

Optimising Block Matching Algorithms By Redundancy Removal For Fast Motion Estimation

Richa Bali

Department of Computer Science and Engineering,
National Institute of Technology Calicut, India

V K Govindan

Department of Computer Science and Engineering,
National Institute of Technology Calicut, India

Abstract- Motion Estimation is a field in which maximum researches are being done to get reduced computation complexity in video compression. Block matching algorithms are very useful in achieving efficient and acceptable motion estimation in this regard. Total computational cost can be efficiently controlled by introducing basic operation like image subtraction and properly modifying block matching algorithms. This paper proposes a new approach for the reduction in computation of existing Block Matching Algorithms but still achieve the same PSNR and hence better quality of compression. The algorithm attempts to reduce the computational cost by applying basic operations on the image before applying the block matching algorithm. Motion estimation is the most time consuming operation in a typical video encoder. The proposed algorithm reduces time for motion estimation. Results show that with the help of the new proposed algorithm, cost of computations has been reduced to 8.14. The PSNR obtained is equivalent to exhaustive search. The advantages of the algorithm is that we can increase/ reduce the number of computations while increasing / decreasing the picture quality depending upon our need. The experimental results based on video frames were compared to demonstrate the advantages of proposed fast motion estimation algorithm.

I. INTRODUCTION

The Motion estimation is defined as a process which determines the motion between two or more frames of video. Extraction of motion parameters by using effective and optimized Block Matching Algorithms (BMA) is one of the best techniques to handle motion estimation problem in today's faster multimedia technology. Motion estimation (ME) is defined as searching the best motion vector, which is the displacement of the coordinate of the best similar block in previous frame for the block in current frame. In a video sequences; there exists a high level of redundancy between consecutive frames which means the changes from one frame to the other are minimal. To reduce the computational complexity of ME algorithms, a variety of methods such as block matching algorithm (BMA) have been presented by many researchers.

1.1. Literature Survey

An important work in this topic is that by Aroh Barjatya [1] who reviewed various block matching algorithms in the paper. It describes the advantages and disadvantages of various block matching algorithms. The paper implements and compare seven different types of block matching algorithms

including the very basic Exhaustive Search to the recent well known algorithms like Adaptive Rood Pattern Search. The main issue in this topic of research is to find the best and fastest Block Matching Algorithm. The algorithms are being compared on the basis of their speed, efficiency and PSNR Value. The author concludes that ARPS takes less computations and hence is the best of the fast block matching algorithms studied in this paper.

Manjunatha [2] presented four block matching motion estimation algorithms, namely, Exhaustive Search (ES), Three Step Search (TSS), New Three Step Search (NTSS), and Diamond Search (DS) algorithms. The author implemented all the four algorithms and compared the performances for different distances between the frames of the video. They proved that Diamond Search (DS) algorithm is the best matching motion estimation algorithm that achieve best tradeoff between computational complexity versus picture quality.

Eric Chan and Sethuraman [3] states the features, advantages and disadvantages of Block matching Algorithms. The authors presented a review matching criterion like MAE, MSE, PDC for Block Matching. The author divided and compared these algorithms into three categories, namely fast algorithms, layered structure algorithms and inter-block motion field prediction algorithms on the basis of candidate blocks and computation complexity.

Sohail Zafar and Ya-Qin Zhang [4] proposed a new motion estimation/ compensation scheme which is based on the "inertia" effect of video scenes. The inter block motion information is found using minimum absolute difference. The Proposed scheme decreases the search area hence increases the efficiency of the algorithm.

Shan Zhu and Kai-Kuang Ma [5] proposed a new diamond search (DS) algorithm for fast block-matching motion estimation.. It has been proved by simulation results that the proposed DS algorithm is better than three-step search (TSS) algorithm and new three-step search (NTSS) algorithm. The proposed algorithm is able to achieve close performance and require less computation.

Yao Nie, and Kai-Kuang Ma [6] proposed a new approach, called adaptive rood pattern search (ARPS), which has two sequential search stages: (1) initial search and (2) refined local search. The proposed algorithm is considered to be an ideal for

large and/or complex motion contents. An adaptive rood pattern (ARP) is proposed for the initial search stage whereas in the second stage of refined local search, a unit-size rood pattern (URP) is used to find the final MV. The author has also included the concept of zero-motion prejudgment (ZMP) which is helpful in small motion contents.

Chun-Ho Cheung, and Lai-Man [7] proposed a new algorithm which is a hybrid of cross search and diamond search algorithm. The small cross-shaped pattern basically counter for the cross-center-biased MVD characteristics and diamond shape pattern helps in finding the motion vector at the centre point. Thus proposed small-cross-diamond search algorithm (SCDS) which also used a halfway-stop technique and could find small motion vectors.

The literature survey was done to thoroughly understand the seven basic block matching algorithms, their advantages and disadvantages. It has been found that most of the pixel don't change between two successive videos frames, but computations is being done on those points also. So my algorithm incorporate the redundancy removal by image subtraction, by which we can reduce the no. of computations on dead blocks where there is no motion between previous and current frame.

The rest of the paper is organized as follows: Section II and III deal with block matching algorithm used, the performance measures and cost functions employed in this paper: Section IV presents the proposed algorithm for motion estimation. Section V discusses the simulation results, and finally Section VI concludes the paper.

II. BLOCK MATCHING ALGORITHMS

In BMA, the current image frame is partitioned into fixed-size rectangular blocks. The motion vector for each block is estimated by finding the best matching block of pixels within the search window in the previous frame according to matching criteria. Although Full Search algorithm (FSA) finds the optimal motion vector by searching exhaustively for the best matching block but it is not fit for practical applications. There are many BMAs like 3SS algorithm, which is used in our algorithm and explained below:-

A. Three Step Search (3SS)

This algorithms was introduced in mid 1980s. The search starts from the centre with 'step size' $S = 4$, and search parameter value of 7. We have to search for these eight locations $\pm S$ pixels around location (0,0). From these nine locations it picks the one giving least cost and makes it the new search origin. In next step, the new step size $S = S/2$, and in third step $S = 1$. Hence the search parameter of 7 is traversed in three steps At $S=1$, least cost function is found at one of these nine points and the macro block with least cost at that location is the best match.

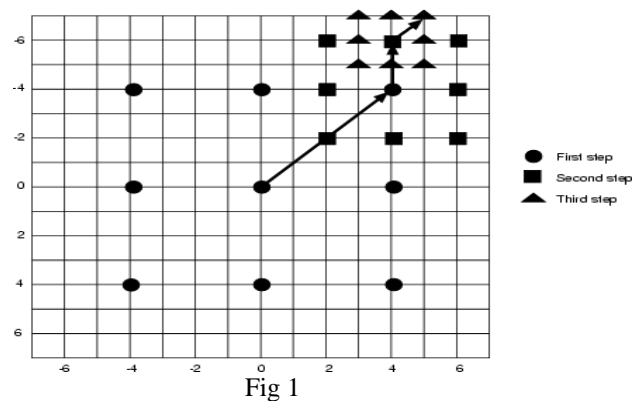


Fig 1

III. PERFORMANCE METRICS AND COST FUNCTIONS

Motion estimation is that the macroblocks in a previous frame move within the frame to form another pattern corresponding objects on the current frame. Cost functions mainly predict the matching of the blocks. The least cost macro block. Two of the main cost functions are: Mean Absolute Difference (MAD) given by equation (i). Another cost function is Mean Squared Error (MSE) given by equation (ii).

i.

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

ii

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

where N is the side of the macro block, C_{ij} and R_{ij} are the pixels being compared in current macroblock and reference macro block, respectively.

In block matching, current frame is divided into matrix of macro blocks that are then compared with corresponding blocks of previous frame and its adjacent neighbors in the previous frame to find the best match. The movement calculated for all the macro blocks comprising a frame, constitutes the motion estimated in current frame. PSNR stated below given by equation (iii) characterizes the motion compensated image

$$PSNR = 10 \log_{10} \left[\frac{(\text{Peak Value Of Original Data})^2}{MSE} \right] \quad (iii)$$

Where

MSE=Mean Squared Error

IV. PROPOSED TECHNIQUE

In this proposed algorithm, we use concept of image subtraction or pixel subtraction for detecting changes between two images. This detection of changes can be used to tell if something in the image moved.

MATLAB Variable Editor: l		Page 1	
Feb 17, 2014		6:19:04 PM	
1	1	2	3
2	14	14	15
3	56	56	57
4	54	55	55
5	53	54	54
6	49	50	51
7	54	54	55
8	52	53	53
9	53	54	54
10	53	54	54
11	53	54	54
12	53	54	54
13	53	54	54
14	53	54	54
15	53	54	54
16	53	54	54
17	49	51	53
18	50	52	52
19	51	54	55
20	52	55	56
21	52	55	56
22	51	54	55
23	50	52	52
24	49	51	53
25	51	52	52
26	52	53	53
27	52	53	53
28	51	52	52
29	51	52	52
30	51	52	52
31	51	52	52
32	51	52	52
33	52	53	53
34	52	53	53
35	52	53	53
36	52	53	53
37	52	53	53
38	52	53	53

Fig. 2: Current Frame

MATLAB Variable Editor: l2		Page 1	
Feb 17, 2014		6:28:15 PM	
1	1	2	3
2	13	14	15
3	58	58	59
4	54	55	55
5	54	54	55
6	52	53	53
7	54	54	55
8	55	55	55
9	53	54	54
10	53	54	54
11	53	54	54
12	53	54	54
13	53	54	54
14	53	54	54
15	53	54	54
16	53	54	54
17	52	53	53
18	52	53	53
19	52	53	53
20	52	53	53
21	52	53	53
22	52	53	53
23	52	53	53
24	52	53	53
25	52	53	53
26	52	53	53
27	52	53	53
28	52	53	53
29	52	53	53
30	52	53	53
31	52	53	53
32	52	53	53
33	52	53	53
34	52	53	53
35	52	53	53
36	52	53	53
37	52	53	53
38	52	53	53

Fig.3 Previous Frame

MATLAB Variable Editor: k		Page 1	
Feb 16, 2014		8:43:24 PM	
1	1	2	3
2	1	1	2
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	0
14	0	0	0
15	0	0	0
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0
20	0	0	0
21	0	0	0
22	0	0	0
23	0	0	0
24	0	0	0
25	0	0	0
26	0	0	0
27	0	0	0
28	0	0	0
29	0	0	0
30	0	0	0
31	0	0	0
32	0	0	0
33	0	0	0
34	0	0	0
35	0	0	0
36	0	0	0
37	0	0	0
38	0	0	0

Fig. 4 Difference Frame

The algorithm is given below.

Algorithm:

{

First Step : Read the previous frame.

Second Step : Read the current frame

Third Step: Subtract the previous frame matrix from the current frame matrix to get the difference image matrix

Fourth Step :Divide the current and previous frame and the difference image into macro blocks(Each macro block means 16x16 pixel matrix)

Fifth Step : For current macro block, Calculate the MAD(mod sum of Difference, which is the cost of current macro block

Sixth Step: Depending upon cost, three cases arise

Case-1: When the cost= α (where α = very small value nearly equal to zero). It shows that there is no change from previous frame to current frame at that macroblock, hence we need not to find the motion vectors at that point. Most of the macroblocks will lie in this case, because in most of the videos the background does not change.

if (cost $\leq \alpha$) (figure-2) // Best possible search.

Else // No second and third step required.

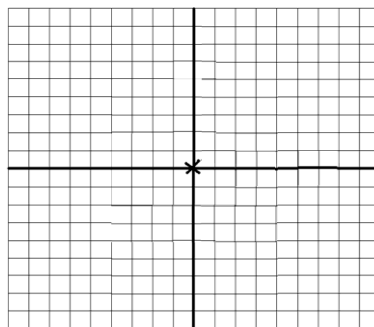


Figure 1:case-1

Case-2: When the cost is greater than α and less than β (where β is the threshold value beyond which we assume that there is a major change between the two frames in our algorithm $\beta = 10$), it shows that there is a small change from the previous frame to current frame for that macroblock, So, we compare the cost of the macro block in current frame and previous frame and will found that the cost of the current macro block is nearly equal to any one the eight neighbour macro block. Hence, we can find the best match of that macroblock in those eight neighbour points.

If((costs > α) and (costs \leq β)

Take $s=1$. (Figure-3). Find the macroblock in current frame which matches the macro block in reference frame, hence find the motion vector and return.

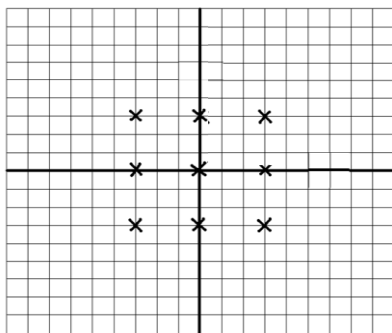


Figure 2: case-2

Case-3: When the cost is greater than β (in our algorithm $\beta=10$), it means there is a major change in the position of macro block from current frame to previous frame, so then we need to apply any block matching Algorithm, in our algorithm, we apply Three Step Search (Figure-4)

if (costs > β)

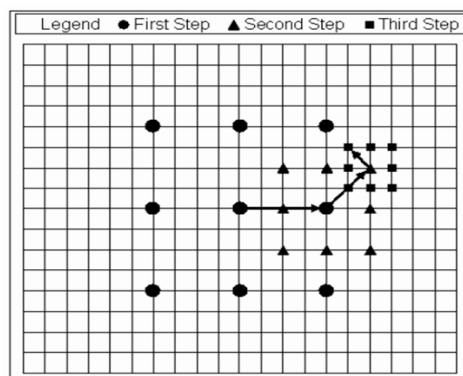


Figure 3: case-3

V. SIMULATION RESULTS

Algorithm	Computation	PSNR
DS	20.2121	41.7401
ES	195.2323	41.7613
NTSS	23.6667	41.6818
Proposed Algo	8.3838	40.0404
SES	15.0152	41.3868
SS4	20.3030	41.5945
TSS	22.8939	41.5851

Table 1: Results for Miss America Frame

Algorithm	Computation	PSNR
DS	12.6970	40.0479
ES	195.2323	40.0722
NTSS	17.4495	40.0536
Proposed Algo	8.1414	40.0415
SES	16.4343	40.0386
SS4	15.9747	40.0445
TSS	22.5909	40.0473

Table 2 : Results for Mother Daughter Frames

Algorithm	Computation	PSNR
DS	19.0606	27.9727
ES	184.5556	28.0208
NTSS	23.6667	27.9617
Proposed Algo	15.6061	27.8870
SES	15.5556	26.5866
SS4	20.1111	27.6172
TSS	21.8788	27.9528

Table 3: Results for Foreman Frames

The proposed algorithm was tested for three different frames, viz., Miss America Frame, Mother Daughter Frame and Foreman Frame. The proposed algorithm produced best results for all three frames. It has been found that computations have been reduced and are equal to one third of the primitive TSS for Miss America and Mother Daughter Frame. For foreman frame also, we can observed that computation has been reduced by 30%. When compared to Diamond search, the new method has been found to decrease the computation by more than 50% where as the PSNR has remained nearly the same. The proposed algorithms gives the PSNR comparable to Exhaustive search with reduction of computations. The proposed algorithms proved to be even better than New Three Step Search in terms of computations. The proposed algorithms has proved to give the least number of computations in comparison to all algorithms for all of the frames. The efficiency of the proposed algorithms lies in its flexibility that we can still reduce the number of computations by compromising a bit in PSNR Value.

VI. CONCLUSION

In this paper, a concept of inter frame redundancy removal is introduced. The major applications of motion estimation algorithms include traffic movement tracking, studying plant root growth, hand posture analysis, human posture analysis, lip movement for user authentication, robotic heart surgery,

breathing motion estimation etc. And all such application have only minor change between two successive frames. Hence, our algorithm has proved to be the best fit for such small motions. The test results demonstrate its efficiency in terms of computation and PSNR. The strength of the algorithm lies in giving acceptable results (with improvement) in terms of Quality (PSNR) and the drastic reduction in the no. computations. The results of computations are better than that of other well known algorithms proposed in the literature. We have observed that with the help of new proposed algorithm, value of computations has been reduced to 8.14. We can change the values of the threshold alpha and beta, and correspondingly increase/ reduce the number of computations while increasing / decreasing the picture quality depending upon our need. This is the main advantage of the proposed Algorithm. Efficiency of the algorithm can be best realizable when more number of dead spaces comes into action.

VII. REFERENCES

- [1] Aroh Barjatya, "Block matching algorithms for motion estimation" ACM Transactions on Programming Languages and Systems, 2004.
- [2] D.V. Manjunatha, "Comparison and implementation of fast block matching motion estimation algorithms for video compression", International Journal of Engineering Science and Technology (IJEST), 3(10), October 2011,1-6.
- [3] Eric Chan, and Sethuraman Panchanathan, "Review of block matching based motion estimation algorithms for video compression" IEEE, 19(3), May 1993, 151-290.
- [4] Sohail Zafar, and Ya-Qin Zhang, "Predictive Block-Matching Motion Estimation Schemes for Video Compression—Part I: Inter-Block Prediction", Southeastcon '91 Conference, Williamsburg, Virginia, April, 1991,7-10
- [5] Shan Zhu, and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE 9(2), February 2000,287-290.
- [6] Yao Nie, and Kai-Kuang Ma, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation", IEEE Trans. Image Processing, 11(12),December 2002,1442-1448.
- [7] Chun-Ho Cheung, and Lai-Man Po, "A Novel Small Cross-Diamond Search Algorithm for Fast Video Coding and Video Conferencing Applications",IEEE ICIP, September 2002

IJERT