

# Fast motion estimation with adaptive search range adjustment

Si-Woong Lee,<sup>a</sup> Seong-Mo Park,<sup>b</sup> and Hyun-Soo Kang<sup>c</sup>

<sup>a</sup>Hanbat National University, Division of Information Communication and Computer Engineering, Dukmyung-dong, Yusong-gu, Daejeon, 305-719, Korea

<sup>b</sup>Electronics and Telecommunications Research Institute, IT Convergence & Components Laboratory, 161 Gajeong-Dong, Yuseong-Gu, Daejeon, 305-350, Korea

<sup>c</sup>Chungbuk National University, School of Electrical and Computer Engineering, Gaesin-dong, Heungduk-gu, Cheongju, Chungbuk, 361-763, Korea

**Abstract.** A fast motion estimation algorithm based on the adaptive search range adjustment (ASRA) is presented. The size of search range for each block is adaptively controlled according to the initial cost at the search center, and then the full search algorithm (FSA) is performed to find the optimal matching block in the adjusted search range. The proposed method is implemented in H.264/AVC reference software (JM) to demonstrate its efficiency over the conventional FSA. © 2007 Society of Photo-Optical Instrumentation Engineers. [DOI: 10.1117/1.2721767]

Subject terms: video coding; motion estimation; search range; full search algorithm (FSA); H.264/AVC.

Paper 060912LR received Nov. 23, 2006; revised manuscript received Jan. 11, 2007; accepted for publication Jan. 29, 2007; published online Apr. 5, 2007.

## 1 Introduction

Block matching motion estimation is the core process in various video coding standards because of its efficiency in removing the temporal redundancy. However, motion estimation is very computationally intensive, if the exhaustive fast search algorithm (FSA) is employed to find the optimal matched block in the reference frame. To alleviate the heavy computation, many fast algorithms, such as the well-known three-step search (3SS),<sup>1</sup> the four-step search (4SS),<sup>2</sup> the diamond search (DS),<sup>3</sup> and the hexagon-based search (HEXBS),<sup>4</sup> have been proposed. Except for the successive elimination algorithm (SEA),<sup>5</sup> these algorithms achieve fewer computations by sampling the search points in the search range, i.e., searching over a subset of possible candidate points with certain search patterns. Although algorithms in this category considerably reduce the computational complexity, they occasionally produce ambiguous search directions, and hence result in a local minimum.

This letter proposes a new fast motion estimation algorithm, which focuses on reducing computation not by sampling the search points but by reducing the size of the search range. Based on the assumption of convexity of the unimodal error surface, the proposed method makes use of the fact that the search center is probably close to the optimal matching point when the initial cost at the search center is efficiently low. Thus, in such cases, it's possible to

reduce the size of the search range for increased speed without much increase in matching error, which is the basic concept of the proposed method.

## 2 Proposed Method

In general, motion vectors of spatially adjacent blocks are highly correlated if they belong to the same moving area. Having a predictive motion vector ( $MV_p$ ) for a current block as the search center, as in the H.264/AVC JM,<sup>6,7</sup> the initial cost at the search center  $J(MV_p)$  is a good estimate for correlation between the motion vector of the current block ( $MV_c$ ) and  $MV_p$ . If  $J(MV_p)$  is low, it means that  $MV_p$  is probably close to  $MV_c$ , and hence the size of the search range can be reduced to a proper size without missing the global minimum. On the other hand, if  $J(MV_p)$  is high, it means that  $MV_p$  is not so correlated with  $MV_c$ , and thus the size of the search range should be larger in order to include the global minimum in the search range. Based on this background, the block matching process for each block is performed as follows:

1. The  $MV_p$  for a current block is determined in the same way as H.264/AVC, and the search center is set to  $MV_p$ .
2. Using  $MV_p$ ,  $J(MV_p)$  is measured according to the definition of the cost.
3. With  $J(MV_p)$ , the size of the search range for a current block,  $SR_{var}$ , is adjusted as

$$SR_{var} = SR_{fixed} / 2^{L-k}, \quad \text{if } T_k < J(MV_p) \leq T_{k+1}, \\ k = 0, 1, \dots, L, \quad (1)$$

where  $SR_{fixed}$  denotes the size of the fixed search range used in JM, and  $L+1$  represents the number of search range sizes in the adaptive search range adjustment (ASRA)-based method. Note that  $SR_{var}$  reduces to the power of 2 as  $J(MV_p)$  decreases. For  $SR_{fixed} = 16$  and  $L=2$ , there are three search range sizes of  $\pm 16$ ,  $\pm 8$ , and  $\pm 4$ .

4. The FSA is performed to estimate  $MV_c$  in the adjusted search range.

## 3 Threshold Decision

With  $SR_{fixed} = 16$  and  $L=2$ , two thresholds  $T_1$  and  $T_2$  ( $T_1 < T_2$ ) must be given to discriminate the ranges of  $J(MV_p)$  for the three kinds of search range such as

$$SR_{var} = \begin{cases} 4, & \text{if } J(MV_p) < T_1 \\ 8, & \text{else if } J(MV_p) < T_2 \\ 16, & \text{else} \end{cases} \quad (2)$$

Taking advantage of the similarity of the minimum costs among the neighboring blocks in the same moving region, we suggest determining the thresholds using the costs of already coded neighboring blocks as follows;

$$T_1 = \alpha \cdot \text{Med}[J_A, J_B, J_C], \quad (3)$$

**Table 1** Performance comparison between the proposed approach and the conventional FSA.

Method	Foreman		Stefan		Soccer	
	PSNR	Complexity	PSNR	Complexity	PSNR	Complexity
FSA( $\pm 16$ )	35.34	1	28.39	1	31.52	1
FSA( $\pm 8$ )	34.98	0.265	27.07	0.265	30.95	0.265
ASRA( $\alpha=1.0$ )	35.29	0.490	28.29	0.503	31.43	0.506
ASRA( $\alpha=1.5$ )	35.27	0.343	28.20	0.348	31.36	0.333
ASRA( $\alpha=2.0$ )	35.25	0.273	28.18	0.261	31.32	0.275

$$T_2 = \alpha \cdot \text{Max}[J_A, J_B, J_C], \tag{4}$$

where  $\alpha$  is a scaling constant, and  $J_A$ ,  $J_B$ , and  $J_C$  are stored minimum costs of the left (A), the upper (B), and the upper-right (C) blocks of the current block, respectively.  $\text{Med}[a, b, c]$  and  $\text{Max}[a, b, c]$  are functions returning the median and maximum values among  $a, b$ , and  $c$ . Figure 1 illustrates the relative locations of the neighboring blocks to the current block (X). For all block partitioning modes including  $16 \times 8$  and  $8 \times 16$  modes, the locations follow the definition used in JM to find the predictive motion vector by median prediction. If block C is unavailable, the upper-left block D replaces C. Note that when the partitioning mode of a current block is different from that of neighboring blocks, the stored costs of neighboring blocks are scaled in accordance to the size of the current block.

#### 4 Experimental Results

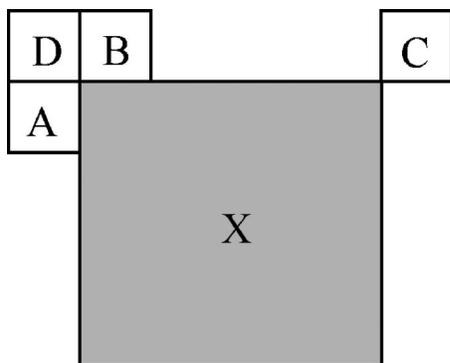
To evaluate the performance of the proposed technique, Foreman, Stefan, and Soccer sequences with the CIF format were tested with H.264/AVC JM9.4. The encoding parameters are as follows: number of frames—200, frame rate—15 Hz, target bit rate—256 kbit/s; RD optimization off;  $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 16$ , and  $8 \times 8$  block modes on; number of reference frames—1.

Table 1 shows the performance comparison between the proposed algorithm and the conventional FSA. Note that the spiral search method with  $MV_p$  as the initial search

point is used in both methods. The computational complexity is measured relative to the FSA with a  $\pm 16$  search range. The results are summarized as follows:

- In terms of computational complexity, the proposed method has a significant gain. When  $\alpha=2.0$ , the computational complexity is approximately 30% compared to the FSA with a  $\pm 16$  search range and nearly equivalent to the FSA with a  $\pm 8$  search range.
- In terms of reconstructed image quality, the proposed method keeps up with the FSA with a  $\pm 16$  search range. The average PSNR decreases 0.2 dB at most. In contrast, the FSA with a  $\pm 8$  search range exhibits severe degradation from  $-0.4$  dB to  $-1.1$  dB. This is because those blocks having large motion cannot be accurately motion-compensated with a small fixed search range. On the contrary, for such cases, the proposed ASRA-based approach may provide the full search range of  $\pm 16$ , so the accuracy of the motion estimation can be maintained. Table 2 shows the distributions of selected search ranges in the proposed method.

Figure 2 shows PSNR and complexity graphs for a range of  $\alpha$ . We can observe that as  $\alpha$  grows, the reduction ratio of the complexity decreases, and it is nearly saturated over  $\alpha=2.0$ , whereas the PSNR drops continually, although the degree of drop is negligible. From this result, a value about 2.0 seems proper as the empirical value for  $\alpha$ .



**Fig. 1** Determination of the neighboring blocks.

**Table 2** Distribution of selected search ranges in the proposed approach.

Method	Foreman (%)			Stefan (%)			Soccer (%)		
	$\pm 16$	$\pm 8$	$\pm 4$	$\pm 16$	$\pm 8$	$\pm 4$	$\pm 16$	$\pm 8$	$\pm 4$
ASRA ( $\alpha=1.0$ )	34.4	36.1	29.5	26.1	37.3	36.6	36.2	35.7	28.1
ASRA ( $\alpha=1.5$ )	22.0	23.0	55.0	16.2	23.7	60.1	21.6	20.3	58.1
ASRA ( $\alpha=2.0$ )	16.5	14.8	68.7	10.7	15.4	73.9	17.1	12.3	70.6

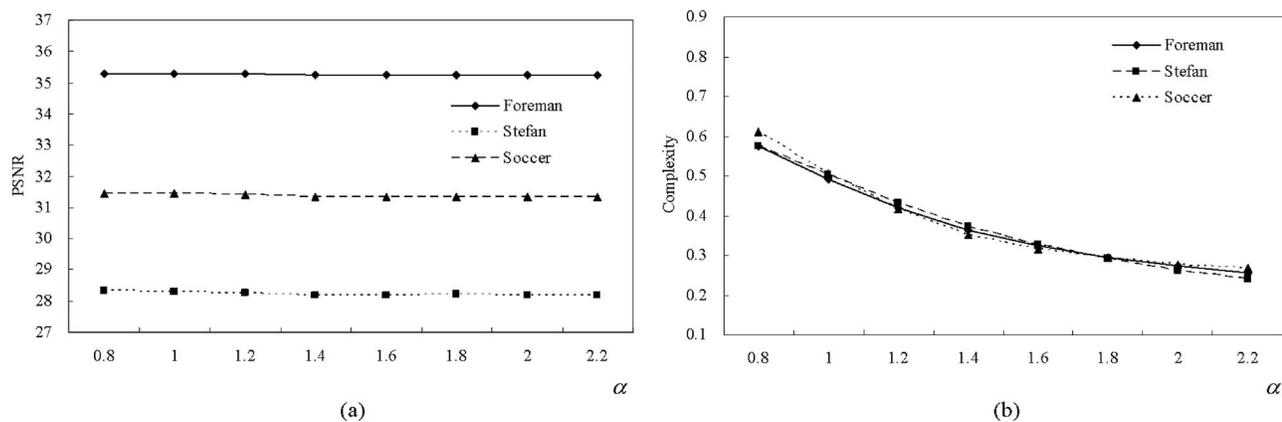


Fig. 2 PSNR and complexity versus  $\alpha$ : (a) PSNR; (b) complexity.

## 5 Conclusions

This letter has presented a fast motion estimation algorithm based on the ASRA. The size of the search range is adaptively adjusted according to the initial cost at the search center. Experimental results demonstrate that the proposed method saves a significant part of computations while providing comparable performance to the conventional FSA. If we combine our method with a fast FSA such as SEA, additional reduction in computation can be simply obtained. In addition, the ASRA-based approach has a strong point with respect to the hardware implementation, since it has a regular structure like the FSA.

### Acknowledgments

This research was supported by the Ministry of Information and Communication (MIC), Korea, under the Information Technology Research Center (ITRC) support program su-

pervised by the Institute of Information Technology Advancement (IITA), IITA-2006-(C1090-0603-0002).

### References

1. J. Jain and A. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.* **29**(12), 1799–1808 (1981).
2. L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.* **9**(2), 313–317 (1996).
3. S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block matching motion estimation," *IEEE Trans. Image Process.* **9**(2), 287–290 (2000).
4. C. Zhu, X. Lin, and L. P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.* **12**(5), 349–355 (2002).
5. W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Process.* **4**, 105–107 (1995).
6. ITU-T Recommendation H.264 | ISO/IEC 14496-10 AVC, "Draft text of final draft standard for advanced video coding" (2003).
7. ISO/IEC JTC1/SC29/WG11, "Working draft of reference software for advanced video coding" (2003).