance the estimation of the true SAD by allowing more candidates (whose signs of the difference sequence are different) in each level. To this end, we generate more candidates which are generalized sum norms to enhance the estimation accuracy. We mean a generalized sum norm as the summation of the pixels according to the predefined basic (addition or subtraction) arithmetic.

Since the systematic and efficient calculation of the sum norms is one of the main reasons for computational complexity reduction in the SEA, the number of possible candidates for the generalized sum norms which can be calculated efficiently is limited. Fig. 1 shows the structured image arithmetic templates taking into consideration the computational complexity. We call the generalized sum

norms of Fig. 1 (a) as original sum norms, Fig. 1 (b) as horizontal sum norms, Fig. 1 (c) as vertical sum norms, and Fig. 1 (d) as diagonal sum norms.

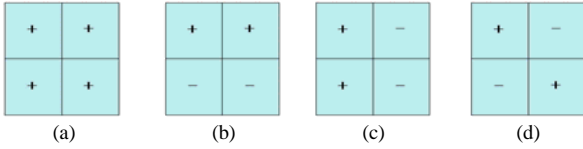


Figure 1. Structured image arithmetic templates (a) template used in SEA and MSEA (b), (c), and (d) structured image arithmetic templates horizontal, vertical and diagonal, respectively.

Due to the increased candidates (horizontal, vertical, and diagonal), the inequality in (4) is changed as:

$$\sum_{k=0}^{N_l-1} \max_{z \in \{o,h,v,d\}} |CB_{l,z}^{(k)} - RB_{l,z}^{(k)}| \leq \sum_{k=0}^{N_{l+1}-1} \max_{z \in \{o,h,v,d\}} |CB_{l+1,z}^{(k)} - RB_{l+1,z}^{(k)}| \quad (9)$$

where  $CB_{l,z}^{(k)}$  and  $RB_{l,z}^{(k)}$  are the generalized sum norms ( $o$  represents original sum norms,  $h$  represents horizontal sum norms,  $v$  represents vertical sum norms, and  $d$  represents diagonal sum norms) of the  $k$ th sub-blocks at the  $l$ th level in the CB and the RB, respectively. The proposed algorithm is almost the same as the MSEA except the following two differences. The first one is that the proposed algorithm gives more stop conditions than the typical MSEA (in the first level, there are 3 more stop conditions than the typical MSEA. And since these stop conditions are of the first level, if one of these stop conditions is satisfied, it reduces the total computational complexity more). The second one is that due to (9), we can attain closer SAD bound than the typical MSEA at the same level reducing the computational complexity.

Note that there is a trade-off between the increased computational complexity of calculating the increased generalized sum norms and the reduced computational complexity by pruning out the bad motion vectors in an early stage. And the increased stop conditions can be pros or cons. To estimate the actual effects of the generalized sum norms in computational complexity, we calculated the computational complexity of the proposed algorithm in terms of the searched points.

TABLE I. COMPUTATIONAL COMPLEXITY OF THE GENERALIZED SUM NORMS CALCULATIONS WHEN THE MOTION BLOCK SIZE IS 16x16 AND THE SEARCH RANGE IS ±16

	MSEA	Proposed - All	Proposed - H only	Proposed - V only
Overhead (points)	3.74	11.21	5.60	5.60

Table I shows the computational complexity of the generalized sum norms in terms of the SAD calculations when the motion block size is 16x16 and the search range is set to ±16. To be specific, the table shows the number of

total operations divided by one SAD calculation operations, in this case we assume that the computational complexity of addition and subtraction operations and that of the absolute operations are the same.

Table II shows the average search points of the proposed algorithms when all the generalized sum norms are used (3<sup>rd</sup> column), only the horizontal sum norms are used (4<sup>th</sup> column), and only the vertical sum norms are used, respectively. Note that the original sum norms were also used in all of the proposed algorithms. The test sequences are of CIF-size and 100-frame long. The motion block size is 16x16 and the search range is set to ±16. The computational complexity of the table I is also considered.

TABLE II. AVERAGE SEARCH POINTS OF ALGORITHMS FOR CIF SEQUENCES WHEN THE MOTION BLOCK SIZE IS 16x16 (100-FRAME, SEARCH RANGE IS ±16)

	MSEA	Proposed - All	Proposed - H only	Proposed - V only
stefan	63.65	82.62	58.81	75.91
football	62.36	81.54	60.53	72.17
foreman	51.87	65.92	48.92	53.58
mobile	40.62	49.62	40.46	40.52
coastguard	63.25	76.65	46.83	92.75
container	42.00	53.15	37.00	47.66
flower	109.42	128.72	89.35	112.45
<b>Avg.</b>	<b>61.88</b>	<b>76.89</b>	<b>54.56</b>	<b>70.72</b>

From the table, we can see that using all the generalized sum norms and using only the vertical sum norms does not provide any computational reduction. Therefore, the final proposed algorithm uses the following inequality for additional stop conditions for the typical MSEA.

$$\sum_{k=0}^{N_l-1} \max_{z \in \{o,h\}} |CB_{l,z}^{(k)} - RB_{l,z}^{(k)}| \leq \sum_{k=0}^{N_{l+1}-1} \max_{z \in \{o,h\}} |CB_{l+1,z}^{(k)} - RB_{l+1,z}^{(k)}| \quad (10)$$

#### IV. EXPERIMENTAL RESULTS

The performance of the proposed algorithm was compared with the MSEA in terms of the SAD calculation points. The full frames of the 7 CIF (352 x 288) sequences, 4 SD sequences (704 x 576), and 4 HD sequences (1280 x 720) were used as test sequences. Motion block sizes were all 16x16 and the searching processes were in spiral order.

Tables III and IV show the average searched points of CIF and SD sequences when the search range is ±16 and ±32, respectively. The proposed algorithm outperforms the MSEA. To be specific, the performance of the proposed algorithm is better than that of the MSEA by 20.0% on average when the search range is ±32 for SD sequences. Table V shows the average searched points of HD sequences when the search range is ±16, ±32 and ±64, respectively. Note that since the ME accuracy of the

proposed algorithm is the same as that of the MSEA, we omit the ME accuracy in terms of the peak signal to noise ratio (PSNR) here.

TABLE III. AVERAGE SEARCH POINTS OF ALGORITHMS FOR CIF SEQUENCES WHEN THE MOTION BLOCK SIZE IS 16x16 (FULL-FRAME, S.R. = SEARCH RANGE)

	MSEA		Proposed - H only	
	S.R. = ±16	S.R. = ±32	S.R. = ±16	S.R. = ±32
Stefan	100.33	170.26	90.20	148.87
football	57.10	126.27	56.28	118.21
Foreman	57.75	97.84	57.9	93.77
Mobile	35.06	78.55	34.52	72.73
Coastguard	54.61	128.21	41.21	89.51
Container	43.25	84.04	37.94	73.68
Flower	91.46	162.33	77.57	138.36
<b>Avg.</b>	<b>62.79</b>	<b>121.07</b>	<b>56.52</b>	<b>105.02</b>

TABLE IV. AVERAGE SEARCH POINTS OF ALGORITHMS FOR SD SEQUENCES WHEN THE MOTION BLOCK SIZE IS 16x16 (FULL-FRAME, S.R. = SEARCH RANGE)

	MSEA		Proposed - H only	
	S.R. = ±16	S.R. = ±32	S.R. = ±16	S.R. = ±32
ICE	157.23	378.44	119.9	279.72
CITY	48.25	94.85	48.00	91.16
CREW	115.55	223.4	107.25	207.32
SOCCER	93.08	176.96	81.9	150.71
<b>Avg.</b>	<b>103.53</b>	<b>218.41</b>	<b>89.26</b>	<b>182.23</b>

TABLE V. AVERAGE SEARCH POINTS OF ALGORITHMS FOR HD SEQUENCES WHEN THE MOTION BLOCK SIZE IS 16x16 (FULL-FRAME, S.R. = SEARCH RANGE)

	MSEA			Proposed - H only		
	S.R. = ±16	S.R. = ±32	S.R. = ±64	S.R. = ±16	S.R. = ±32	S.R. = ±64
Big-Ships	53.57	100.53	174.95	45.26	78.89	158.67
Crew	178.57	365.27	718.07	156.08	321.34	648.33
Preak-ness	25.00	49.69	91.71	24.55	43.97	108.6
Sheriff	39.30	77.06	173.64	35.45	64.23	134.22
<b>Avg.</b>	<b>74.11</b>	<b>148.14</b>	<b>289.59</b>	<b>65.34</b>	<b>127.11</b>	<b>262.46</b>

As can be seen from the tables, the proposed algorithm outperforms the MSEA in terms of the computational complexity. To be specific, the proposed algorithm reduces the computation complexity by 19.85% when the search range is ±32 for SD sequences. Since the proposed algorithm can be easily plugged in other SEA based ME algorithms, additional computational reduction for this seemingly marginal gain can be expected.

V. CONCLUSIONS

By reinterpreting the typical SAD measure, we proposed an extended SEA for fast optimal block matching ME in this paper. By allowing more candidates in estimating the true SAD, we can obtain additional decision criteria whether to discard the impossible candidate motion vectors. Experimental results show that the proposed algorithm reduces the computational complexity up to 19.85% on average comparing with the typical MSEA. Since the proposed algorithm can be easily adopted in other SEA based ME algorithms, additional computational reduction can be expected. Therefore, our future research will be focused on merging the proposed algorithm with the previous SEA based ME algorithms to reduce the computational complexity a lot without degrading the 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 Syst. Video Technol., vol. 10, no. 5, Aug. 2000, pp. 669-678.
- [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] C. Cheung and L. Po, "Adjustable partial distortion search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 13, Jan. 2003, pp. 100-110.
- [5] Y. Wang, Y. Wang, and H. Kuroda, "A Globally Adaptive Pixel-Decimation Algorithm for Block-Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 6, Jan. 2000, pp. 1006-1011.
- [6] C. Choi and J. Jeong, "New sorting-based partial distortion elimination algorithm for fast optimal motion estimation," IEEE Trans. Consumer Electron., vol. 55, no. 4, Nov. 2009, pp. 2335-2340.
- [7] X. Jing and L. Chau, "An efficient three-step search algorithm for block motion estimation," IEEE Trans. Multimedia, vol. 6, no. 3, Jun. 2004, pp. 435-438.
- [8] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," IEEE Trans. Image Process., vol. 8, Jan. 1995, pp. 105-107.
- [9] X. Q. Gao, C. J. Duanmu, and C. R. Zou, "A multilevel successive elimination algorithm for block matching error estimation," IEEE Trans. Image Process., vol. 9, Mar. 2000, pp. 501-504.
- [10] C. Choi and J. Jeong, "Enhanced Two-bit Transform Based Motion Estimation via Extension of Matching Criterion," IEEE Trans. Consumer Electron., vol. 56, no. 3, Aug. 2010, pp. 1883-1889.