An Improved Search Algorithm for Motion Estimation Using Adaptive Search Order

Lung-Chun Chang, Kuo-Liang Chung, and Tsung-Cheng Yang

***Abstract*—An adaptive search order (ASO) algorithm is presented in this letter to speed up the block motion estimation in digital video coding. According to the motion trend, a table of the adaptive search order is defined. For each searching iteration, a better search order is derived and then the best matched block can be found in the early search stage. Some experimental results demonstrate the computational advantage of the proposed improved algorithm when compared to previous algorithms, such as the full search algorithm, the successive elimination algorithm, the block sum pyramid algorithm, and the multilevel successive elimination algorithm.**

***Index Terms*—Adaptive search order, motion estimation, motion vector, pyramid data structure.**

I. Introduction

MOTION estimation plays an important role in video coding [4], [5]. The block matching (BM) algorithm is a well known method in motion estimation. In BM algorithm, the encoder first divides the current image frame into some fixed-size blocks, and the motion vector for each block is estimated by finding the closest block in the reference image frame according to the defined matching criterion. That is, for each block in the current image frame, the best matched block within a search window in the reference image frame is to be determined. Let the search window be of size $(2P+1) \times (2P+1)$ and each block be of size $N \times N$ with $N = 2^m$. The matching criterion used in this letter is the accumulate absolute difference (AAD) which is expressed as

$$\mathrm{AAD}(v_x, v_y) = \sum_{i=1}^{N}\sum_{j=1}^{N} |B_c(i,j) - B_r(i+v_x, j+v_y)|$$

for $-P \le v_x, v_y \le P$, where $B_c(i,j)$ denotes the gray value of the pixel at position (i,j) in the current block, and $B_r(i+v_x, j+v_y)$ denotes the gray value of the pixel at position $(i+v_x, j+v_y)$ in the reference block. The best matched block within the search window in the reference image frame is found to be the one with the minimum match error, i.e., minimal $\mathrm{AAD}(v_x, v_y)$.

The full-search (FS) algorithm finds the closest block among all the possible search positions in a search window. Although the FS algorithm can indeed obtain the global optimal result, however the considerable computational cost limits its practical applications. To reduce the computational complexity and obtain the global optimal result, several more efficient search algorithms, such as the multilevel successive elimination (MLSE) algorithm [1], the block sum pyramid (BSP) algorithm [2], and the successive elimination (SE) algorithm [3] have been developed.

Manuscript received October 11, 2000. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. J. Apostolopoulos. This work was supported by the National Science Council, R.O.C., Contract NSC89-2213-E011-062.

The authors are with the Department of Information Management, Institute of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10672, Taiwan, R.O.C. (e-mail: klchung@cs.ntust.edu.tw).

Publisher Item Identifier S 1070-9908(01)02792-4.

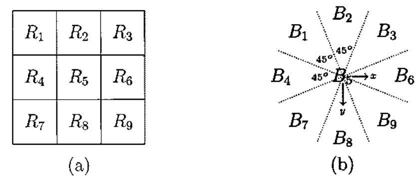

Fig. 1. (a) Nine subregions within the search window and (b) nine bands of the direction of the predicted initial motion vector.

TABLE I
NINE SEARCH ORDER

band	the search order of the nine subregions								
B_1	R_5	R_1	R_2	R_4	R_3	R_7	R_6	R_8	R_9
B_2	R_5	R_2	R_1	R_3	R_4	R_6	R_8	R_7	R_9
B_3	R_5	R_3	R_2	R_6	R_1	R_9	R_4	R_8	R_7
B_4	R_5	R_4	R_1	R_7	R_2	R_8	R_6	R_3	R_9
B_5	R_5	R_4	R_6	R_2	R_8	R_1	R_3	R_7	R_9
B_6	R_5	R_6	R_3	R_9	R_2	R_8	R_4	R_1	R_7
B_7	R_5	R_7	R_4	R_8	R_1	R_9	R_2	R_6	R_3
B_8	R_5	R_8	R_7	R_9	R_4	R_6	R_2	R_1	R_3
B_9	R_5	R_9	R_6	R_8	R_3	R_7	R_2	R_4	R_1

In this letter, a faster search algorithm, called the ASO algorithm, is presented to improve the previous algorithms [1]–[3] while still obtaining the global optimal result. According to the motion trend, we define a table of the adaptive search order within the search window. For each searching iteration, a better search order is derived and then the best matched block can be found in the early search stage. This reduces the search time for obtaining the motion vector. Experimental results reveal that the execution time of the proposed improved algorithm, which includes the preprocessing time and the searching time, is much less than the previous algorithms [1]–[3].

II. Improved ASO Algorithm

Before presenting the improved ASO algorithm for motion estimation, we first sketch the idea used in the proposed ASO algorithm.

Initially, for each current block in the current image frame, the search region within the search window is segmented into nine square nonoverlapping subregions as shown in Fig. 1(a). Further, we predict the initial motion vector of the current block in the current image frame using the approach [6]. The predicted initial motion vector of the current block is used to determine a better search order within the nine subregions.

According to the direction of the predicted initial motion vector, the nine bands of the adaptive search order are defined

TABLE II
COMPARISON OF THE NUMBER OF OPERATIONS

Algorithm	salesman			garden			Susie		
	ADD	ABS	COMP	ADD	ABS	COMP	ADD	ABS	COMP
FS	500222	250601	978	500222	250601	978	500222	250601	978
SE	123014	61625	1218	217007	108709	1402	94727	47454	1162
BSP	11681	6493	1305	50120	25985	1849	12303	6784	1261
MLSE	10708	5516	1305	49155	25007	1849	11322	5806	1261
ASO	7904	4090	1245	10985	5673	1330	7997	4111	1194

as shown in Fig. 1(b). For example, if the predicted initial motion vector is (3, -3), the band B_3 is selected. If the predicted initial motion vector is (0,0), the band B_5 is selected. From the direction of the predicted initial motion vector, the corresponding band can be determined. After determining that band, the nine search order in the search window are listed in Table I. As shown in Table I, the search order of the nine subregions is defined according to the motion trend and the position correlation between the center subregion and its eight neighboring subregions. For example, if the predicted initial motion vector is (3,3), the band B_9 is selected, and the search order of the nine subregions in the search window is arranged as $(R_5, R_9, R_6, R_8, R_3, R_7, R_2, R_4, R_1)$.

According to the idea mentioned previously, and combining some existing techniques, our formal algorithm is presented as shown below.

Algorithm: ASO algorithm

Input: A video sequence

Output: Best motion vectors

Preprocessing: Constructing all block sum pyramids for the reference image frame [1], [2]

begin

for each current block in the current image frame

Step 1: The search region within the search window is segmented into nine square nonoverlapping subregions. In addition, the initial motion vector is predicted using the approach [6]. According to the predicted initial motion vector, the current minimal AAD is calculated.

Step 2: According to the direction of the predicted initial motion vector, one of the nine bands is selected and then a search order of the nine subregions is followed as shown in Table I.

Step 3: After selecting the search order of the nine subregions, the multilevel successive elimination algorithm [1] is employed in the current block and each reference block of the nine subregions. Step 3 is repeated until all reference blocks within the nine subregions are checked.

Step 4: The best motion vector with the minimal AAD is obtained.

end for

end.

Since the adaptive search order method as mentioned in Table I keeps a better search order for these nine subregions, the best matched block can be obtained using less computational effort. Experimental results (See Section III) will confirm the computational saving advantage for finding the best matched block.

III. EXPERIMENTAL RESULTS

In this section, some experiments are carried out to demonstrate the performance among the FS algorithm, the SE algorithm [3], the BSP algorithm [2], the MLSE algorithm [1], and our proposed ASO algorithm. All the concerning algorithms are implemented using Borland C++ builder 4 and the AMD K6-II 350-based PC on the Windows 98 environment. Three typical video sequences, salesman, garden, and Susie, are used as the benchmarks in the experiments. Here, the first 21 frames of the salesman sequence and the first 30 frames of the other two sequences are used. Each image frame is of size 352 $\times$ 256. The block size and the size of the search window are selected as 16 $\times$ 16 and 33 $\times$ 33, respectively.

TABLE III
EXECUTION TIME COMPARISON

Algorithm	salesman	garden	Susie
FS	204.66	296.96	296.99
SE	51.72	129.81	60.38
BSP	10.17	38.56	15.01
MLSE	9.75	37.19	14.34
ASO	8.83	14.57	12.81

Table II shows the average number of operations required in each current block for finding the best matched block in terms of number of additions, namely, ADD, number of absolutes, namely, ABS, and number of comparisons, namely, COMP. Here, we assume that the time required to perform one addition is equal to that of one subtraction. As can be seen from this table, the proposed ASO algorithm has least ADD and ABS when compared to that of the FS algorithm, the SE algorithm, the BSP algorithm, and the MLSE algorithm. For COMP, the proposed ASO algorithm outperforms the BSP algorithm and the MLSE algorithm, but is in competition with the SE algorithm. However, the total execution time required in the proposed ASO algorithm is the least one among all the concerning algorithms. Table III lists the execution time comparison. Each execution time includes the preprocessing time, the time of the motion prediction, and the searching time for the three given video sequences, where the time unit is "second." In Table III, it is observed that the execution time of the ASO algorithm is the least.

REFERENCES

[1] 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, pp. 501–504, Mar. 2000.

[2] C. H. Lee and L. H. Chen, "A fast motion estimation algorithm based on the block sum pyramid," *IEEE Trans. Image Processing*, vol. 6, pp. 1587–1591, Nov. 1997.

[3] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Processing*, vol. 4, pp. 105–107, Jan. 1995.

[4] K. R. Rao and J. J. Hwang, *Techniques and Standards for Image, Video, and Audio Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[5] A. M. Tekalp, *Digital Video Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[6] J. B. Xu, L. M. Po, and C. K. Cheung, "Adaptive motion tracking block matching algorithms for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1025–1029, Oct. 1999.