I.J. Image, Graphics and Signal Processing, 2015, 7, 60-68
Published Online June 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2015.07.07

Multi Point Search Pattern for Fast Search Motion Estimation of High Resolution Video Coding

Nehal N. Shah
Sarvajanik College of Engineering and Technology, Surat, 395001, India
Email: nehal.shah@scet.ac.in

Upena D. Dalal[#1] and Priyank H. Prajapati[#2]
Surat, India
Email: [#1]udd@eced.svnit.ac.in, [#2]priyank.prajapati@rediffmail.com

Abstract—Block matching algorithm (BMA) based motion estimation (ME) is most accepted method for removal of temporal redundancy between frames in video coding. With recent advancement in resolution of video, the need of search pattern covering most of macroblocks within search area in frame is increasing. Existing search patterns are tiny and take plenty of time to reach at edge or corner of the search window. With aim of covering nearly every probable candidate macroblocks in all direction and to speed up the search process, multipoint search patterns are presented in this paper. Initial candidate macroblocks are chosen on grid of 12x12 and then search progresses like traditional diamond or hexagon search. Due to multipoint, chances of trapping in incorrect direction is very less and method can exhibit better quality of encoding with optimum number of search points.

Index Terms—Motion Estimation, Fast search block matching algorithm, Multi point search, H.265, HD video sequences.

I. INTRODUCTION

Video consists of huge amount of data that if not compressed, then occupies large bandwidth in transmission and vast storage space. For example raw, 1080p (1920x1080) full HD video at frame rate of 30 fps (frames per second) require 1.49 Gbps bit rate. High efficiency video coding (HEVC / H.265) [1] standard supports large picture size up to 8192x4320 resolutions, which raise demand of huge storage space, if not compressed. A typical system needs to send dozens of individual frames per second to create an illusion of a moving picture. Each individual frame is coded such that redundancy is removed with a motion estimation (ME) and motion compensation (MC) system. An ME is the most computational intensive operation in the coding hence, Block Matching Algorithm (BMA) [2] based ME and MC techniques have been widely accepted to alleviate the computational burden. BMAs estimate motion on the basis of rectangular blocks and assume that all the pels (picture elements) within a block have the same motion activity, therefore produce one motion vector (MV) for each block.

Real-time and low computation intensive encoding requirements create great challenges for design engineers. Traditional fast search ME algorithms designed for H.264 video coding standards start from the center of search window and assumes most of MVs are located in 0 to 2 radiuses. But with larger size of video frames MVs are located much faraway which demands ME algorithm which takes care of MV located at distant location in all directions in search window and avoid trapping in local minima. In this paper section II discuss block matching method for motion estimation. Literature review of existing fast search BMAs is presented and BMAs used for hardware implementation are also evaluated. Section III describes multipoint search patterns with diamond and hexagon shapes. Multipoint search pattern with a lesser amount of search point is compared with existing BMAs. Experimental results are presented in section IV which is followed by conclusion.

II. BLOCK MATCHING METHOD AND ALGORITHMS

In motion estimation (ME) previously coded frames are used to generate MV which contributes significant bit saving. For ME current frame is divided in to block of MxN called macroblock (MB). From reference frame, BMA selects a prediction region, identify predicted block and subtract this from the original block of samples to form a residual. It utilizes multiple prediction block sizes i.e. 64×64 down to a 4×4, multiple reference frames and special modes such as direct and weighted prediction in H.265. By selecting the best prediction options for an individual macroblock, an encoder can minimize the residual size to produce a highly compressed bitstream.

Block matching method is shown in fig. 1; where MB from current frame is compared within search window in reference frame. Size of search window is (N+2p)x(M+2p) with horizontal and vertical displacement dx=dy={-p,p}. A similarity measure is calculated for all candidate MBs in the search window and the correlation corresponding to the largest similarity becomes the best match of the block under consideration in current frame. The relative position between the block and its best match gives the motion vector. Different matching criterions (cost function) have been proposed in the literature to calculate the distortion between the blocks are MSE (Mean Square Error), MAD (Mean absolute difference) and SAD (Sum of Absolute Difference). SAD is mostly used in video processing techniques because of its less computational complexity than MSE and MAD.

Full search block matching algorithm (FSBMA) is highly computational intensive operation as there are $(2p + 1)^2$ CMBs to be matched for each current MB to deliver global optimum solution with good accuracy. To overcome this drawback, fast BMAs have been developed which lighten the computational burden. Simple methods with fixed number of iterations like TSS [3] and 4SS [4] are not suitable for identifying motion everywhere in the search window therefore methods with unrestricted iterations are proposed. Block based gradient decent search (BBGDS) [5] use square pattern in grid of 3x3. Diamond Search (DS) [6] [7] and Hexagon Search (HEXBS) [8] use diamond and hexagon shapes in grid of 5x5 and finally use small diamond shape in grid of 3x3. For reduction in number of search points, at cost of quality of encoding, modified version of DS and HEXBS are available as Cross Diamond Search (CDS) [9], Directional Diamond Search (DDS) [10], Cross Diamond Hexagonal search (CDHS) [11], Predict Hexagon Search (PHS) [12] etc. All these search patterns are really little for HD video sequences hence take lot of time to reach at distant location in search window.

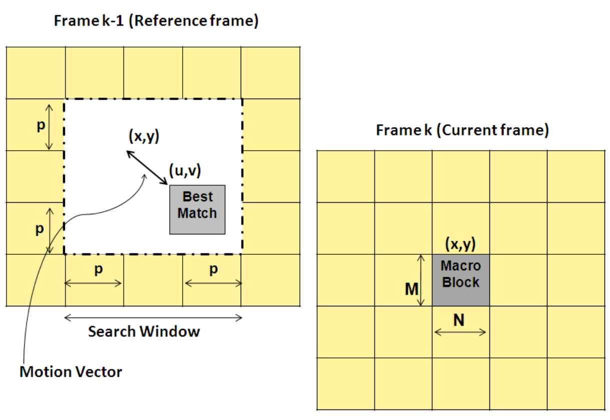

Fig.1. Block matching method

BMAs like Efficient Three Step Search (E3SS) [13] choose among TSS and small diamond. Hybrid motion compensation technique (H-MCT) [14] is based on E3SS and CDS. Switching Search Patterns (SSP) adaptively [15] [16] claims to cover wide range of motion by using combination of search patterns as well variable step size and exhibit speedup or improvement in quality of encoding at the cost of other parameter. Algorithms with switching patterns are complex in implementation and unable to cover wide area of search window. Unsymmetrical cross Multi Hexagon grid Search (UMHS) [17] and its variant [18] covers most of area of search window and claims quality comparable to FSBMA but use several variety of search patterns and many search points. Taking into account two search paths [19] with first and second minima provides better result from error perspective at cost of almost double computation load. In directional search [20] initial search pattern covers all direction but for refining search, BBGDS is used which is slow down the search process. Fuzzy logic based TSS and 4SS are implemented in [21] [22] respectively, which demonstrate reduction in search points among actual candidate macroblocks of both methods. TSS and 4SS are archaic methods and supplementary computations are required for identification of useful candidate macroblocks using fuzzy approach.

Algorithms with highly irregular shapes and toggling search pattern based on result of previous iteration are complex in hardware realization due to access of macroblocks, several steps and additional computation involved in decision making. In hardware implementation of fast search BMAs, most of architecture uses DS, HEXBS and their variants. Architecture based on DS algorithm with 4:1 pixel sub-sampling technique is presented in [23] by Porto et al. which demonstrate fast implementation but degraded quality compared to DS due to sub-sampling. Architecture based on PHS is presented in [24] which is complex due to switching search pattern among small and large PHS. Architecture presented in [25] is configurable for mapping five different BMAs (BBGDS, DS, CDS, HEXBS, and TSS). Hardware oriented Modified Diamond Search (HMDS) [26] use two shapes having 17 and 13 search points and provides better PSNR for center biased low motion sequences.

Search patterns with repeating shape are preferred in hardware implementation. By keeping that in mind, in this paper multi point search pattern which covers wide area in search window during first step and use diamond or hexagon shape for refining search are proposed and compared for quality of video and number of search points. Hexagon shape use less number of search points compared to diamond while diamond shape provides better quality by covering MV in all directions. Hence according to requirement either of them can be employed.

III. MULTIPOINT SEARCH PATTERN

In multipoint search pattern, search begins from center of the search window and big diamond or big hexagon is used. As indicated in fig. 2, multipoint search pattern with big outer diamond (MPBDS) utilize grid of 12x12 and 9 search points indicated with orange color are evaluated in initial stage. Around match of first stage, diamond is created with grid of 5x5 and search progresses further as per diamond search [7] algorithm as indicated with blue color. To reduce first stage as well as remaining stage search points from 9 to 7 hexagon shape is employed in which search progresses like HEXBS [8] after initial search and called multipoint search pattern with big hexagon (MPBHS) as indicated in fig. 3.

Video sequences generated during video conference or video call has low motion content and most of MVs are around center. To take care of such motion, more search points are required at the center.

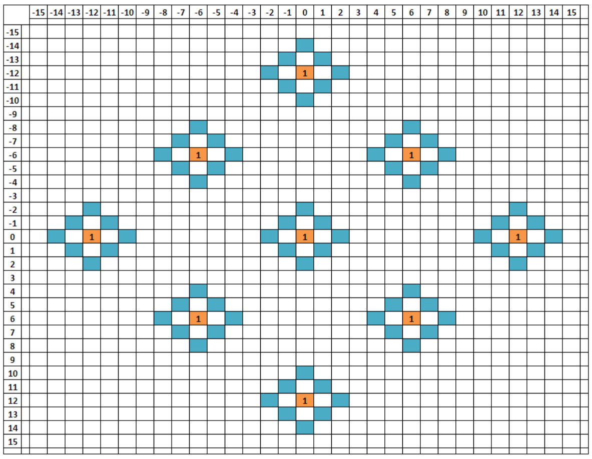

Fig.2. Multipoint search pattern with big outer diamond (MPBDS)

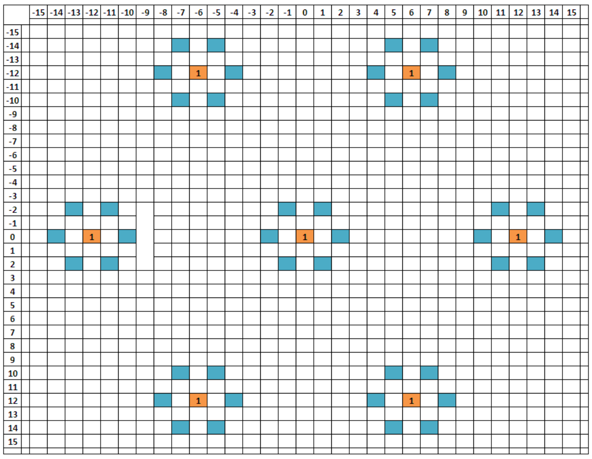

Fig.3. Multipoint search pattern with big outer hexagon (MPBHS)

Therefore, in both the variants of multipoint search patterns, diamond and hexagon at grid of 5x5 are added around center as indicated in (MPDS) fig. 4 and (MPHS) fig. 5 respectively. Number of search points required to reach at any location in search window of [-16, +16] is indicated in fig. 6 to fig. 9 for all four variants of multi point search patterns and also compared in table 1. Among existing fast search BMAs, average number of checking points considering equal probability of MV at all locations in search area of [-16, +16] are least in HEXBS as indicated in table 1. These search points can be further reduces by using MPBHS which is variant of HEXBS. Other variants of multipoint search patterns like MPBDS, MPHS and MPDS also use relatively less search points compared to DS, PHS, UMHS, and HMDS etc. Multi point search patterns are exhaustively tested and results are presented in section IV.

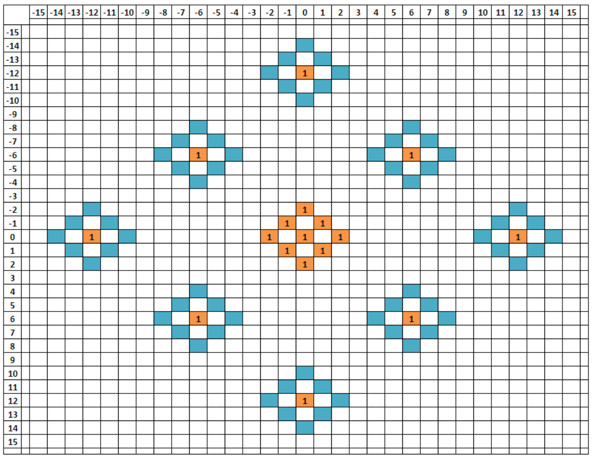

Fig.4. Multipoint search with inner and big outer diamond (MPDS)

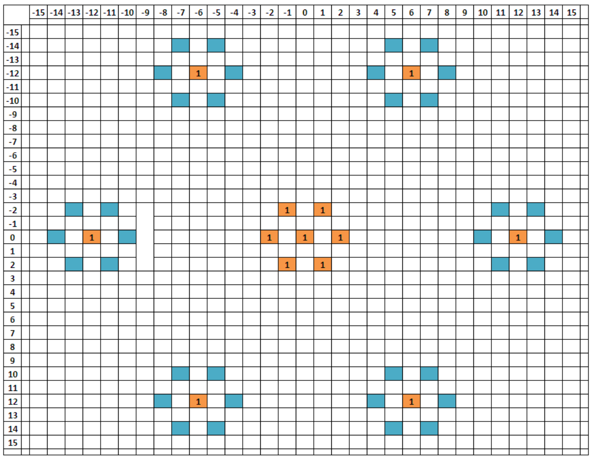

Fig.5. Multipoint search with inner and big outer hexagon (MPHS)

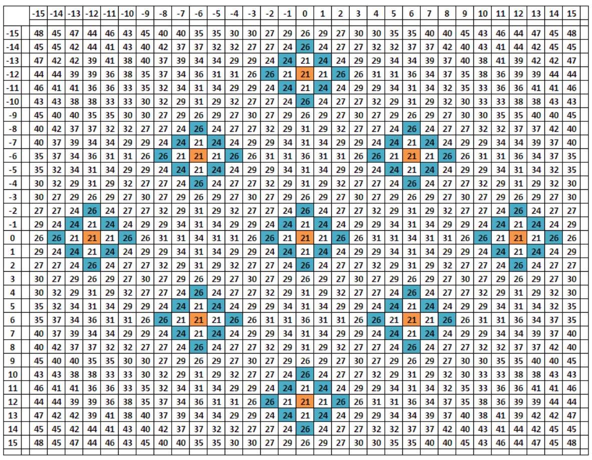

Fig.6. Minimum number of search points required to reach at any location in search window of [-16, +16] using MPBDS algorithm

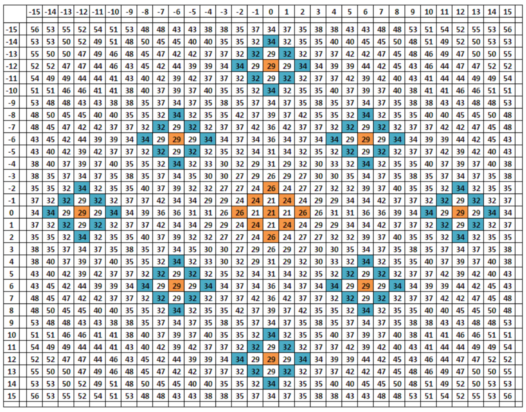

Fig.7. Minimum number of search points required to reach at any location in search window of [-16, +16] using MPDS algorithm

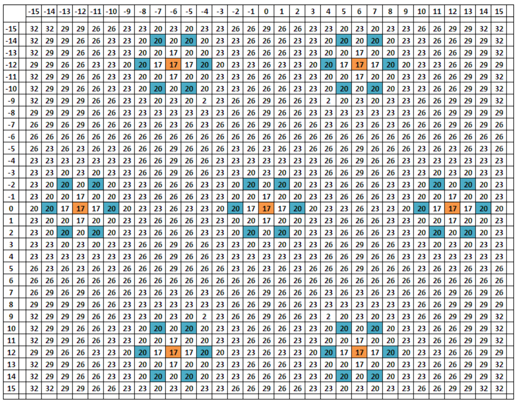

Fig.8. Minimum number of search points required to reach at any location in search window of [-16, +16] using MPBHS algorithm

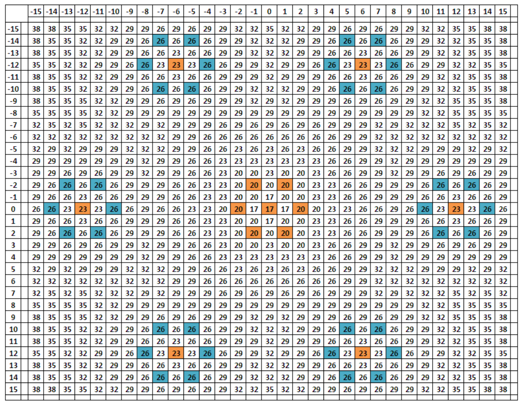

Fig.9. Minimum number of search points required to reach at any location in search window of [-16, +16] using MPHS algorithm

IV. Experimental Results

To evaluate the performance of proposed multipoint search patterns, experiments are conducted with video sequences having diverse characteristics. 14 HD resolution (1920x1080) video sequences are used form YUV video repository [27]. MB size of 8x8 and search window area is chosen as [-16, +16]. For the performance evaluation average PSNR value of the luma component, number search points required per MB and average structural similarity index (SSIM) are measured and tabulated in table 2, 3, 4 respectively. PSNR and SSIM are calculated for each frame and numbers of search points are measured for each macroblock and then their average is calculated.

Compared to HMDS, all multi point search patterns outperforms for all three measurable quantity like quality of encoding measured by PSNR as well as SSIM and number of search points. For slightly better quality in UMHS huge computation is involved as indicated in table 3. Compared to HEXBS all proposed patterns uses more number of search points due to nature of random probability of motion vectors and most of MVs are around center but in terms of quality of encoding they offer better performance compared to both DS as well as HEXBS. Bigger search area results in better quality at cost of more number of computations. By increasing search window to [-32, +32], proposed search patterns are tested and improvement in PSNR as well as increased number of computation are indicated in table 5. Average PSNR enhancement is 0.24 for all 14 sequences and computation increase is 0.56 per macroblock. For moderate to high motion sequences like pedestrian, rush_hour and speed_bag such improvement is significant as indicated in table. Average Y-PSNR for Speed bag, Pedestrian and Rush hour video sequences of 1080p resolution is shown in fig. 10, fig. 11 and fig. 12 respectively.

Fig. 13, fig. 14 and fig. 15 indicates number of search points required in proposed multi point search patterns in comparison with UMHS and HMDS for Speed bag, Pedestrian and Rush hour video sequences. All proposed search patterns use less number of search points in comparison with UMHS and HMDS. Similar results are available for other test sequences also.

Table1. Comparison of required search points for various locations in search window of [-16, +16] for existing and proposed fast search BMAs

Search points location	DS	HEXBS	PHS	UMHS	HMDS	MPDS	MPBDS	MPHS	MPBHS
For center point (0,0)	13	11	5	109	21	21	21	17	17
For corner points (-15,-15), (-15,15), (15,15), (15,-15)	59	45	58	119	41	56	48	38	32
For Edge points (-15,0), (15,0)	48	32	33	119	41	34	26	26	20
For Edge points (0,-15), (0,15)	48	35	33	119	41	34	26	35	29
Average number of checking Points considering equal probability of MV at all locations in search area of [-16, +16]	40.2	29	34.5	119	41	38	31	29	24

Table2. Comparison of Average Y-PSNR in multipoint search patterns with existing fast search BMAs for HD sequences.

Video Sequence	DS	HEXBS	UMHS	HMDS	MPBHS	MPBDS	MPHS	MPDS
blue_sky	26.401	25.765	29.013	24.148	24.226	25.851	25.286	26.668
Riverbed	24.377	24.340	26.616	25.010	25.808	25.962	25.931	26.007
Station	35.312	35.009	35.336	35.003	33.524	33.951	34.751	35.156
Aspen	35.285	34.985	35.773	34.653	33.431	34.287	34.890	35.468
crowd_run	25.861	25.476	26.436	25.513	24.094	24.711	25.452	26.106
Pedestrian	29.970	29.776	31.895	30.465	31.029	31.262	31.153	31.362
ducks_takeoff	25.778	25.612	25.941	25.828	25.610	25.745	25.678	25.854
into_tree	30.100	29.835	30.305	29.983	28.171	28.408	29.746	30.037
old_town_cross	30.535	30.228	30.457	30.576	29.242	29.501	30.205	30.545
park_joy	24.060	23.757	24.264	24.522	21.753	21.777	23.896	24.116
rush_hour	34.365	34.132	35.154	34.081	33.947	34.548	34.481	34.880
sunflower	34.351	33.848	35.211	33.441	32.110	33.269	33.769	34.481
speed_bag	24.883	24.819	25.839	24.953	25.260	25.425	25.447	25.572
controlled_burn	34.219	34.130	34.602	34.233	34.158	34.286	34.326	34.438

Table3. Comparison of Number of search points per macroblock in multipoint search patterns with existing fast search BMAs for HD sequences.

Video Sequence	DS	HEXBS	UMHS	HMDS	MPBHS	MPBDS	MPHS	MPDS
blue_sky	25.840	18.548	108.056	38.870	20.750	26.576	24.743	32.562
riverbed	30.544	20.915	109.113	39.207	21.850	29.007	25.640	34.612
Station	19.456	14.802	111.790	37.806	19.613	25.226	21.153	27.651
aspen	21.948	14.802	109.451	39.904	20.209	26.043	22.736	29.945
crowd_run	19.364	14.728	111.292	38.703	19.603	25.192	20.971	27.281
pedestrian	24.238	17.185	107.868	33.468	19.300	25.705	22.120	29.661
ducks_takeoff	16.582	13.327	111.778	37.782	18.971	24.244	19.487	24.696
into_tree	19.335	14.856	110.514	39.973	19.514	25.037	21.208	27.651
old_town_cross	17.372	13.816	110.078	38.802	18.933	24.129	20.543	26.330
park_joy	20.794	15.682	109.181	39.558	19.867	25.697	21.818	28.693
rush_hour	21.650	16.221	110.111	39.679	20.215	25.984	22.384	29.281
sunflower	24.766	17.648	109.052	40.469	21.539	28.401	23.267	31.429
speed_bag	27.640	19.165	106.819	37.104	20.423	27.714	23.919	32.698
controlled_burn	18.216	13.916	112.826	26.522	18.339	23.590	20.173	26.002

Table4. Comparison of Average SSIM in multipoint search patterns with existing fast search BMAs for HD sequences.

Video Sequence	DS	HEXBS	UMHS	HMDS	MPBHS	MPBDS	MPHS	MPDS
blue_sky	0.953	0.946	0.980	0.917	0.915	0.943	0.939	0.957
riverbed	0.842	0.841	0.915	0.861	0.890	0.895	0.894	0.897
Station	0.984	0.983	0.985	0.981	0.970	0.975	0.981	0.984
aspen	0.981	0.981	0.987	0.979	0.975	0.980	0.982	0.984
crowd_run	0.970	0.968	0.978	0.967	0.950	0.957	0.969	0.974
pedestrian	0.946	0.944	0.964	0.951	0.957	0.959	0.958	0.959
ducks_takeoff	0.963	0.961	0.967	0.964	0.961	0.963	0.963	0.965
into_tree	0.960	0.959	0.971	0.959	0.932	0.936	0.959	0.961
old_town_cross	0.975	0.975	0.978	0.976	0.965	0.968	0.975	0.976
park_joy	0.941	0.938	0.958	0.951	0.887	0.889	0.940	0.941
rush_hour	0.979	0.979	0.985	0.980	0.980	0.982	0.982	0.983
sunflower	0.991	0.990	0.994	0.988	0.987	0.990	0.991	0.993
speed_bag	0.902	0.901	0.913	0.903	0.909	0.909	0.909	0.910
controlled_burn	0.975	0.975	0.980	0.975	0.977	0.978	0.978	0.978

Table5. Improvement in PSNR as well as raised number of search points per MB for proposed multipoint search patterns with bigger search window of [-32, +32]

Video Sequence	MPBHS	MPBDS	MPHS	MPDS	MPBHS	MPBDS	MPHS	MPDS
blue_sky	0.044	0.017	0.026	0.012	0.278	0.153	0.182	0.116
riverbed	0.243	0.221	0.220	0.212	0.814	0.831	0.651	0.740
Station	0.018	0.007	0.006	0.003	0.129	0.058	0.045	0.019
aspen	0.083	0.053	0.064	0.049	0.213	0.134	0.120	0.095
crowd_run	0.020	0.007	0.010	0.004	0.114	0.053	0.047	0.023
pedestrian	1.473	1.291	1.484	1.305	1.306	1.675	1.257	1.631
ducks_takeoff	0.010	0.009	0.008	0.007	0.036	0.035	0.020	0.015
into_tree	0.010	0.005	0.003	0.001	0.100	0.049	0.029	0.014
old_town_cross	0.008	0.004	0.001	0.000	0.061	0.034	0.015	0.005
park_joy	0.079	0.065	0.108	0.100	0.365	0.358	0.267	0.306
rush_hour	0.443	0.407	0.447	0.416	0.442	0.427	0.314	0.342
sunflower	0.095	0.058	0.060	0.049	0.252	0.164	0.133	0.115
speed_bag	0.917	0.828	0.913	0.828	2.929	4.380	2.842	4.319
controlled_burn	0.170	0.155	0.163	0.152	0.545	0.697	0.507	0.664
Average	**0.258**	**0.223**	**0.251**	**0.224**	**0.542**	**0.646**	**0.459**	**0.600**

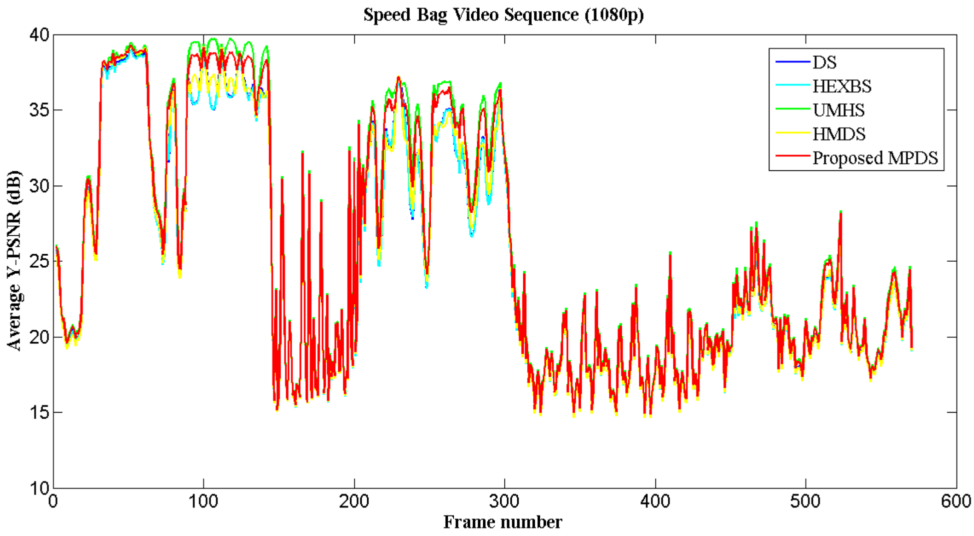

Fig.10. Comparison of Average Y-PSNR with existing fast search BMAs using Speed bag video sequence

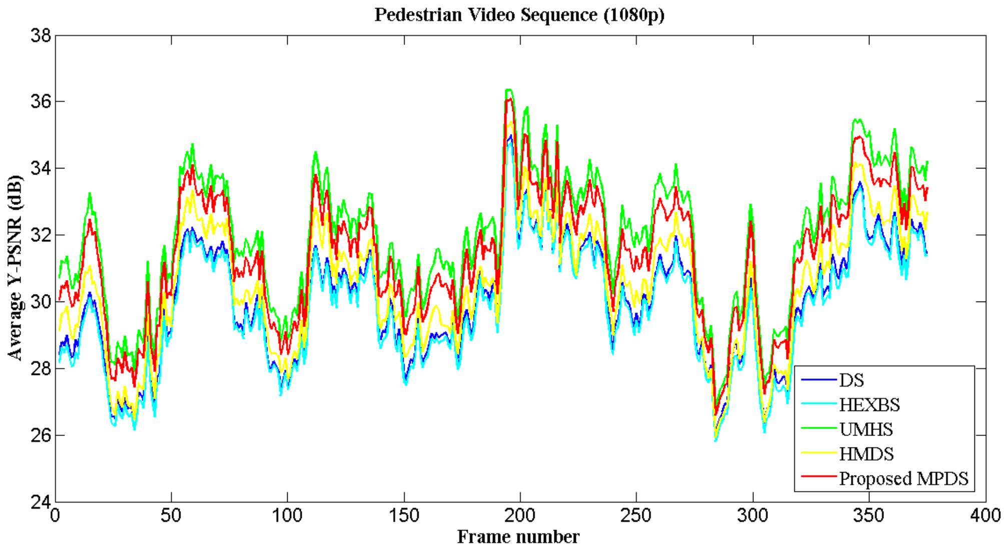

Fig.11. Comparison of Average Y-PSNR with existing fast search BMAs using Pedestrian video sequence

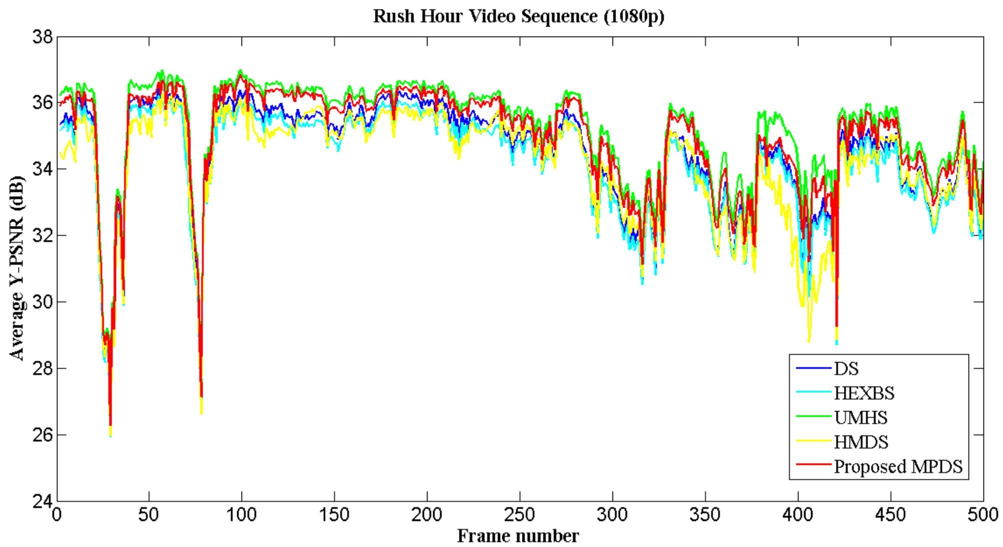

Fig.12. Comparison of Average Y-PSNR with existing fast search BMAs using Rush hour video sequence

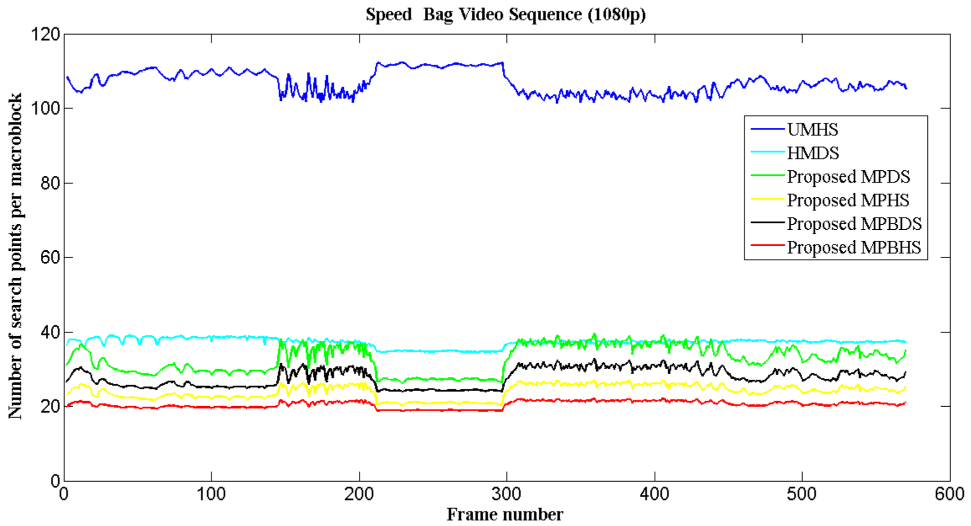

Fig.13. Comparison of number of search points / macroblock with existing fast search BMAs using Speed bag video sequence

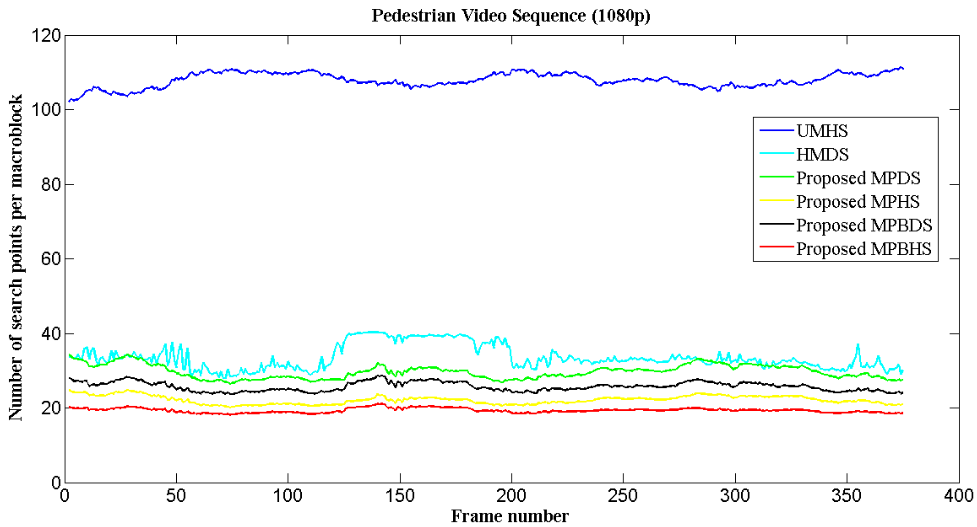

Fig.14. Comparison of number of search points / macroblock with existing fast search BMAs using Pedestrian video sequence

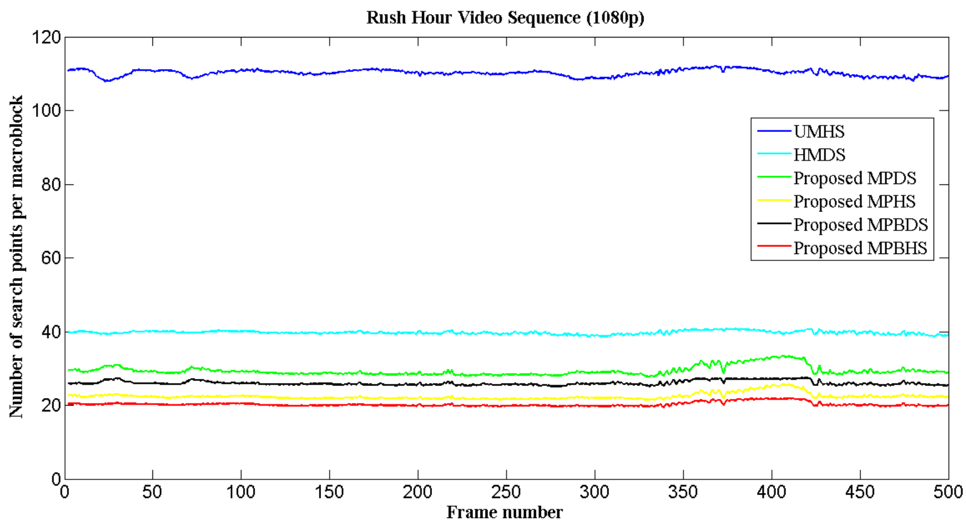

Fig.15. Comparison of number of search points / macroblock with existing fast search BMAs using Rush hour video sequence

For better quality of encoding MPDS and for fast execution MPBHS search patterns are preferable.

V. Conclusion

In this paper multipoint search patterns are suggested for high resolution video coding. Search shape are chosen such that all direction in search area are covered uniformly while progressing search and identification of MV at edge or corner of the window is faster compared to existing fast search BMAs. Variants are proposed such that initial candidate CMBs are at same location and from second iteration onwards search shape is repeated hence hardware implementation is uncomplicated. MPDS utilize maximum number of search points to offer best quality of encoding among MPBHS, MPBDS, MPHS and MPDS and also outperform compared to existing fast search BMAs like UMHS and HMDS for all measurable parameters. For HD and QFHD video sequences due to bigger size of frame, motion is in wide range. To tackle such MVs, multipoint search patterns do better than existing BMAs for reaching at far location in search window.

References

[1] H. Standard, G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding," vol. 22, no. 12, pp. 1649–1668, 2012.

[2] M. E. Rizkalla, P. Salama, M. El-sharkawy, and M. Sushmitha, "Hardware implementation of Block-based Motion Estimation for real time applications," *J. VLSI Signal Process.*, pp. 139–159, 2007.

[3] J. Kim and T. Choi, "A fast Three-step search algorithm with minimum checking points unimodal error surface assumption," *IEEE Trans. Consum. Electron.*, vol. 44, no. 3, pp. 638–648, 1998.

[4] W.-C. M. Lai-Man Po, "A novel Four-step search algorithm for fast block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–317, 1996.

[5] E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 4, pp. 419–422, 1996.

[6] A. A. K. Jo Yew Tham, Surendra Ranganath, Maitreya Ranganath, "A Novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 369–377, 1998.

[7] S. Zhu, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, 2000.

[8] C. Zhu, X. Lin, and L. Chau, "Hexagon-Based Search Pattern for Fast Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 349–355, 2002.

[9] C. Cheung and L. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1168–1177, Dec. 2002.

[10] H. Jia and L. Zhang, "Directional diamond search pattern for fast block motion estimation," *Electron. Lett.*, vol. 39, no. 22, pp. 2–3, 2003.

[11] C. Cheung and L. Po, "Novel Cross-Diamond-Hexagonal Search Algorithms for Fast Block Motion Estimation," *IEEE Trans. Circuits Syst.*, vol. 7, no. 1, pp. 16–22, 2005.

[12] T. Tsai and Y. Pan, "A Novel 3-D Predict Hexagon Search Algorithm for Fast Block Motion Estimation on H.264 Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1542–1549, 2006.

[13] X. Jing, L. Chau, and S. Member, "An Efficient Three-Step Search Algorithm for Block Motion Estimation," *IEEE Trans. Multimed.*, vol. 6, no. 3, pp. 435–438, 2004.

[14] Y. Yan and S. Meng, "A New Hybrid Search Scheme for Video Motion Estimation," *J. Converg. Inf. Technol.*, vol. 6, no. 3, pp. 106–112, 2011.

[15] S. Huang, C. Cho, and J. Wang, "Adaptive Fast Block-Matching Algorithm by Switching Search Patterns for Sequences With Wide-Range Motion Content," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1373–1384, 2005.

[16] K. Ng, L. Po, K. Wong, C. Ting, and K. Cheung, "A Search Patterns Switching Algorithm for Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 753–759, May 2009.

[17] Z. Chen, J. Xu, Y. He, and J. Zheng, "Fast integer-pel and fractional-pel motion estimation for H.264/AVC," *J. Vis. Commun. Image Represent.*, vol. 17, no. 2, pp. 264–290, Apr. 2006.

[18] H. J. Hsieh, C. C. Lin, and Y. Lin, "Multi-direction search algorithm for block motion estimation in H.264/AVC," *IET Image Process.*, vol. 3, no. 2, pp. 88–99, Apr. 2009.

[19] W. Hsu, T. Yu, and J. Guo, "Enhanced Block Motion Estimation Based on Threshold-Aware Two-Path Search Method," *J. Converg. Inf. Technol.*, vol. 5, no. 5, pp. 99–110, 2010.

[20] L. Po, K. Ng, K. Cheung, K. Wong, and C. Ting, "Novel Directional Gradient Descent Searches for Fast Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 8, pp. 1189–1195, 2009.

[21] A. Nayak, B. Biswal, and S. K. Sabut, "Evaluation and Comparison of Motion Estimation Algorithms for Video Compression," *Int. J. Image, Graph. Signal Process.*, vol. 4, no. 2, pp. 9–18, 2012.

[22] S. Acharjee and C. Sheli, "Fuzzy Logic Based Four Step Search Algorithm for Motion Vector Estimation .," *Int. J. Image, Graph. Signal Process.*, vol. 4, no. 4, pp. 49–55, 2012.

[23] M. Porto, A. Silva, S. Almeida, E. Costa, and S. Bampi, "Motion Estimation Architecture Using Efficient Adder-Compressors for HDTV Video Coding," *J. Integr. Circuits Syst.*, vol. 5, no. 1, pp. 78–88, 2010.

[24] T.-H. Tsai and Y.-N. Pan, "High Efficiency Architecture Design of Real-Time QFHD for H.264/AVC Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 11, pp. 1646–1658, 2011.

[25] J. Vanne, E. Aho, and K. Kuusilinna, "A Configurable Motion Estimation Architecture for Block-Matching Algorithms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 4, pp. 466–476, 2009.

[26] O. Ndili and T. Ogunfunmi, "Algorithm and Architecture Co-Design of Hardware-Oriented, Modified Diamond Search for Fast Motion Estimation in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1214–1227, 2011.

[27] "YUV video repository." [Online]. Available: http://media.xiph.org/video/derf/.

Authors' Profiles

Nehal Shah is a PhD student at the Sardar Vallabhbhai Patel National Institute of Technology (SVNIT), Surat. She received her postgraduate degree in Electronics Design and Technology from Indian Institute of Science (IISc) Bangalore and Electronics engineering from SVNIT, Surat. Her research interests include Video processing, ASIC Design, Microprocessors and Embedded Systems. She is working in Sarvajanik College of Engineering and Technology, Surat since 1998 and currently she is PG Incharge and Associate Professor in Electronics and Communication department. She has guided several PG and UG students for their project work and coordinated conference, STTPs, workshops and several technical programs for students, faculty members and working professional.

Dr Upena Dalal is currently Associate Professor and Head at Electronics and Communication Engineering department, at the Sardar Vallabhbhai Patel National Institute of Technology (SVNIT), Surat. She received her doctoral degree in era of wireless communication and post graduation in Electronics and Communication Systems in which she was gold medalist. She has vast teaching experience of 24 years and her major subjects are Cellular Technology, Wireless Communication and Fiber optic networks. She is author of well-known book on "Wireless Communication" by Oxford University press (2009) and “Selectively Imposed Pilot based Channel Estimation" by VDM publications, Germany (2010). She is also co-author of book on "WiMAX Developments" from Intechweb, Viena, Austria (2010). She has guided several PhD and MTech students and coordinated conference, STTPs, workshops and numerous technical programs for betterment of students, faculty and research fraternity.

How to cite this paper: Nehal N. Shah, Upena D. Dalal, Priyank H. Prajapati,"Multi Point Search Pattern for Fast Search Motion Estimation of High Resolution Video Coding", IJIGSP, vol.7, no.7, pp.60-68, 2015.DOI: 10.5815/ijigsp.2015.07.07