Optimization of Real Time Edge Enhanced Object Tracking Algorithm on Video Processor

S. Kanwal[1], M. H. Yousaf [2]

[1,2]Computer Engineering Department, University of Engineering and Technology Taxila, Pakistan
[2]haroon.yousaf@uettaxila.edu.pk

Abstract-This research work provides an efficient methodology that is strongly comparable with other techniques. Real time objects can be tracked efficiently. The work undertaken in this dissertation is mainly focused on development of a reliable and robust tracking system that can track any object of interest in the video acquired from a stationary or moving camera. The steps included in the proposed algorithm are target smoothing by applying Gaussian smoothing filter, edge enhancement and template matching. The proposed visual tracking system has been implemented in RGB space. This scheme augments the feature set values of template iteratively, by taking weighted sum of template and the current best match. The proposed algorithm is a real time algorithm that operates in more than 25 frames per second. Frame rate is dependent on template size. Adjustable window size control is also provided. The proposed algorithm is more invariant to varying illumination conditions, and is performing well in presence of background clutter and variations in shape and size of the object. Template drift is significantly minimized in the proposed scheme. Target loss is also significantly minimized. Furthermore, in presence of fog and haze the algorithm proves itself to be efficient. The algorithm can handle partial and fast occlusions.

Keywords-Visual Tracking, Normalized Cross Correlation, Edge Normalization, Efficient Template Updation

I. Introduction

Object tracking is an important task in the field of computer vision. Object tracking is the process of finding the motion of an object in an image sequence. Object tracking can be used in animation and interaction, navigation, recognition of objects, video surveillance, medical applications like medical therapy and computer assisted living. Tracking is also very important in behavioral research in animals. Existing methods depends on limited field of view from a fixed camera and requires lots of human interaction.

Video tracking have been focused by many researchers. Significant efforts have been made in this field. It attracts the attention of industry. Any algorithm designed to perform object tracking analyze frames of video sequentially. Various algorithms have been designed by many researchers that have their own strengths and weaknesses. In the industry, the demand of accurate video tracking algorithms has been increased dramatically over the last few decades. The algorithms designed by the researchers based on high robustness, accuracy and throughput.

For machines efficient object tracking is a difficult task. The computational complexity of the algorithm is critical for most applications. Several target tracking techniques have been presented by many researches in past. Most of the techniques are dependent on limited field of view from a stationary camera and requires lots of human computer interaction. MPEG-4 Video coding techniques [i] imposes a constraint that the object moves small between successive frames. This will put restriction on speed of the object. The optical flow based tracking [ii] schemes can provide accurate locations of moving object but this approach is computationally expensive and suitable for stationary background conditions.

Motion estimation based video trackingapproach is presented in [iii]. This technique is not computationally expensive but if background motion exists than this application fails. In [iv] authors used an adaptive correlation tracking technique to track objects of varying sizes. The algorithm was variant to intensity changes. With light variations the algorithm lost the track. A Fast Normalized Cross Correlation method is introduced in [v] to implement a tracker based on feature matching techniques. The algorithm was not able to lock an object that has low contrast difference with the background. Although it is a fast technique that takes less computational power, but it is using Phase Correlation for template matching which is difficult to map on a fixed point Digital Signal Processor due to the spectrum size of Fast Fourier Transform.

Temporal consistency, temporal differencing and correlation matching techniques are used in [vi]. The technique is not able to track a small sized object. It is also not applicable to cluttered environment. Merge-split (MS) and straight-through (ST) schemes for object tracking are presented in [vii]. The main difficulties in this approach are to re-establish object identities following a split and assignment to a specific object of pixels that could belong to several objects. Another technique presented in [viii] exploited dynamic template updation scheme. This scheme is efficient and accurate but is computationally very expensive.

Mean Shift and Kernel Histogram Filtering is used in [ix] to track an object. This model optimizes similarity measure between target model and candidate model in consecutive frames but is not able to track an object with varying scales and rotation angles. It is also variant to intensity changes.

A binary classifier for object tracking is introduced in [x]. Objects and background are discriminated by a Support Vector Machine (SVM) classifier. They estimate transformation parameters to increase the SVM score. Authors extended this idea who allows propagating observation by a formulation of probabilistic domain. Seed images are used to map image space to transformation parameter space. These methods are robust and efficient; however they rely on the classifier appearance variation, which is not updated during tracker operation. Authors proposed discriminative features to update a classifier in [xi]. It is assumed that background clutter is not present. The idea of classifier updation based on AdaBoost [xii] implemented tracking during large variations in appearance. However these methods are not able to prevent drift.

Eigenspace tracking approach is proposed in [xiii]. Several methods were proposed by the researcher to adapt an eigenspace. Eigenspace updating schemes are also available. Authors exploited an appearance model to a specific number of images in [xiv]. Eigenspaces are updated for updating the model. Outlier handling is not addressed in any of these papers. Sobel edge detection algorithm and soft threshold wavelet de-noisingis used in [xv] to perform edge detection on image occluded with Gaussian noise. Soft threshold wavelet is used to remove noise, and then Sobel operator is used to perform edge extraction. This method is mainly used on the images which includes White Gaussian noises. The algorithm is accurate and performing well in noisy images but it is computationally very expensive and requires more memory because of wavelet transform.

A genetic algorithm and improves sobel operator based approach for edge detection proposed in [xvi] provides effective edges but it extracts too many edges from the image which is not appropriate in task of tracking. The fuzzy-canny logic based magnified edge detector presented in [xvii] uses gray level histogram and global canny edge detection operator. Entropy optimization is also utilized. The technique used global and local edge detection so two edge detection operators are applied which increases computational complexity. Also this type of technique is useful for application like fingerprint identification in which every minute edge is of interest.

An edge detection algorithm based on edge-adapted reconstruction techniques is presented in [xviii]. The technique does not incorporate traditional filtering and thresholding. It utilizes comparison of non linear multi resolution quantities. The algorithm is invariant to illumination changes. This technique misses certain important edges from the image. MIL (Multiple Instance Learning) algorithm based object tracking presented in [xix] achieves real time performance. The algorithm trains a classifier to separate the object. Slight inaccuracies in tracker can degrade the classifier. The classifier training causes drift in the template.

Some researchers utilize blob based tracking while considering environment free from occlusion, and only considers that Particle Filter approaches are suitable for occlusion handling [xx, xxi]. It is clear from the technique itself that it has a limitation that it needs blobs which is not suitable when the scene is crowded. The technique is not suitable for crowdy environment.

The self similarity concept for visual detection and considered it as a feature was proposed in [xxii]. In this research they worked on object detection, object recognition, object tracking and image registration. In [xxiii] authors use self similarity plots of images (ISSP) for the first time for recognition. Self similarity metrics for recognition and tracking are proposed in [xxiv, xxv] proposed. The same self similarity metrics concept is also utilized in [xxvi]. Self similarity metrics as a biometric signature in object tracking is also utilized in [xxvii] as in [xxiii]. Pedestrian tracking using self similarity measures is employed in [xxviii]. The same technique for pedestrian tracking using multiple cameras is employed in [xxix]. Normally self similarity techniques are more suited towards detection and identification. They are not suitable for tracking purposes.

II. Problem Formulation

Numerous works describes the tracking of various objects. As the human being is an intelligent entity, so tracking an object with complex appearance is not a difficult task for human being. But for machines, object detection and object tracking requires complex mathematical models. The accuracy of tracking algorithm is the key problem in object tracking domain.

The central challenge for any tracking algorithm is to discriminate the locked target with the background. In many applications target and the background are not much different in their intensity values. The accuracy of locking is another issue in target tracking. Whenever a target is locked by sending the Lock command, the exact location where the command is sent is not being tracked. Some algorithms lock some other target in the near vicinity of the locked target. In cluttered environment it is very difficult for an algorithm to track an object. Clutter refers to detections or returns from nearby objects, clouds, electromagnetic interference, acoustic anomalies, false alarms, etc. Whenever the template is updated, the new template contains the clutter and the object becomes a secondary part in the template.

A big challenge which has to be faced by the tracking algorithms is the drift of target window with an incoming object in the periphery of target, eventually resulting in loss of the target. For example in a people tracking scenario, a person was tracked by the algorithm and another person enters in the tracking area with velocity greater than the tracked person. The tracking window than drifts with the incoming occlusion and leaves the original target.

Another situation arises with the changing appearance of object that needs to be addressed. Fast moving objects are also a major issue. If the scene illumination changes frequently over time, tracking task becomes more difficult to be achieved. When the intensity of the target changes the template matching algorithm give false peaks which eventually results in the loss of target. All tracking algorithms assume that object motion is smooth with no abrupt changes. If the object size and rotation angles changes with time, it will add difficulty for the tracking algorithm to achieve desired results.

In the images cluttered with fog, haze or any some other type of noise, tracking task becomes difficult to achieve. In a scenario where two or more objects are combined together, tracking algorithms got confused when objects move away. Template window size is difficult to be decided.

Implementation on Digital Signal Processor (DSP) is also an important challenge. Many algorithms are not able to meet the processing requirements of the processor. Optimization is a major constraint in this field. In order to design an algorithm that carefully manages all the issues defined above, a complex design will be required. Complex algorithm designs often need bulk of computational power which is difficult to be achieved in case of DSP processors.

This paper addresses all the problems mentioned above and helps in increasing the efficiency, accuracy and robustness of previously defined approaches. Edge enhancement effectively increases both the efficiency and accuracy of the proposed scheme. Dynamic range adjustment and threshold operation decreases the object lost probability. The computationally complex process of correlation is optimized by using summed area table. A summed area table is a data structure that effectively and optimally generates the sum of values in a rectangular subset of a grid. Summed area table computes normalization factor for cross correlation in just four addition operation. The effective template updating decreases the chance of template drift and allows the algorithm to work with partial occlusion and clutter.

III. OPTIMIZED OBJECT TRACKING METHODOLOGY

In this section the basic methodology of the research work is described. Figure 1 shows the block diagram of the hardware used in proposed research work. A CCD camera provides NTSC/PAL video input to DSP. Track command and location attributes are provided through a serial channel. The DSP based on the received commands tracks the object and show results on video display.

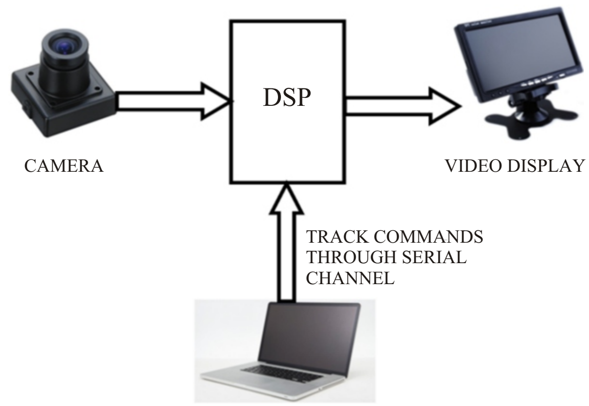

Fig. 1. Block Diagram of Hardware used in Proposed Research Work

Fig. 2 depicts the flow diagram of Edge Enhanced Correlation based Tracking.

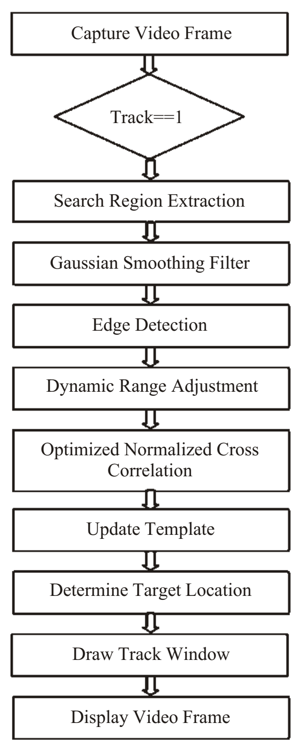

Fig. 2. Flow Chart of Proposed Algorithm

A. Search Region Extraction

To minimize the computational complexity instead of searching the object in the whole frame, a search region is extracted from the video frame, near the vicinity of object. The size of the search window should be kept smaller to reduce the computational power and to avoid false correlation peaks that may arises due to noise in the background.

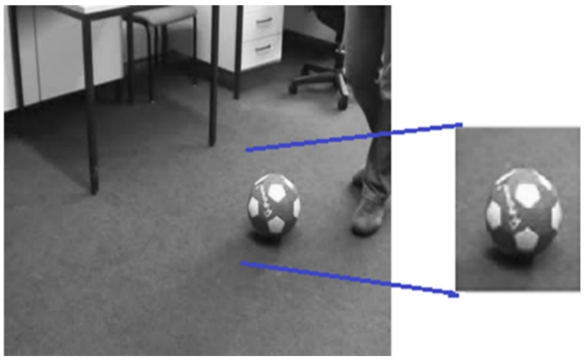

Fig. 3. Search region Extraction

B. Gaussian Smoothing Filter

The search image should be smoothed by applying some smoothing filter. The Gaussian Smoothing before applying edge detection algorithm typically reduces level of noise in the image. This effectively enhances the performance of gradient algorithms. Apply ?x? Gaussian filter for smoothing undesirable artifacts. Gaussian filter minimizes the rise and fall time. Smoothing will reduce undesired edges to be extracted in the edge image. Gaussian filter is normally used with edge extraction algorithms. Most algorithms for detecting edges are sensitive to noise so the noise should be removed prior to edge detection.

Standard deviation of the Gaussian function determines the degree of smoothing. Smoothing increases signal to noise ratio. Standard deviation of 1.85 is used in the proposed technique. Fig. 4 shows the result of applying a Gaussian smoothing filter an image.

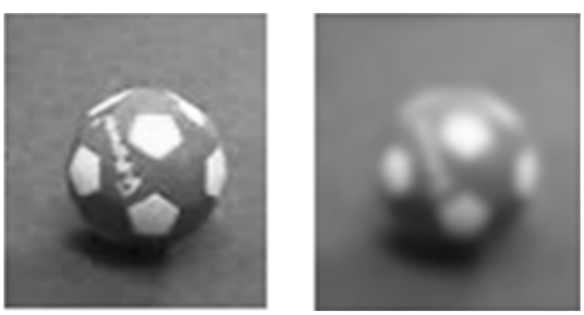

Fig. 4. Gaussian Smoothing

C. Edge Detection

Edge detection is the process of identifying points at which image intensity changes abruptly. The purpose of edge detection is to identify important events. Various techniques are available for edge detection. The well known horizontal and vertical sobel masks are applied to the smoothed image.

$$L_v(x,y) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * L \quad (1)$$

$$L_h(x,y) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * L \quad (2)$$

The magnitude of edge image is than calculated by the following equation:

$$L(x,y) = |L_h(x,y)| + |L_v(x,y)| \quad (3)$$

D. Dynamic Range Adjustment

The dynamic range of edge image is towards darker side in the resulted image obtained after applying edge detection. The available range is from [0,255]. Low contrast imagery subjects more to dynamic range errors. For increasing the robustness of edge detection, normalization process is used which is given by:

$$L_n(x,y) = \left(\frac{255}{L_{max} - L_{min}}\right)\{\, L(x,y) - L_{min} \,\} \quad (4)$$

This histogram of the image is now linearly stretched in whole dynamic range [0, 255] by this normalization process. The contrast of the image effectively enhances.

The next step to dynamic range adjustment is thresholding. The threshold process decides whether the edge point is a strong edge which is to be added to the edge image. Weak edges are discarded from the edge image after applying threshold process. If a low threshold value is selected than more edges will be detected and the resulting image is more susceptible to noise. If a high value of threshold value is selected than there is a high risk of missing subtle edges. So threshold value should be selected with care. After experiments threshold is selected to be □ = 50, which removes the edges produced by noise and artifacts. Thresholding is specified by:

$$L_{nt}(x,y) = \begin{cases} L_n(x,y) & if\ L_n(x,y) > \\ 0 & otherwise \end{cases} \quad (5)$$

Where L_{nt} is the normalized and thresholded image. Fig. 5 depicts the result of applying edge detection and dynamic range adjustment to the ball example.

Fig. 5. Edge Extraction

E. Optimized Normalized Cross Correlation

Cross correlation is the measure of similarity of two images. The size of search window should be greater than the size of template. Consider a template T and a search window SW. Match every template-size portion of search window with template to calculate the position of target in the current video frame. There are many template matching schemes available. Here Normalized Cross Correlation is used. Normalized Cross correlation of the template T(x,y) and SW(x,y) is given by:

$$c = \frac{\sum_{i=o}^{W-1}\sum_{j=0}^{H-1} SW(m+i,n+j)T(i,j)}{\sqrt{\sum_{i=0}^{W-1}\sum_{j=0}^{W-1} SW^2(m+i,n+j)}\sqrt{\sum_{i=0}^{W-1}\sum_{j=0}^{W-1} T^2(i,j)}}$$

A correlation surface is built after applying normalized cross correlation between target and search window. Locate the maximum value c_{max} in the correlation surface. The location where the maximum value found is the best match location located at the top left corner of target.

F. Template Update

In every video frame whenever a best match is found, the target image must be updated. Template drift problem can also be avoided by proper template update. The objects may changes size, shape and orientation during tracking so template updating process should be intelligent enough to accommodate the changes in object shape. If the template is updated in every frame and new best matched object is taken as the new template than it may leads to template drift. If the new template is corrupted with clutter, than the results become more severe.

Here we propose a more advanced template updating scheme which uses two coefficients, β and (1-β). β Can be defined as $\beta = \lambda c_{max}$, where λ is recommended to be in the range [0.0, 0.3].

$$T[n+1] = \begin{cases} \beta B_n + (1-\beta)T_n & if\ c_{max} > 0.84 \\ T[n] & otherwise \end{cases} \quad (6)$$

If maximum correlation value c_{max} is greater than 0.84, than updated template is the weighted sum of current best match times β and previous template times (1-β). If c_{max} is less than or equal to 0.84 than template is not supposed to be updated and the previous template is taken as the new template.

G. Determine Target Location

The top left corner of the search window is represented by x_T and y_T in the frame, than the location of center of the current best match is determined with respect to top left coordinates of frame which are (0 , 0).

IV. EXPERIMENTATION RESULTS AND DISCUSSION

The algorithm is tested on various data sets and real world image sequences in order to test the effectiveness and accuracy of proposed object tracking scheme. Same datasets are also tested on a classic technique[xxx] having some motion segmentation and blob detection algorithms applied. MATLAB is used to test the datasets. Other factors like frame rate and CPU load are obtained from the implementation on a development platform DM6437 EVM. TMS320DM6437 is a fixed point multimedia processor. The DSP is operating at 600 MHz with a performance of up to 5600 million instructions per second (MIPS). 128 Mbytes of RAM is provided. DSP/BIOS operating system is operating on the processor [xxxi].

The issue of evaluating the performance of video object tracking algorithm is becoming more and more important. Any tracking algorithm can be evaluated by measuring the performance of algorithm based on different performance metrics. To understand performance metrics, we should first know what qualities we expect from an ideal object tracking algorithm. It should estimate position of object precisely [xxxii].

Processing time is one of the widely used characteristic of performance metric. Processor utilization is the percentage of DSP resources that an algorithm can use [xxxiii]. By increasing the window size the CPU load can be computed by Real Time Analysis (RTA) Tool. The CPU load is the percentage of time in which the algorithm is not working in the idle loop. The processing speed of digital signal processor can fall below the frame rate display of any video device during peak CPU loading that result in missing frames. When the processor becomes overloaded than frame dropping can occur. Table I illustrates the CPU percentage for varying target window sizes in proposed and classic techniques.

TABLE I
CPU LOAD PERCENTAGE COMPARISON

Window Size	Proposed Technique	Classic Technique
20 x 20	20%	46%
25 x 25	23%	48%
30 x 30	25%	50%
35 x 35	28%	54%
40 x 40	30%	58%
45 x 45	33%	62%
50 x 50	36%	63%
55 x 55	38%	67%
60 x 60	41%	70%

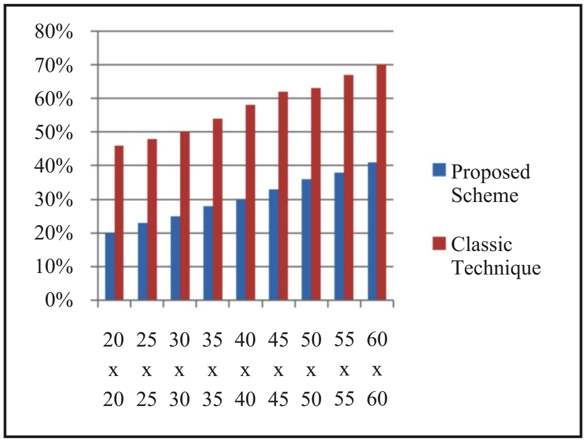

Fig. 6. CPU percentage on various window sizes

The processing time is calculated by toggling a GPIO of DSP and measuring the execution time on oscilloscope. The loop time of the DSP is 40ms so it achieves the required time for 70 x 70 target size. But the other technique which is lacking in optimization did not meet the required criteria. The Table II shows the comparison of the execution time for varying window sizes.

TABLE II
COMPARISON OF PROCESSING TIME ON DIFFERENT WINDOW SIZES

Window Size	Proposed Technique	Classic Technique
20 x 20	40ms	40ms
25 x 25	40ms	40ms
30 x 30	40ms	40ms
35 x 35	40ms	43ms
40 x 40	40ms	44ms
45 x 45	40ms	45ms
50 x 50	40ms	47ms
55 x 55	40ms	47.5ms
60 x 60	40ms	49ms

Frame rate is the rate at which frames are captured, processed and displayed. For a target window size of 60 x 60 the proposed scheme provides 30 frames per second frame rate and the classic one provides 20 frames per second. The minimum frame rate required to achieve a flicker less video is 25 frames per second so the classic object tracking algorithm fails to achieve a real time video effect.

The main goal of tracking algorithm is to find the trajectory of object in a video scene precisely. Accuracy is another measure to evaluate the object tracking algorithms. The algorithm should be able to determine the exact location of object and the template drift should be minimized to achieve the goal of accuracy.

A comparison of number of instructions required to perform the algorithms for different window sizes are listed in the Table III. It is clear from the Table III that the classic algorithm requires more instructions than the available number of instructions.

TABLE III
COMPARISON OF INSTRUCTIONS PER FRAME

Window Size	Proposed Technique	Classic Technique
20 x 20	201212542	321235610
25 x 25	232111232	356412010
30 x 30	302145652	430225412
35 x 35	322254120	462023015
40 x 40	365015542	492301224
45 x 45	392500330	532145697
50 x 50	410213565	590001213
55 x 55	436542021	652120312
60 x 60	521453562	692120012

The memory requirement is another important feature for object tracking algorithms that are to be implemented on any processor. Buffering frames in memory requires substantial amount of memory. A PAL standard image is of 720 by 576 pixels and requires 1.2 Mbytes of memory to store a single frame. In DSP for real time operation three frames are saved at a time so that when DSP finishes processing on a frame it doesn't starve for the upcoming frame. The frame must be available in memory beforehand so that real time requirements should be achieved. For storing three frames 3.5 Mbytes of memory is required. The proposed scheme requires more memory because of the application of Gaussian smoothing filter and edge enhancement operations. The memory requirements for both the techniques are given in the Fig 7.

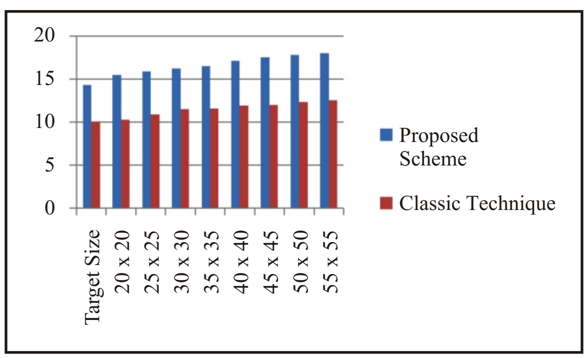

Fig. 7. Memory Requirements on different Window Sizes

A number of datasets for evaluating the performance is taken from "BoBoT (Bonn Benchmark on Tracking)". All the datasets are licensed under Creative Common License. The datasets are tested in the MATLAB simulation. All the datasets are of 320 x 240 resolution and provides 25 frames per second. They are of MPEG2 encoding.

The proposed algorithm is applied on a dataset in which a football is passed to each other by two players. The dataset contains 602 frames. The video sequence is comprised of an indoor activity in which the camera is moving. The target is moving, rotating and achieves fast speed at certain locations. Fig. 8 shows the results. The inner white box shows the template and the outer white box depicts the search area. The proposed object tracking algorithm is accomplished with an advanced technique for template updation which reduces the template drift and make the tracker accurate and efficient. The algorithm is able to track the football even when it moves very fast. Whereas when the same dataset is tested with the classic technique with simple template updating and edge detection schemes, the algorithm suffered from template drift and when the target moves fast the track is lost. The results are depicted in the Fig. 9.

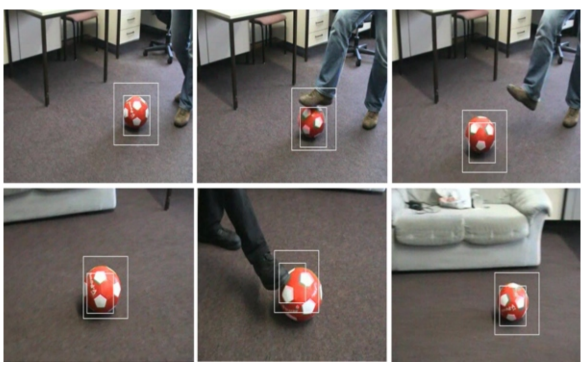

Fig. 8. Result of proposed algorithm on dataset 1 with fast moving target

Fig. 9. Result of a previous algorithm on dataset 1 on fast moving target

Fig. 10 shows the trajectory of target followed by both the schemes. Original target trajectory is also given. It is clear from the figure that the trajectory of target calculated by the proposed scheme is much closer to the original one.

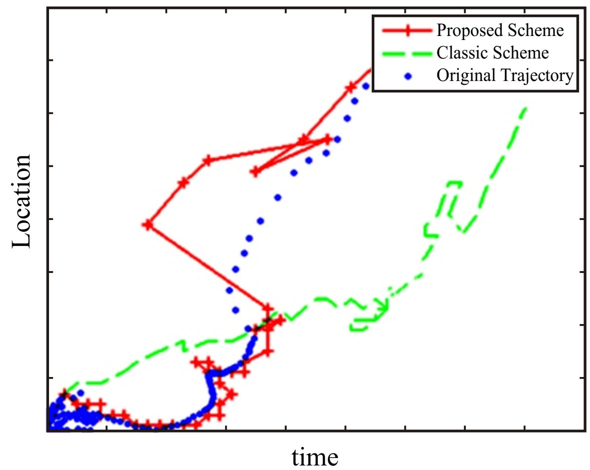

Fig. 10:. Target Trajectory

Another dataset contains a blue colored mug. The mug is moving around changing backgrounds. The contrast of the object and background is almost same at certain locations. The camera is moving and the target is also moving. The background is changing and the scale of the object is also changing. So that would be a challenging task for any tracking algorithm to track the object in such a varying environment. This dataset is passed through both the algorithms. The proposed algorithm succeeded in tracking the mug along the way in 629 frames as shown in Fig 11. But when the dataset is tested on the other technique the algorithm fails to track and the track is lost when the blue mug is passed from the low contrast background. The results are shown in Fig. 12.

Fig. 11. Results of proposed algorithm on dataset 2 with changing scale and cluttered background

Fig. 12. Results of previous algorithm on dataset 2 with changing scale and cluttered background

The algorithm is also tested in foggy environment. An environment is chosen that contains cars moving in a row. Car tracking example is shown in Fig. 13. The forth most car is chosen as a target. The car moves along the road. On the way the car subjects to various other movements other than straight motion, which changes shape, orientation and size of car. The proposed scheme continues to track when the car changes its orientation angle. Whereas the classic technique in Fig. 14 is failed to track the car as soon as it starts turning and started changing size and shape.

Fig. 13. Results of proposed algorithm on foggy dataset 3

Fig. 14. Result of previous algorithm on foggy dataset 3

Another outdoor example is taken in which a person in moving on the road. The object is moving, camera is also moving in the scenario. The target is non-rigid and changing its orientation. The color of shirt of the person is almost same as that of background. When tested with the proposed scheme the person is tracked accurately as shown in Fig. 15. But with the classic tracker there is an adverse effect of template drift. The template gradually drifts off the target as soon as the person starting turning the position.

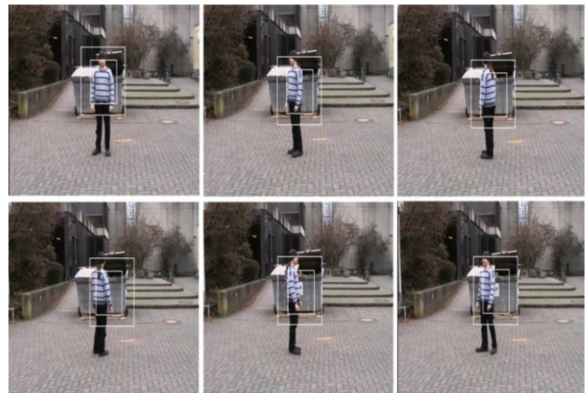

Fig. 15. Result of proposed algorithm on dataset 4 with non rigid object changing orientation

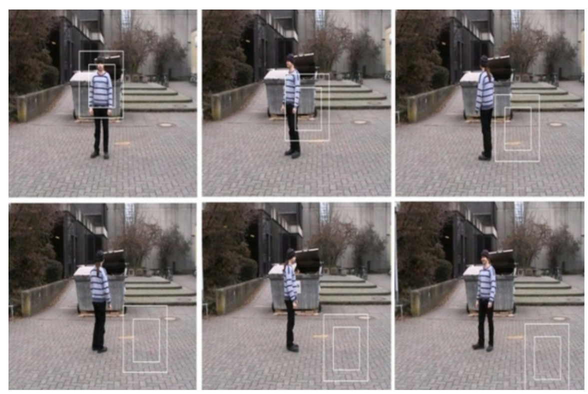

Fig. 16. Result of previous algorithm on dataset 4 with non rigid object changing orientation

An outdoor example of partial occlusion where the person is standing stationary and the camera is moving. The object become partially occluded behind the pillar and appears again. The proposed algorithm keeps on tracking the tracking on the way along partial occlusion. It handles the partial occlusion efficiently. The advanced template update technique allows the object to be tracked in partial occlusion. The other technique fails to handle the partial occlusion and loses the track. The results are clear from the Fig. 17 and 18.

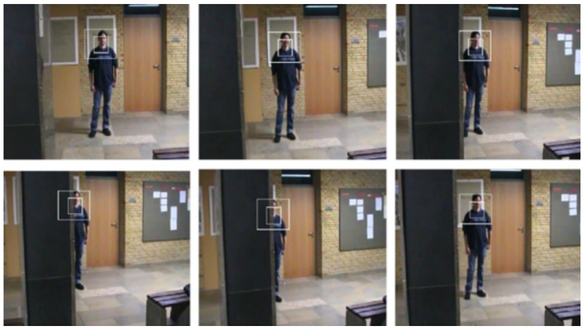

Fig. 17. Results of proposed algorithm on dataset 5 with partial occlusion

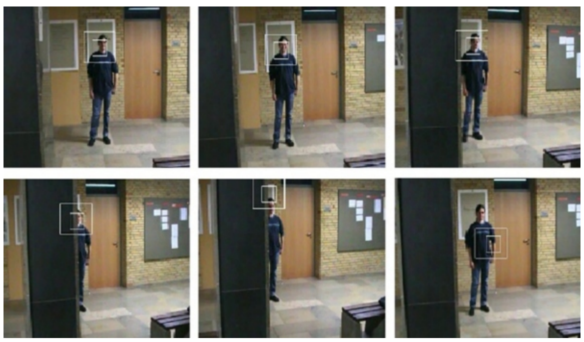

Fig. 18. Results of previous algorithm on dataset 5 with partial occlusion

V. CONCLUSION

An accurate, efficient and robust real time object tracking algorithm is introduced in this research. The algorithm can be implemented with moving camera and on different field of views. The algorithm is implemented by using various image processing schemes. First of all Gaussian smoothing is applied to minimize noise and artifacts. Edges are extracted from the smoothed image using Sobel edge detector. Dynamic range adjustment is performed. Normalization and thresholding are following steps. Correlation process is optimized by Summed area table (SAT). An effective and robust model for template updation is provided. The algorithm is compared with another object tracking technique. The results prove that the proposed object tracking technique is more accurate, efficient and more computationally compact. The proposed algorithm is more invariant to varying illumination conditions, and is performing well in presence of background clutter. It also provides effective results when shape and size of the object subjects to change. Template drift is minimized in the proposed scheme. Furthermore, in presence of fog and haze the algorithm proves itself to be efficient. Accuracy is increased.

In future Multiple Object Tracking can be exploited. Multiple object tracking is the process of tracking multiple objects simultaneously. Objects might move in different directions and with different velocity. Targets may differ in their appearance and other characteristics. Clear discrimination of objects must also be carried out in multiple object tracking scenario. Some objects might appear for short period of time and some might appear for longer period. If one object gets lost than the tracker will not leave the track of other targets.

Automatic object detection is another major feature for object tracking. Whenever a moving object comes in periphery of vision the moving object will automatically be detected and tracked by the tracker. There are various object detection techniques available like background subtraction, training HAAR cascade, SIFT and SURF keypoints, template matching and optical flow.

Occlusion detection and avoidance will also be a great achievement in this perspective. For example in people tracking scenario a person is moving on the road and another person or any other object comes in between tracked person and camera than occlusion occurs. This occlusion should be detected and avoided. An estimation and prediction technique should be used for this purpose. The tracker will track the object using template matching technique when the object is visible to it. Whenever the object will hide behind the occlusion the tracker shifts to the prediction and starts predicting the next location of target until object come in vision again.

REFERENCES

[i] A. Neri, S. Colonnese, G. Russo, P. Talone, “Automatic moving object and background separation”, Signal Processing, vol 66 , pp. 219-232, 2001

[ii] B. K. P. Horn and B. G. Schunck, “Determining Optical Flow”, Artificial Intelligence, 17(3), pp. 185-203, 2000

[iii] A. Reid, V Markandy and S. Wang, “Global Optimum Multiple Object Tracking”, University of Southern California, Vol. AES 32, No.2, 2008.

[iv] F. Bashir, F. Porkli, “Performance Evaluation of Object Detection and Tracking Systems”, Mitsubishi Electric Research Laboratories, TR2006-041, June 2006.

[v] J. P. Lewis “Fast Normalized Cross Correlation”Industrial Light & Magic 1995.

[vi] A. J. Lipton, H. Fujiyoshi, R. S. Patil, “Moving Target Classification and Tracking from Real-time Video”, The Robotics Institute, Carnegie Mellon University, 2002

[vii] P. F. Gabriel, J. G Verly, J. H. Piater, A. Genon, “Multiple Object Tracking Under Occlusion in Video Sequences”, Advanced Concepts for Intelligent Vision Systems (ACIVS), 2003.

[viii] Z. ZHOU, J. ZHANG, L. FANG, "Object Tracking Based on Dynamic Template and Motion Prediction"Journal of Computational Information System, vol. 5, pp. 1841-1846, 2009.

[ix] S. Q. Abhari, "Target Tracking Based on Mean Shift and KALMAN Filter with Kernel Histogram Filtering",Academin Journal of Computer & Information Science, vol.4, Issue 2, pp. 152, March 2011

[x] S. Avidan. "Support vector tracking", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 26 no. 8, pp. 1064-1072, August 2004.

[xi] R. T. Collins,Y. Liu and M. Leordeanu. "Online selection of discriminative tracking features", IEEE Trans on Pattern Analysis and Machine Intelligence, vol. 2, issue 10, pp. 1631-1643, Oct 2005.

[xii] H. Grabner and H. Bischof, "Online boosting and vision", IEEE Trans on Computer Vision and Pattern Recognition, vol. 1, pages 260267, June 2006.

[xiii] M. J. Black and A. D. Jepson, "Robust matching and tracking of articulated objects using a view-based representation", International Journal of Computer Vision, vol. 2, issue 1, pp. 6384, 2010.

[xiv] K. C. Lee and D. Kriegman, "Online learning of probabilistic appearance manifolds for video-based recognition and tracking". In Proc. Conf. Computer Vision and Pattern Recognition, vol. 2, Sep 2011.

[xv] W. Gao, L. Yang, X. Zhang, H. Liu "An Improved Sobel Edge Detection," IEEE Int. Conference on Computer Science and Technology, pages 67-71, 2010

[xvi] Z. Jin-Yu "Edge Detection of images based on improved Sobel operator and genetic algorithms," IEEE International Conference on Image Analysis and Signal Processing Taizhou, pp. 31-35, April 2009.

[xvii] H. Kapoor, P.Singla, "Magnified Edge Detection using Fuzzy-Canny Logic", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, pp. 376-378, May 2012.

[Xviii]F. Arandiga, A. Cohen, R. Donat, B. Matei, "Edge detection insensitive to changes of illumination in the image", Image and Vision Computing 28 (2010), pp. 553-562, 2009

[xix] B. Babenko, M. H. Yang, S. Belongie"Robust Object Tracking with Online Multiple Instance Learning", IEEE Trans on Pattern Analysis and Machine Intelligence, vol. 33, pp. 1619-1632, August 2011.

[xx] S. L. Tang, Z. Kadim, K. M. Liang, M.K. Lim "Hybrid blob and particle filter tracking approach for robust object tracking", Proc. Computer. Sci. vol. 1, pp. 2549-2557, 2010

[xxi] Q. Zhong, Z. Qingqing, G. Tengfei, "Moving object tracking based on code book and particle filter", Proc. Eng, vol. 29, pp 174-178, 2012

[xxii] E. Shechtman, M. Irani "Matching local self-similarities across images and videos", IEEE Conference on Computer Vision and Pattern Recognition, pp. 18, 2007

[xxiii]C. Ben Abdelkader, R. Cutler, H. Nanda, L. Davis: "Motion-based recognition of people using image self-similarity" Technical report 2001

[xxiv] Dexter, E., Junejo, I., Pérez, P., Laptev, I., "Cross-view action recognition from temporal self similarities" Proceedings of the European Conference on Computer Vision 2008

[xxv] I. N. Junejo, E. Dexter, I. Leptev,P. Pérez, "View-independent action recognition from temporal self-similarities", IEEE Trans. Pattern Anal. Mach. Intell, vol. 33(1), pp. 172-185, 2011

[xxvi] P. Pérez, E. Dexter, I. Laptev, "Multi-view synchronization of human actions and dynamic scenes", Proceedings of the British Machine Vision Conference 2009, British Machine VisionAssociation, pp. 122.1122.11, 2009

[xxvii]M. Rani, G. Arumugam "An efficient gait recognition system for human identification using modified ICA" Int. J. Comput. Sci. Inf. Technol, vol. 2(1), pp. 55-67, 2010.

[xxviii]S. Walk, N. Majer, K. Schindler, B. Schiele"New features and insights for pedestrian detection" IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1030-1037, 2010

[xxix] Y. Cai, M. Pietikäinen"Person re-identification based on global color context" Asian Conference on Computer Vision Workshops, 2011.

[xxx] M. Narayana "Automatic Tracking of Moving Objects in Video for SurveillanceApplications" Postgraduate Thesis, School of CS, UMASS, 2007

[xxxi] TMS320DM6437 datasheet

[xxxii]F. Yin, D. Makris, S. Velastin., "Performance Evaluation of Object Tracking Algorithms", Digital Imaging Research Center, vol . 1, 2010

[xxxiii]T. Ellis, "Performance Metrics and Methods for Tracking in Surveillance", Third IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, June, Copenhagen, Denmark, 2002, pp 26-31.