Object Detection and Tracking using Modified Diamond Search Block Matching Motion Estimation Algorithm

Apurva S. Samdurkar[1], Shailesh D. Kamble[2], Nileshsingh V. Thakur[3], Akshay S. Patharkar[4]

[1] PG Scholar, Computer Science & Engineering, Yeshwantrao Chavan College of Engineering (India)
[2] Computer Science & Engineering, Yeshwantrao Chavan College of Engineering (India)
[3] Computer Science & Engineering, Prof Ram Meghe College of Engineering & Management (India)
[4] Computer Technology, K.D.K. College of Engineering (India)

UNIR
LA UNIVERSIDAD
EN INTERNET

Abstract

Object tracking is one of the main fields within computer vision. Amongst various methods/ approaches for object detection and tracking, the background subtraction approach makes the detection of object easier. To the detected object, apply the proposed block matching algorithm for generating the motion vectors. The existing diamond search (DS) and cross diamond search algorithms (CDS) are studied and experiments are carried out on various standard video data sets and user defined data sets. Based on the study and analysis of these two existing algorithms a modified diamond search pattern (MDS) algorithm is proposed using small diamond shape search pattern in initial step and large diamond shape (LDS) in further steps for motion estimation. The initial search pattern consists of five points in small diamond shape pattern and gradually grows into a large diamond shape pattern, based on the point with minimum cost function. The algorithm ends with the small shape pattern at last. The proposed MDS algorithm finds the smaller motion vectors and fewer searching points than the existing DS and CDS algorithms. Further, object detection is carried out by using background subtraction approach and finally, MDS motion estimation algorithm is used for tracking the object in color video sequences. The experiments are carried out by using different video data sets containing a single object. The results are evaluated and compared by using the evaluation parameters like average searching points per frame and average computational time per frame. The experimental results show that the MDS performs better than DS and CDS on average search point and average computation time.

Keywords

I. Introduction

From the viewpoint of tracking in videos, the use of block matching algorithm can be made for the purpose of motion estimation and object tracking. Detection of an object is typically the first step when starting the tracking process. The object detection mechanism is carried out either in the sequence of frames or when the object first appears in the frame of the video for tracking [1, 2]. Background subtraction is a process of subtracting the second frame (current frame) from the first background (reference) frame, thus the dissimilarity between two frames and the position of moving object can be obtained [3, 4]. Object tracking means the process of locating the object of interest from a video sequence. The object is tracked by continuously monitoring the motion of an object in the video. Each of the frames can be divided into two set of objects, foreground and background objects [5]. An adaptive background subtraction technique can also be used for object detection [6]. The complete process of object tracking can be categorized into

* Corresponding author.

E-mail addresses: aapurva2009@gmail.com (A. S. Samdurkar), shailesh_2kin@rediffmail.com (S. D. Kamble), thakurnisvis@rediffmail.com (N. V. Thakur), akshay.patharkar7@gmail.com (A. S. Patharkar).

the following three steps, object detection, object classification and object tracking [7]. Many application of tacking object using different approaches and techniques are implemented. Video surveillance is the basic and widely used application of object tracking [8]. The block matching algorithm is used for the purpose of tracking an object. The algorithms for motion estimation by block matching are used where similar blocks in a sequence of frames of the video are located for the purposes of motion vector estimation. Motion estimation is the process of finding the motion vectors from all frames in a video sequence [9]. The block matching algorithm aims at finding a matching block from a frame in some other frame. Block matching involves partitioning the current frame into a number of macro blocks and compares each macro block with the corresponding macro block. A vector is created that maps the movement of a macro block from one location to another [10]. These motion vectors provide the displacement in the block which can be used for object tracking. Block matching technique also helps in removing the redundancy in the frames. This process can also be carried out in real-time environment along with some hardware [11, 12]. The matching process is performed by minimizing a certain matching criterion (Mean of Absolute Difference), and the best-matched block, where the motion vector is found, which gives the minimum block-matching distortion (BMD) [13]. There are different approaches of block matching algorithm (BMA) amongst which some are advanced algorithms of the already

existing BMAs [14]: Three step search (TSS); novel four step search [15], New three step search (NTSS) [16]; Simple and efficient search; Four step search (FSS) [17]; Diamond Search (DS) [18] & Cross Diamond search (CDS) [19] etc. In this paper, the modified diamond search algorithm (MDS) is presented by introducing the capabilities of DS and CDS. It starts by initializing the small cross shape in the first step, which grows into the large diamond shape (LDSP). Estimation of Motion vector (MV) by reducing the size of the blocks can also make the tracking process fast [20]. The various applications of block matching algorithm other than tracking are used in video compression [21], block matching motion estimation using automata theory for fractal coding [22,23] and also modified Three-Step Search (TSS) algorithm using fractal coding [24]. The various sections of this paper are organized as follows: Algorithm and analysis of DS and CDS are given in Section II. Section III describes the Proposed MDS algorithm. Section IV reports the significant experimental results. The conclusion and future scope is given in Section V.

II. DS and CDS Algorithm

From the entire existing block matching algorithms the behavior of diamond search and cross diamond search are studied. These two algorithms are selected for study because these algorithms require less searching points than the other algorithms as seen from the studied literature survey. The DS and CDS algorithms are explained in detail below with its steps.

A. Diamond Search Algorithm

The steps of DS algorithm with different cases are explained below.

Step 1: Start at the center by initializing 9 search points on the search window. Check all the 9 search points, if the minimum cost found to be at the center of the search window then go to Step 3, else go to Step 2.

 Step 2: The point marked with minimum cost in the Step 1 is re-set as the new center by forming LDSP around it. If the newly obtained minimum cost point is located at the center position then go to Step 3, else repeat Step 2.

Step 3: Shift the searching pattern from LDSP to Small Diamond Search Patter (SDSP). The minimum cost point found in Step 3 is said to be the final solution for the motion vector, which points to the best matching block.

Diamond search pattern has different cases depending upon the point of concern. The presence of the least cost point at different location is processed differently as shown in Fig. 1.
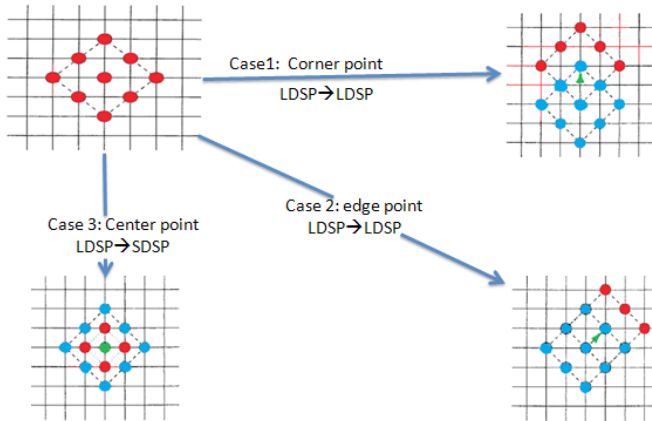

Fig. 1. Search Pattern of Diamond Search Algorithm.

B. Cross-diamond Search Algorithm

Cross diamond shaped search (CDS) pattern implies a cross search pattern (CSP) over the diamond search (DS) pattern as shown in Fig. 2.
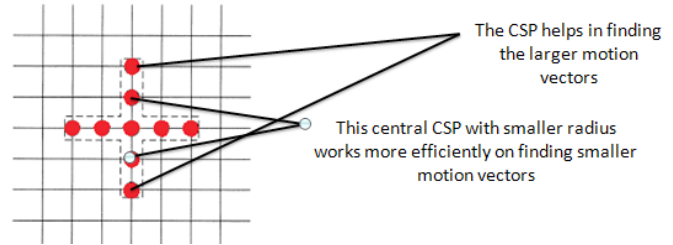

Fig. 2. Cross (+) Shaped Search Pattern.

The CDS Algorithm works as follows:

Step 1: The minimum cost of each of the 9 search points of the CSP is found. If the point with minimum cost is found at the center of the CSP then stop the search as shown in Fig. 3(a), else go to Step 2.

Step 2: 2 more closest search points to the current minimum cost point of the central CSP are checked. If the minimum cost point from the previous step 1 is located at the center of the CSP and if the new minimum cost point found in this step coincides with this point then, stop the search as in Fig. 3(b) else go to Step 3.

Step 3: The point with minimum cost found in previous Step 2 is the new center of the LDSP. If the new point with minimum is still at the center of the newly formed LDSP then go to step 4. Else repeat this step.

Step 4: A new SDSP is formed with the minimum cost point in the previous step as the center. The new minimum cost point, in this step, is the final solution to motion vector.
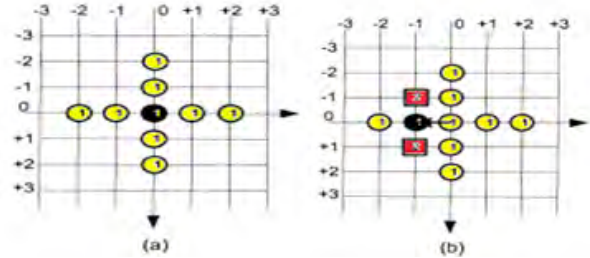

Fig. 3. (a) First step stop   (b) Second step stop

For an in-depth analysis of the block matching motion estimation (BMME) algorithm, the experiments are carried out using the existing block matching algorithm: Diamond search (DS) and Cross diamond search (CDS). The experiments are performed on eight video sequences listed in Table I, among which the first four are the well-known sequences such as "Traffic", "Pet", "Walk", "Ant", and the remaining four "Bottle1", "Bottle2", "Walk1", "Walk2" are the user defined video data sets. All these data sets consist of only a single object moving throughout the frames. Also for all the video data sets used, the camera is stationary. The video sequences such as "Traffic", "Walk", "Walk1" involves higher motion than other videos. The other remaining video sequences involve gentle or slower motion. The DS and CDS algorithms are performed on a 16x16 macro block, with a window size (w) of -7 to +7. The diamond search algorithm starts by initializing 9 points at the center of the search window. It then carries out the steps according to the algorithm and finds out the motion vectors. The motion vectors are found using the block matching criterion such as MAD. The Cross diamond search algorithm also starts by initializing 9 points in the first step, but it differs from the diamond search, in the shape of searching points initialized. The

results, after performing DS and CDS on the eight video sequences, the searching points per frame and computational time per frame are tabulated as required by the DS and CDS algorithms, given in Table II and Table III.

The diamond search pattern starts by initializing nine points, arranged in a diamond shape pattern, on 5x5 grids, to cover the search points in both the directions (up and down). The diamond search pattern may require 13 search points at its best case and 30 search points in the worst case. On an average, it may range between 13 to 30 search points. Table II are the results of the searching points and computational time computed, using the diamond search algorithm, on the video sequences. In the same way, the cross diamond search algorithm also requires nine points in the first step but the search points are arranged such that it forms a cross like structure (+). In the best case, CDS may require 9 points and 29 search points in the worst case. On an average, it may range between 9 to 29 search points. Table III are the results of the searching points and computational time computed, using the Cross Diamond search algorithm, on the video sequences. The experiments are performed on the following sequences of video, "Traffic", "Pet", "Walk", "Ant","Bottle1", "Bottle2","Walk1", "Walk2". The first 10 frames of each of these video sequences are considered for analysis. The obtained results are studied and analyzed for further designing of modified diamond search algorithm. Fig. 4 and Fig. 5 represent the graph of the obtained results in terms of searching points and computational time, for DS and CDS, respectively. The main aim is to design an algorithm that could match or to improve the performance of the DS and CDS, in terms of the search points required and speed up the tracking process.

TABLE I. Video Sequence with Specification used for Experimentation

Sr. No	Name of video sequence	No. of frames	Frame Rate	Frame Height	Frame Width
1	Traffic	120	15	120	160
2	Pet	520	25	288	352
3	Walk	113	25	120	160
4	Ant	430	8	288	352
5	Bottle1	123	15.08	240	320
6	Bottle2	76	15.08	240	320
7	Walk1	220	30	320	240
8	Walk2	180	30	240	240

TABLE II. Comparison of All Videos in Terms of Searching Points per Frame and Computational Time per Frame using Diamond Search Algorithm

| Diamond Search Algorithm																
Search Points per Frame								Computational Time per Frame								
Fr. No	Traffic	Pet	Walk	Ant	Bottle1	Bottle2	Walk1	Walk2	Traffic	Pet	Walk	Ant	Bottle1	Bottle2	Walk1	Walk2
2	11.56	12.52	13.74	13.59	13.93	13.92	14.13	13.65	8.75	49.46	8.59	47.79	36.36	36.2	36.53	36.52
3	11.77	12.52	14.67	13.48	12.65	12.87	13.76	13.57	8.62	50.24	8.54	48.23	36.08	36.38	36.32	36.37
4	11.7	12.43	16.62	12.93	12.7	12.8	14.06	13.46	8.74	51.04	8.56	48.01	36.08	36.15	36.35	36.39
5	11.65	12.41	15.25	13.39	13.18	12.89	13.65	13.7	8.81	48.32	8.54	48.72	36.64	36.15	36.28	36.44
6	11.65	12.39	14.12	12.92	12.75	12.53	13.75	13.24	8.99	481	8.59	48.7	37.44	36.09	36.31	36.39
7	11.65	12.36	14.32	13.34	12.51	12.59	13.61	13.49	8.59	47.61	8.53	48.69	40.25	36.08	36.38	36.46
8	11.7	12.45	12.94	13.01	12.76	12.63	13.95	13.74	8.81	47.92	8.61	47.98	43.06	36.14	36.43	36.43
9	12.17	12.39	15.6	13.28	12.86	12.59	13.55	13.34	8.73	47.64	8.58	47.83	42.92	36.28	36.32	36.39
10	8.73	12.41	15.18	13.09	12.89	12.77	13.81	13.19	8.5	47.74	8.65	47.76	36.5	37.03	36.34	36.39
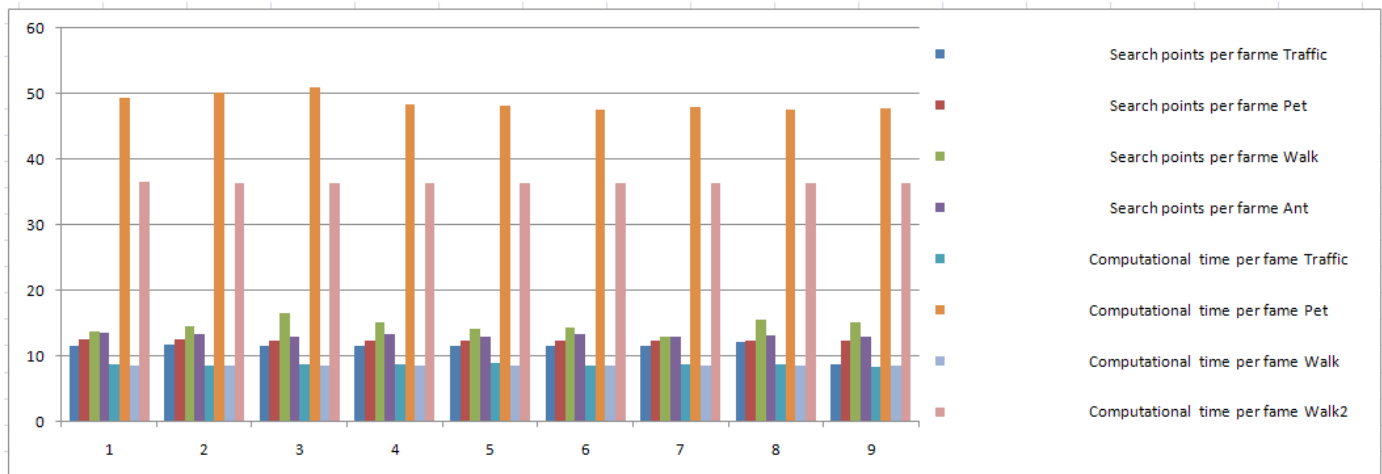
Fig. 4. Graphical representation of videos in terms of search points/frame and computational time/frame of DS algorithm.

TABLE III. Comparison of All Videos in Terms of Searching Points per Frame and Computational Time per Frame using Cross Diamond Search Algorithm

|	Cross-Diamond Search Algorithm															
	Search Points per frame								Computational time per frame							
Fr. No	Traffic	Pet	Walk	Ant	Bottle1	Bottle2	Walk1	Walk2	Traffic	Pet	Walk	Ant	Bottle1	Bottle2	Walk1	Walk2
2	8.35	9.54	12.67	11.65	12.6	12.65	12.49	11.79	7.25	40.25	7.23	39.78	34.78	32.01	31.2	31.16
3	8.4	9.48	14.05	11.38	10.46	10.64	12.56	11.74	7.66	39.65	7.33	39.87	29.95	29.22	31.26	30.96
4	8.45	9.36	17.87	10.44	10.47	10.48	12.72	11.39	7.24	39.44	8.32	39.1	29.26	29.38	30.75	30.24
5	8.31	9.37	15.41	11.31	11.52	10.44	11.97	11.9	8.1	38.96	8.15	39.41	30.93	29.12	30.26	31.21
6	8.38	9.74	14.2	10.47	10.42	9.68	11.83	10.9	7.99	43.26	7.86	39.64	29.2	28.92	30.41	29.96
7	8.31	9.08	13.39	11.18	9.73	9.71	11.54	11.16	11.46	43.45	7.91	39.45	28.4	28.33	29.82	29.82
8	8.45	9.25	12.91	10.58	10.6	9.8	12.2	11.91	8.52	40.16	7.61	38.96	29.48	28.37	29.87	30.57
9	8.94	9.27	15.27	11.25	10.81	9.73	11.44	10.81	7.2	41.45	8.26	39.37	29.42	28.35	29.83	29.58
10	8.68	9.16	15.5	10.63	10.72	10.1	12.12	10.86	7.79	45.41	8.28	38.72	29.54	28.57	30.22	29.88
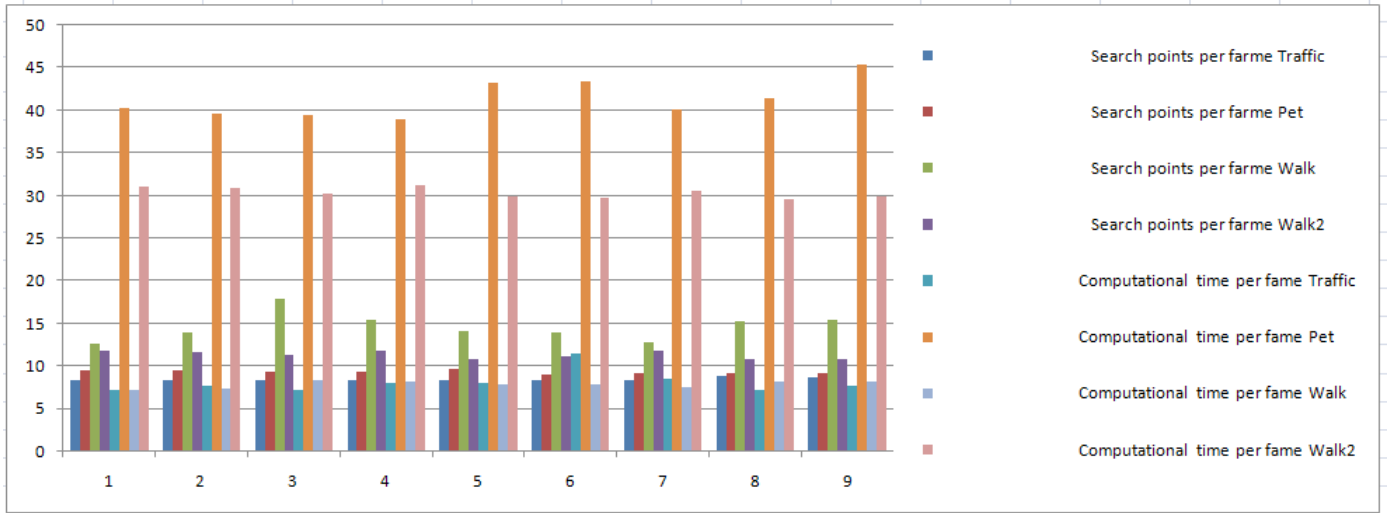
Fig. 5. Graphical representation of videos in terms of search points/frame and computational time/frame of CDS algorithm.

III. Proposed Work

The proposed work is concerned with designing an algorithm that could match or improve the performance of the DS and CDS, in terms of the search points required and speed up the tracking process. The tracking process can be made fast if the search points required by the proposed algorithm can be reduced.

Flowchart of the object tracking system is shown in Fig. 6. That helps in visualization of the stepwise working of the proposed approach. It represents the process for single object tracking using block matching motion estimation techniques.

- As a preprocessing step video is converted into YUV color space, where only the "Y" component (luminance) is considered as it is sensitive to the human eye.

- Later the video is divided into many frames. Only two frames are considered at a time of processing. From these two frames one is current frame and the other is reference frame.

- On all the frames of the video sequence the background subtraction method for object detection is applied. In background subtraction, the current frame gets subtracted from the reference frame. The result after subtraction is the object in the foreground.

- After detecting the object using background subtraction, the Block Matching Algorithm (BMA) is applied.

- BMA divides the block into number of macroblocks.

- BMA calculates the motion on a block by block basis. For every current frame block, a block from previous frame is found and matching is based on certain criteria. (MAD here to find best match).

- A vector is created that keeps track of the movement of a macro block within different locations. Motion vector provides displacement in the block in terms of MVx and MVy.

- Using all these motion vectors MVx and MVy the object is tracked from the video.

A. Diamond Search Pattern

The diamond search algorithm uses a Large Diamond Shape Pattern (LDSP) and a Small Diamond Shape pattern (SDSP) as shown in Fig. 7. The Fig. 7 (b) is used as the initial step to the Modified Diamond Search (MDS) algorithm, followed by the Fig. 7 (a).
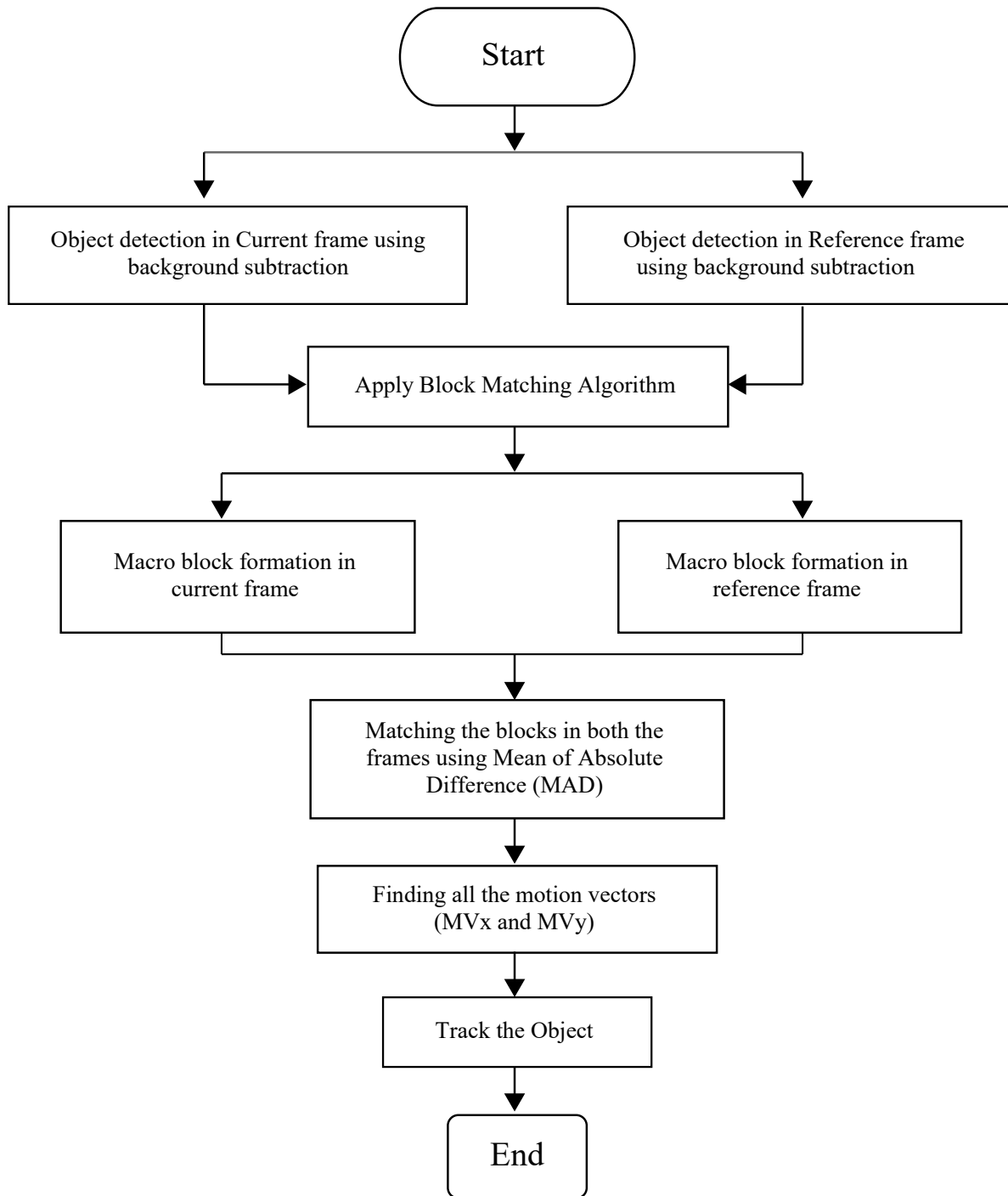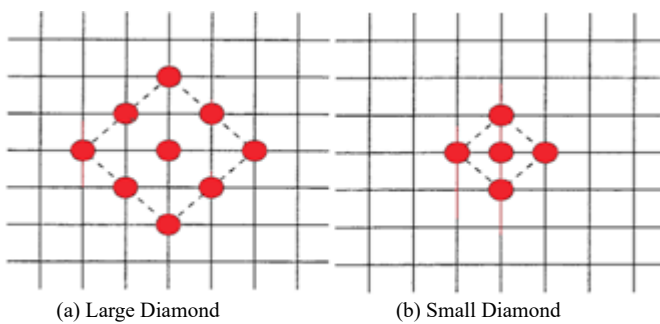
Start

Object detection in Current frame using background subtraction

Object detection in Reference frame using background subtraction

Apply Block Matching Algorithm

Macro block formation in current frame

Macro block formation in reference frame

Matching the blocks in both the frames using Mean of Absolute Difference (MAD)

Finding all the motion vectors (MVx and MVy)

Track the Object

End

Fig. 6. Flow of object tracking system.

(a) Large Diamond   (b) Small Diamond

Fig. 7. Search Pattern used in MDS Algorithm.

1) The Modified Diamond Search Algorithm

Following are the steps of MDS algorithm:

Step 1: Start with five points of SDSP as the initial step from the center of the search window and the 5 checking points of SDSP, are tested. If the cost with minimum point after calculation is located at the center position then stop the search. Otherwise, go to Step 2.

Step 2: The minimum cost point found in the Step 1 is re-positioned as the center point and forms a new LDSP. If the new point with minimum cost obtained is located at the center position then go to Step 3; else repeat this step in recursion until the minimum point occurs at the center of the LDSP and then go to Step 3.

Step 3: With the new minimum point as the center, shift the search pattern, from LDSP to SDSP. The minimum cost point, found in this step, is the final solution of the motion vector, which is the best matching block.

Fig. 8 shows the flowchart of the MDS algorithm and the various cases of MDS algorithm are shown in Fig. 9.
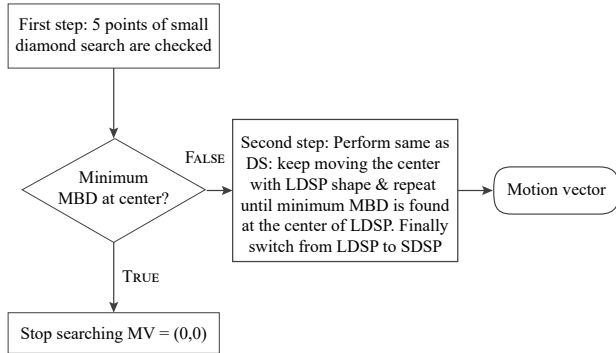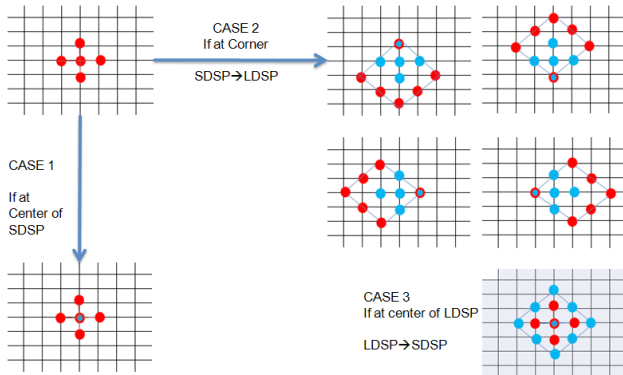

Fig. 8. Flowchart of the MDS algorithm.

Fig. 9. Different cases of searching pattern in MDS algorithm.

B. Analysis of MDS Algorithm

Fig. 9 shows the various cases of MDS algorithm, in which the modified diamond search algorithm starts, by initializing only five points in the first step in a 3x3 window and nine points in the second step on the 5x5 search window respectively, when compared to the nine searching points used in the first step of DS and CDS. With this characteristic, it can facilitate the optimization of computation of MDS over DS and CDS. The design of MDS also helps in finding smaller motion vectors more efficiently. MDS algorithm behaves like DS only if the min MBD is not found at the center in the first step and keeps on modifying between successive LDSP, by three points or five points for searching. Thus the total number of search points, in the best case, are 5 and varies from 5 to (5+8) =13, in the worst case, whereas, on an average, it requires search points in between 5 and 13.

The MDS algorithm is quite different from any other fast BMAs. i.e. (i) The search patterns used in MDS have the minimum number of points; (ii) the directional search patterns are used, and (iii) the switching strategy of the diamond search patterns is adopted in the last stage. MDS starts with a small searching pattern and then grows into a large shape and again shrinks to the small shape, replacing the methods, in cross search pattern and simple diamond pattern.

IV. Experimental Results

A. Experimental Results on MDS

The proposed MDS algorithm is simulated and tested using the various standard video sequences and use created video sequences consisting of different motion types. For all the video sequences we have used the Mean of Absolute Difference (MAD) as the block matching measure (criterion), block size of 16x16, and search window of size 7 (i.e. w = -7 to +7). All these video sequences used for the experiment, consist of only a single object moving throughout the frames. Also for all the video data sets used, the camera is stationary. The video sequences, such as "Traffic", "Walk", "Walk1", involves higher motion than other videos. The other remaining video sequences involve a slow moving object. The results of the Modified diamond search algorithm are listed in Table IV.

The results in Table IV are graphically represented in Fig. 10. A lot of improvement is seen in the result of MDS over the DS and CDS. The modified algorithm for diamond search is compared against the existing diamond search and cross diamond search algorithm, in the aspect of searching points.

TABLE IV. Comparison of All Videos in Terms of Searching Points per Frame and Computational Time per Frame using MDS

| Modified Diamond Search Algorithm																
Search points per frame								Computational time per frame								
Fr. No	Traffic	Pet	Walk	Ant	Bottle1	Bottle2	Walk1	Walk2	Traffic	Pet	Walk	Ant	Bottle1	Bottle2	Walk1	Walk2
2	8.35	9.12	10.04	10.07	10.83	11	10.56	10.5	8.57	48.39	8.68	49.36	36.76	36.51	36.41	36.44
3	8.41	9.08	10.31	9.98	9.84	9.79	10.53	10.37	8.57	48.04	8.60	89.16	35.99	36.44	36.35	36.75
4	8.41	9.05	11.6	9.57	9.96	9.7	10.46	10.04	8.53	47.94	8.68	49.02	36.01	36.92	36.3	36.43
5	8.31	9.07	11.64	9.84	10.62	9.75	10.21	10.25	8.56	47.93	8.68	48.23	36.03	36.43	36.41	36.38
6	8.34	8.86	10.98	9.56	9.91	9.21	10.18	9.73	8.58	48.24	8.64	48.75	36.63	36.67	36.4	36.38
7	8.31	8.84	11.24	9.87	9.4	9.24	9.91	9.69	8.54	47.87	8.73	48.96	36.05	36.32	36.39	36.47
8	8.34	9.02	10.67	9.63	9.84	9.24	9.88	9.99	8.68	48.10	8.68	49.01	36.04	36.25	36.41	36.42
9	8.48	8.94	11.64	9.89	9.87	9.28	9.88	9.7	8.57	48.04	8.69	48.69	36.13	36.43	36.31	36.36
10	8.44	8.92	11.67	9.61	9.91	9.49	10.15	9.73	8.56	47.85	8.58	48.69	36	36.2	36.37	36.54
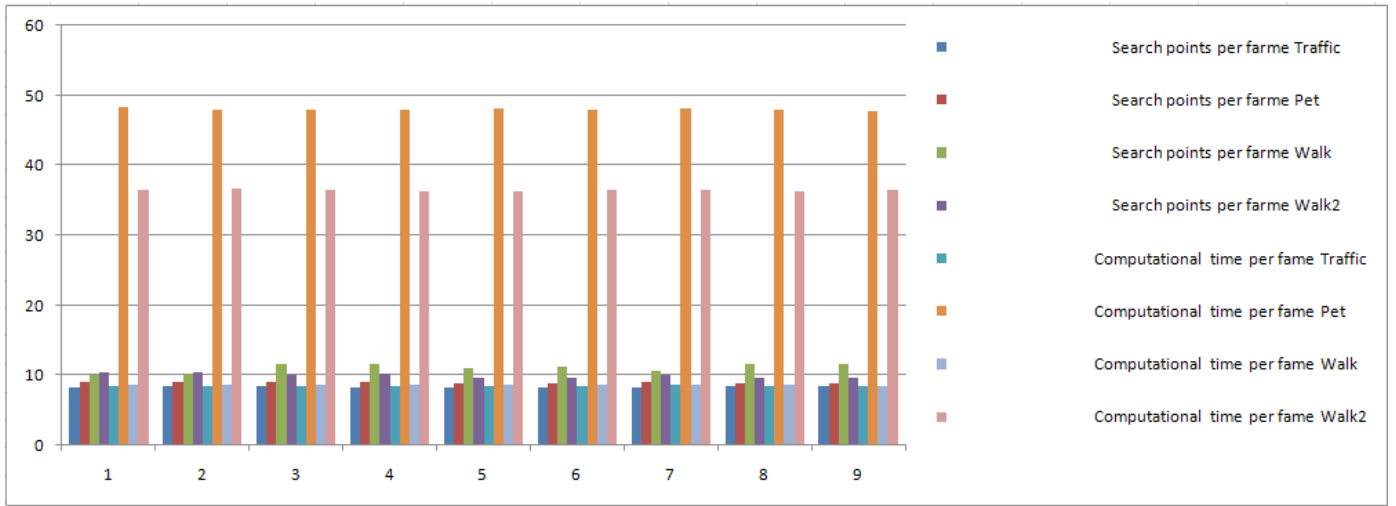
Fig. 10. Graphical representation of videos in terms of search points/frame and computational time/frame of MDS algorithm.

Table V shows the comparison of MDS with DS and CDS algorithm.

TABLE V. Performance Comparisons of DS, CDS and MDS

Using sequence "Traffic"			
BMA	DS	CDS	PDS
Avg. SP	11.4	8.47	8.38
Using sequence "Pet"			
BMA	DS	CDS	PDS
Avg. SP	12.43	9.36	8.99
Using sequence "Walk"			
BMA	DS	CDS	PDS
Avg. SP	14.71	14.63	11.09
Using sequence "Ant"			
BMA	DS	CDS	PDS
Avg. SP	13.22	10.98	9.78
Using sequence "Bottle1"			
BMA	DS	CDS	PDS
Avg. SP	12.91	10.81	10.02
Using sequence "Bottle2"			
BMA	DS	CDS	PDS
Avg. SP	12.84	10.35	9.63
Using sequence "Walk1"			
BMA	DS	CDS	PDS
Avg. SP	13.8	12.07	10.19
Using sequence "Walk2"			
BMA	DS	CDS	PDS
Avg. SP	13.48	11.38	10

The Table V compares the searching points among different block matching algorithms. It shows that the Modified diamond search algorithm (MDS) always consumes the smaller number of search points, as compared to the Diamond Search (DS) and Cross Diamond Search (CDS). Compared to DS, CDS approximately saves about 1.98 – 2.24 search points. When CDS is compared to MDS, it also saves 0.36 – 3.55 search/check points, which is much more than the other two BMAs. The average search points, per frame with observations, MDS < CDS < DS, is detected for the video sequences, with w= +7 to -7. With such an improvement in the searching point, the experimental results, using the video sequence, shows that the MDS algorithm achieves less searching points, than that of DS

and CDS. For the video sequence "Traffic", the MDS algorithm saves up to 0.36 and even 3.02 search/check points per frame, as compared to CDS and DS respectively. For sequence "Pet" the proposed algorithm saves 0.37 to 3.44 search points, "Walk" saves 3.55 to 3.62 searching points, "Ant" search points ranges from 1.2 to 3.44, "Bottle1" sequence saves 0.79 to 2.89 searching points, video sequence "Bottle2" saves 0.72 to 3.21 search points, and the video sequences "Walk1" and "Walk2" saves 1.88 to 3.61 and 1.38 to 3.48 searching points per frame. This implies that the searching pattern of proposed Diamond Search (MDS) algorithm, with a search area of 3x3, in the first step, starts with 5 searching points, instead of with 9 search points as in DS and CDS. This feature outperforms the existing Diamond search and Cross Diamond search that use a larger search area and more search/check points in the first step. The smaller shape of MDS helps in finding smaller Motion Vector (MV) efficiently. MDS starts with a small searching pattern and then grows into a large shape and again shrinks to the small shape, replacing the methods in cross search pattern and simple diamond pattern. Thus, MDS performs better than DS and CDS. Fig. 11 and Fig. 12 plots the average searching points per frame using sequence "Traffic" and "Bottle1" respectively, on each frame. These two figures clearly show the superior performance of MDS over DS and CDS algorithm, in terms of the search points used. Fig. 13 shows the performance comparison of DS, CDS, and MDS in terms of average search points.

B. Experiment Results using MDS for Object Tracking

This research has applied the capabilities of Modified diamond search (MDS) block matching algorithm for the purpose of tracking an object. The block matching motion estimation algorithm is used, where similar blocks in a sequence of frames of the video are located, for the purposes of motion vector estimation. The purpose of a block matching algorithm is to find a matching block, from a frame, in some other frame. Block matching involves partitioning the current frame into a number of macro blocks and compares each macro block, with the corresponding block. A vector is created that maps the movement of a macro block, from one location to another. These motion vectors provide the displacement in the block, which can be used for object tracking. In the preprocessing, the RGB video sequence is converted into YUV space and all the experimentation is done, using the luminance of video sequence. Thus this research performs background subtraction for detection of the object and then applies the MDS algorithm to it, where the motion vectors are calculated and the object is tracked. Fig.14 shows the step wise results of the tracking. Final tracked object is shown by the blue bounding box.
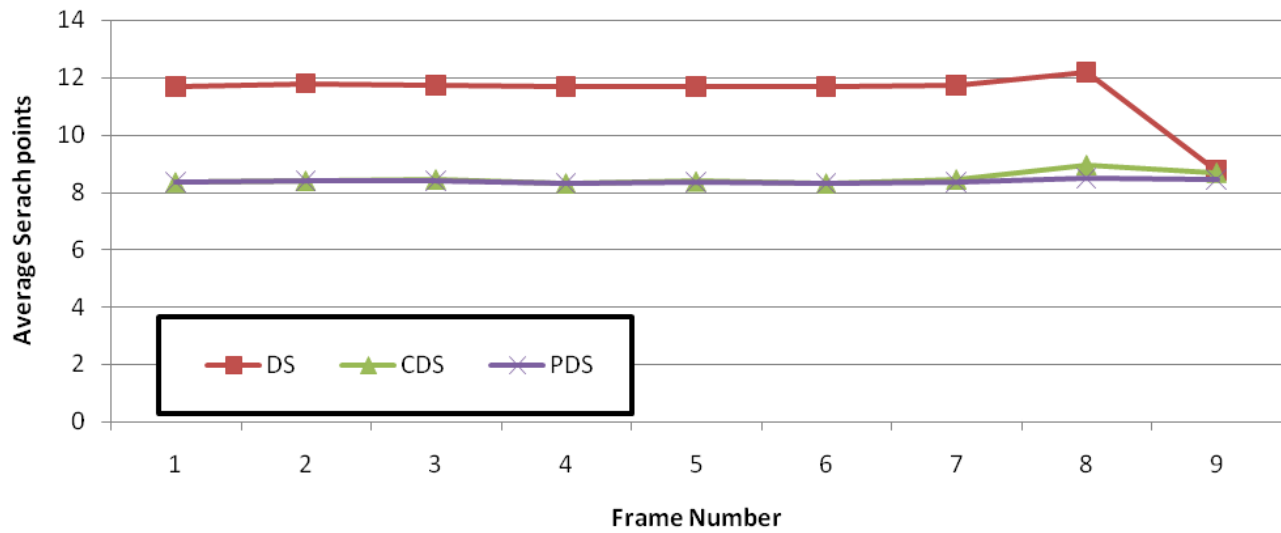
Fig. 11. Frame-wise performance comparison between DS, CDS and MDS on "Traffic" sequence by average search point.
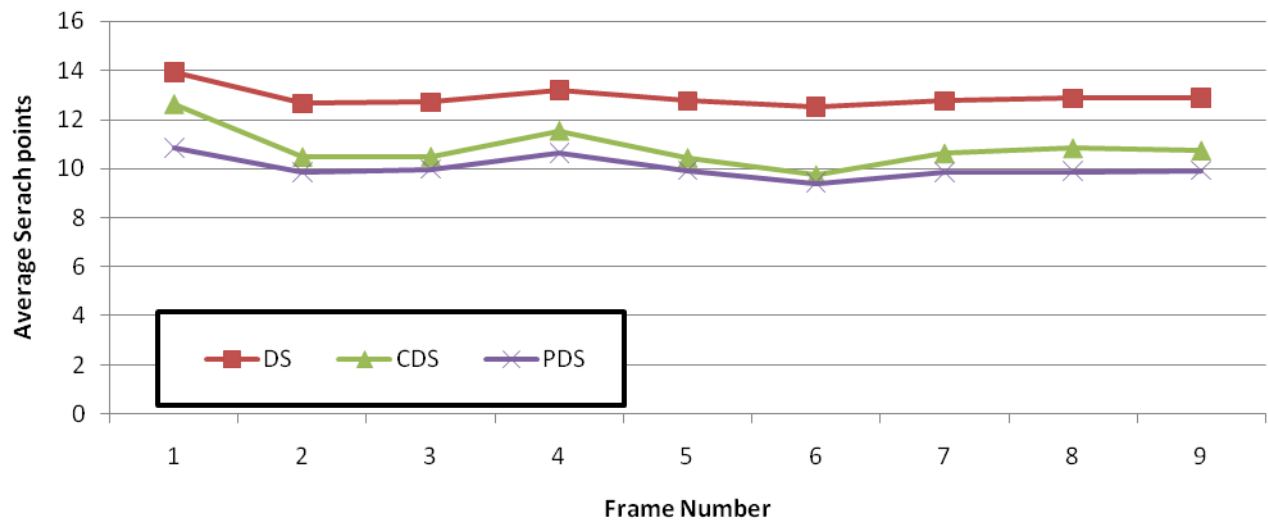

Fig. 12. Frame-wise performance comparison between DS, CDS and MDS on "Bottle1" sequence by average search points.
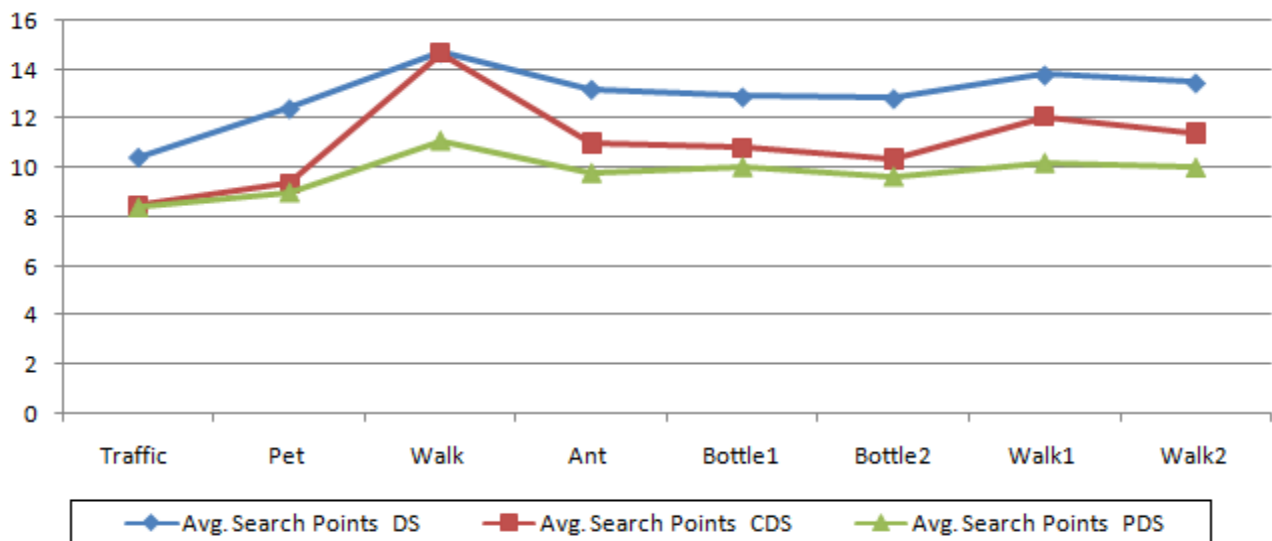

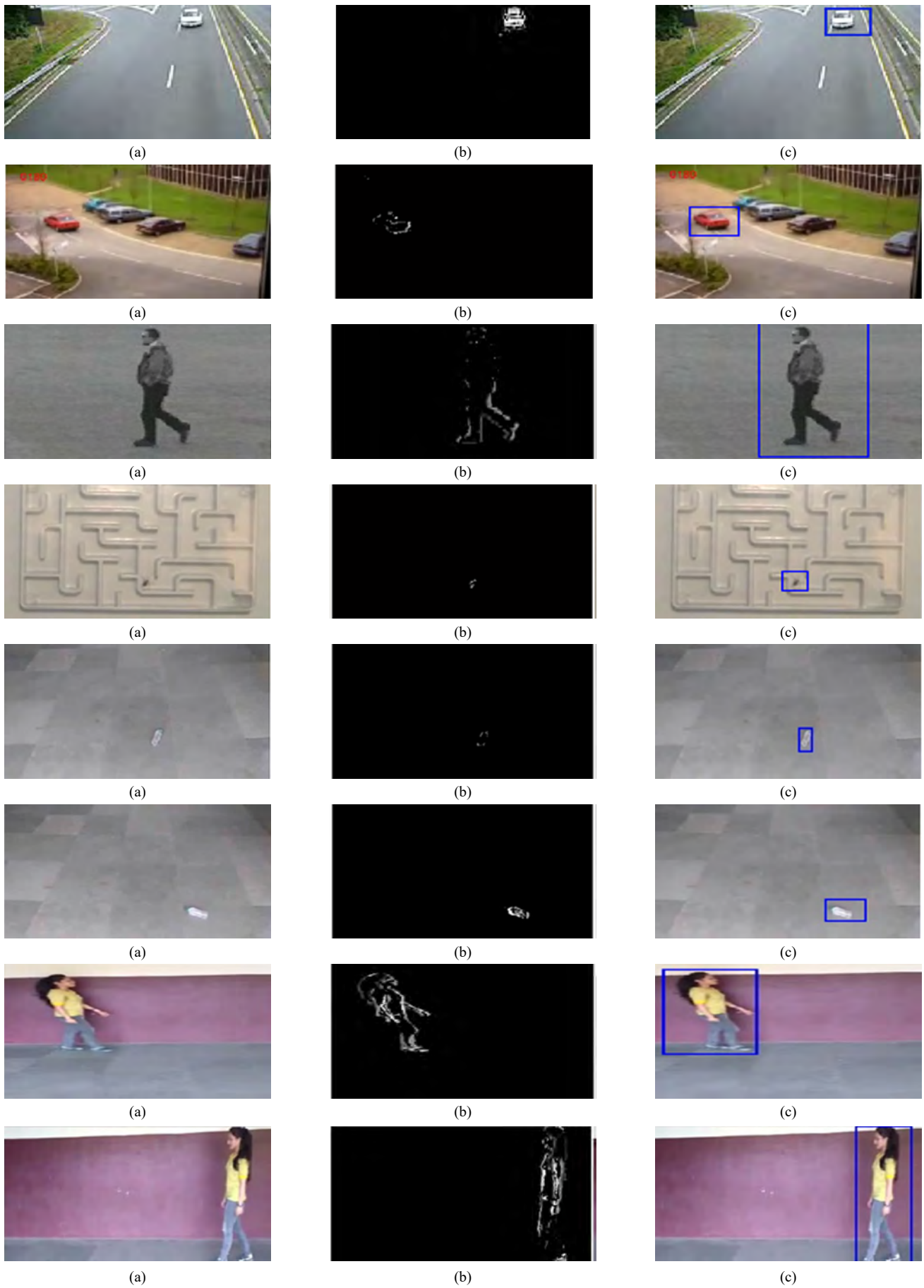Fig. 13. Performance comparison of DS, CDS and MDS by average search points.

Fig. 14. (a) Original video Frame (b) Background Subtraction (c) Object Tracking Result of the video sequences.

(a)   (b)   (c)   (d)

Fig. 15. (a) Original video frame, (b) YUV converted frame, (c) Output after background subtraction, (d) Result of object tracking.

C. Results of Object Tracking

The experiments start with the background subtraction and have implemented the Modified Diamond search algorithm on all the video sequences, to obtain the results. The first 20 frames of the "Walk1" sequence are processed and the step wise output is given, as shown in Fig. 15. The first image is the original video frame, followed by its YUV converted frame. Third image shows the output after the background subtraction and the last image shows the result after object tracking.

D. Simulator

The proposed approach is implemented using (MATLAB 8.1.0.604) (R2013a). The experiments are carried out on Intel (R) Core (TM) 3 Duo T6570, 2.10 GHz processor. The RAM of 4GB is used. The operating system is 32-bit installed on Windows 7 platform.

V. Conclusion and Future Scope

In this paper, a modified diamond search algorithm (MDS) is proposed by combining the capabilities of Diamond search (DS) and Cross diamond search (CDS) block matching algorithm for finding motion vectors between two macro blocks. These motion vectors are then used to track the object in video sequence. The proposed algorithm uses a 3x3 area as the first most step and diamond shape as the further steps. The proposed work is concerned with designing an algorithm that could match or improve the performance of the DS and CDS, in terms of the search points required and speed up the tracking process. The experimentation performed and obtained results show that the proposed algorithm requires fewer searching points as compared to DS and CDS algorithms. The tracking process can be made fast if the search points required by the proposed algorithm can be reduced. Hence this research concludes that the Modified diamond search Algorithm is better, when compared with DS and CDS, based on the experimental results.

The literature on object tracking is very rich. Developing a new way of tracking object from the video scene by combining the object detection approach with the block matching algorithm gives a scope of improvement to existing object tracking techniques. In future this research can extend the block matching algorithm for object tracking applications using real-time object tracking system.

References

[1] H.S. Parekh, U.K. Jaliya, D.G. Thakore, "A Survey on Object Detection and Tracking Methods", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 2, 2014.

[2] B. Karasulu and S. Korukoglu, "Moving Object Detection and Tracking in Videos", Springer Briefs in Computer Science, 2013.

[3] S. H. Shaikh, N.Chaki, K. Saeed, "Moving Object Detection Using Background Subtraction", Springer Briefs in Computer Science, 2014.

[4] Helly, M. Desai, V. Gandhi, "A Survey: Background Subtraction Techniques", International Journal of Scientific & Engineering Research, Vol. 5, Issue 12, 2014.

[5] M.Piccardi, "Background subtraction techniques: a review", IEEE International Conference on Systems, Man and Cybernetics, 2004.

[6] R.Zhang, J.Ding, "Object Tracking and Detecting Based on Adaptive Background Subtraction", International Workshop on Information and Electronics Engineering, 2012.

[7] Duc Phu Chau, Francois Bremond, Monique Thonnat, "Object Tracking in Videos: Approaches and Issues", the International Workshop "Rencontres UNS-UD", 2013.

[8] A.K.Chauhan and D.Kumar, "Study of Moving Object Detection and Tracking for Video Surveillance", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, Issue 4, 2013.

[9] Chia-Ming Wu and Jen-Yi Huang, "A New Block Matching Algorithm for Motion Estimation", Applied Mechanics and Materials, Trans Tech Publications, 2017.

[10] N.Singh, A.Mishra, "Block Matching Algorithm for Motion Estimation using Previous Motion Vector Pattern", International Journal of Computer Applications, Vol. 150, No.8, September, 2016.

[11] H. A.Surrah, Mohd. J. Haque, "A Comparative Approach for Block Matching Algorithms used for Motion Estimation", International Journal of Computer Science Issues, Vol. 11, Issue 3, No. 2, 2014.

[12] Hsiang-Kuo Tang, Tai-Hsuan Wu, Ying-Tein Lin, "Real-time object Image Tracking based On Block Matching Algorithm", Project Report available at https://homepages.cae.wisc.edu/~ece734/project/s06/lintangwuReport.pdf

[13] M.H.Sherie, I.Ashimaa, Mahmoud Imbaby and A. Elam, "Experimental Comparison among Fast Block Matching Algorithms (FBMAs) For Motion Estimation and Object Tracking", in Proceeding of 28th National Radio Science Conference (NRSC 2011),National Telecommunication Institute, Egypt, 2011.

[14] B.N. Tejas, S.H. Bharathi, K.N. Vidya Sagar, "Block Based Algorithms for Estimating Motion", International Research Journal of Computer Science, Vol. 3, Issue 05, 2016.

[15] B.Sugandi, H.Kim., J.K.Tan and S.Ishikawa, "A Block Matching Technique for Object Tracking", 2008 International Conference on Computer and Communication Engineering, Kuala Lumpur, 2008.

[16] N. Verma, T. Sahu, P.Sahu, "Efficient Motion Estimation by Fast Three Step Search Algorithms", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 1, Issue 5, 2012.

[17] Bassam, Ait-Boudaoud, Djamel, "Improved diamond half-pel hexagon search algorithm for block-matching motion estimation", *EI 2017 proceedings.* Society for Imaging Science and Technology, IS&T International Symposium on Electronic Imaging, 2016.

[18] S.Banchhor, D.Shukla, "An Improved Diamond Search Pattern for Motion Estimation", IEEE Transaction on Image Processing, Vol. 9, Issue 2, 2016.

[19] C.H. Cheung, L.M.Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation", i-manager's Journal on Pattern Recognition, 2002.

[20] S. Acharjee, N. Dey, D. Biswas, P. Das, and S. Chaudhuri,"A novel Block Matching Algorithmic Approach with smaller block size for motion vector estimation in video compression", 12thIEEE International Conference on Intelligent Systems Design and Applications (ISDA), 2012.

[21] S.Kamble, N.Thakur, L.Malik, P.Bajaj,"Color video compression based on fractal coding using quad tree weighted finite automata", Information system design and Intelligent application, Proceedings of Second International Conference INDIA 2015, Advances in Intelligent System and Computing, Springer India,Vol. 340, pp. 649-658, 2015.

[22] S.Kamble, N.Thakur, L.Malik, P.Bajaj, "Quad Tree Partitioning and Extended Weighted Finite Automata Based Fractal Color Video Coding", Int.J.Image Mining, Vol. 2, No. 1, pp. 31-56, 2016.

[23] S.Kamble, N.Thakur, P.Bajaj, "A Review on Block Matching Motion Estimation and Automata Theory based Approaches for Fractal Coding", International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 4, No. 2, pp. 91-104, 2016.

[24] S.Kamble, N.Thakur, L.Malik, and P.Bajaj, "Fractal Video Coding using Modified Three-Step Search Algorithm for Block Matching Motion Estimation", Computational Vision & Robotics in Proceedings of International Conference on Computer Vision & Robotics, Advance in Intelligent Systems & Computing, Vol. 332, pp. 151-162, 2015.

Apurva S. Samdurkar

Apurva S. Samdurkar received Bachelor of Engineering degree in Computer Science and Engineering from Pankaj Laddhad Institute of Technology and Management Studies Buldana, India under the Sant Gadge Baba Amravati University, Amravati, India. She received Master of Technology degree from Yeshwantrao Chavan College of Engineering, Nagpur, India. Presently she is working as a Software Engineer at MRR Soft LLC. She is member of CSI, India and IEANG.

Shailesh D. Kamble

Shailesh D. Kamble received Bachelor of Engineering degree in Computer Technology from Yeshwantrao Chavan College of Engineering, Nagpur, India under the Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur, India. He received Master of Engineering degree from Prof Ram Meghe Institute of Technology and Research, formerly known as College of Engineering, Bandera under Sant Gadge Baba Amravati University, Amravati, India. Presently, he is Ph.D. candidate in department of Computer Science and Engineering from G.H. Raisoni College of Engineering (An Autonomous Institution Affiliated to Rashtrasant Tukadoji Maharaj Nagpur University), Nagpur, India. He is working as the Assistant Professor in Department of Computer Science and Engineering at Yeshwantrao Chavan College of Engineering, Nagpur, India. He is having over 15 years of teaching and research experience. His current research interests include image processing, video processing and language processing. He is the author or co-author of more than 20 scientific publications in International Journal, International Conferences, and National Conferences. He is the life member of ISTE, India, IE, India.

Nileshsingh V. Thakur

Nileshsingh V. Thakur received Bachelor of Engineering degree in Computer Science Engineering from Government College of Engineering, Amravati, India and Master of Engineering degree from College of Engineering, Bandera under Amravati University and Sant Gadge Baba Amravati University in 1992 and 2005 respectively. He received Ph.D. degree in Computer Science Engineering under Department of Computer Science Engineering from Visvesvaraya National Institute of Technology, Nagpur, India on 1st February, 2010. His research interest includes image and video processing, sensor network, computer vision, pattern recognition, artificial neural network and evolutionary approaches. He is having over 24 years of teaching and research experience. Presently, he is Professor and Head in Department of Computer Science and Engineering and Dean, PG Studies at Prof Ram Meghe College of Engineering and Management, Bandera- Amravati, Maharashtra, India. He is the author or co-author of more than 70 scientific publications in refereed International Journal, International Conferences, National Journal, and National Conferences. He is a member of editorial board of over eight International journals; also, he is the life member of ISTE, India, IAENG and IAEME. He also worked as the reviewer for refereed international journals and conferences.