ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2019.0283

PERFORMANCE-BASED STUDY ON DIAMOND SEARCH AND ITS VARIATIONS FOR EFFICIENT MOTION ESTIMATION IN VIDEO

Hussain Ahmed Choudhury[1], Nidul Sinha[2], Tanvir H. Sardar[3] and Archana Wagh-Mare[4]

[1,3,4]Department of Computer Science and Engineering, Jain University, India
[2]Department of Electrical Engineering, National Institute of Technology, Silchar, India

Abstract

The motion estimation is a crucial process of video encoding process which needs to be dealt with high importance. There has been a lot of research works on the topic and various block-based motion estimation techniques were designed and implemented for the same. Out of the existing developed techniques, Diamond Search (DS) is one of the efficient motion estimation technique and most no. of new algorithms are designed based on this DS only. In this paper, we will be critically reviewing the Diamond Search (DS) and few existing variations of block matching algorithms based on DS which varies according to search criteria, search pattern, size of the search window, macro block size and PSNR values. The algorithms are being applied to different video sequences to do a comparative study among them and check out which variation of DS is efficient for which video sequences.

Keywords:

Block Matching, DS, MDSS, Motion Estimation, MV

1. INTRODUCTION

Video coding is very essential as the video is considered most important in every field of communication starting from communication to entertainment to online education, robotics etc. Coding a video will help it to transfer fast, securely and efficiently. With the invention of High Definition (HD) videos, the storage requirement becomes very high which is a great constraint. Different coding standards have reduced the bit required to send and encode videos and lowers bit rate also. Videos are a collection of still images where the consecutive frames have both temporal redundancy and spatial redundancy. The concept behind decreasing the bit requirement is to remove temporal and spatial redundancies. The important and crucial step to reduce temporal redundancy is known as motion estimation which is a search process by which best match in the reference frame is found. Since matching is done block wise, so it is called Block-Based Motion Estimation (BBME) Technique and algorithms used for BBME are called Block Matching Algorithm (BMA) [1] [9]. The first BMA used was Full Search (FS) [9] which was very accurate in finding Motion Vector (MV) [1] but it is highly computationally expensive. So the Fast Block Matching Algorithms (FBMA) were developed and Diamond Search (DS) [1] is one of them. DS was proposed by Zhu and MA [1] and one of the best FBMA amongst existing ones. Thereafter many types of research carried out and various improved versions of DS were proposed.

In this paper, we have discussed DS in details and all other FBMA which were either using the DS concept or modified version of it in details. We also compared their performances based on a number of computations, time and quality of the compensated image.

The remaining part of the paper is designed as follows. The section 2 describes the MV distribution and matching criteria used by existing BMAs. The DS and other variations of it are explained in section 3. The simulation results obtained by applying the algorithms on different video sequences are mentioned in section 4 followed by conclusion in section 5.

2. MV DISTRIBUTION AND MATCHING CRITERIA

The size of the search window and the size of that greatly affects the performance of the BMAs. The series of frames of a video has high similarity. When we search the motion vector in the reference frame, there is a huge probability that the minimum point or best MV corresponds to the location around the center. This is called center biased MV. We select the search window as rectangular one initially and it gradually changed. The search window has many search locations and only FS searches each and every location but FBMA searches a few random locations. As the search progresses, depending on the motion type, the search window size can be adjusted. Diamond Search (DS) is one of fast motion estimation Technique but it is good for large motion only. The unrestricted search pattern became a drawback for DS and many BMA were developed on this fact. The algorithms which use search window of size 3×3 are good for small motion and a window size of 5×5 are good for fast motion. The matching between the blocks in consecutive frames must be done using some criteria. The common search criteria used in motion estimation of video encoding are Sum of Absolute Difference (SAD) [9], Mean Absolute Difference (MAD) [9], Mean Square Error (MSE) [1] and to check the quality of compensated image we use Peak Signal to Noise Ratio (PSNR). While implementing the below algorithms we have used MAD as matching criteria and PSNR ([1]-[10], [28], [30]-[32]) to check the quality.

3. DIAMOND SEARCH AND ITS VARIATION.

With the advancement of technologies, many FBMA were proposed. Among them the Diamond Search (DS) ([1]-[10], [28], [30]-[32]) was one of the efficient search techniques which were introduced in the year 2000. After that various research works were carried out on DS and as a result, DS became the base of most numbers of BMA in recent time. So we will discuss DS along with the versions of it further.

3.1 DIAMOND SEARCH (DS)

This algorithm is introduced by Zhu and MA [1] gives better performance than many other existing fast BMA. Based on the pattern of search, this algorithm is named as DS [14] [15] [20]-[23]. The algorithm has the following steps:

- We start searching at 8 locations around the center in the form of Large Diamond with step size 2 and including center, we calculate at 9 locations for minimum BDM. If the center has the minimum BDM, jump to step 3 else follow 2.
- We make the minimum BDM point of 1st step as a center and search in LDSP pattern again. If the minimum lies at the center, follow step 3 else repeat step 2.
- Change the search pattern from LDSP to Small Diamond Shape Pattern (SDSP) and the location corresponding to minimum BDM in this step will give us the best MV.

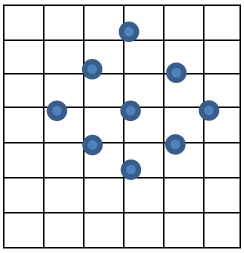

Fig.1(a). LDSP pattern (1st step) of DS

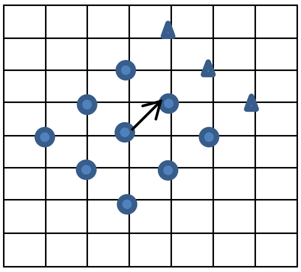

Fig.1(b). Shift center and new LDSP (2nd step)

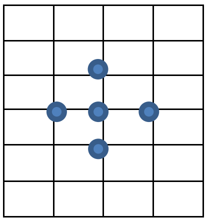

Fig.2. 3rd Step of DS (LDSP to SDSP change)

3.2 CROSS DIAMOND SEARCH (CDS)

Cheung and Po [2] introduced the new algorithm named as CDS [2] - [5], [8], [32] based on existing Diamond Search (DS). It chooses 9 points including the center in the direction along plus (+) with a window size 5×5. This algorithm has 2 stopping criteria which help to terminate early if some conditions get satisfied. The algorithm has the following steps:

- The search starts at the locations (+1,0), (0,+1), (+2,0) & (0,+2) with (0,0). If the minimum cost function found at the center, the search stops else go to 2nd step.
- Two new points around the minimum point out of four points i.e. (+1,0) and (0,+1) are checked whichever is nearer. If the minimum cost location matches with that of the previous step, search stop else follows step 3.
- Shift the center to the new minimum and form LDSP with it. If the minimum is at the center, follow step 4 else repeat the current step.
- We form SDSP with minimum cost point as the center and check. If the minimum cost point at this step will give us the best MV.

3.3 SMALL CROSS DIAMOND SEARCH(SCDS)

Cheung and Po [3] have introduced this SCDS [3] [4] [8] [32] algorithm too. This algorithm starts with 5 initial search points along a horizontal and vertical direction forming Small Cross Search Pattern (SCSP). This algorithm is also based on DS but modified version of CDS. The algorithm has the following steps:

- The search starts with an initial 5 points across “+” known as SCSP. If the best match found to be at center, the search stops else move to next step.
- The new 4 points are searched at (+2,0) and (0,+2) around the center and shift the center to minimum cost position.
- Two new points nearer to the minimum position out of four points (+1,+1) are checked. If the point of a minimum of the 1st step and this step coincides at any of (+1,0) and (0,+1), then search stops and corresponding MV will be final else follow the next step.
- The minimum cost location of the previous step will be the center of new LDSP. If the center is the point of minimum, then go to step 5 else repeat.
- The minimum point of the 4th step will be the center of new SDSP and the new minimum found in this step will be the point of the best match and best MV.

3.4 NEW CROSS DIAMOND SEARCH (NCDS)

Based on the concept of DS and its variations CDS and SCDS [5] [8] [32] Lam et al. [5] proposed it. The algorithm has the procedure as mentioned below.

- The search starts with initial 5 points across “+” known as (+1,0) and (0,+1) with center (0,0). If the minimum block distortion measure (BDM) is found to be at the center, the search stops else move to next step.
- With the minimum BDM point, we search in the same pattern. If the new minimum is at the center, the search stops else follow the next step.
- The remaining 3 points of (+2,0) and (0,+2) are checked where calculations are not done. Then go to step 4.
- The LDSP is followed by taking the center as minimum BDM point. If the minimum is at the center, go to step 5 else repeat.
- Follow the SDSP with minimum BDM as the center. The location with the minimum cost function value will give us the final motion vector.

3.5 NOVEL REDUCED DIAMOND SEARCH (RDS)

As the name suggests, RDS [27] is the reduced version of DS in terms of checking points. It also uses the concepts of the threshold value of PSNR for early termination. The algorithm has the following steps:

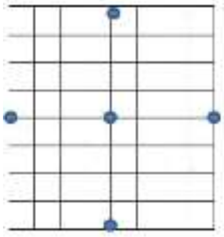

Fig.3(a). first pattern

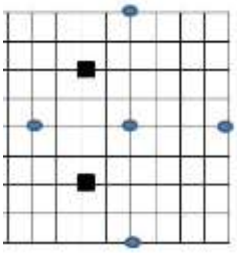

Fig.3(b). half LDSP

- The search starts with 5 points around the center in "+" pattern shown in Fig.4. If the center is the point of minimum, jump to step 4 else move to next step.
- Chose extra two points around the point of minimum to form half LDSP as shown in Fig.5 and check.
- The minimum point found in the 2nd step is chosen as the center of newly formed LDSP. If the minimum lies at the center, go to step 4 or else repeat. Depending on the new minimum if at the corner or at the edge, search points are shown in Fig.6 and Fig.7.
- Check the points of SDSP and check 5 points. The minimum position will give us the final MV.

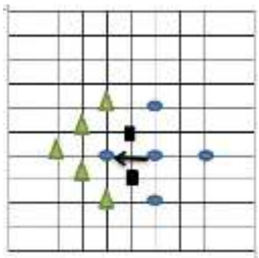

Fig.4(a). Minimum is at Corner

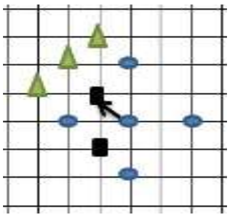

Fig.4(b). Minimum is at edge

3.6 ORTHOGONAL DIAMOND SEARCH (ODS)

This algorithm ODS [35] [36] combines the features of orthogonal search and small diamond shape search. This algorithm is efficient to detect small motion.

- The search starts with 3 horizontal search positions centered at origin i.e. in orthogonal direction shown in Fig.8. If the minimum lies at center, follow 2nd step.
- Else, taking the center, form SDSP with one extra point in horizontal direction and follow step 2.
- With the point of minimum as new center, search at vertical direction 2 points at vertical direction with step size 2 and proceed to next step.
- Now with the step size 1, we check two horizontal positions around minimum and go to step 4.
- Now again with same step size two vertical points are checked and the resultant point of minimum will give the best MV.

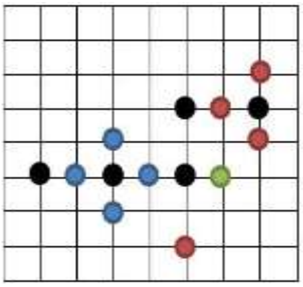

Fig.5. ODS pattern

● Orthogonal positions

🔵 SDSP

🟢 Extra points

🔴 vertical search locations

3.7 MODIFIED DIAMOND SEARCH PATTERN (MDSP)

Yoon et al. [12] and [13] have introduced this algorithm which is another form of modification of DS. Here, the algorithm uses a little different search pattern and adaptively. The flow chart of the algorithm is shown in Fig.6 and algorithm run as follows:

- If the MV of the current block in the reference frame is (0,0), then start with SDSP pattern and check and if not follow step 2.
    a) If the minimum cost lies at the center of SDSP, then the minimum position will give the best MV.
    b) If the minimum is not at SDSP, repeat this step.
- If not (0,0), then
    a) We will shift origin to that point and start our search taking it as the center.
    b) We start our search at 13 points in a pattern known as MDSP and is shown in Fig.7. If the minimum cost lies at the center of MDSP or any point of SDSP, the search stops and results in best MV else follow (c)
    c) We will shift the center to the minimum and MDSP is followed until we get the minimum cost at the center or any point of SDSP.

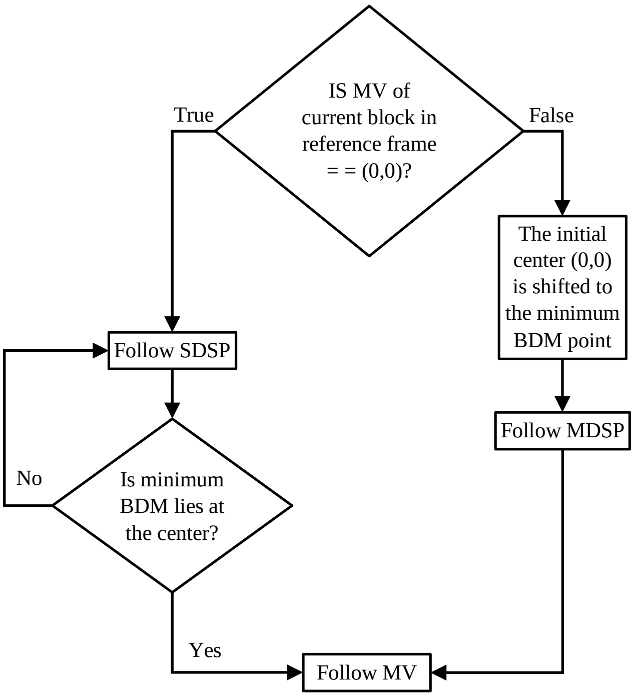
IS MV of current block in reference frame = = (0,0)?
True
False
The initial center (0,0) is shifted to the minimum BDM point
Follow SDSP
Follow MDSP
No
Is minimum BDM lies at the center?
Yes
Follow MV

Fig.6. MDSP algorithm flow chart

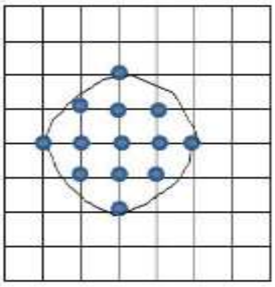

Fig.7. Modified Diamond Search Pattern (MDSP)

3.8 MODIFIED DIAMOND –SQUARE SEARCH (MDSS)

The algorithm MDSS [30] was introduced by Devi et al. [30] changes the shape of MDSP to Modified large search pattern (MLSP) as shown in Fig.8(a) and SDSP to Modified Small Square pattern (MSSP) as shown in Fig.8(b) which is a square shape. By doing so, it also reduces the number of search points and computational complexity.

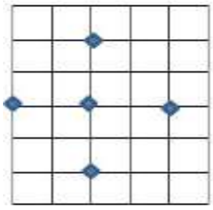

Fig.8(a). MLSP (LDSP with 5 points)

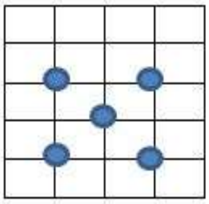

Fig.8(b). MSSP

The algorithm has the steps:

- Initially, the search is done at the MLSP along with center (0,0) position. If minimum MBD lies at the center, go to 3rd step else follow the next step.
- Shift the center to new minimum cost location and search at MLSP. If the minimum is at the center, go to the next step or repeat this step.
- We follow MSSP and the new minimum cost location found at this step will give the best MV.

The total process of MDSS is shown in Fig.8(c).

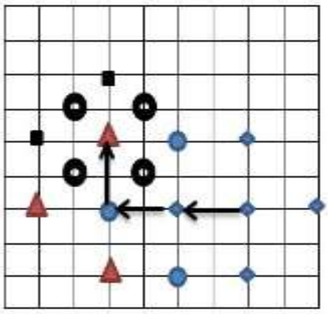

Fig.8(c). MDSS search strategy with 17 Search points

3.9 DIAMOND TO ARC STRATEGY FOR MOTION ESTIMATION (DTA)

The Diamond to Arc Strategy [19] [24] is very much useful for detecting large MV because of its sense of direction and pattern of search. The algorithm has the following steps:

- Initial search point starts with LDSP but with lesser points i.e. only 5 points.
- If the minimum MBD is at any point except center, the center is shifted to the new minimum and two points are selected for checking so that its center and two points form Arc like structure.
- We check the nearest point to the point of minimum on the horizontal line joining new minimum and original center which is called the look-back step. If the new point is the MBD, we check the points around SDSP to get best MV else the arc pattern is followed repeatedly and 4 points are checked all the time. If the new MBD lies at the earlier MBD point, SDSP is checked for finding best MV. In another case, if the MBD is one of the new arcs and this step 3 has to be repeated to find the best MV. The DTA search pattern is shown in Fig.9(b).

3.10 DIAMOND-ARC-HEXAGON ALGORITHM

This DAHS [19] [24] algorithm is introduced by Chiang et al. [24] which combines the concept of DTA and New Predictable Hexagon Search (NEW-PHS). The algorithm is as follows:

- The first step searches in both LDSP and SDSP but in LDSP, it searches only in 4 points so it is known as Double DS pattern (DDSP) is shown in Fig.14. If the minimum cost point is at the center, stop and result is final MV else go to step 2.
- If the minimum MBD lies at the surface points of SDSP, NEW PHS algorithms will be used. This algorithm searches remaining positions on the vertical and horizontal directions until we find the MBD at either one of the centers of a hexagon or till search reaches the boundary of the search window. Else we move to the next step.
- The step searches for MBD and uses DTA to find the new and final MV.
- The total process of DAHS is shown in Fig.9(c) and Fig.9(d).

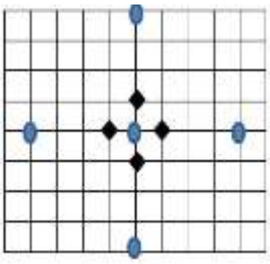

Fig.9(a). Double Diamond Search Pattern (DDSP) used in DAHS

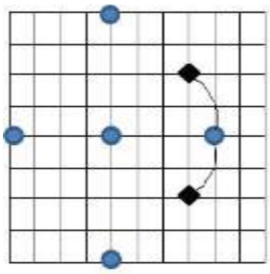

Fig.9(b). Diamond-Arc pattern

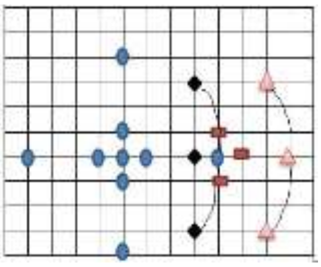

Fig.9(c). DAHS search with 18 search points to get best MV

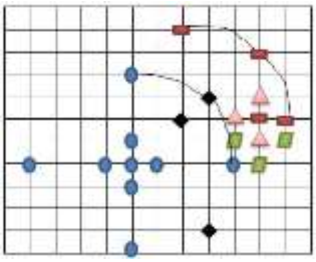

Fig.9(d). DAHS search with 22 search points to get best MV

3.11 MODIFIED SMALL-CROSS DIAMOND SEARCH (MSCDS)

This algorithm MSCDS [32] was proposed by Singh and Ahamed [32] uses both small search pattern and large search patterns to enhance the quality of the search algorithm and it is also a modified version of DS. The algorithm follows the following steps:

- The search starts with initial SDSP and if MBD is at the center (0,0), the search stops as shown in Fig.10(a). Else go to step 2.
- Center is shifted to minimum MBD and checking is done at the small cross pattern. Here we need to compute only at (1,1), (1,0) and (1,-1) points. If MBD lies at any of (+1,0) or (0,+1) as shown in Fig.10(b) then we stop and take the best MV from this step or else move to next step.
- If the MBD lies except the center position, we make the minimum location as our new center of LDSP search. We repeat this step till the boundary position is not reached or minimum lies at the center of LDSP and we further go to the next step to add some improvements.
- We follow SDSP with minimum MBD as a new center. The minimum found in this will give us Best MV. The whole process of MSCDS is shown in Fig.10(c).

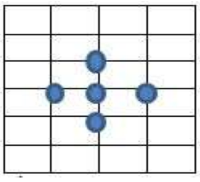

Fig.10(a). 1st step

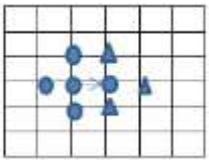

Fig.10(b). 2nd step of MSCDS

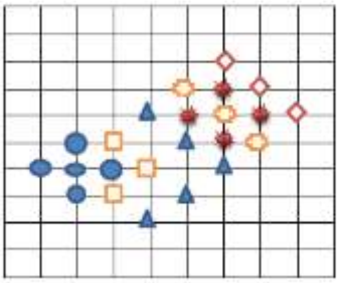

Fig.10(c). ● - 1st step □ - 2nd step ▲ - 3rd step ⊞ - 4th step ◆ - 5th step ✱ - 6th Step

3.12 T SHAPE DIAMOND SEARCH(TDS)

Another variation of Diamond Search named as T shape Diamond search based on the pattern of motion vector distribution is introduced. It has been observed in many cases that the probability of finding MV around the center and the cross positions around the center is maximum and this algorithm considers the three directions around the center for covering these positions which forms a T shape i.e. top, down, left and right positions.

The algorithm progresses as mentioned below:

- TDS algorithm search starts with the original position (0,0) and the points included in the TDS pattern is tested as shown in Fig.11 for best match and if found at the center, go to step b or else follow step c.
- Choosing the movement direction: A new point at a distance of 2 in the same direction of previously found point ‘2’ least cost point is added and calculation is done. If the new MBD point ‘2’ is not matching with the old MBD point i.e. ‘1’, we need to carry out this step again and again until we match the previous step MBD. Else two extra points with step size 2 i.e point ‘3’, ‘4’ will be added as shown in Fig.11, if the new MBD is same as previous MBD, follow step c. If not, another point with the same direction of newly found MBD is considered.
- If the distortion error corresponding to the location of minimum MSE is less than the threshold, then this location will be the best position. Else follow step d.
- Final step:
  a) We shift the center to the new minimum and 4 locations denoted by ‘4’ are included and calculated for MBD. If the minimum MBD is at the center, we stop here and if not follow ii.
  b) Any one of the points denoted by ‘5’ through ‘8’ will be considered and evaluated. Now if the sum of MSE values at point ‘1’ and ‘2’ denoted by MSE12 is less than ‘1’+’3’ denoted by MSE ‘13’, then ‘5’ location is added else ‘6’ will be evaluated. This selected point will be the best MV location.

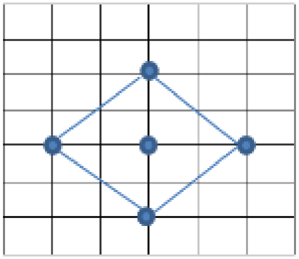

Fig.11. 1st step of TDS

3.13 KITE CROSS DIAMOND SEARCH (KCDS)

The KCDS [8] name itself derived from three words-Kite, cross and diamond as it follows all three shaped search pattern. The line that connects the corners of a kite is the line of symmetry.

The algorithm follows as mentioned below:

- We follow a “+” (cross) pattern and check 5 points at a distance of 1 around the center. If the minimum lies at the center, we stop and else follow the next step. The step is shown in Fig.24.
- With the minimum BDM as the center, a kite search pattern (KSP) is used as per MV of step 1. E.g. if the top vertical point has the MBD, we follow up-kite shaped KSP. There are 4 cases like up-kite as Fig.12(a), down-kite as Fig.12(b), right-kite as Fig.12(c) and left-kite as Fig.12(d) depending on the direction and position of minimum BDM of the last step. If we find the minimum at the center of KSP, we will stop the search else to proceed to the next step. This step is shown in Fig.12(e).
- *Diamond pattern* - We will make the minimum BDM point of last step center of new LDSP. If the minimum BDM is found to be at the center, move to next step else continue this step.
- *Converging* - Follow SDSP by taking minimum BDM of the last step as a center and the new minimum which will be found, will give the best MV.

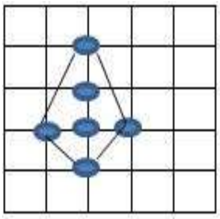

Fig.12(a). Up-kite shape pattern

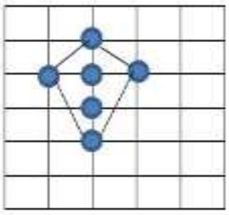

Fig.12(b). Down-kite shape pattern

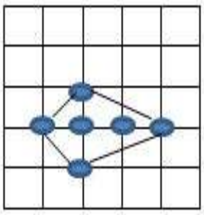

Fig.12(c). Right kite shape pattern

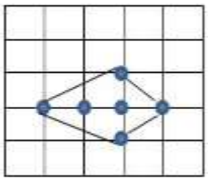

Fig.12(d). Left kite shape pattern

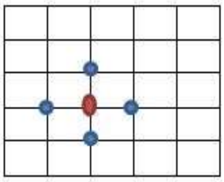

Fig.12(e). 1st step of KCDS with center (0,0) i.e.●

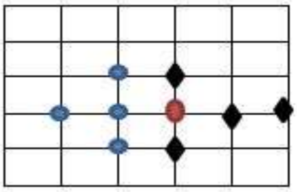

Fig.12(f). 2nd step with right-KSP and center (1,0) i.e. stopping criteria

3.14 ENHANCED CROSS DIAMOND SEARCH (ECDS)

Enhanced Cross Diamond Search (ECDS) proceeds as mentioned below:

- The first step is as the name suggests i.e. cross diamond search pattern and searching is done in 5 points. If the minimum cost lies at the center, we will stop else will proceed to step B.
- If the minimum cost lies at (1,0) (or (-1,0), (0,1), (0,-1)), we will consider four other additional points i.e. (2,0), (3,0), (1,-1) and (1,1). If we minimum costs lie at the same point as the previous one, we consider this new minimum point location as best location and search will stop here. Else we will start following step C.
- There are following cases which can exist:

Case i: If minimum cost lies at (2,0) (or (-2,0), (0,2), (0,-2)), we will check (2,-1) and (2,1). If we find the minimum at (2,0), we stop. Else proceed to step D.

Case ii: If the location (1,-1) (or (-1,-1), (-1,1), (1,1)) has the least cost in last step, two extra locations with (1,-2), (2,-1) are need to be checked. If the same point has the least cost, we won't proceed and the (1,-1) will be the best matched location. Else go to step D.

Case iii (a): If the least point is (3,0) (or(-3,0),(0,3), (0,-3)), search points (0,-3), (-3,0), (0,3) are checked for best match. If minimum cost lies at (3,0) again we move to step D.

Case iii (b): If the least cost lies at (0,-3) (or (-3,0), (0,3)), we will check (0,-2) additionally. If we find the best match at the same (0,-3), go to step d. If the least cost is at (0,-2), we will check (-1,-2) and (1,-2) in addition to that. If the minimum BDM lies at (0,-2) is still at (0,-2), we will stop. Else we will follow step D.

- The minimum point found in earlier steps will be made as a center of LDSP. If the best match is found to be in the center, go to step E. Else step D will be performed repeatedly.
- The least cost location of the previous step will be taken as the center of SDSP.

3.15 STAR DIAMOND-DIAMOND SEARCH(SDDS)

Satish et al. [38] had proposed the most recent BMA based on the DS which is known as Star Diamond-Diamond Search(SDDS). The algorithm has the following steps:

In the star diamond pattern, 9 points including center are searched for best match i.e. SAD_{min} and go to step 2.

The search pattern changes from star to SDSP and continues until SAD_{min} is located at the center. To detect the direction of MV if either stationary or vertical or horizontal or diagonal, star pattern runs only once.

The center is always shifted to the point or location with SAD_{min} and steps continue till the center will be minimum.

3.16 SIMULATION RESULTS AND ANALYSIS

All the above-discussed algorithms vary on the type of search pattern or a number of search points required to find the best MV and all BMA use the concept of DS. The various types of search patterns used by all BMAs are small diamond search pattern (SDSP), Large Diamond Search Pattern (LDSP), Cross search Pattern (CSP), Small CSP (SCSP), Large CSP (LCSP), Kite shaped pattern (KSP), Diamond to Arc pattern (DTA), Double Diamond Search Pattern (DDSP) and Modified Diamond Search Pattern with 13 points (MDSP). All BMAs are different based on complexity i.e. no. of search points required to find Best MV, the quality of the compensated image. All algorithms use the same search criteria and few BMAs are efficient for large motion videos whereas few BMAs are efficient for slow and fast motion. To see and compare the performances of different versions of DS, we implemented all BMA using MATLAB on different video sequences both slow and fast. The various video sequences are in the form of CIF, QCIF, SIF, mp4 etc. The size of the macro blocks is of size 16×16 taken. The search parameter is taken as +7 pixels, the search area is 15×15. For different BMA, search window varies from 5×5 to 5×3, the reference frame is taken as the past frame and frame gap between two is taken as 1. The results obtained from various experiments based on the above criteria are listed in Table.1. The efficiency of algorithms is based on the no. of search points required to search to find the best match. The algorithm with fewer search points is regarded as more efficient. The higher the value of Peak Signal to Noise Ratio (PSNR), the better the algorithm it is. The simulation results in respect of a number of search points required per MB and the corresponding PSNR values are listed below in Table.2-Table.4.

Table.1. Video sequence details

Video formats	Average frame speed	Frame difference
CIF, QCIF, mp4, SIF, etc.	15fps-30fps	1 (immediate past frame is used for reference)

Table.2. Average no. of search points per MB required by all BMA's w.r.t. different video sequences

Videos	DS	CDS	SCDS	NCDS	MSCDS	MDSP
Tennis	16.351	15.46	13.8	13.28	12.44	8.33
Caltrain	16.45	13.86	13.63	13.21	11.53	8.78
Claire	12.45	8.91	5.41	5.18	6.78	5.3
Flower	18.31	18.24	17.118	16.52	12.75	17.39
Football	13.71	10.92	8.29	7.91	7.67	9.31
Ms America	12.43	11.68	6.45	5.78	4.60	9.32
Stefan	16.25	12.67	13.33	13.168	12.13	8.96
Foreman	16.81	15.95	14.67	14.61	12.66	7.21

4. CONCLUSION

We have experimented with almost all variations of DS in our review paper and the following observations can be concluded as follows:

The CDS has almost 40% more speed than DS. It searches 9 locations at the end of the 1st step stop and 11 Locations at the end of the 2nd step stop before finding the best match. SCDS achieves a speed up the ratio of 146% for Akiyo video sequence as reported by Cheung and Po [1]. NCDS achieves a speed up ratio even to 58% for Claire video as per the report of Lam et al. [5]. It is more efficient to detect small motion than the other two as it may repeat the small cross search pattern of 5 points if required. In MDSP the pattern of search is selected based on coordinates of candidate block in the reference frame and adaptively. It is more efficient than both in case of search points and quality of the compensated image. KCDS uses three search patterns namely Kite, cross and Diamond which makes it easier to detect small motion. In various results and analysis, it is found that KCDS searches 58% fewer points as compared to DS and 41% less as compared to CDS. It has the almost the same quality of the compensated image. From the prospect of a number of search points, we can conclude that KCDS <NCDS <SCDS <CDS <DS i.e. KCDS has the least computational complexity among the above NCDS, SCDS, CDS and DS. Horizontal DS [18] focuses on the horizontal and translational movement of the object. Its result is almost similar to CDS. The DAHS combines the diamond and hexagon search pattern with Arc pattern. The diamond to arc strategy is deployed for large motion on the initial DDSP. The DAHS has smaller MSE and solves the problem of 3SS (Three step search). The RDS reduces search points by 20% - 60% as compared to other BMA and depending on the motion type. Upon applying a threshold, the results showed an improvement of 10% - 30% search points. RDS improves search points by 28% as compared to DS. Hexagonal DS [28] uses two patterns and has the improvement of up to 56% in case of computations without compromising the image quality. HDS achieves 20.8% speedup ratio for fast video sequences than that of DS.

Table.3. Average PSNR (dB) for selected BMA over few video sequences

BMA Videos	DS	CDS	SCDS	NCDS	MSCDS	MDSP
Tennis	24.37	24.54	24.435	24.541	24.543	24.512
Caltrain	34.37	34.452	34.543	34.435	34.442	34.453
Claire	34.88	34.78	34.82	34.39	34.42	34.993
Flower	23.33	23.14	22.89	22.65	22.711	22.675
Football	19.89	19.84	19.78	19.76	19.69	19.681
Ms America	39.38	39.33	39.32	39.32	39.241	39.342
Stefan	22.78	22.69	22.45	22.66	22.52	23.16
Foreman	28.67	28.56	28.43	28.12	27.962	29.61

Table.4. Average PSNR (dB) for selected BMA over few video sequences

BMA Videos	KCDS	DAHS	RDS	HDS	ODS
Tennis	24.542	25.06	24.332	24.37	24.37
Caltrain	34.253	35.658	34.37	34.37	34.37
Claire	34.783	36.333	34.88	34.88	34.88
Flower	22.65	24.032	23.33	23.33	23.33
Football	19.871	25.16	19.89	19.89	19.89
Ms America	39.293	39.978	39.38	39.38	39.38
Stefan	23.23	24.305	22.78	22.78	22.78
Foreman	29.78	30.225	28.67	28.67	28.67

REFERENCES

[1] S. Zhu and K.K. Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", *IEEE Transactions on Image Processing*, Vol. 9, No. 2, pp. 287-290, 2000.

[2] C.H. Cheung and L.M. Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 12, pp. 1168-1177, 2002.

[3] C.H. Cheung and L.M. Po, "A Novel Small-Cross-Diamond Search Algorithm for Fast Video Coding and Video Conferencing Applications", *Proceedings of International Conference on Image Processing*, pp. 1-5, 2002.

[4] H. Jia and L. Zhang, "A New Cross Diamond Search Algorithm for Block Motion Estimation", *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 231-237, 2004.

[5] C.W. Lam, L.M. Po and C.H. Cheung, "A New Cross-Diamond Search Algorithm for Fast Block Matching Motion Estimation", *Proceedings of International Conference on Neural Networks and Signal Processing*, pp. 1262-1265, 2003.

[6] W.I. Choi, B. Jeon and J. Jeong, "Fast Motion Estimation with the Modified Diamond Search for Variable Motion Block Sizes", *Proceedings of International Conference on Image Processing*, pp. 364-371, 2003.

[7] H. Jia and L. Zhang, "Directional Diamond Search Pattern for Fast Block Motion Estimation", *Electronics Letters*, Vol. 39, No. 22, pp. 1581-1583, 2003.

[8] C.W. Lam, L.M. Po and C.H. Cheung, "A Novel Kite-Cross-Diamond Search Algorithm for Fast Block Matching Motion Estimation", *Proceedings of International Conference on Circuits and Systems*, pp. 723-729, 2004.

[9] A. Barjatya, "Block Matching Algorithms for Motion Estimation", *IEEE Transactions Evolution Computation*, Vol. 8, No. 3, pp. 225-239, 2004.

[10] G. Ding and Q. Dai, "A Line-Diamond Parallel Search Algorithm for Block Motion Estimation. In Image and Graphics", *Proceedings of 3rd International Conference on Image and Graphics*, pp. 320-323, 2004.

[11] L.M. Po and C.H. Cheung, "Novel Cross-Diamond-Hexagonal Search Algorithms for Fast Block Motion Estimation", *IEEE Transactions on Multimedia*, Vol. 7, No. 1, pp. 16-22, 2005.

[12] H.S. Yoon, G.S. Lee, S.H Kim and J.Y. Chang, "Motion Estimation based on Temporal Correlations", *Proceedings of IEEE International Conference on Information and Communication Technology*, pp. 75-83, 2002.

[13] X. Yi and N. Ling, "Rapid Block-Matching Motion Estimation using Modified Diamond Search Algorithm", *Proceedings of IEEE International Conference on Circuits and Systems*, pp. 5489-5492, 2005.

[14] H. So, J. Kim, W.K. Cho and Y.S. Kim, "Fast Motion Estimation using Modified Diamond Search Patterns", *Electronics Letters*, Vol. 41, No. 2, pp. 62-63, 2005.

[15] G. Ding, Q. Dai and W. Xu, "Fast Motion Estimation Algorithm based on Predictive Line Diamond Search Technology", *Proceedings of IEEE International Conference on Circuits and Systems*, pp. 523-527, 2005.

[16] H. Liu, Y. Lie and C. Xie, "Fast Block-Matching Motion Estimation based on an Improved Cross-Diamond Search Algorithm", *Proceedings of IEEE International Conference on Circuits and Systems*, pp. 1-4, 2005.

[17] A. Moradi, R. Dianat, S. Kasaei and M.T.M. Shalmani, "Enhanced Cross-Diamond-Hexagonal Search Algorithms for fast Block Motion Estimation", *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 558-563, 2005.

[18] A. Samet, N. Souissi, W. Zouch, M.A. Ben Ayed and N. Masmoudi, "New Horizontal Diamond Search Motion Estimation Algorithm for H. 264/AVC", *Proceedings of IEEE International Conference on Communication, Control and Signal Processing*, pp. 13-15, 2006.

[19] H.T. Lin and J.S. Chiang, "A New Diamond-Arc-Hexagon Search Algorithm for Fast Block Motion Estimation", *Proceedings of IEEE International Conference on Communication, Control and Signal Processing*, pp. 811-816, 2006.

[20] H.S. Lee, J.H. Jung and D.J. Park, "An Efficient Diamond Search with Large Kite Search Patterns for Fast Block Motion Estimation", *Proceedings of IEEE International Conference on International Conference on Applied Science and Engineering*, pp. 3137-3141, 2006.

[21] Y.W. Huang, C.Y. Chen, C.H. Tsai, C.F. Shen and L.G. Chen, "Survey on Block Matching Motion Estimation Algorithms and Architectures with New Results", Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology, Vol. 42, No. 3, pp. 297-320, 2006.

[22] G. Jeon, J. Kim and J. Jeong, "Enhanced Cross-Diamond Search Algorithm for Fast Block Motion Estimation", *Proceedings of IEEE International Conference on Image Analysis and Recognition*, pp. 481-490, 2007.

[23] C.J. Duanmu, X. Chen, Y. Zhang and S. Zhou, "Mixed Diamond, Hexagon, and Cross Search Fast Motion Estimation Algorithm for H. 264", *Proceedings of IEEE International Conference on Multimedia and Expo*, pp. 761-764, 2008.

[24] J.S. Chiang, H.T. Lin and C.H. Hsia, "Novel Fast Block Motion Estimation using Diamond-Arc-Hexagon Search Patterns", *Journal of the Chinese Institute of Engineers*, Vol. 31, No. 6, pp. 955-966, 2008.

[25] Y. Ismail, J. McNeelly, M. Shaaban and M.A. Bayoumi, "Enhanced Efficient Diamond Search Algorithm for Fast Block Motion Estimation", *Proceedings of IEEE International Conference on Circuits and Systems*, pp. 3198-3201, 2009.

[26] X. Wang, W. Wan, J. Zhang and Y. Ma, "Research on the Motion Estimation with a Novel Octagon Cross Diamond Search Algorithm", *Proceedings of IEEE International Conference on Microelectronics and Electronics*, pp. 89-92, 2010.

[27] I.A. Hashad, A.R. Sadek and S.K. Mandour, "A Novel Reduced Diamond Search (RDS) Algorithm with Early Termination for Fast Motion Estimation", *International Journal of Video and Image Processing and Network Security*, Vol. 10, No. 4, pp. 32-39, 2010.

[28] R.A. Manap, S.S.S. Ranjit, A.A. Basari and B.H. Ahmad, "Performance Analysis of Hexagon-Diamond Search Algorithm for Motion Estimation", *Proceedings of IEEE International Conference on Computer Engineering and Technology*, pp. 152-155, 2010.

[29] T. Swati and M.V.R. Vittal, "Estimation of Motion Vector Parameters using Modified Diamond Search", *International Journal of Computer Applications*, Vol. 15, No. 8, pp. 38-41, 2011.

[30] A.A. Devi, M.R. Sumalatha, N.M. Priya, B. Sukruthi and M. Minisha, "Modified Diamond-Square Search Technique for Efficient Motion Estimation", *Proceedings of IEEE International Conference on Recent Trends in Information Technology*, pp. 1149-1153, 2011.

[31] S. Gupta and S.M. Arora, "Motion Estimation Method between Diamond Search and Three Step Search", *Proceedings of National Conference on Web and Knowledge-Based Systems on Emerging Trends and Perspectives*, pp. 23-33, 2011.

[32] K. Singh and S.R. Ahamed, "Modified Small-Cross Diamond Search Motion Estimation Algorithm for H. 264/AVC", *Proceedings of IEEE International Conference on Recent Trends in Information Technology*, pp. 1-5, 2013.

[33] J. Luo and J. Peng,"An Unsymmetrical Diamond Search Algorithm for H. 264/AVC Motion Estimation",

Proceedings of IEEE International Conference on Chinese Conference on Image and Graphics Technologies, pp. 54-65, 2013.

[34] S. Banchhor and D. Shukla, "An Improved Diamond Search Pattern for Motion Estimation", *i-manager's Journal on Pattern Recognition*, Vol. 3, No. 3, pp. 1-19, 2016.

[35] N.A. Hamid, A.M. Darsono, N.A. Manap, R.A. Manap and H.A. Sulaiman, "A New Orthogonal-Diamond Search Algorithm for Motion Estimation", *Proceedings of IEEE International Conference on Computer, Communications, and Control Technology*, pp. 1-6, 2014.

[36] N.A. Hamid, A.M. Darsono, N.A. Manap, R.A. Manap and H.A. Sulaiman, "Performance Evaluation of Orthogonal-Diamond Search of Block Matching Algorithm for Video Coding", *Jurnal Teknologi*, Vol. 76, No. 1, pp. 61-66, 2015.

[37] S.K. Sahu and D. Shukla, "A New Approach of Block Matching Motion Estimation Algorithm for H. 264/AVC Video Codec", *i-manager's Journal on Pattern Recognition*, Vol. 4, No. 2, pp. 1-10, 2017.