International Journal of Multidisciplinary Research and Development
Online ISSN: 2349-4182, Print ISSN: 2349-5979, Impact Factor: RJIF 5.72
www.allsubjectjournal.com
Volume 3; Issue 3; March 2016; Page No. 450-456

Fast motion estimation algorithms

Shaifali M Arora

Asst Prof, ECE Dept. Maharaja Surajmal Institute of Technology, New Delhi, India

Abstract
Usage of HD (High Definition) Video is increasing day by day in applications like television, video over internet, gaming and video surveillance, videoconferencing, video email, and mobile video, demanding intense video processing. Considerable storage space is required to store such information. Computational requirement to process large amount of data is also very high. Video is nothing but large number of frames transmitted per second. A large similarity exists between various frames, which are exploited in spatial and time domain to achieve compression. Block based motion estimation algorithms provide the hardware friendly solutions and hence are most widely used in existing video standards like MPEG, H.26x etc. Full search algorithm provides the best video quality at the cost of very high computations. Various algorithms have been proposed in literature to reduce this computational burden. This paper focuses on the implementation and study of these algorithms. Results of comparison of Full search algorithm with other fast algorithms have been presented in this paper.

Keywords: Block based motion estimation, fast motion estimation, block matching criteria

1. Introduction

The increasing usage of videos in all spheres of life and advancements in video display and capturing techniques, have led to an overwhelming need of processing video data at higher data rates. From the past three decades two groups are continuously working on the development of new standards for videos to meet the agile technological changes. These groups are – ITU (ITU-I h.261/262/263/264 for telecom based applications) and MPEG (MPEG-1//2/4 for applications in computer domain) [1]. Image and Video compression continues to be an active research field as new research on high efficiency video coding is jointly initiated by two groups to reduce the bit coding rates further by 50% [2].

Raw real time video signal captured is an analog signal. To transmit the analog video signal it is required to first convert it to the digital domain. For example NTSC has 30 frames per second, 858 x 525 luminance samples, 429 x 525 x 2 chrominance samples with 8 bits per sample. Therefore, the bit rate = 30 x 8 x ((858 x 525) + (429 x 525 x 2)) = 216.216 Mbps is required to transmit the video. So there is a strong need to process the raw video data to use the resources effectively. This can be done by exploiting the spatial and temporal redundancies present in the consecutive frames of a video. Removal of these redundancies leads to large compression rates. The key factor in video compression is motion estimation. As only some pixels or objects may be displaced in two adjacent frames. So motion vectors of the current frame are computed w.r.t. the reference frame. This is the most time consuming part of video processing.

Many different systems and algorithms have been proposed to enhance video compression in last few years. These algorithms can be categorized as pixel matching algorithms, region based algorithms and block motion estimation based algorithms. The pixel matching algorithms are thresholding based and also they do not give an adaptive solution for different scenes thus cannot be used in practice. Region based algorithms are very complex and involve large amount of computations. The most versatile and robust algorithms are block based motion estimation algorithms and is the field of my research.

In most of the video encoding standards like H.261, H.263, MPEG-1, MPEG-2, MPEG-4, and H.264, block- matching based algorithms are preferred because of its effectiveness and simplicity in hardware implementation as compared to other techniques of motion estimation.

In Block Based algorithms, a frame may have multiple objects moving randomly in different directions, motion of these objects cannot be accurately predicted if entire frame is processed as a single unit. Therefore, each frame of input video is divided into fixed size blocks, called macro blocks (MB) and motion of objects is predicted at macro block level. A current frame block (or macro block) of size mXm is searched for its best match in the reference frame within a search window of size ±p. The displacement of best matching current block in the reference search window gives the motion vector (MV) for the current block. At the decoder, these MVs along with the reference frame are used for predicting the current frame. So the quality of the decoded video depends upon accurate calculation of the MVs.Fig1 depicts the motion estimation based on block matching.

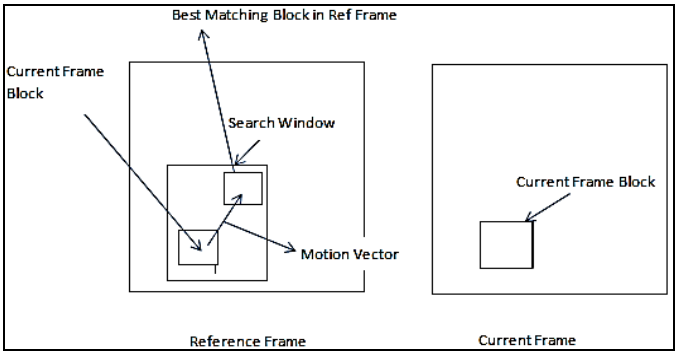

Fig 1: Motion Estimation Based on Block Matching

2. Issues in Estimation of Motion Vectors by Block Matching Technique:

The Performance of the video encoder in Video data Processing can be measured using different criteria such as the quality of the produced bit stream, encoder computational complexity and the resulting compression ratio.

1. Quality of the Produced Bit stream: The quality of the produced bit stream can be measured in terms of both qualitative and quantitative measures. Avg. PSNR is often used quantitatively to measure the processed video bit stream. Higher the PSNR, better the quality. So PSNR plays a big role in developing the Motion estimation Algorithm.

   PSNR = 10 Log (255^2)/MSE

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

   Where C_{ij} and R_{ij} are the current frame and the reference frame respectively.
2. Motion Estimation Computational complexity is another major issue in developing the algorithm. Complexity depends upon the number of search operations performed to find the approximate motion vector for each block in a defined search window. So lesser will be the search points better will be the algorithm.
3. Time taken to estimate the approximate motion vector.

3. Review of Fast Motion Estimation Algorithms

After completing my course work, I initiated my literature survey in the field of Block Matching based motion estimation Algorithms. To achieve the objectives of my research I started looking for the factors that can fasten the process of Motion Estimation and can reduce the complexity. From my study till date I found that there are a no. of factors, like- Search Window Size Fixed Search Window Size or Variable/ Adaptive Search Window Size Selection for Motion Estimation, Block Size- Fixed Block Size and Varying Block Size for Motion Estimation, Motion Estimation Based on Detecting the Edges of Objects in Video Data, Accelerating Motion Estimation by Zero Motion Prejudgment, that can reduce the no. of computations in finding the motion vectors for the current block.

In fixed size block matching algorithms the search range plays a very important role in finding motion vectors. I focused on estimating the search range for estimating the motion vectors. I had done literature survey on the adjustment of Search Range. For Slow motion sequences the motion vectors exists between a smaller search windows and moderate and fast motion sequences large search windows are required. I am trying to find the relation between motion activity and accordingly selection of search window. For this I started my work with the analysis and study of Fast Motion Algorithms based on fixed search pattern based motion estimation algorithms. During this period I communicated three Research papers in various conferences. The results for the two is awaited and one of the papers is accepted and details are as follows:

"Comparative Analysis of Motion Estimation Algorithms on Slow, Medium and Fast Video Sequences", Accepted in IEEE International Conference: ICROITS 2014, by Manav Rachna International University, Faridabad.

Fixed Search Pattern Based Motion Estimation Algorithms

In this approach, it is assumed that motion estimation matching error decreases monotonically as the search moves toward the position of the global minimum error and that the error surface is uni-modal. A fixed set of search pattern is used for finding the motion vectors (MV) of each block.

J.R.Jain and A.K.Jain [3] introduced a 2-D logarithmic search which is accomplished by successively reducing the search area to half in each of the iteration. Each step consists of searching of five locations which contain center of the search window and the mid points between center and the four boundaries of the area along the axes passing through the center. Three step search (TSS) algorithm given by T Koga [4] is also one of the famous algorithms in which the search is accomplished in three steps only. A step size of 2Log(p+1)-1 is adopted for a search window (SW) of size ±p.

A refinement of the TSS named new three step search (NTSS) was next introduced by Li [5] which exploits the center biased characteristics and half way stop to quickly identify the stationary and quasi stationary blocks. In this along with eight points as in TSS eight more points next to the immediate neighbors of the center of search window are checked. If the minimal point is amongst these eight near neighbors, search in the next iteration is processed by taking minimal point as center otherwise the algorithm works as normal TSS. This algorithm searches 17 & 33 locations in its best and worst cases respectively.

Next 4SS was introduced by Po [6] which possesses the center biased and half way stop features of the NTSS. Its computational complexity is less and it is claimed to have better performance for complex motions like camera zooming, fast motion etc. 4SS sets a fixed pattern size of 2 and looks at 9 locations in a 5x5 window. It then proceeds depending on the location of the least weight. It has the best performance of 17 and worst performance of 27 checking points.

One at a time search (OTS) algorithm [7] works in two stages- horizontal stage and vertical stage. Horizontal stage finds the points of minimum distortion on the horizontal axis. Then this becomes the starting point for vertical stage and the minimum distortion in the vertical direction is found.

Orthogonal search algorithm [8] given by Puri is a hybrid of three step search and 2-d logarithmic search. It was modified by Metkar and Talbar [18] to reduce its computational complexity. The method uses the Centre biased search point pattern for estimating small motions and also a half way stop technique to improve the speed of the algorithm over the existing algorithms.

Gradient descent search algorithm (GDSA) given by Liu [9] uses 3x3 size checking blocks and all points of this block are evaluated for minimum point. If distortion is found at center, search is stopped otherwise minimum point is treated as center for the next 3x3 checking block. Number of new search points to be evaluated in successive steps is 3 or 5 depending on whether minimum point is the edge point or corner point of the checking block. Algorithm always moves the search in the direction of optimal gradient descent search where one expects the distortion function to approach its minimum. The algorithm gives a competitive performance with reduced computational complexity.

Zhu [10] analyzed the motion vector distribution of various video sequences using full search and found that up to 98%

motion vectors are enclosed within a radius of 2 from the center of search window. Using these observations a new algorithm called diamond search (DS) was proposed. This algorithm employs two search patterns– large diamond search pattern (LDSP) comprising nine checking points in total and small diamond search pattern (SDSP) with five checking points. For searching, LDSP is repeatedly used until the center point becomes the minimum distortion point. Several modifications of this algorithm have been suggested by many authors for small motion based sequences like videoconferencing.

Hexagon based search algorithm (HEXBS) [11] is based on the same search pattern as in diamond search but in this a large hexagon is used instead of large diamond search pattern.

Further a New Adaptive Fast BMA was proposed by Huang [12] which is capable of efficiently removing the temporal redundancies of sequences with slow, medium and fast motion content.

The most important advantage of the fixed search based methods is their simplicity and regularity. These algorithms reduce the computational loads. But the main disadvantage of these methods is that they are less adaptive and less efficient for tracking large motions and only provide suboptimal solutions. These algorithms tend to be trapped in local minima when motion does not match the predefined pattern.

In paper [13] various block matching algorithms are compared by using two 100 frames QCIF (174 X144) ("Foreman" and "Mother and Daughter") and four 300- frame SIF (352X240) ("Football, "Flower Garden, "Table Tennis", and "Mobile Calendar") video sequences. These sequences contain different type of motion activity from low through medium to high motion. It is clear from the table 1 that there is no single algorithm which is performing equally for all type of motions However BBGDS (block based gradient descent search) and diamond search (DS) are performing better in terms of number of computations as compared to other algorithms. A tabulated performance of results is shown in table 2

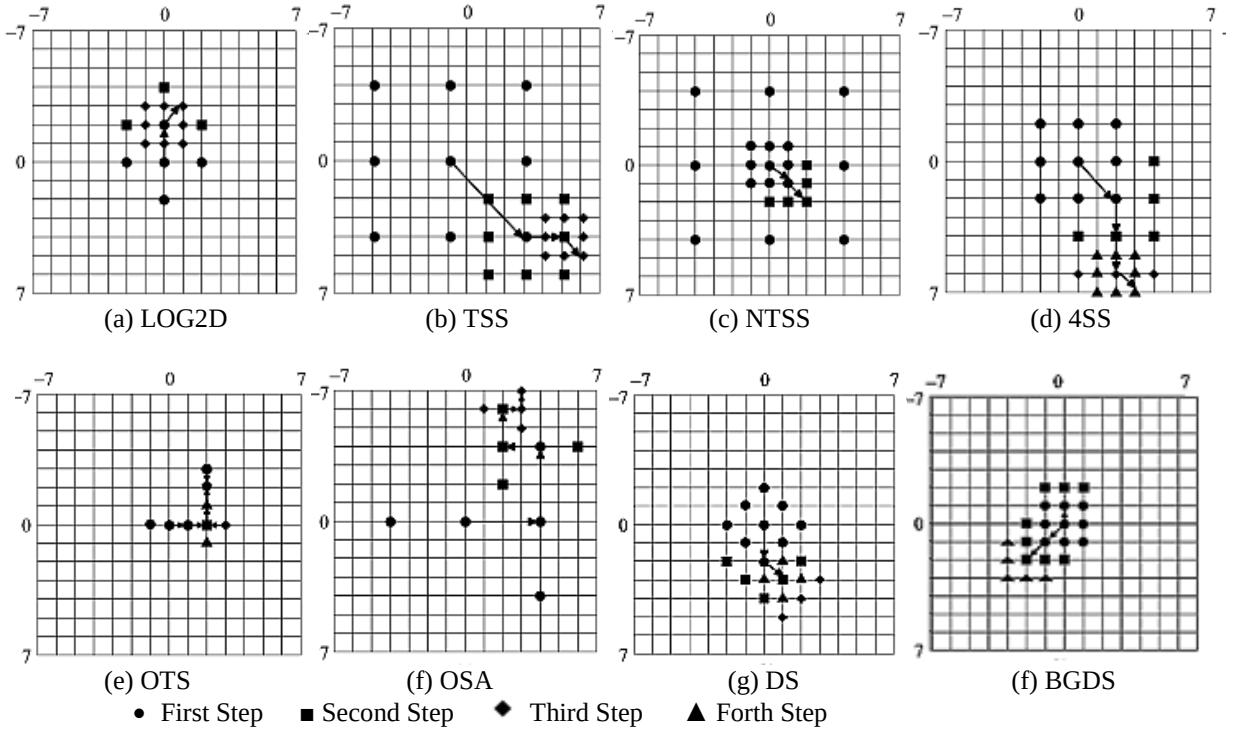

Fig 2: Examples of Search Path

Table 1: Performance Comparison of FS, TSS, 4SS, NTSS, BBGDS and DS [11]

Sequence	Video Type and SR	Motion Type	Comparison Strategy	Algorithms					
				FS	TSS	4SS	NTSS	BBGDS	DS
Flower Garden	CIF /±15	Medium	SPs	859.45	30.52	19.03	20.54	14.03	16.89
			PSNR[dB]	29.92	21.18	23.39	21.93	23.52	23.61
Football	CIF /±15	High	SPs	859.45	30.52	19.19	21.66	15.15	17.11
			PSNR[dB]	23.08	21.82	22.27	22.02	22.08	22.24
Foreman	QCIF/±7	Medium	SPs	184.56	21.66	16.70	18.37	12.47	14.50
			PSNR[dB]	30.65	30.16	29.83	30.23	29.73	29.83
Mobile Calendar	CIF /±15	Medium	SPs	859.45	30.52	15.88	18.04	10.73	12.60
			PSNR[dB]	22.61	22.31	22.51	22.58	22.58	22.54
Mother & Daughter	QCIF/±7	Low	SPs	184.56	21.66	14.94	15.42	8.43	11.85
			PSNR[dB]	39.71	39.59	39.63	39.65	39.62	39.63
Table Tennis	CIF /±15	High	SPs	849.45	30.52	18.31	20.17	12.71	15.54
			PSNR[dB]	29.49	25.82	26.42	26.26	26.25	26.61

In the current report I am basically showing my two simulations. One is the comparative analysis of FS, TSS, NTSS, DS, ARPS algorithms for slow fast and moderate motion sequences. Second is the variation of the size of the search window and to look out its impact on the PSNR for the above algorithms.

1. Based on the motion activity in the consecutive frames video sequences could be broadly divided in to three categories- slow, medium and fast video sequences. I had done the comparative analysis of various motion estimation algorithms on all the three categories. The motion estimation algorithms which are analyzed are Full Search (FS), Three Step Search (TSS), New Three Step Search (NTSS), Diamond Search (DS) and Adaptive Rood Pattern Search (ARPS). The performance of these algorithms is considered in terms of Peak to Signal Ratio (PSNR), number of computations and the time required to find motion vectors per frame. All the algorithms are compared and also the suitability of various algorithms is established for each of the categories of slow, medium and fast video sequences.

Two test sequences representing slow, moderate and fast motion activity in consecutive frames are taken for comparing various block matching algorithms. The test sequences considered for each type of motion activity are Akiyo and Container sequences for slow motion, Foreman and Car Phone sequences for medium motion and Flower and Car Phone for fast motion. The algorithms compared are Full Search (FS), Three Step Search (TSS), New Three Step Search (NTSS), Four Step Search, Diamond Search (DS) and Adaptive Rood Pattern Search (ARPS). A fixed search window size of ±7 and a fixed block size of 16x16 is taken for performing all simulations. For slow and moderate motion first 28 frames are taken and for fast motion 60 frames are considered. Figure 3 shows the difference of two adjacent frames in slow, medium and fast motions. It is clear from this difference figure that there is very slight difference in adjacent blocks in current and the reference frames. Most of the background is constant only small movement is visible in certain blocks. In case of medium motion there is appreciable difference between two frames. For fast motion sequence very large no. of blocks have differences i.e. large motion activity. Mean absolute error matching criteria is opted for finding best matching blocks.

In terms of PSNR, for slow motion DS gives the best results, for medium motion NTSS gives the best results and for fast motion TSS gives the best results. But ARPS outperforms all the algorithms in terms of number of computations and thus the computation time, and the compromise that is done in terms of quality is very less. So ARPS algorithm gives optimized results for slow, medium and fast motion sequences and is optimum for practical implementations. Further for fast motion activity there is large reduction in the PSNR as compared to slow and moderate motion sequences. So this gives a new research direction to design a technique (criteria and algorithm) to improve the quality of fast motion sequences.

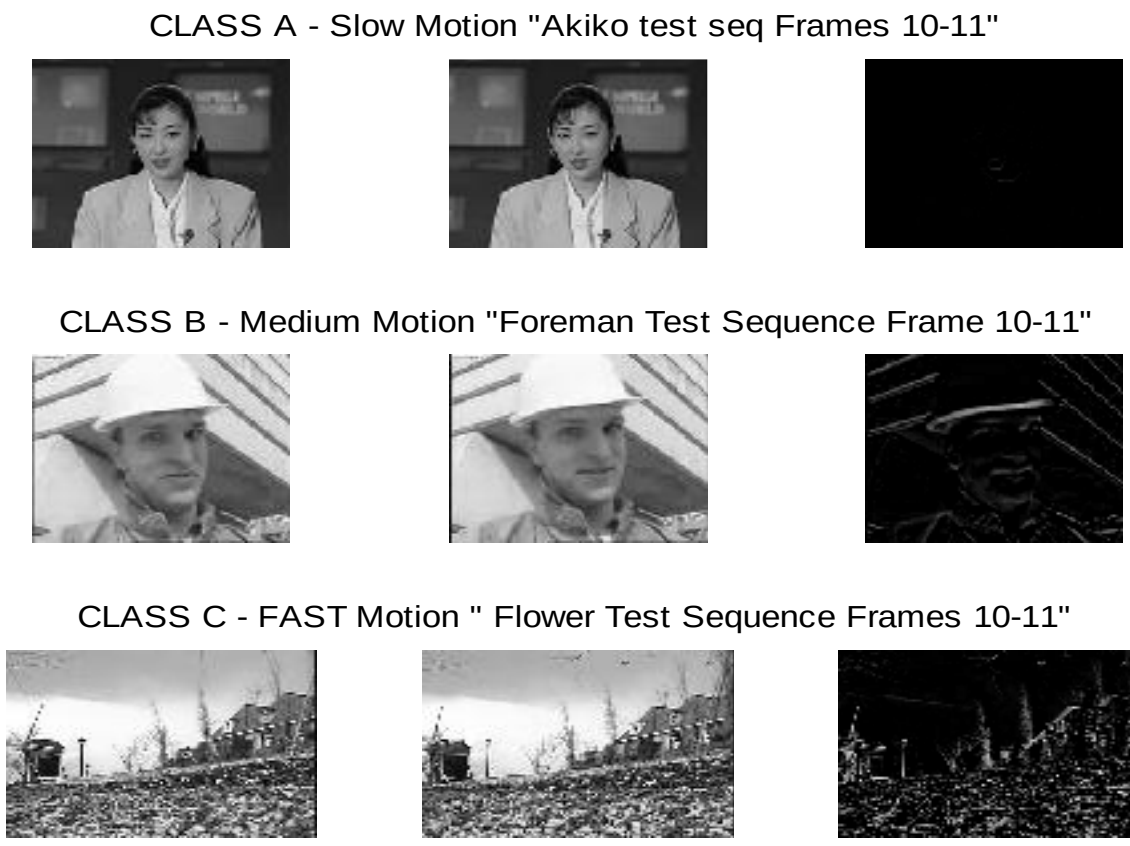

Fig 3: Difference between two consecutive frames in Slow, Moderate and fast Motion Video Sequences

Table 2: Average No. of Computations Required to Find Motion Vectors

Motion Estimation Techniques	Class A (28 Frames)		Class B (28 Frames)		Class C (60 Frames)	
	Akiyo	Container	Foreman	Car Phone	Flower	India Pride
Full Search	177.7201	177.7202	188.7051	177.7202	200.5836	197.7439
Three Step Search	20.6891	20.8556	22.1677	21.2798	23.0702	22.8877
New Three Step Search	14.141	16.3427	25.064	21.4276	27.5954	28.617
Four Step Search	14.141	15.3423	25.064	18.2174	27.5954	22.9855
Diamond Search	11.0127	12.9357	21.3497	17.73696	22.5836	26.49
Adaptive Rood Pattern Search	4.6988	6.9431	12.6988	11.2798	13.4917	14.85

Table 3: Average PSNR per Frame

Motion Estimation Techniques	Class A (28 Frames)		Class B (28 Frames)		Class C (60 Frames)	
	Akiyo	Container	Foreman	Car Phone	Flower	India pride
Full Search	40.9346	30.0907	20.5190	22.6994	13.5073	18.7888
Three Step Search	40.9346	30.065	20.1347	22.5229	13.4342	18.5824
New Three Step Search	40.9346	30.0834	20.1829	22.5198	13.4082	18.5679
Four Step Search	40.9346	30.0726	20.094	22.4221	13.2644	18.383
Diamond Search	40.9346	30.0856	20.0519	22.4767	13.2073	18.4158
Adaptive Rood Pattern Search	40.9346	30.0816	20.1832	22.4278	13.3071	18.64

Table 4: Time Required for Calculating MVs per Frame

Motion Estimation Techniques	Class A		Class B		Class C	
	Akiyo	Container	Foreman	Car Phone	Flower	India Pride
Full Search	0.2379	0.2467	0.5259	0.2376	0.9062	0.7810
Three Step Search	0.0345	0.0400	0.0706	0.0337	0.1206	0.1051
New Three Step Search	0.0605	0.0350	0.0455	0.0403	0.1365	0.1174
Four Step Search	0.0324	0.0402	0.0835	0.0318	0.1057	0.1231
Diamond Search	0.0521	0.0747	0.1129	0.0455	0.1729	0.1903
Adaptive Rood Pattern Search	0.0263	0.0446	0.0745	0.0360	0.1177	0.1255

2. Simulation by Varying the Size of Search Window - SW plays a very important role in the determination of Motion Vectors in motion estimation. Correct choice of the window size may lead to reduction in the number of computations required to find motion vectors without compromising with the quality. In case of Fast motion large search window may be required to find accurate motion vectors and for slow or medium motion smaller search window may give good results. Table 5 below shows the various search window sizes for slow, medium and fast motion sequences and the corresponding PSNRs.

Standard sequences for slow, medium and fast motion namely Container sequence, Car Phone sequence and Flower sequence are taken for simulation. These sequences are tested using the fixed search based motion estimation algorithms namely Full Search (FS), Three Step Search (TSS), New Three Step Search (NTSS), Diamond Search and Adaptive rood pattern search. PSNR and average number of computations per frame are calculated for 30 frames. Search window sizes that are used in the analysis are ±2, ±3, ±4 and ±7.

Table 5: Comparison of PSNR and No. of Computations for Different Motion Activity Sequences using Variable Search Window

Motion Estimation Techniques	Class A		Class B		Class C	
	Container		Car Phone		Flower	
	Computations Required per Frame	Avg PSNR per Frame	Computations Required per Frame	Avg PSNR per Frame	Computations Required per Frame	Avg PSNR per Frame
FullSearch p=7	177.7202	30.0907	177.7202	22.6994	200.5836	13.2073
p=2	20.3389	30.06979	20.3389	20.9981	21.9613	12.5419
p=3	39.3647	30.0784	14.4137	21.5296	42.7555	12.8431
p=4	60.2234	30.08331	60.2234	21.9199	64.3423	13.1252
TSS p=7	20.8556	30.065	21.2798	22.5229	23.0702	13.4342
p=2	7.5383	30.0184	7.5383	20.3258	8.0305	12.1818
p=3	14.2106	30.0711	18.7298	21.4765	15.3361	12.7805
p=4	16.3423	30.0711	19.2248	21.4765	17.2134	12.7805
NTSS p=7	16.3427	30.0834	21.4276	22.5198	27.5954	13.4082
p=2	16.3427	25.4441	19.2313	19.2313	11.6854	11.6854
p=3	16.1597	30.0774	18.7298	21.5046	21.578	12.7762
p=4	16.9987	30.0774	18.9298	21.5046	21.688	12.7762
FSS p=7	15.3423	30.0726	18.2174	22.4221	27.5954	13.2644

Motion Estimation Techniques	Class A		Class B		Class C	
	Container		Car Phone		Flower	
	Computations Required per Frame	Avg PSNR per Frame	Computations Required per Frame	Avg PSNR per Frame	Computations Required per Frame	Avg PSNR per Frame
p=2	15.3423	30.07256	18.2173	22.4221	21.4713	13.3644
p=3	15.3423	30.07256	18.2173	22.4221	21.4713	13.3644
p=4	15.3423	30.07256	18.2173	22.4221	21.4713	13.3644
DS p=7	12.9357	30.0856	17.73696	22.4767	22.5836	13.2073
p=2	11.3445	30.0693	11.5712	20.9815	12.3789	12.52
p=3	12.2779	30.07668	13.5914	21.4969	15.3341	12.77
p=4	12.4279	30.0803	13.2114	21.8622	15.5341	12.9595
ARPS p=7	6.9431	30.0816	11.2798	22.4278	13.4917	13.3071
p=2	6.4579	30.067	8.0056	20.9583	9.2755	12.5083
p=3	6.722	30.0735	8.9326	21.4616	10.6213	12.7481
p=4	6.833	30.0759	8.9326	21.8137	10.6213	12.9549

4. Conclusion

Various techniques for fastening the process of motion estimation have been discussed in this paper. Effective ways to reduce the search points to fasten motion estimation are by zero motion prejudgment, initial search center prediction.

5. References

1. Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-TRec.H.264/ISO/IEC14496-10AVC in Joint Video Team (JVT) of ISO/IECMPE Gand ITU-TVCEG, JVT G050, 2003.
2. Francesca De Simone, Lutz Goldmann, Jong-Seok Lee and Touradj Ebrahimi, "Towards high efficiency video coding: Subjective evaluation of potential coding technologies", Journal of Visual Communication and Image Representation. 2011; 22(8):734-748.
3. Jain JR, Jain AK. "Displacement measurement and its application in interframe image coding", IEEE Transactions on Communications. 1981; 29(12):1799-1808.
4. Koga T, Iinuma K, Hirano A, Iijima Y, Ishiguru T, "Motion compensated interframe coding for video conferencing", in Proc of NTC81. New Orleans, LA. 1981; C9.6.1-C9.6.5.
5. Li R, Zeng B, Liou ML. "A new three step search algorithm for block motion estimation", IEEE Transactions on Circuits and Systems for Video Technology. 1994; 4:438-442.
6. Po L, Ma W. "A novel four step search algorithm for fast block Motion estimation", IEEE Transactions on Circuits and Systems for Video Technology. 1996; 6(3):313-317.
7. Mei-Juan Chen, Liang-Gee Chen, Tzi-Dar Chiueh. "One-dimensional full search motion estimation algorithm for video coding", IEEE Transactions on Circuits and Systems for Video Technology. 1994; 4(5):504-509.
8. Puri A, Hang HM, Schilling DL. "An efficient block-matching algorithm for motion compensated coding", in Proc. IEEE – ICASSP. 1987; 25.4.1-25.4.4.
9. Metkar S, Talbar S. "Fast motion estimation using modified orthogonal search algorithm for video compression", Signal, Image and Video Processing. 2010; 4:123-138.
10. Liu L, Feig E. "A block based gradient descent search algorithm for block motion estimation in video coding", IEEE Transactions on Circuits and Systems for Video Technology. 1996; 6(4):419-422.
11. Zhu S, Ma K. " A new diamond search algorithm for fast block Matching motion estimation" IEEE Transactions on Image Processing. 2000; 9(2):287-317.
12. Zhu C, Lin X, Chau L, "Hexagon based search pattern for fast block motion estimation", IEEE Transactions on Circuits and Systems for Video Technology. 2002; 12(5):349-355.
13. Huang SY, Cho CY, Wang JS. "Adaptive fast block matching algorithm by switching search patterns for sequences with wide range motion content", IEEE Transactions on Circuits and Systems for Video Technology. 2005; 15(11):1373-1384.
14. Hsieh CH, Lu PC, Shyn JS, Lu EH. "Motion estimation algorithm using interblock correlation", IEE Electron. Lett. 1990; 26(5):276-277.
15. Wenbin Jiang, Manli Zhou, Fuyuan Peng, Yiping Xu. "Novel block-matching algorithms by subsampling both search candidates and pixels", IEEE Journal of Systems Engineering and Electronics. 2005; 16(3):533-537.
16. Chung KL. "A new predictive search area approach for fast block estimation", IEEE Transactions on image processing. 2003; 12(6):648-652.
17. Yamada T, Ikekawa Masao, Kuroda I. "Fast and accurate motion estimation algorithm by adaptive search range and shape selection", in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing. 2005; 2:897-900.
18. Shih Chang Hsia, Wei Chih Hsu, Shih Ren Wu. "A fast rate distortion optimization algorithm for H.264/AVC codec", Signal, Image and Video Processing. 2013; 7(5):939-949.
19. Ben Ayed A, Samet A, Masmoudi N. "Toward an optimal block motion estimation algorithm for H.264/AVC", International Journal of Image and Graphics. 2006; 5(2):23-26.
20. Yang L, Yu K, Li J, Li S. "An effective variable block size early termination algorithm for H.264 video coding," IEEE Trans. on Circuits and Systems for Video Tech. 2005; 15(6):784-788.
21. An-Chao Tsai, Bharanitharan K, Jhing Fa Wang, Kuan I, Lee. "Effective Search Point Reduction Algorithm and Its VLSI Design for HDTV H.264/AVC Variable Block Size Motion Estimation", IEEE Transactions on Circuits and Systems for Video Technology. 2012; 22(7):981-988.

22. Yui Lam Chan, Wan Chi Siu, "Block motion vector estimation using edge matching: an approach with better frame quality as compared to full search algorithm", in Proc. IEEE International Symposium on Circuits and Systems. 1997; 2:1145-1148.
23. Phat Nguyen Huu, Vinh Tran Quang, Miyoshi T. "Efficient Motion Estimation Algorithm Using Edge Feature and Arithmetic Coding for Video Compression on WVSNs", in Proc. of IEEE International Conference on Software, Telecommunications and Computer Networks, 2011, 1-5.
24. Amrita Ganguly, Anil Mahanta. "Improved mode decision algorithm based on residues and early zero block detection in H.264/AVC", Signal, Image and Video Processing, Springer, Online Publication, 2012.