An Efficient Parallel Motion Estimation Algorithm for Digital Image Processing

Liang-Gee Chen, Wai-Ting Chen, Yen-Shen Jehng, and Chih-Ta Chuch

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 10764, R.O.C.

This paper presents an efficient parallel block matching algorithm called the parallel hierarchical one-dimensional search for motion estimation. This algorithm is based on the assumptions made by the one-at-a-time search and 2-D logarithmic search. Instead of finding the two-dimensional motion vector directly, it finds two one-dimensional displacements in parallel on the two axes independently within the search area. The major feature of this algorithm lies in the fact that its search speed for the motion vector is faster than that of the other search algorithms on account of its simpler computations and parallelism.

I.Introduction

In digital image processing, e.g., video conferencing application, the significantly high correlation between consecutive frames can be exploited more efficiently by considering the displacements of moving objects in the coding process. Therefore, in any motion-compensated coding scheme, the performance of the real-time system heavily depends on the accuracy and speed of the motion estimation.

Though, the block matching algorithm(BMA) is much more realizable according to its computational simplification [2]. However, one difficulty with the BMA is the extensive computations required by the full search(FS). To alleviate the computational overhead of the FS, several techniques such as the three-step hierarchical search(3SHS)(adopted by [5]), the one-at-a-time search(OTS) [1], the 2-D logarithmic search(LOGS) [2] and a modified version of it (MLOGS) [1], have been developed. Although the strategies of these algorithms could reduce the computational complexity, they also suffer from the cost of irregular control flow when hardware realization is taken into consideration.

The objective of this paper is to propose an efficient hardware-oriented BMA, which involves considerably simpler arithmetic and regular control flow. The search

CH 3006-4/91/0000 – 0670 $1.00 © IEEE

procedure and properties of the proposed algorithm are described in Section II. The simulation results from applying the proposed algorithm and other BMA's for four measurements on a video sequence are reported in Section III. Conclusions are presented in Section IV.

II.The Proposed Algorithm

To reduce the computational complexity, all the search algorithms, except the FS, assume that the distortion increases monotonically as the searched point moves away from the direction of minimum distortion [4]. Based on the assumptions made by the OTS [1] and the LOGS, the parallel hierarchical one-dimensional search(PHODS) is presented to reduce the number of sequential steps and search points. Under such assumptions, we need not make too much effort to find out the exact two-dimensional displacement of the moving object. Instead of searching the two-dimensional motion vector directly, the PHODS locates two one-dimensional displacements in parallel on two axes(say x and y) independently within the search area. Thus the motion vector $(\Delta x,\Delta y)$ is obtained from the results of the x-axis search $(\Delta x,0)$ and the y-axis search $(0,\Delta y)$ concurrently. The mean absolute error(MAE) is chosen as the matching criterion [2],[3] for the PHODS and all the algorithms being compared in the simulation because of its simpler computational complexity [1].

The search procedure of the PHODS, as illustrated in Fig. 1, is described as follows:

initialization: $S=2^{\lfloor \log_2 p \rfloor}$;

/* step size S=4 when p=4-7 */

$\Delta x=0$; $\Delta y=0$;

while (S is larger than zero)

/* Search loop, executed $\lfloor \log_2 p \rfloor+1$ times */

IN PARALLEL

/* $\lfloor . \rfloor$ is a lower integer truncation function */

x-axis : $(\Delta x,0)$<-- the location of

min($D(\Delta x-S,0)$, $D(\Delta x,0)$, $D(\Delta x+S,0)$);

y-axis : $(0,\Delta y)$<-- the location of

min($D(0,\Delta y-S)$, $D(0,\Delta y)$, $D(0,\Delta y+S)$);

S=S/2;

end while_loop;

the motion vector is $(\Delta x,\Delta y)$;

Table 1 presents a comparison of the number of search points and sequential steps required by the algorithms mentioned above. For a real-time hardware realization, the number of the required sequential steps can be a more important feature than the number of search points, since some of these can be evaluated by parallel computations .

Furthermore, the PHODS possesses the following features: 1) regularity: static sequential steps easily bring into control and regular data flow when real-time hardware implementation is considered; 2) simplification: the less required number of search points per sequential step can reduce the computational overhead; 3) parallelism: the x- and y-axis searches could be parallel processed, a characteristic that can shorten the motion vector search time.

Since the computations of all the algorithms, except the FS, are not continuous processes, any step must complete its computation before the next step begins. From the hardware point of view, the inherent feature of the discontinuity of all the algorithms will delay the search time for the motion vector because of the hardware latency. Fortunately, owing to the parallelism of the PHODS, the problem could be solved easily by pipeline interleaving the x- and y-axis searches when hardware realization is taken into consideration.

III.Simulation Results and Performance Comparisons

To test the performance of the PHODS, a sequence(a speaker with slow movements) containing 16 frames with a frame sampling rate at 12.5 Hz, with each frame possessive of 256*256 pels and 8-bit resolution for each pel, is used in the simulation. The simulation is performed on an IBM PC/AT personal computer. The comparisons are made using six search algorithms, namely i)FS ii)3SHS iii)LOGS iv)MLOGS v)OTS vi)PHODS, in terms of the following four measures: 1)PSNR(peak-to-peak SNR), shown in Fig. 2. 2)entropy, shown in Fig.3. 3)the percentage of unpredictable pels(pels with absolute prediction errors larger than three, over a range of 255, are classified as unpredictable pels [2]), shown in Fig.4. 4)search time(CPU time calculated only), shown in Fig. 5.

IV. Conclusions

In this paper, an efficient parallel search algorithm is presented for motion estimation. It is computationally simpler and performs about the same as the other motion estimation techniques reported. Most of the hardware proposed recently is applicable to the FS only [6] [7], since the search strategies require a control overhead for the non-regular dataflow and will delay the search times for the motion vectors caused by the hardware latency. Thanks to the parallelism of the PHODS, the problem brought about by the inherent feature of the discontinuity of the search strategies could be relieved. Furthermore, the regularity, simplification and parallelism of the PHODS strongly imply that it is quite suitable for and efficient in VLSI implementation when used as a low-bit rate video coder. At last, a systolic architecture based on the PHODS is currently under development for real-time motion estimation.

V.References

1. R. Srinivasan and K. R. Rao, "Predictive Coding Based on Efficient Motion Estimation," IEEE *Trans. Commun.*, vol. COM-33, NO. 8, pp. 888-896, Aug. 1985.

2. H. G. Musmann, P. Pirsch, and Hans-Joachim Grallert, "Advances in Picture Coding," *Proc.* IEEE, vol. 73, NO. 4, pp. 523-548, 1985.

3. H. Gharavi and Mike Mills, "Blockmatching Motion Estimation Algorithms-New Results," IEEE *Trans. Circuits and Systems*, vol. 37, No. 5, pp. 649-651, May 1990.

4. A. N. Netravali and B. G. Haskell, *Digital Pictures Representation and Compression.* New York: AT & T Bell Lab., 1988.

5. Ronald Plompen, Yoshinori Hatori, Wilfried Geuen, Jacques Guichard, Mario Guglielmo, and Harald Brusewitz, "Motion Video Coding in CCITT SG XV-The Video Source Coding," (Globecom88), pp. 997-1004, 1988.

6. T. Komarek and P. Pirsch, "Array Architectures for Block Matching Algorithms," IEEE *Trans. Circuits and Systems*, vol. CAS-36, No. 10, pp. 1301-1308, Oct. 1989.

7. K. M. Yang, M. T. Sun, and L. Wu, "A Family of VLSI Designs for the Motion Compensation Block-Matching Algorithm," IEEE *Trans. Circuits and Systems*, vol. CAS-36, No. 10, pp. 1317-1325, Oct. 1989.

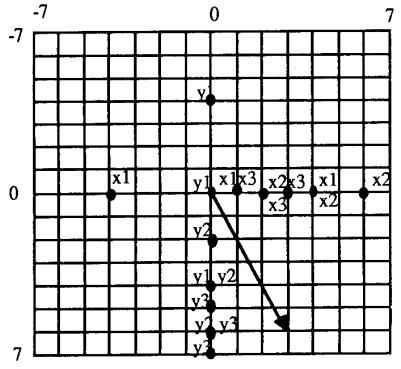

Fig. 1. The parallel hierarchical one-dimensional search procedure. The motion vector is (3,6) in this example.

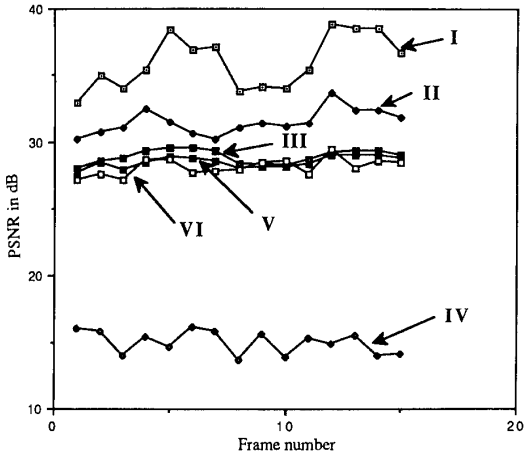

Fig. 2. PSNR comparison on prediction errors for algorithms I-VI.
I) FS. II) 3SHS. III) LOGS. IV) MLOGS. V) OTS. VI) PHODS.

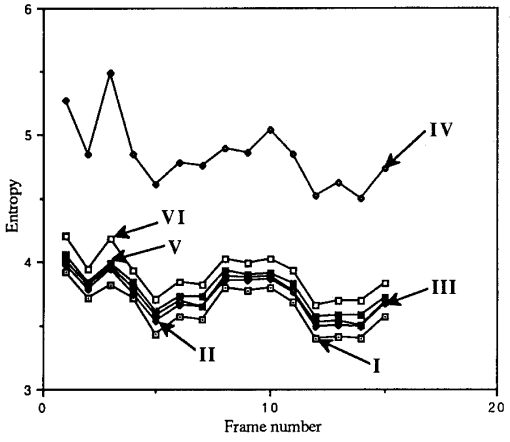

Fig. 3. Entropy of prediction errors for algorithms I-VI.
I) FS. II) 3SHS. III) LOGS. IV) MLOGS. V) OTS. UI) PHODS.

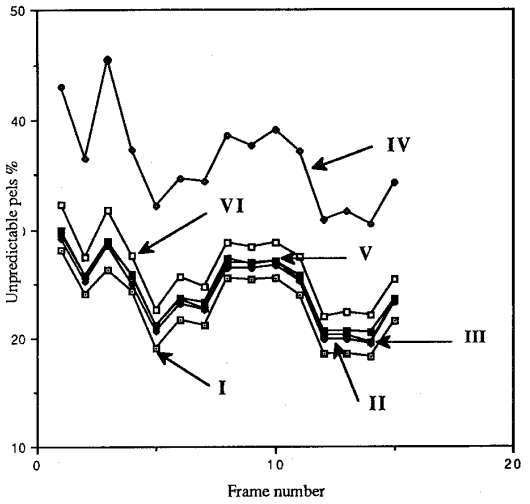

Fig. 4. Percentages of unpredictable pels for algorithms I-VI.
I) FS. II) 3SHS. III) LOGS. IV) MLOGS. V) OTS. VI) PHODS.

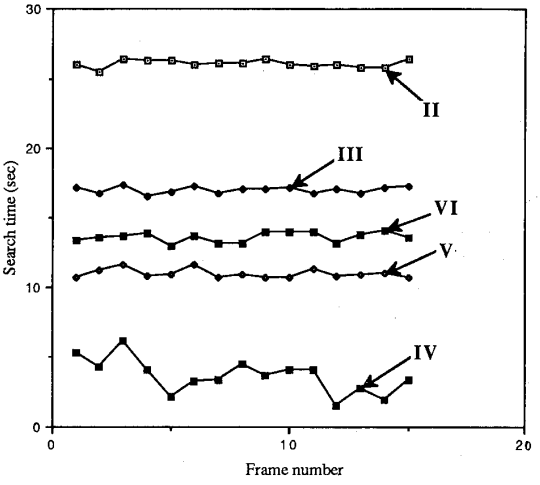

Fig. 5. Search times for algorithms II-VI on IBM PC/AT.
*The search time of the PHODS should be reduced by 1/2, when parallel processing is considered.
II) 3SHS. III) LOGS. IV) MLOGS. V) OTS. VI) PHODS.

case / algorithm	examples in Fig. 1-5 a	examples in Fig. 1-5 b	worst case for p=7 a	worst case for p=7 b
II.3SHS	3	25	3	25
III.LOGS	6	22	8	28
IV.MLOGS	6	17	7	19
V.OTS	11	14	14	17
VI.PHODS	3	13	3	13

a) Required number of sequential steps.
b) Required number of search points.

Table.1. Comparisons of the number of sequential steps and search points for algorithms II-VI.