Varible Blocksize Motion Estimation for H.264

Shafee Vunnisa Sayyad, V.S.K. Reddy

Abstract--- Digital video coding plays a vital role in Transmission of videos. A still image will not contain any temporal content but possesses only spatial information but video does. Main key point for video compression is Motion Estimation. The clarity of the reconstructed video depends on how best we estimated the motion between two successive frames. Motion Estimation is done by Block matching algorithms. In this paper six algorithms are explained & applied on different styles of videos like slow motion, medium motion, fast motion. The comparison results will give the summary and the best algorithm suitable for different category of input videos.

Keywords—ME, EBMA, ARPS, DS, SAD, PSNR

I. INTRODUCTION

A still image will not contain any temporal information but possesses only spatial information. In the standards of image compression, only an image frame is to be sent and it need to be encoded at transmitter-end by an encoder, to be decoded by a decoder and is reconstructed by the receiver at receiver-end. Considering the case of a video, a stream of continuous images that can be made available from video camera or any video medium are to be transmitted, processed and reconstructed.

The basic important difference is that a video contains temporal information, that is, it is varying both in space and in time. A signal which varies with respect to time possesses high similarity between its neighboring frames. As an example, suppose that a video is being transmitted with a frame rate of 30 frames per sec. This means that each frame is separated by 33 milliseconds. In this very short span, the successive or the neighboring frames possess a high rate of similarity among them because the movement between them physically is not much noticeable. If keenly observed in some videos, the background in each frame is exactly same it doesn't need not be changed every frame and is stationary. If in a video, a person is talking, his lip, eye and shoulder movement is highly noticeable and hence there is a great effect of temporal redundancy. Hence Each frame is not encoded but the difference between the two successive frames. i.e estimating the motion between two consecutive frames.

To determine the motion estimation we make use of Different BMA's explained in section II. Block Matching techniques are applied on slow motion, fast motion, and medium motion videos and compared. In section III Matching critera's and various input videos data is given. Lastly summary for all BMA's with variable block size.

Revised Manuscript Received on March 02, 2019.

SHAFEE VUNNISA SAYYAD, Research scholar, Rayalaseema University, kurnool, Associate Professor, SNIST, AP, INDIA. (E-mail: Shafiunnisa8@gmail.com)

DR. V.S.K. REDDY, Director, MRCET, Telangana, INDIA. (E-mail: Vskreddy2003@gmail.com)

II. BLOCK MATHCING ALGORITHMS

BMA's have been looked into since mid-1980's. Numerous calculations have been produced, however just the absolute most essential or usually utilized have been depicted underneath

Exhaustive Search

In the real-time motion estimation, in a full search block algorithm, there is quite a high computational complexity. There are only two ways available where this can be reduced. The first one, getting adoptive to some of the specialized hardware which was designed using the VLSI technology which has the capability of estimating the motion vectors by overcoming this complexity. The other way is to adapt a fast or a quick motion estimation techniques where, unlike the full search block, only specific areas or position of the block are searched in order to increase the speed of computation.

But in the full search block, although there is a high computational complexity, there is always a guarantee that the optimal solution would be obtained for sure because here, the entire search window is searched. In the case of fast motion estimation techniques, there is no such surety that the optimal solution can be obtained because we perform the search operation only at a specific specified areas, which gives only a sub-optimal solution. However, the complexity is going to be reduced to a greater extent.

Three Step Search

Another popular and important fast search technique is the three step search algorithm and can be denoted as TSS in short. This is a combination of 2-D logarithmic algorithm and the cross search algorithm. 2-D logarithmic search is carried out at the ends of a plus and cross search, search is carried at the ends of cross whereas, in this search, all the nine positions are searched after defining the initial value of p. The entire search range, in this case, is (2p+1) and the nine positions are (p, -p), (0, -p), (p, -p), (-p, 0), (0, 0), (p, 0), (-p, p), (0, p), (p, p). At all these positions, the search is done and anywhere if a minimum is encountered without waiting for any future iterative, the step size factor is reduced by a factor of 2. The next iterative search is made with p/2 and then all the new nine positions along with center are searched. As a third iterative search, again the step size is divided by 2 and its value becomes 1. So, again for all the nine positions, the search is carried and in whichever position the minimum occurs that is concluded as a final minimum.

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

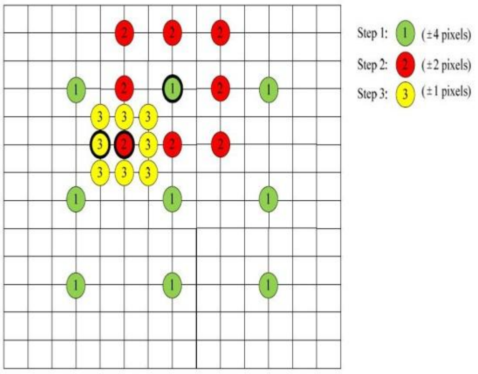


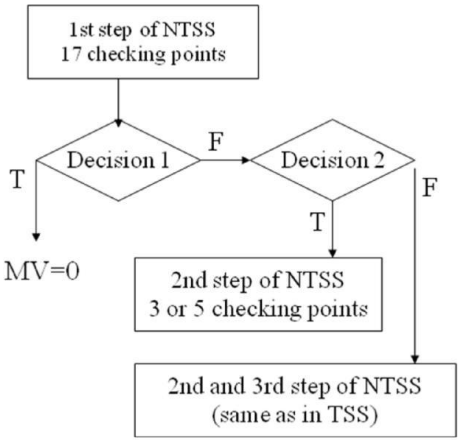
Fig.1. TSS search -procedure

New Three Step Search

Uniformly allocated checking pattern is the methodology used by TSS and has the drawback of missing small motions. NTSS is the solution for it. It follows the scheme of a center biased search and has an option to stop halfway to reduce computational cost. New Three Step Search(NTSS) is one of the first widely accepted fast algorithms and used for implementing standards like MPEG 1 and H.261.

Decision 1: minimum at the search window center ?

Decision 2: minimum at one neighbor of center ?



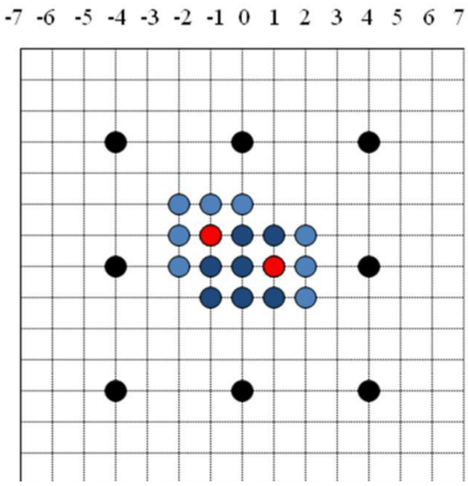


Fig.2. NTSS search -procedure

Simple and Efficient Search

In TSS, the error surface due to motion in every macro block is unimodal. A unimodal surface is a bowl shaped surface such that the cost function generates the weight, which increases monotonically from the global minimum. However, a unimodal surface will snot be having two minimums in its opposite directions. So, TSS can be furtherly modified to SES and save no. of computations.

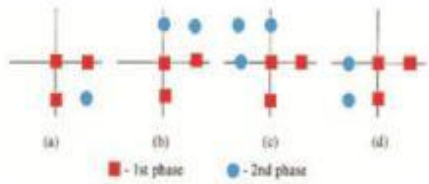


Fig. 3 SETS Search – Quadrant

Four Step Search

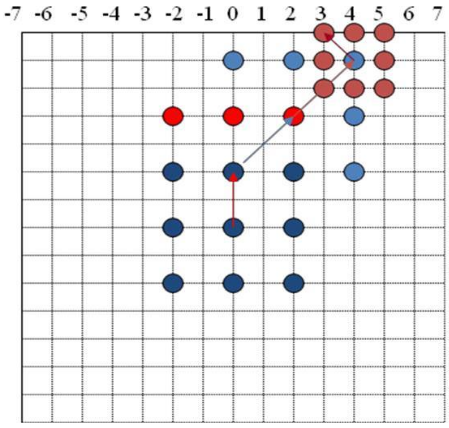


Fig. 4. 4SS Search Procedure

Four Step Search is a better version of TSS in terms of computational cost which is lower than TSS and has the better peak signal-to-noise ratio(PSNR). Like NTSS, FSS also employs the center biased searching and possesses the halfway stop.

Diamond Search

This algorithm uses a pattern of diamond search and the algorithm is in the similar way of how FSS is. However, the number of steps that this algorithm can take has no limit.

In diamond search, there are two types of patterns:

- Large Diamond Search Pattern (LDSP)
- Small Diamond Search Pattern (SDSP)

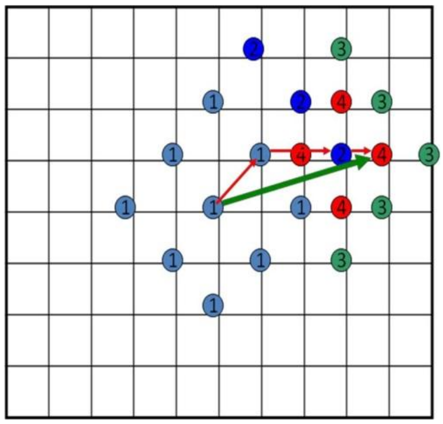


Fig. 5. DS Search Procedure

This algorithm is said to have the highest degree of accuracy in finding the global minimum, because the search pattern is neither too small nor too big. This algorithm has the peak signal-to-noise ratio close to that of the Exhaustive Search algorithm with significantly lesser computational complexity.

Adaptive Rood Pattern Search

Adaptive Rood Pattern Search (ARPS) algorithm highlights the fact that, when you consider a frame, it's motion is coherent in nature, i.e. the macro blocks which are around the current macro block moving in a particular direction will be having similar motion vector as that of the current block and such probability is high. In order to predict its own motion vector, this algorithm uses the macro block's motion vector towards its immediate left. In Rood pattern search, there is a high probability of finding a good matching block. The advantage of ARPS is, if the predicted motion vector is

(0, 0), it directly proceeds with SDSP, rather than doing LDSP. This reduces the computational time. Furtherly, if the predicted motion vector is away from the center, ARPS saves computations by directly choosing that vicinity and directly uses SDSP.

III. Matching criteria and Input Data

Motion estimation s carried out two adjacent frames. The type of motion estimation we employ is a block-based one, in which the image is divided into several blocks that are non-overlapping and for each of this block, estimation of motion is done. The block that has to be searches is known as the candidate block and to search this, a search space is defined. After defining, the block is searched and is compared with the position in the reference block. To perform this comparison, there has to be a measure for matching. The matching criteria are mentioned bellow:

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

$$MAD = \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}|C_{ij} - R_{ij}|$$

$$PSNR = 10Log_{10}\left[\frac{255^2}{MSE}\right]$$

Input Videos

	Duration (sec)	Frame rate	No. of frames	Width	Height
Suzie	9.933	15	149	176	144
Rhino	7.6	15	114	320	240
Stephan	2.933	30	88	352	288

IV. RESULTS & CONCLUSION

The following tables will give the comparison of various Block matching algorithms for different bock sizes and different categories of inputs. Three inputs have taken suzie(slow motion), Rhinos (Medium Motion) and Stephan(Fast Motion). All algorithms are executed at the fixed optimal search range of 7. ES algorithms will give the best PSNR but with the cost of high Search points. Among all ARPS algorithm will give the best PSNR (approximately same as ES) at the less computational cost.

The Graphs for PSNR & Search Range are plotted for all three inputs & six BMA for block size 4 and search Range '7. Also ARPS and NTSS algorithms PSNR and search range graphs are plotted for different Block sizes for the ease of explanation.

All BMA's results in increment in PSNR as reducing the block size with less increment in computational cost. If we are comparing the difference in the increment of the PSNR for various inputs slow motion, medium motion and fast motion are 0.7 dB, 1.7dB and 2dB respectively when BS changes from 16 to 8. Same proportional increment in PSNR is present when BS changes from 8 to 4. For fast motion videos reducing the BS makes considerable increment in PSNR but not for slow motion.

Table1. PSNR and SP for input Suzie

Suzie, p=7	**BS -16X16**		**BS- 8X8**		**BS- 4X4**	
Algorithm	PSNR	SP	PSNR	SP	PSNR	SP
ES	34.37	184	35.43	204	36.88	210
ARPS	34.25	6.21	35.14	6.45	36.16	7.0
NTSS	34.34	16.11	35.05	17.55	36.49	19
DS	34.31	11.42	35.21	12.23	36.17	13.2
4SS	34.23	15.24	35.07	16.54	36.02	17.7
SETSS	33.99	15.88	34.58	16.69	35.26	16.9

Stephan p=7	**BS -16X16**		**BS- 8X8**		**BS- 4X4**	
Algorithm	PSNR	SP	PSNR	SP	PSNR	SP
ES	23.98	204	25.64	214	27.40	217
ARPS	23.67	8.04	25.02	8.29	26.16	9.05
NTSS	23.54	22.13	24.59	23.21	26.16	24.1
DS	23.00	19.71	23.57	19.79	24.55	19.5
4SS	23.05	18.24	23.72	19.08	24.73	19.8
SETSS	22.67	16.28	22.96	16.55	23.81	16.5

Rhino p=7	**BS -16X16**		**BS- 8X8**		**BS- 4X4**	
Algorithm	PSNR	SP	PSNR	SP	PSNR	SP

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

ES	26.12	199.3	28.06	211	30.19	214
ARPS	25.82	12.85	27.19	13.23	28.17	12.9
NTSS	25.56	27.28	27.13	28.03	28.41	28.0
DS	25.82	23.57	26.55	23.46	27.14	22.3
4SS	25.76	22.21	26.66	22.70	28.41	22.6
SETSS	25.72	15.30	25.68	15.52	26.49	15.5

Table 3. PSNR and SP for input Rhinos

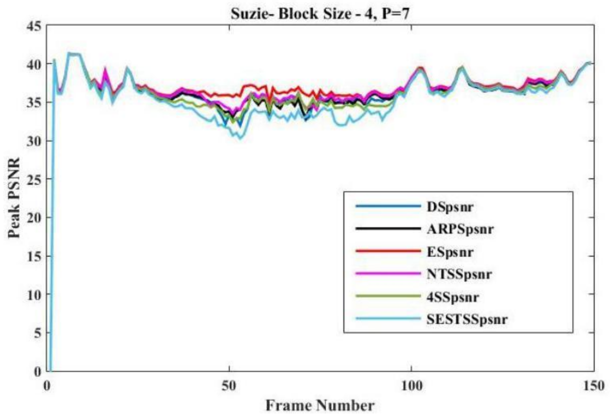


Fig.6 PSNR GRAPH for the input suzie

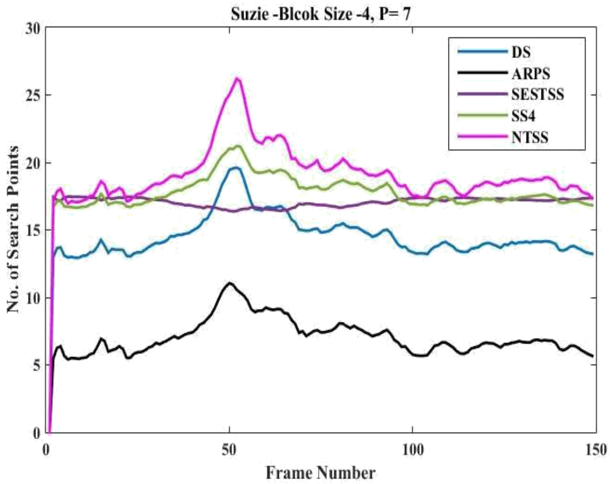


Fig.7. No. of Search points for input Suzie

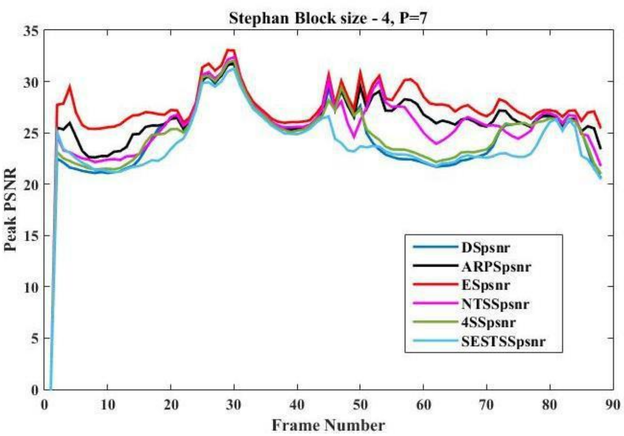


Fig.8. PSNR GRAPH for the input Stephan

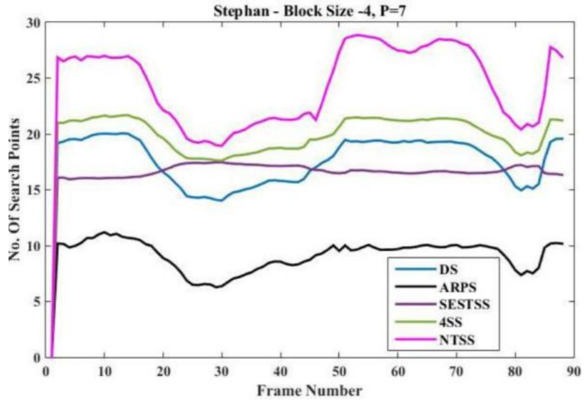


Fig. 9.No. of Search points for input Stephan

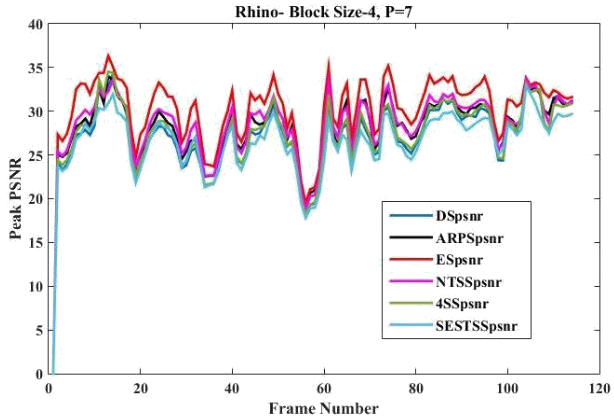


Fig.10.PSNR GRAPH for the input Rhino

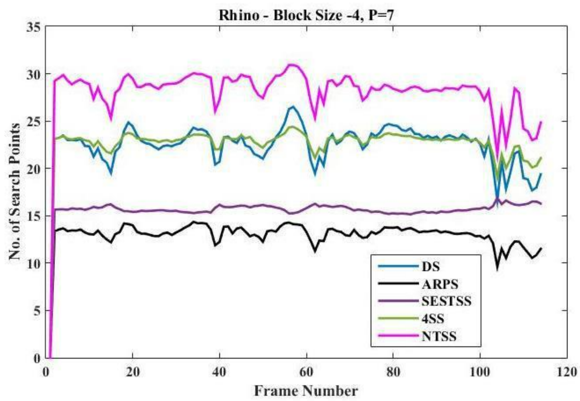


Fig. 11,No. of Search points for input Rhino

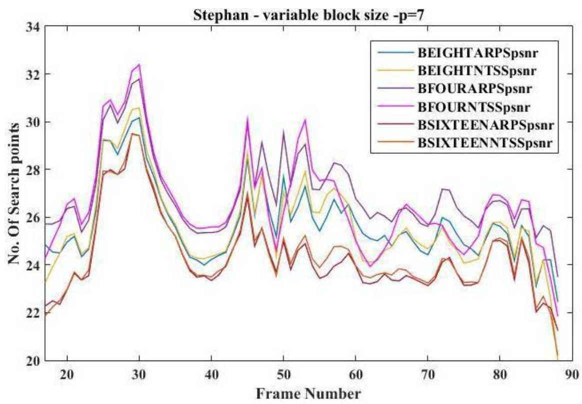


Fig. 12,PSNR plot for different BS for stephan

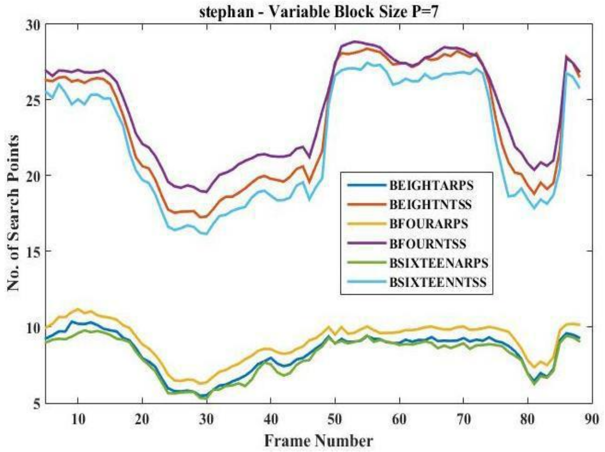


Fig. 13, Search range plot for different BS for stephan

REFERENCES

http://en.wikipedia.org/wiki/H.264
http://en.wikipedia.org/wiki/Video_compression
http://en.www.wikipedia.com http://www.chiariglione.org
http://www.vodex.com

[1] I.E.G. Richardson, (2003). *H.264 and MPEG-4 Video Compression: Video Coding for Next generation Multimedia*, John Wiley & Sons, Ltd, 0-470-84837-5, UK

[2] shafee vunnisa sayyad(2016), *"Flow Management for Video applications in Wireless Local Area Network"ccsn 2016 IEEE conference, west Bengal, December 2016*

[3] Iain E.Richardson, (2010). *The H.264 Advanced Video Compression Standard* John Wiley & Sons ltd, 978-0-470-51692-8, UK.

[4] Lai Mingche, Dai Kui, Lu Hong-yi, Wang Zhi-ying, 2006, A Novel Data-Parallel Coprocessor for Multimedia Signal Processing",IEEE Computer Society.

[5] Kue-Hwan Sihn, Hyunki Baik, Jong-Tae Kim, Sehyun Bae, Hyo Jung Song, (2009), Novel Approaches to Parallel H.264 Decoder on Symmetric Multicore Systems, *Proceedings of IEEE*.

[6] Kun Ouyang Qing Ouyang Zhengda Zhou, (2009), Optimization and Implementation of H.264 Encoder on Symmetric Multi- Processor Platform, *Proceedings of IEEE Computer Society, World Congress on Computer Science and Information Engineering*.

[7] Nuno Roma and Leonel Sousa, 2002, Efficient and Configurable Full-Search Block-Matching Processors, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 12, December 2002.

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*