An Improved Algorithm for Fast Block Motion Estimation

FLORIN ALEXA[1] , ZORAN BOJKOVIC[2] DAN L. LACRÃMÃ [3] and NIKOS MASTORAKIS[4]
[1,3)] Communication Department
"Politehnica" University o Timisoara
Bd. V. Pârvan, no.2, 1900 Timisoara
ROMÂNIA
[2)]Faculty of Traffic and Transport Engineering, University of Belgrade, YUGOSLAVIA
4) Military Inst.of University Education, Hellenic Naval Academy, Terma Hatzikyriakou, Piraeus, GREECE

Abstract: - Experimental results show that the block motion field of real image sequence is usually gentle, smooth and varies slowly, as result in a center-biased global minimum motion vector distribution instead of an uniform distribution. Based on these characteristic of image sequence a new four step search (FSS) algorithm with center biased checking point pattern for fast block motion estimation is proposed in this paper. Variable searching step technique is employed in the proposed algorithm with minimum searching steps of 2 and 4 for the maximum. The total number of points is varied from 17 to 27. Simulation results show that, as compared to the well-known three-step search the proposed FSS produces smaller motion computational requirement. Also, the proposed algorithm possesses the regularity and simplicity of hardware oriented features.

Key-Words: - Algorithm, Fast block motion estimation, search *CSCC'99 Proceedings,* pp.7251-7254

1 Introduction

Motion estimation (ME), in general, can improve the prediction accuracy between adjacent frames/ pictures. This technique falls into two categories:

- pel-by-pel ME, called pel-recursive algorithms (PRA)
- block-by-block ME, called block matching algorithms (BMA).

PRAs have been rarely used because they are inherently complex and the ME algorithms sometimes run into convergence problems. Object based ME is tailored for image coding. The ME is dependent on identifying or locating the object, or at least its boundaries, in both frames pictures.

In block matching algorithm, motion of a block of pels, say (N×N), within a frame/field interval is estimated. The range of the motion vector is constrained by search window. The current image frame is firs partitioned into equal-sized rectangular blocks. The motion vector for each block is estimated by searching for its peak correlation with an associated block within a search window in the previous frame.

Block motion estimation had been used in different video compression standards such as: H.261, H263, H263+, MPEG-1, MPEG-2, MPEG-4 [1], [2]. One common feature of these standards is that both of which use DCT transform coding to reduce spatial redundancy and block based motion estimation/ compensation to reduce the temporal redundancy. In addition, the encoder complexity of these standards is dominated by the motion estimation if the full search algorithm is used as the block-matching algorithm (BMA). Full search matched every possible displaced candidate block within the search area in the reference frame in order to find the block with minimum distortion. Massive computation is, therefore, required in the implementation of full search algorithm [3]. On the other hand H.261 and MPEG standards have not specified the BMA for motion estimation at the encoder. As a result many fast BMA's have been developed to alleviate the heavy computation of full search algorithm.

Among the proposed BMA's the three-step search (TSS) [4], becomes the most popular one and RM8 of the H.261 and SM3 of MPEG also recommend it owing to its simplicity and effectiveness. However, the TSS uses a uniformly allocated checking point pattern in its first step, which becomes inefficient for the estimation of small motion. In addition, experimental results shows that the block motion field of real image sequence is usually gentle, smooth, and varies slowly. It results in a center biased global minimum motion vector distribution instead of a uniform distribution [4]. For "Mobile" sequence, there are 80% blocks that can be regarded as stationary or quasi-stationary blocks and most of the motion vectors are enclosed in central 5×5 area Fig. 1.

Fig. 1. Motion vector for Mobile4 sequence

For the "Miºa" sequence Fig.2, which consists of faster motion, although the motion vector distribution is more diverse as compared to "Mobile", it is still highly center biased.

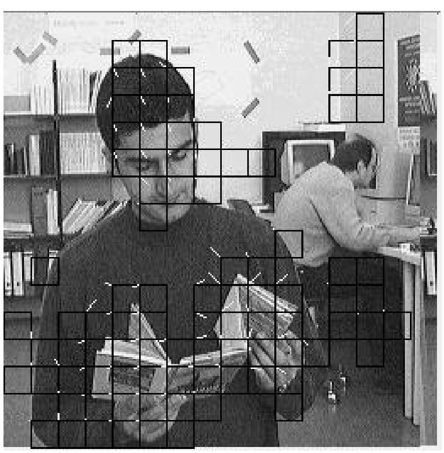


Fig. 2. Motion vector for Miºa 15 sequence

Based on the characteristic of center biased motion vector distribution, a new three step search (NTSS) algorithm for improving the performance of the TSS on the estimation of small motion was proposed in [5]. The NTSS algorithm employed a center biased checking point pattern in the first step, that combined the original checking point used in TSS algorithm and 8 extra neighboring points of the search window's center.

The NTSS used a halfway-stop technique to speed up the stationary or quasi-stationary blocks matching. Simulation results in [5] show that the NTSS is much more robust, produces smaller motion compensation errors as compared with three-step search. For the maximum displacement of ± 7, however, the NTSS algorithm, in the worst case, requires 33 block matches while TSS need only 25 block matches. Therefore, the computational requirement of NTSS for some image sequences may be higher than TSS even in terms of average computational complexity. On the other hands, for the real time or VLSI implementation of motion estimation the worst case computational requirement should be considered instead of the average computation [6]. In general moving scenes, it is very likely that a homogenous area of the picture frames moves in the same direction with similar velocities. Therefore, the displacements between neighboring blocks are highly correlated or dependent [7]. In application to image sequence coding, the BMA has systematic limitation owing to rotation, deformation, etc. [8]. Another problem of the BMA is the restricted search range, which may result in inaccurate results when the motion of the object extends beyond the maximum search region. One solution for this problem is to enlarge scope of the search region. In this paper, a new four step search (FSS) algorithm based on the center-biased motion vector distribution characteristic is proposed. The proposed FSS produces better performance than TSS and have similar performance as compared with the NTSS. While the worst case computational requirement of the FSS is 27 block matches that is just 2 more matches as compared with TSS.

2 Four step search algorithm

For the maximum motion displacement of ±7, a motion vector with minimum distortion is to be found out according to a predefined block distortion measure (BMD). The mean absolute error (MAE) is the most commonly used BMD for block based motion estimation.

The proposed FSS algorithm utilized the center biased search pattern with 9 checking points on a 5×5 window as shown in Fig. 3(a) is used in the first step of the FSS algorithm instead of 9×9 window in the TSS. The search window size of the next two steps is depended on the location of the minimum BMD point. If the minimum BMD point is found at the center of the previous search window, the search will go to the final step with 3×3 search window. Otherwise, the search window size is maintained in 5×5 in the second/third step. In this case, the search pattern is depended on the position of the previous minimum BMD point. If the minimum BMD point is at the corner of the previous search pattern, 5 additional checking points as shown in Fig. 3(b) are used. If the minimum BMD pint is along the horizontal or vertical axis of the previous search pattern, 3 additional checking points as shown in Fig. 3(c) are used. In the final step, the search window is reduced to 3×3 and the search stops at this small search window. It implies that the intermediate steps of the FSS may be skipped and jumps to the final step with 3×3 window if at any step the minimum BMD point is found at the center of the search window.

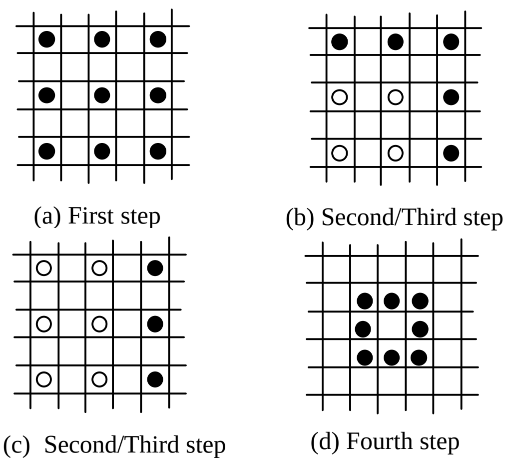


Fig. 3. Search patterns of the FSS

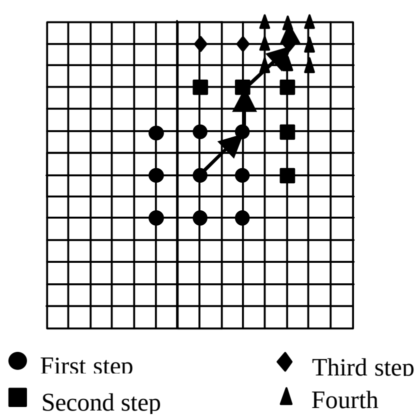
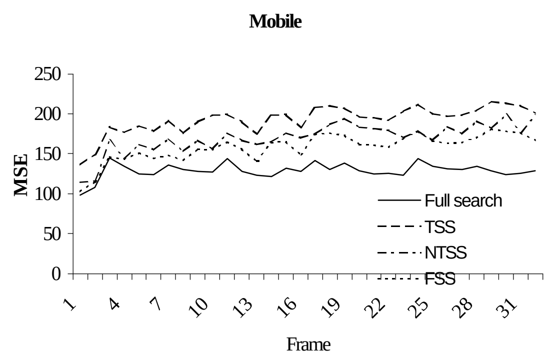


Fig. 4. Search path of FSS

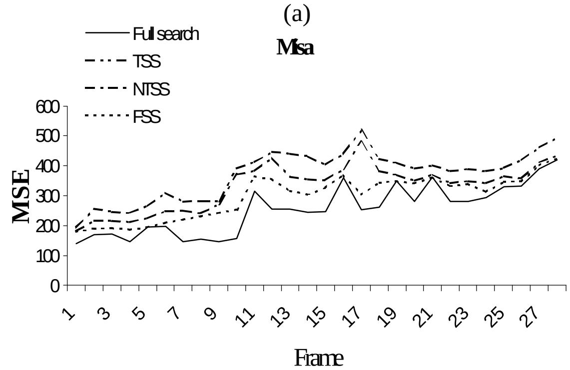
Based on this four step search patterns, we can cover the whole 15×15 displacement window even only small search window 5×5 and 3×3 are used. In Fig. 4, search pattern is shown. The search used a total of 25 checking points and the motion vector is (4, -7). The computational complexity of the FSS is only two more search points than the TSS and 6 block matches less than NTSS in the worst case.

3 Simulation results

The FSS algorithm is simulated using luminance components of the 32 frames of the "Mobile" sequence and 26 frames of the "Miºa" sequence. These two sequences consist of large displacement and fast motion. The size of each individual frame is 256×256 pixels quantised uniformly to 8 bits. The MAE distortion is used as BMD. The maximum displacement of the search area is ±7 pixels in both horizontal and vertical direction for a block size of 16×16. The performance between estimated frames and original frames for FSS, TSS, NTSS and full search in terms of mean square error (MSE) are shown in Fig.5. The MSE comparisons show that the FSS provides the better performance than TSS and similar to the NTSS, especially in the area where fast motion is involved.



(a)



(b)

Fig. 5. MSE comparisons of FSS, TSS, NTSS and Full Search: (a) Mobile sequences (b) Miºa sequences

Table 1. Average MSE of frames

Searching algorithm	Mobile	Miºa
Full search	128,95	253,67
TSS	190,03	370,54
NTSS	167,57	330,78
FSS	156,89	298,35

Table 2. Average searching points needed for each motion vector

Searching algorithm	Mobile	Miºa
Full search	225	225
TSS	25	25
NTSS	19,98	22,66
FSS	18,16	19,56

Table 3. Speed up ratio for fast algorithm

Searching algorithm	Mobile	Miºa
TSS	9	9
NTSS	11,26	9,92
FSS	12,38	11,50

Table 4. Average distance to the motion vector found by the Full search method

Searching algorithm	Mobile	Miºa
TSS	0,871	0,6214
NTSS	0,884	0,7059
FSS	0,908	0,7354

Tables 1 to 4 give some statistical performances comparisons of four BMA's. Table 1 shows the average MSE of the first 32 frames for "Mobile" sequence and for the first 26 frames for "Miºa" sequence, using different algorithms. From the table, we can see that FSS gives the smallest MSE among the other algorithms except full search. Table 2 shows the average searching points needed for each motion vector. From the table, we found that the average searching point for FSS is very near to its minimum 17. However, the average searching points of NTSS is dependent very much on the motion content of the image sequence. The speed up ratio in Table 3 is measured as ratio between searching point for full search algorithm and searching point for fast algorithm.

Table 4 gives us a measure of the accuracy of different algorithms using the motion vectors found by full search as reference. The NTSS and FSS give similar accuracy and they are much better than the conventional TSS.

Those experimental results show that the FSS always provides a better performance with the minimal computational requirement among the three fast BMA's.

4 Conclusion

A new FSS algorithm for fast block based motion estimation is proposed in this paper. The proposed algorithm performs better performance than the well-known TSS and has similar performance to NTSS in terms of mean square error measure. The average computation requirement of FSS is less than TSS and NTSS. In addition, the FSS is much more robust as the performance. It is nearly independent of the motion pattern of a given image sequence. When the sequence involves much complex movement, the improvement in performance is much more obvious. On the other hand, the FSS also possesses the regularity and simplicity of hardware oriented features.

References:

[1] K.R. Rao and J.J. Hwang, *Techniques and Standards for Image, Video and Audio Coding*, Prentice Hall, 1996

[2] Z.S. Bojkoviæ , C.I. Toma, V. Gui and R. Vasiu, *Advanced Topics in Digital Image Compression*, "Politehnica" Press, Timiºoara, 1997

[3] J.R. Jain, A.K. Jain, Displacement measurement and its application in interframe image coding, *IEEE Trans. on Communications*, vol. COM-29, No.12, pp.1799-1808, Dec.1981

[4] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Isiguro, Motion Compensated Interframe Coding for Video Conferencing, *Proceedings of National Telecommunication Conference,* New Orleans, pp. G5.3.1-G5.3.5, 1981

[5] R. Li, B. Zeng and M.L. Liou, A new three-step search algorithm for block motion estimation, *IEEE Trans. on Circuits and Systems for Video Technology*, vol.4, No. 4, pp. 438-442, Aug. 1994

[6] S.Y. Kim, J.H. Lee and S.D. Kim, VLSI Realisation of Symmetric Block Matching Algorithm, *Signal Processing of HDTV IV*, Elsevier Science Publishers, pp.301-308, 1993

[7] J.Ch. Tsai, C.H. Hsieh, S.K. Weng and N.F. Lai, Block-Matching Motion Estimation Using Correlation Search Algorithm, *Signal Processing: Image Communication*, vol.13, No.2, pp.119-134, Aug. 1998

[8] D.W. Kim, J.S. Choi and J.T. Kim, Adaptive Motion Estimation Based on Spatio-Temporal Correlation, *Signal Processing: Image Communication*, vol.13, No.2, pp.161-170, Aug. 1998