RESEARCH ARTICLE OPEN ACCESS

Fast Computational Four-Neighborhood Search Algorithm For Block matching Motion Estimation

Ibrahim Nahhas*
*(Faculty of Electrical and Electronic Engineering, University of Aleppo, Syria)

ABSTRACT
The Motion estimation is an effective method for removing temporal redundancyfound in video sequence compression. Block Matching algorithm has been widely used in motion estimation and a number of fast algorithms have proposed to reduce the computational complexity of BMA. In this paper we propose a new search strategy for fast block matching based on Four-Neighborhood Search (FNS) and fast computational strategy, this new algorithm can significantly speed up the computation of the block matching by reducing the number of checked points and the time computational. Results have been shown that 89% to 93% of operations can be saved while maintaining the quality of video relative to full search algorithm.
Keywords– Block matching, Motion estimation, Four-neighborhood search.

I. INTRODUCTION

The videocompression depends on removing the redundancy in the temporal, spatial and/or frequency domains [1]. The goal of the motion estimationis to reduce temporal redundancy between successive frames. Changes between two successive video frames may be caused by object motion, camera motion and lighting changes. It is possible to estimate the trajectory of each pixel between successive video frames, producing a field of pixel trajectories known as map of motion vectors. To have a practical method of motion estimation, we segment the actual frame into non-overlapped, equally paced, fixed size small rectangular sections, called „blocks“, and determine all pixels inside the block to have the same motion vector to estimate the movement of „blocks“ of the current frame. The block matching technique is simple, straightforward, and very efficient. It has been by far the most popularly utilized motion estimation technique in video coding. In fact, it has been adopted by all the international video coding standards: ISO MPEG-1 and MPEG-2, and ITU H.261, H.263 and H.264.This paper presents new strategy in BMA depending on combining of Four-Neighborhood search[2] and fast computational Full search[3] to reduce the complexity and time of computational which benefit for real time video applications.

II. BLOCK MATCHING ALGORITHMS

The block matching algorithms segment the current frame into blocks and determine of all pixels inside the block have the same motion vector, which was estimated by finding its best-matched counterpart in the previous frame. The block size needs to be chosen properly. In general, the smaller the block size, the more accurate, but leading to more motion vectors to be estimated and encoded, which means an increase in time computational and side information. As a compromise, a size of 16×16 pixels is considered to be a good choice – this has been specified in the international video coding standards such as H.261, H.263 and MPEG-1, MPEG-2 [4][5]. The Fig. 1 illustrates the principle idea of block matching technique, where segment an frame at the moment$t_n$into non-overlapped $p\times q$ rectangular blocks.

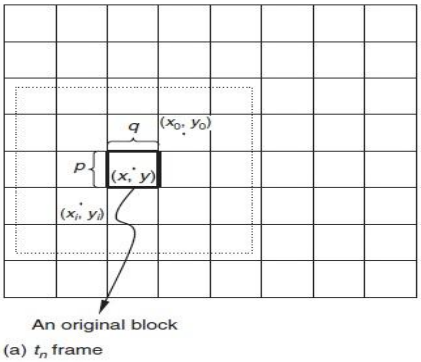

Fig.1 Principle idea of Block Matching.

Consider one of the blocks centered at (x, y). It is assumed that the block is translated as a whole. Consequently, only one motion vector needs to be estimated for this block. In order to estimate the motion vector, a rectangular search window is opened in the frame t_{n-1} and centered at the pixel (x, y) as in Fig. 2, than a rectangular correlation window of the same size $p\times q$ is opened with the pixel located in its center. A certain type of similarity measure (correlation) is calculated. After this matching process has been completed for all candidate pixels in the search window, the correlation window corresponding to the largest similarity becomes the

best match of the block under consideration in frame t_n and there are various weighted functions to calculate the best matching, computationally expensive is Mean AbsoluteDifference (MAD) [6] given by equation (1), Mean Squared Error (MSE) [4] given by equation (2) and Sum of Absolute Difference (SAD) [3] given by equation (3).

$$MAD = \frac{1}{N^2}\sum_{j=0}^{N-1}\sum_{i=0}^{N-1}\left|C_{ij} - R_{ij}\right| \qquad (1)$$

$$MSE = \frac{1}{N^2}\sum_{j=0}^{N-1}\sum_{i=0}^{N-1}\left(C_{ij} - R_{ij}\right)^2 \qquad (2)$$

$$SAD = \sum_{j=0}^{N-1}\sum_{i=0}^{N-1}\left|C_{ij} - R_{ij}\right| \qquad (3)$$

Where N is the side of the macro bock, C_{ij} and R_{ij} are the pixels being compared in current block and reference block, respectively. The relative position between these two blocks (the block and its best match) gives the motion vector.

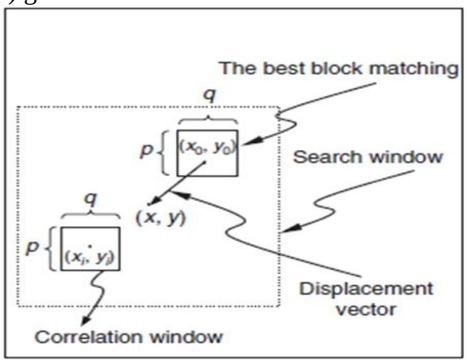

Fig2. Matching process

2.1 Fast Block matching algorithms

Full Search [10] algorithm is the first and simple algorithm for block matching. In a FullSearch (FS) the correlation window moves to each candidate position within the search window and the minimum dissimilarity gives the best matching. FS provides the highest PSNR but at the same time suffers from long computational time so needs an improvement with maintaining the same PSNR and there are number of block matching algorithms which have been developed to accelerate the block matching process to reduce the search time which poses great challenge for real-time codec implementation. We classify these techniques into three categories:

- Partial Search Set techniques reduce the number of the searched points.
- Partial-Matching-Error techniques reduce the computational cost of the matching error for each search points.
- Hybrid techniques are combination of first and second categories to further improve the efficiency of search techniques.

III. FOUR-NEIGHBORHOOD SEARCH ALGORITHM

Many fast block matching algorithms had been developed to reduce the time and complexity of computationalrelative to Full Search (FS). For example the three-step search (TSS) [7], Four step search (4SS) [8] and Diamond search (DS) [9], etc. Among the proposed block matching algorithms, the TSS became the most popular one and it is also recommended by RM8 of H.261 and SM3 of MPEG 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 motions. It results in a center-biased global minimum motion vector distribution instead of a uniform distribution. A new four-neighborhood search (FNS) algorithm with center-biased checking point pattern for fast block motion estimation is proposed[2]**.** Halfway-stop technique is employed in the new algorithm with searching steps of 2 to 5 and the total number of checking points is varied from 5 to 25 with variable pattern of size step of 1 to 4. The basic idea behind the proposed Four-Neighborhood search (FNS) is to reduce the number of checked position and computational cost of the matching error [2]. Four-Neighborhood Search depends on center biased searching with a variable pattern size of step. For the maximum motion displacements of ±7, the proposed FNS algorithm utilizes a center-biased search pattern with 5 checking points on a 3×3 window in the first step instead of a 9×9 window in the TSS. The center of the search window is then shifted to the point with minimum matching measure. The size of search window in the next steps depends on the location of the point which has the minimum matching measure. If the minimum matching measure point is found at the center of the search window, the search will go to the second step with 5×5 search window and the size of search window still increase two by two as long as the center of the search window has the minimum value of matching measure however if the minimum matching measure point is found at one of four-neighborhood points, the center of search window will shift to this point and the size of search window return to 3×3 again to repeat this procedures even have best matching. In this strategy of searching, there will be reduced the total number of checked points relative to full search and other fast search algorithms. To reduce the computational cost of the matching measure, this algorithm uses two values of matching measure and compares the value of matching measure in each checked point to decide continuation in calculating of matching measure or stop to move on next point.These two values, the first one is the ideal value and when matching measure achieve this value the searching procedure stops directly while the second one is the updated value

which has initial value from matching measure between the centered point at actual frame and the same position on at reference frame and this value changes during the matching process depending on the minimum matching measure at every checked point. The FNS algorithm is summarized as follows:

Step 1: Open a 3×3 search window with size step is 1 located at the center of the 15×15 searching area.

Step 2: Find a minimum matching measure point from a 5 checking points (center points and its four neighborhood points) on a search window. Determine the minimum matching measure as initial value of updated threshold and compare it with the ideal value to decide, if it achieves the ideal value go to step 5 and if otherwise continue searching procedure and now if the minimum matching measure point is found at the center of the search window, go to Step 4; otherwise go to Step 3.

Step 3: Move the center of search window to checked point which has minimum matching measure and go to step 2.

Step 4: Increase the size of search window two by two and the size step by 1and go to step 2.

Step 5: The updated value of matching measure is minimal value and its checked point is best matching point.

Fig.3 and 4 illustrate the principle of the FNS. In these scenarios, each step involves a comparison between five points; the size and position of search window depend on the value of matching measure during matching process as described in FNS above.

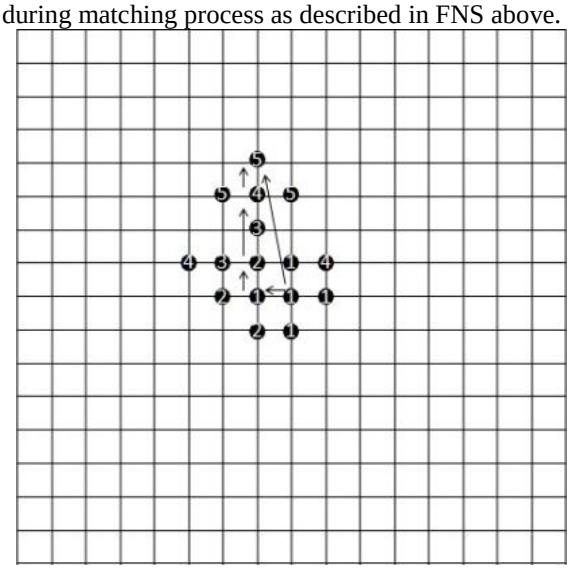

Fig3. Scenario 1

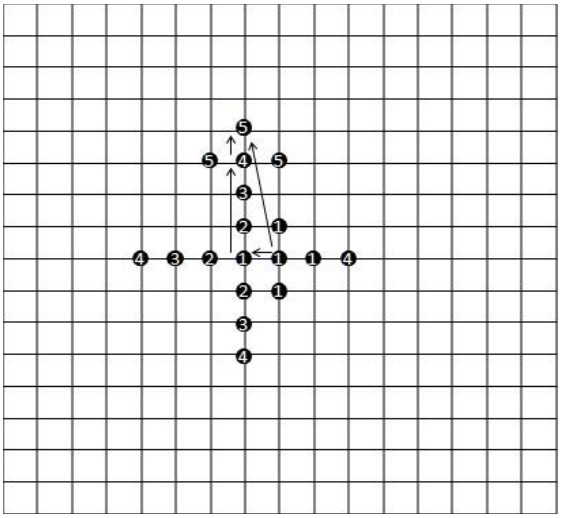

Fig.4 Scenario 2

IV. FAST COMPUTATIONAL FULL SEARCH (FCFS)

Fast computation Full Search [3] used in Full search algorithm but we can implement it in FNS Algorithm where keeps the same resolution of decompression video as FNS block-matching algorithm while decreasing the computational time required to determine the matching block from the reference frame to the current block. The principle of this algorithm is: first make a simple check to detect whether a candidate block is possible to be the best matching one and stop the calculation of cost function between the pixels when the current uncompleted sum absolute value is greater than the previous calculated one, so only the potential candidate blocks are further processed on detailed distortion calculation – so a large part of unnecessary computation for impossible candidate block can be avoided. The proposed algorithm works as follows:

Step 1: Compute the sum of absolute difference (SAD_{min}) between the current macro-block MB and the block at the same location in the reference frame, put the amount to be the minimum SAD.

Step 2: Compute the sum of absolute difference between pixels of next candidate block and the current block, if the summation exceeds the SAD_{min} then stop computing the SAD for the rest of the pixels and go to step 3, otherwise continue the process of computing SAD for the rest of pixels until the pixels of the -block are finished go to step 4.

Step 3: Move to the next macro-block in the search area and go to step 2.

Step 4: Assign the new SAD from step 2 to the SAD_{min} and move to the next candidate MB then go to step 2.

Step 5: The last SAD_{min} will give the matching block.

Fast Computational technique algorithm improves the computational time todetermine the matching block without compromising the quality of the FNS Search maintaining the same number of searched points and the modified algorithm is called Fast Computational Four Neighborhood Search (FCFNS) .

V. Resutls

To assess the performance of the (FCFNS) algorithm, we compared these algorithms with other fast search algorithms. The comparisons of fast block matching search algorithms were based on implementation the different algorithms on using luminance popular video sequences of 50 frames using CIF formats (352×288) with large motion activity "Stefan" video sequence as shown in Fig.5. Table 1 indicates the PSNR difference (ΔPSNR) between Full Search and other fast search algorithms for "Stefan" video for the first 50 frames, average number of searched positions using the 7 block matching algorithms and speed up of all 7 algorithms.

Fig.4 Stefan Video Frame

	FS	FCFS	TSS	NTSS	FSS	DS	FNS	FCFNS
ΔPSNR	0.00	0.00	-0.24	-0.26	-0.29	-0.25	-0.23	-0.23
SP	225	225	25.0	24.0	22.0	20.0	17.5	17.5
Speed up	1.00	1.60	9.00	9.37	10.22	11.25	12.85	14.45

Tabel.1 Results using the "Stefan" video sequence.

VI. Conclusion

A Fast Computational Four-neighborhood search algorithm (FCFNS) was proposed. This algorithm significantly speeds up the block matching procedure and substantially decreases the checked points and computational time, when compared with fast search algorithms, still providing similar quality performances. As a consequence, it was proven as to be specially suited to be implemented in most embedded systems with restricted computational requirements, i.e. it is often adopted by portable devices and for real time applications.This paper presents a new algorithm for fast block matching based on Fast Computational Four-neighborhood search (FCFNS), this algorithm can significantly speed up the computation of the block matching by reducing the number of checked points and improve the performance of FNS by combining with fast computation strategy to reduce the time computational. results has been shown that 89% to 93% of operations can be saved while maintaining the quality of video.

REFERENCES

[1] Iain E.G. Richardson, Video Coding Concepts, in *H.264 and MPEG-4 Video Compression*, Essex, England, Wiley, pp. 27-42,2003.

[2] I. Nahhas, *A New Algorithm for Fast Block-Matching Motion Estimation Based on Four-Neighborhood Blocks*, EEICT, CZ, p. 6, 2013.

[3] Z. Ahmed et al., *Fast Computations of Full Search Block Matching Motion Estimation (FCFS)*, PGNeT Conference, 2011.

[4] A. Barjatya, *Block Matching Algorithms For Motion Estimation*, Dept. of Electrical Engineering, Utah State University, Utah, DIP 6620, 2004.

[5] D.V. Manjunatha et al, *Comparison And Implementation Of Fast Block Matching Motion Estimation Algorithms For Video Compression*, International Journal of Engineering Science and Technology, Vol. 3, No. 10, p. 5, 2011.

[6] C.M. Lin and S.C. Kwatra, *Motion compensated interframe color image coding*, International Conference on Communications,Amsterdam, Part 1, pp. 516 –520, 1988.

[7] T. Koga et al., *Motion compensated interframe image coding for video conference*, Proceedings of NTC81,1998.

[8] L.M. Po and W.C. Ma, *A Novel Four-Step Search Algorithm for Fast Block Motion Estimation*, IEEE Transactions of Circuits And Systems For Video Technology, Vol. 6, No. 3, pp. 313-317,(1996).

[9] S. Zhu and K.K. Ma, *A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation*, IEEE Transactions of Image Processing, Vol. 9, No. 2, pp. 287-290, 2000.

[10] M. Ahmadi and M. Azadfar, *Implementation of fast motionestimation algorithms &*

comparison with full search method in H.264, IJCSNS International Journal of Computer Science & Network Security, Vol. 8, No. 3, pp. 139-143, 2008.

BIBLOGRAPHY

Ing. Ibrahim Nahhas graduated in 2008 at the University of Aleppo, Faculty of Electrical & Electronic Engineering – Communication Systems Department, in Syria. He had his diploma thesis of communication engineering. He is now a Ph.D. Student at the Brno University of Technology, Faculty of Information Technology in the Czech Republic.His research topics include image and video processing, video compression standards and motion estimation and block matching. For more information –see please http://www.fit.vutbr.cz/~inahhas/.