

LOGARITHMIC SEARCH FOR MOTION ESTIMATION

¹SALONI R. MISTRY, ²HENI S. MODI, ³RAHUL N. GONAWALA

^{1,2}B.E. Student, ³Assistant Professor

E-mail: mistrysaloni15@gmail.com, henymodi@gmail.com, rgonawala@gmail.com

Abstract- A video sequence consists of series of frames. The Block Matching is a temporal compression technique used in the video encoding. The main purpose of this method is to determine the displacements of each block of pixels between two successive frames. Motion estimation (ME) is to be done in the encoder side to find the best MV so that it can be applied on stored frames by motion compensated block to generate the predicted video in video compression. Widely motion estimation is used to reduce temporal redundancy in many standards. This technique, performed in the step of motion estimation, consumes the majority of the total time and mathematical complexity of video coding. This paper presents logarithmic search algorithm for block based motion estimation. The performance of the algorithm is evaluated using standard video sequences and the results are compared to a traditional full search algorithm (FSA) and three step search (3SS) algorithm. The estimation considers the accuracy of the matching procedure with respect to the time requires. Moreover, the PSNR of the Logarithmic Search (LS) method is as good as that of the full search method and 3SS. With this, LS can be used as an alternative to FSA and 3SS.

Keywords- Logarithmic Search, Block Matching Algorithm, Video Compression, Motion Estimation

I. INTRODUCTION

Block matching techniques consist of three main components: Block determination, Search methods and Matching criteria. For each reference block in the current frame, BMA searches for the best matched block within a search window of size $(2W+N) \times (2W+N)$ in the previous frame, where W stands for the maximum allowed displacement and $N \times N$ is number pixels in macro-block. Then the relative position between the reference and its best matched block is represented as the motion vector of the reference block. The basic procedure involves coding the initial frame and then tracking the trajectories traversed by the various objects. Through this ample magnitude of compression is achieved with this. A video sequence typically contains temporal redundancy; that is, two successive pictures are often very similar except for changes encouraged by object movement, illumination, camera movement, and so on. Motion estimation and compensation are used to reduce this type of redundancy in moving pictures. The block-matching algorithm (BMA) for motion estimation has proved to be very efficient in terms of quality and bit rate; therefore, it has been implemented by many standard video encoders.

The basic principle of block matching motion estimation and compensation is introduced. Also fast motion search algorithms are simulated. The smoothness of algorithm makes full search interactive algorithm. It requires extremely large computation time which makes it unsuitable for practical and real-time implementation. Based on analysis of variants of BMA, Logarithmic method is simulated and presented in this paper. This paper is divided into

four sections. Section I contains introduction. Section II includes basic concepts related to block matching algorithm and variants of BMA. Section III comprises algorithmic implementation method for Logarithmic block matching algorithms. Section IV embraces result analysis and comparative study.

II. BLOCK MATCHING ALGORITHM AND ITS VARIANTS

The two-dimensional logarithmic search was the first block-matching algorithm to exploit the quadrant monotonic model to match blocks. The initial step size 's' is $\left\lceil \frac{w}{4} \right\rceil$ (where $\lceil \cdot \rceil$ is the upper integer truncation function) where 'w' is the search range in either direction. The block at the center of the search area and the four candidate blocks at a distance 's' from the center on the x and y axes are compared to the target block to determine the best match.

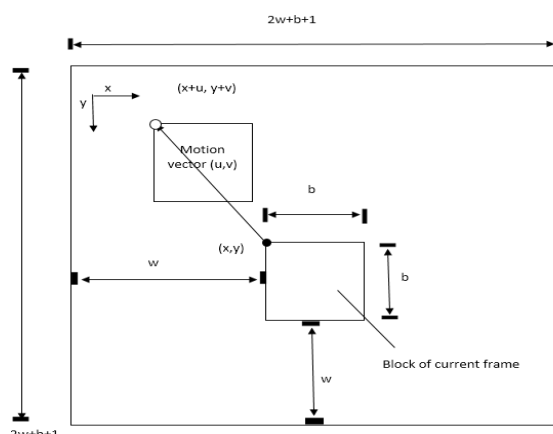


Fig 1. Block Matching Method

Full Search Method

Full search algorithm examines all positions in search window. It is optimal for correctly defined search range to give best matching point. It uses cost function to find motion vector for motion estimation with each possible location in search window.

Full search algorithm occupies more computations and optimal result. Since it occupies more mathematical computation, another approach Fast Search algorithm is surveyed and simulated.

Three Step Search

Compare to Full search, a very popular method for Fast search method is three step search method. It is robust and optimal in performance initial step size is assumed. Eight points around center at a distance of step size are chosen for comparison to find best match. At the end of first iteration the best match is considered as center for next iteration with step size reduced by factor of 2. The process is repeated until we get best match as center.

From literature survey one problem faced in three step search is that uniformly allocation of checking point pattern becomes inefficient. Hence, another variant of fast search BMA called logarithmic search method is identified as a good choice for the motion estimation.

Logarithmic Search Method

The five positions form a pattern similar to the five points of a cross (+). Thus, if the center of the search area is at position $[0, 0]$, then the candidate blocks at position $[0, 0]$, $[0, +s]$, $[0, -s]$, $[-s, 0]$, and $[+s, 0]$ are examined. Fig.2 shows the search pattern of logarithmic search algorithm. The step size is reduced by half only when the minimum distortion measure point of the previous step is the center (x, y) or the current minimum point reaches the search window boundary. Otherwise, the step size remains the same.

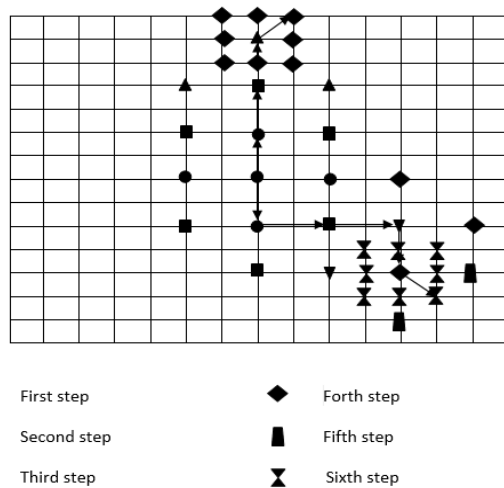


Fig 2. Logarithmic Search

When the step size is reduced to one, all eight blocks around the center position, which are $[x - 1, y - 1]$, $[x - 1, y]$, $[x - 1, y + 1]$, $[x, y - 1]$, $[x, y]$, $[x, y + 1]$, $[x + 1, y - 1]$, $[x + 1, y]$, and $[x + 1, y + 1]$ are examined, minimum distortion measure point of these is determined to be the best match for the target block and then it halts the algorithm. Otherwise (step size greater than one), the candidate blocks at positions $[x, y]$, $[x - s, y]$, $[x + s, y]$, $[x, y - s]$, and $[x, y + s]$ are evaluated for distortion measure.

III. LOGARITHMIC SEARCH METHOD

The two dimensional logarithmic search (TDL), is the first block matching algorithm to exploit the quadrant monotonic model to match blocks. The multi-stage search is accomplished by successively reducing the search area during each stage until the search area is trivially small. The TDL has several stages.

Stage 1

If the maximum displacement allowable is d , then the step size s is equal to $2^{\lceil \log_e d \rceil - 1}$. The block at the center of the search area and the four candidate blocks at distance s from the centre on the x and y axes are compared to the target block to determine which is the best match. The five positions form a pattern similar to the five points of a cross (+). Thus if the centre of the search area is at position $[0,0]$ then the candidate blocks at position $[0,0]$, $[0,+s]$, $[0,-s]$, $[-s,0]$, and $[+s,0]$ are examined.

Stage 2

If the position of the best match is at the center $[x, y]$ then the step size s is halved. If the best match is in one of the four outer positions, then that position becomes the center point of the next stage. That is $[x, y] = [a, b]$ where $[a, b]$ is the position of the best match.

Stage 3

If the step size 's' is equal to one then all nine blocks around the center position are examined and the best match of these is determined to be the best match for the target block. That is $[x-1,y-1]$, $[x-1,y]$, $[x-1,y+1]$, $[x,y-1]$, $[x, y]$, $[x,y+1]$, $[x+1,y-1]$, $[x+1,y]$, and $[x+1,y+1]$ are examined, and the algorithm halts. Otherwise (step size greater than one) the candidate blocks at positions $[x, y]$, $[x+s, y]$, $[x-s, y]$, $[x, y+s]$, and $[x, y-s]$ are compared. The algorithm goes to stage 2.

Fig. 3 shows the flowchart of Logarithmic search method of BMA (Block Matching Algorithm). Flowchart shows various steps included in Logarithmic search method. The steps are as described below.

Step 1

Consider an initial step size. Take block at the Center of the search area and assume four blocks around the center point.

Step 2

If the best match obtained is at the center, then halve the step size. If one of the other four points is the best match, then it becomes the center and again step 1 is repeated.

Step 3

When the step size becomes unity, all the nine blocks around the center are selected for the search and the best among them is picked as the required block.

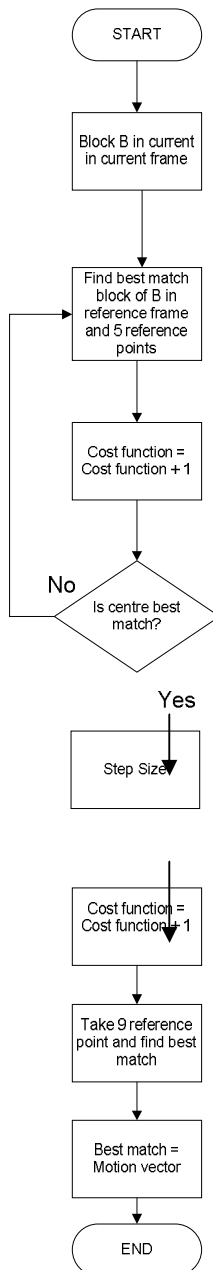


Fig 3. Flowchart of Logarithmic Search Method

IV. SIMULATION ANALYSIS

In this research work, four standard Quarter Common Intermediate File Format (QCIF) video sequences of different motion contents are used for performance

comparison of different algorithms. Simulations are done on Intel(R)Core(TM)i5-3317U CPU@ 1.70GHz configured system. Various macro-block size and search window combinations are used to test motion estimation. Computational cost is calculated for each case and compared for various cases. Results are shown in Table 1. PSNR is calculated for first 30 frames of video sequence. Table 1 indicates data for average PSNR and average computation/Macroblock. Here search window $p=7$ is used.



Akiyo
Fig 4. Frame of 'Akiyo'

Table 1 indicates data for average PSNR and average computation/Macroblock. Here search window is altered by keeping Macroblock size to 16.

TABLE 1
Average PSNR and Computation for first 30 frames

Search window 'P'	PSNR (dB)	Computation/Macroblock
5	30.0146	13.8358
7	30.33815	14.1174
9	30.3688	14.4239

Table 1 shows that as search window increases, computation per macroblock tends to increase. Table 2 indicates data for average PSNR and average computation/Macroblock. Here Macroblock size is altered by keeping search window constant. Table 2 shows that decreasing macroblock size results in increase of PSNR at cost of more computations.

TABLE 2
Average PSNR and Computation for first 30 frames

Macro block size	PSNR (dB)	Computation/Macroblock
16	30.33815	14.1174
8	31.44225	15.232
4	33.6933	16.0985

For the simulation, Search window $p=7$ and Macro-block size = 16 is used to find motion vector and PSNR is calculated and tabulated in Table 3. For mathematical computational, computation per macro-block is calculated and compared.



Fig. 5 First frame of video sequences Foreman, Crew and News

Now, in Table 3 various standard video sequences are tested for motion estimation.

TABLE 3
Average PSNR and Computation for first 30 frames

Frame	PSNR(dB)	Computation/Macroblock
News	30.33815	14.1175
Foreman	26.5877	16.03665
Akiyo	44.66805	13.39645
Crew	24.98955	16.74875

CONCLUSION

Variants of Block Matching Algorithms are surveyed. Full Search and Fast search BMA are reviewed. Full

search algorithm gives best reproduced output at cost of more computational complexity. To develop the scope of architecture development on hardware, goal is to reduce computational complexity. As a result, Logarithmic Search Method is simulated on MATLAB for various combinations of standard test video sequences, search window & macroblock size. Evaluation is done based on PSNR & computation cost for motion estimation.

REFERENCES

- [1] CH.Hufnagl And A.uhl, "Fast Block-Matching Algorithms For High-Resolution Video Compression", 1999
- [2] Hussain Ahmed Choudhury, Monjul Saikia. "Comparative Study of Block Matching Algorithm FOR Motion Estimation." International Journal of Advanced Computational Engineering and Networking, ISSN: 2320-2106 Volume-1, Issue-10, Dec-2013, NERIST, Arunachal Pradesh, India
- [3] Rahul Gonawala and Prof. Nehal Shah, "Full Search and Fast Search Algorithms for Motion Estimation in Video Compression", International conference on Information, Knowledge & Research in Engineering, Technology & Sciences, March 2012, Rajkot.
- [4] Siddhartha Ahluwalia, Sourabh Rungta, Dr. Anupam Shukla, "Circular 2-D Logarithmic Search Algorithm For Motion Estimation" M.Tech Student, Department of Information Technology, ABV-IIITM, Gwalior, India -474010 Ph.D Student, RCET DURG Asso. Prof. , Department Of Information Technology, ABV-IIITM, Gwalior, India -474010
- [5] Y-C. Lin and S-C. Tai. 1996" Fast Full-Search Block-Matching Algorithm for Motion Compensated Video Compression." In Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume III-Volume 7276 - Volume 7276 (ICPR '96), Vol. 7276. IEEE Computer Society, Washington, DC, USA, 914.

★★★