Journal of Graphic Era University
Vol. 4, Issue 2, 66-71, 2016
ISSN: 0975-1416 (Print)

Comparative Study of Three Step Search and Diamond Search Algorithm for Motion Estimation

Ria, Subhash Chandra Yadav*

Department of Electronics and Communication Engineering,
Graphic Era University, Dehradun, India
*Corresponding author: subhash.yadav775@gmail.com

Abstract
Over the last couple of decades, video processing and coding/decoding has grown exponentially. The processing power of the computers available has grown in almost equal proportions. However, with the introduction of newer video standards, which are further more complex, it has become imperative to work on more optimized algorithms and implement them efficiently and judiciously. A number of parallel implementations of the algorithms have also been introduced, including using reconfigurable systolic arrays for the same. In this paper, we compare the two most commonly used search algorithms, Diamond Search and Three Step Search for motion estimation and decide which one is better for a given type of video sequence. This information can help in better decision making regarding the choice of the algorithm for a certain video sequence and save time by as much as 50%.

Keywords – Motion Estimation, SAD, Diamond Search, Three Step Search, VLSI Architecture, PSNR

Abbreviation and Acronyms

ME	Motion Estimation
SAD	Sum of Absolute Difference
PSNR	Peak Signal-to-Noise Ratio
TSS	Three Step Search
DS	Diamond Search
ES	Exhaustive Search
BMA	Block Matching Algorithms
MB	Macroblocks
CB	Current Block
RB	Reference Block
SR	Search Range

1. Introduction

With the advancement in the video processing technology and the introduction of high definition videos, the bandwidth and processor requirements have increased substantially. While the current architecture works well for the latest encoding standards, with the introduction of the H.265 video coding/decoding scheme there will be a need of even better algorithms, mostly parallel.

Motion estimation is based on the premise that the video macrblocks in two consecutive video frames do not change a lot. One video frame can be constructed by the previous one by shifting

Journal of Graphic Era University
Vol. 4, Issue 2, 66-71, 2016
ISSN: 0975-1416 (Print)

(moving) the blocks from the previous frame, and hence the name motion estimation. There have been different algorithms devised for this purpose, including the Exhaustive Search, Three Step Search, 2D Logarithmic Search, Diamond Search, etc. While the exhaustive search scans the entire frame for a match, the rest of the algorithms use selective macroblocks for finding a suitable fit.

We have divided this paper into a number of sections, the first one being the current introduction. The literature survey has been covered in the second section, followed by the review of the DS and TSS algorithms in the third. The fourth section deals with the methods used in the research and the data points collected in it. The results are summarised in the fifth section followed by conclusions thereafter.

2. Literature Survey

A lot of research has been carried out in optimizing the motion estimation algorithms as well as making better suited processors that can reduce the power consumption, both in India and abroad.

In the first decade of the century, there was a huge increase in the number of mobile phone users, and their internet connectivity were increasing at a massive pace. At the National Taiwan University, Lin came up with a Low-Power motion estimation processors for mobile video application (Lin, 2004) where he worked on mobile devices with limited processing power and optimised motion estimation with some trade-off in performance. Lin et al. (2009) have published a book titled “VLSI Design for Video Coding H.264/AVC Encoding from Standard Specification to Chip” (Lin et al., 2009) wherein they have presented VLSI architectural design and chip implementation for high definition H.264/AVC video encoding, using a state-of-the-art video application, with complete VLSI prototype, via FPGA/ASIC. The book goes on to say that advances in ever increasing coding efficiency comes at the expense of great computational complexity and this can only be achieved through massive parallel processing.

With the International Telecommunications Unit deciding to bring out H.265 standards in the coming years and the enormous amount of high-definition videos streamed through internet every minute, it is now much more important than ever to make sure that the bandwidth consumed is limited without hampering the quality of videos being streamed.

3. Review of TSS and DS Algorithm

The Three Step Search (TSS) Algorithm is the most fundamental fast Block Motion Algorithm (BMA) after Exhaustive Search (ES) BMA (Li et al., 1994). Compared to the latter where all the macro blocks (MBs) in the search range (SR) are compared with current block (CB), TSS compares only 25 (9 (1st step) + 8 (2nd step) + 8 (3rd step)) MBs in the SR with CB, thereby saving large computation cost. We divide the video frame into MBs of size N × Np. Usually, N= 4, 8 or 16 as H.264 standard supports variable block size. In ME a CB is matched with

Journal of Graphic Era University
Vol. 4, Issue 2, 66-71, 2016
ISSN: 0975-1416 (Print)

different RBs in the SR according to various cost functions, the coordinates of the block with the least difference is transmitted as a MV for that CB along with the difference between that RB and CB. The cost functions that we use for ME is SAD because it gives the efficient MV at a comparatively low computational cost (Furht et al., 2012). Thus the MV within the SR is given by the following equation:

$$MV(i_m, j_n) = \arg(SAD_{min})$$

In the Diamond Search algorithm, the search starts from the center of the search region, and spreads across eight other positions around the center which then form the shape of a diamond (Yang et al., 1989). This step is called LDS and is continuous; it searches either 3 or 5 positions in the next step and continues doing so until the search reaches the boundary of the search region or the best matched block is the central block. Once LDS finishes, SDS starts wherein it runs just once and the macroblock with the minimum SAD is taken to be the best matching block, and the motion vector is determined using that.

The Sum of Absolute Difference is calculated in the following way (Hsieh and Lin, 1992):

$$SAD(x,y) = \sum_{m=h}^{h+N-1} \sum_{n=v}^{v+N-1} |CB(m,n) - RB(m+x, n+y)|,$$
$$SAD_{min} = \min\big(SAD(x,y)\big), -SR \le x, y < SR$$

The macroblock with the minimum SAD represents the motion vector for the corresponding current block.

The PSNR value of a macroblock is calculated using the following formula (Komarek and Pirsch, 1989):

$$PSNR = 10 \log_{10} \frac{(2n-1)^2}{MSE}$$

with the Mean Squared Error being defined as:

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [O(i,j) - O_r(i,j)]^2$$

where O is the original image and O_r is the reconstructed image.

4. Research Method

We begin with collecting test video sequences from CIPR. The next step is to implement the algorithms for DS and TSS on Matlab. The basic algorithm implementation is done on Matlab, and points taken out for PSNR and the number of computations required per frame. These result points are then plotted on to a graph to derive conclusions from the same.

Journal of Graphic Era University
Vol. 4, Issue 2, 66-71, 2016
ISSN: 0975-1416 (Print)

The process goes as follows:

a. From the current frame, choose a macroblock (current block, CB) to find its motion vector.
b. In the reference frame, find a macroblock (reference block, RB) corresponding to the position of CB.
c. Select other reference search blocks as defined in the respective algorithms.
d. Find the SAD with the CB and the reference search block. Whichever RB gives the lowest SAD, take that as the new search center and select new reference search blocks around it.
e. When the algorithm terminates, the MB with the lowest SAD gives the motion vector.
f. Find the PSNR for the resultant block with the CB. Also note the computations required.

5. Results and Analysis

We marked the PSNR and the number of computations required for both the algorithms, with different video sequences, and used it to make a plot. Different legends are used for easy recognition of the sequences. The results are summarised in Figure 1 and Figure 2.

Figure 1 depicts the number of computations required per frame to get to the ideal matching block. As we can observe, the number of computations required for the TSS algorithm for various sequences (Salesman, Caltrain, Surfside, Missa & MR Chest) vary from 23 to 25, which matches with the number of computations required for TSS, i.e. 25. However, in case of DS algorithm, where we use LDSP and SDSP techniques, the number of computations per frame vary from 12 to 28. Hence, for some sequences, it gives better performance than the TSS algorithm while in others it is worse.

Next we take a look on Figure 2 for the PSNR value for both the algorithms. As we can observe, except for the Caltrain video sequence, the PSNR value for both the algorithms are exactly the same. This also tells us that both the algorithms point to the same matching macroblock. However, in case of Caltrain, different macroblocks are chosen by the two algorithms thereby resulting in different results for the PSNR.

6. Conclusion

The research work carried out in this paper signifies that there is not much difference in the final computed macroblock selected by the Three Step Search and the Diamond Search Algorithm. However, there is a significant difference in the number of computations required by both the algorithms, and a proper selection of algorithm will result in huge computational savings. From the video sequences used, we conclude that the Diamond Search algorithm is a better tool for computing motion vectors for slow-motion video sequences. On the other hand, for video sequences with high-motion (e.g. Surfside), the Three Step Search Algorithm is a better choice. This difference is due to the way the search expands itself in both the algorithms. While the TSS expands in the shape of a square and limits itself to 3 iterations, the DS expands in the shape of a diamond and limits its iteration only when it reaches the end of the frame.

Journal of Graphic Era University
Vol. 4, Issue 2, 66-71, 2016
ISSN: 0975-1416 (Print)

This difference in the search technique changes the search characteristics for different video sequences.

In any case, the number of computations per frame never exceed 28, which is still acceptable in most of the cases. To sum it up, we'd like to say that both the algorithms are fast enough for computation, however a better analysis beforehand can save computation time by as much as 50%.

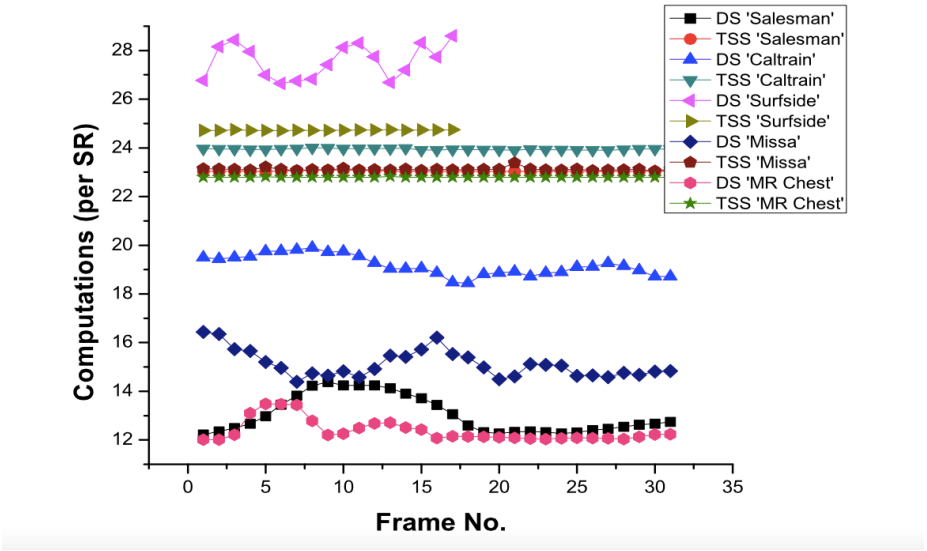

Figure 1. No. of computations per frame for TSS & DS algorithms

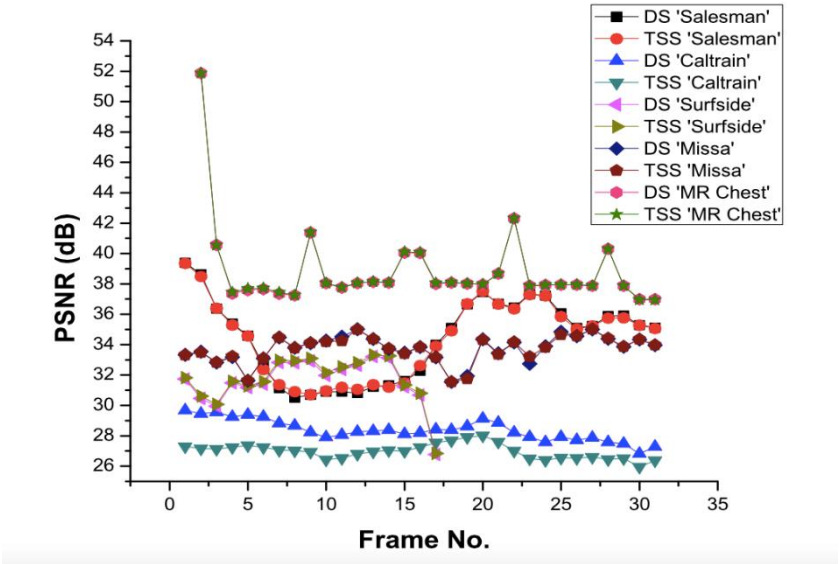

Figure 2. PSNR values per frame for TSS & DS algorithms

References

Furht, B., Greenberg, J., & Westwater, R. (2012). Motion estimation algorithms for video compression (Vol. 379). Springer Science & Business Media.

Hsieh, C. H., & Lin, T. P. (1992). VLSI architecture for block-matching motion estimation algorithm. IEEE Transactions on Circuits and Systems for Video Technology, 2(2), 169-175.

Komarek, T., & Pirsch, P. (1989, May). VLSI architectures for block matching algorithms. In Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on (pp. 2457-2460). IEEE.

Li, R., Zeng, B., & Liou, M. L. (1994). A new three-step search algorithm for block motion estimation. IEEE transactions on circuits and systems for video technology, 4(4), 438-442.

Lin, S. S. (2004). Low-power motion estimation processors for mobile video application (Doctoral dissertation, MS thesis, Graduate Institute of Electronic Engineering, National Taiwan University, Taipei, Taiwan).

Lin, Y. L. S., Kao, C. Y., Kuo, H. C., & Chen, J. W. (2009). VLSI Design for Video Coding: H. 264/AVC Encoding from Standard Specification to Chip. Springer Science & Business Media.

Yang, K. M., Sun, M. T., & Wu, L. (1989). A family of VLSI designs for the motion compensation block-matching algorithm. IEEE Transactions on Circuits and Systems, 36(10), 1317-1325.

About the Authors

Ria received her B.Tech. degree in Electronics & Instrumentation Engineering from Uttar Pradesh Technical University, Lucknow in 2014. She is currently working to get the M.Tech. degree in VLSI Design & System at Graphic Era University, Dehradun, India. Her research interests mainly include signal processing, multimedia processing and reconfigurable computing.

Mr. Subhash Chandra Yadav is an Assistant Professor at the Graphic Era University, Dehradun, India. He completed his M.Tech. Degree from Kurukshetra University, Kurukshetra, India. His area of interest includes VLSI Design, Logic Design and Signal Processing.