Computer Vision Inspired Real-Time Autonomous Moving Target Detection, Tracking and Locking

Javed Iqbal[1], Syed Mustafa Pasha[1], Khelifa Baizid[2], Abdul Attayyab Khan[3], Jamshed Iqbal[1*]

[1]Department of Electrical Engineering, COMSATS Institute of Information Technology, Islamabad, Pakistan
[2]Dipartimento di Ingegneria Elettrica e dell'Informazione, Universita di Cassino e del Lazio Meridionale, Via Di Biasio 43, 03043 Cassino (FR), Italy
[3]Department of Electrical Engineering, King Faisal University (KFU), Kingdom of Saudi Arabia
[*] jamshed.iqbal@comsats.edu.pk

Abstract: Moving object analysis is one of the most important and challenging research subjects in computer vision, video processing and robotics. This analysis is three folds: object detection followed by its tracking and locking. Today's applications demand object analysis in real time, which further complicates the problem. Autonomy is another desired feature of such a system since manual handling of moving targets with a camera platform requires constant attention of humans thus making the tracking task cumbersome, time consuming and erroneous. To addressing these challenges, this research paper presents a complete system for handling moving objects i.e. from their detection to locking. Detection and tracking has been achieved using frame differencing and mean-shift algorithms whereas locking is accomplished using the developed control system. The presented strategy has been validated in simulation environment as well as using a custom-developed hardware platform. The prototype consists of an on-board camera, pan-tilt system and a wooden assembly. The performance of the algorithms has been tested based on recorded video as well as on real time live streaming. Experimental results demonstrated that the system has ability to track moving objects efficiently. The proposed system finds application for simple object detection to complex multiple tracking in various industries like transport, military, sports etc.

[Javed Iqbal, Syed Mustafa Pasha, Khelifa Baizid, Abdul Attayyab Khan, Jamshed Iqbal. **Computer Vision Inspired Real-Time Autonomous Moving Target Detection, Tracking and Locking**. *Life Sci J* 2013;10(4):3338-3345]. (ISSN:1097-8135). http://www.lifesciencesite.com 443

Keywords: Motion estimation; Target tracking; Computer vision; Image processing; Robotics

1. Introduction

Moving target detection and tracking is a recent focus of researchers working in vision and image processing. Autonomous target handling in real time is a challenging task. Numerous domains require implementation of this task. These domains include but are not limited to: human detection and tracking, sports analysis, security in commercial and public domains, military purposes, vehicle detection and tracking, traffic control, air surveillance and Human Machine Interface (HMI). Consider a typical room having a specific seating capacity. An automatic detection and identification system can permit entry of certain number of people and can then close the doors automatically. Pedestrian detection and tracking and observation of athletic performance are other relevant examples. Moving object detection can facilitate sports analysis e.g. in a football match the ball or a player can be tracked easily both in the recorded video and the real-time scenarios. Military applications of such a system include aircraft detection and tracking, guided missiles tracking. In transport industry, automatic identification of trajectory, posture, speed and direction of a vehicle can be determined using moving object analysis. Automatic vehicle detection, accurate identification and tracking can be used to optimize traffic management system. Moving target detection system can be installed in an aircraft for air surveillance and object detection and tracking on ground. Patrolling of highways and railways are also examples of air surveillance using moving target detection system. Similarly, security of commercial and public places is another important sector of such systems' applications.

Detection and tracking in real time is divided into three modes: autonomous, semi-autonomous and manual. It is very difficult for humans to track moving object manually and monitor it constantly. To address these issues in diverse areas of applications, a platform is presented where detection and tracking systems function reliably in an autonomous manner.

The remaining paper is outlined below: Section 2 presents a comparative review of reported algorithms for target detection and tracking. Section 3 details the implemented algorithms. Developed simulation environment and custom-developed hardware prototype are highlighted in Section 4. Section 5 discusses the experimental results. Finally, Section 6 comments on conclusion.

2. Comparative Review of Algorithms

Target analysis has two primary parts i.e. detection and tracking. In order to investigate motion of a body, the first step is to detect the object reliably and efficiently. Based on this detection, the object is then tracked. Locking is an automatic recursive version of tracking in real time. Scientific literature reports numerous techniques and algorithms for moving target analysis. Figure 1 summarizes the state-of-the-art reported algorithms in hierarchal form.

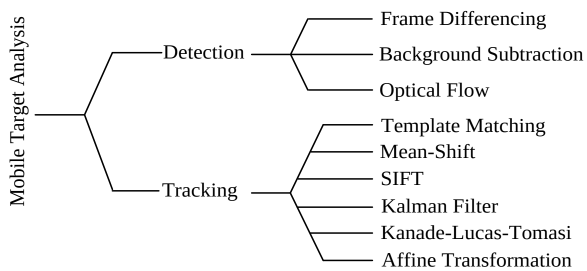


Figure 1. Target handling algorithms

Detection Algorithms: To accurately and efficiently detect moving targets, three algorithms are more popular: Frame Differencing (FD) [1], Background Subtraction (BS) [2] and Optical flow [3]. Each algorithm has its own applicability, specifications, complexity level and limitations. FD algorithm has good detection results but it is very sensitive to light intensity variations and multiple objects. BS algorithm works perfectly under static camera conditions. However, it is very sensitive to background changes. Optical flow method is an efficient detection method under ideal conditions but in real time scenario, it is computationally expensive and is very sensitive to noise. Moreover, its anti-noise performance is very poor.

Tracking Algorithms: Common tracking algorithms mentioned in the literature include Template Matching (TM) [1,4], Mean-Shift (MS) tracking [5], Scale Invariant Feature Transform (SIFT) tracking [6], Kanade-Lucas-Tomasi (KLT) tracker [7], Affine Transformation [8] and Kalman Filtering [9]. TM uses mathematical correlation between the target template and the source image. TM has good tracking results if the target template is completely present in the source image. MS algorithm performs tracking using kernel function and Bhattacharyya coefficients to compute similarity measure and has good tracking results. SIFT finds out features that are invariant to scaling, rotations and light conditions. However, it suffers from implementation complexity especially in real time. KLT tracker is dependent upon optical flow field of the image but is very sensitive to noise and requires huge computations. Kalman filtering removes noise from the resultant image obtained from detection algorithm. More detailed description of tracking algorithms has been reported by Iqbal et al. in [10].

3. Detection and Tracking Algorithms

The present work considers two algorithms: FD and MS. FD algorithm has been chosen because of its simplicity in implementation and strong adaptability to variety of dynamic environments. MS algorithm having high-level complexity in implementation, works effectively when there are multiple targets and we are interested to detect and track a specific target only. These algorithms are detailed below:

Frame Differencing (FD) Algorithm: In FD algorithm, two consecutive frames are taken and compared with each other within certain threshold. Objects that have changed their position with respect to previous frame are treated as moving objects. After image-acquisition (from a recorded video or live streaming) and pre-processing, two consecutive frames are subtracted. Considering two consecutive frames i.e. current frame I_n (frame at time t) and a previous frame I_{n-1} (frame at time t-1). The difference between these two frames is then

$$D = I_n - I_{n-1} \qquad (1)$$

The resultant image obtained after difference operation is then converted to binary image using threshold value (T).

$$B = \begin{cases} 1 & D(x,y) \geq T \\ 0 & \text{elsewhere} \end{cases} \qquad (2)$$

where B is binary image, D(x, y) is the difference at pixel (x, y) and T is threshold value. The ideal value of threshold is 10%. Pixels having values greater than or equal to the pre-defined threshold are turned white and those having less values are turned black. White region is considered as moving objects while black region is taken as background. Applying a median filter of size 3x3 to remove noise and refining the image by performing morphological operations result in object detection. The detected objects are then passed to tracking function to track the moving object. This is done by enclosing the object in a bounding box. Finally, the trajectory of the tracked object is plotted. Figure 2 presents the flowchart of FD algorithm.

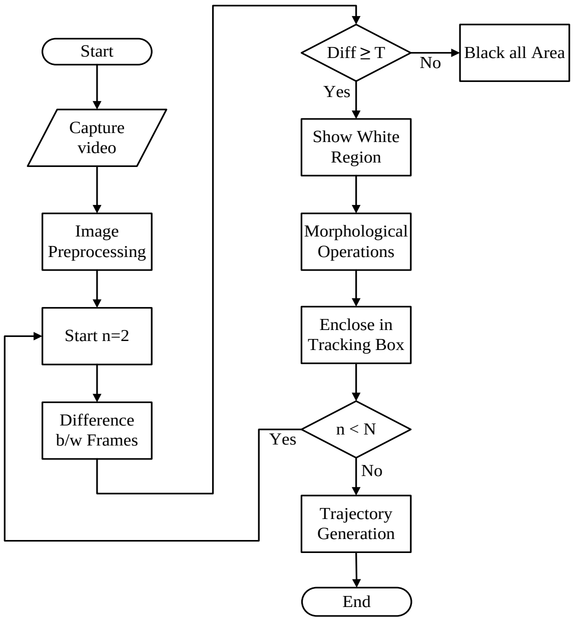


Figure 2. Flowchart of FD algorithm

Mean-Shift (MS) Algorithm: MS algorithm, widely used for moving object tracking, is based on statistical calculations of colour in the target model and the candidate model. Bhattacharyya coefficients are used to measure the similarity. The algorithm iteratively shifts a data point to the average value of points in its neighbourhood. It uses the colour histogram information and statistical calculations to locate the new position of the object. The algorithm uses Kernel function *K(x)* that helps to estimate the mean value. Classically, kernel K is a function of $\|x\|^2$. The simple mean (*m*) at point x with kernel *K(x)* is calculated using (3)

$$m(x) = \frac{\sum_{i=1}^{n} K(x - x_i)x_i}{\sum_{i=1}^{n} K(x - x_i)} \tag{3}$$

The difference calculated by the term *m(x)-x* is known as the mean shift. The mean shift tracking based on colour histogram is divided into three basic steps. In the first step Probability Density Function (PDF) of the target model qu and the candidate model pu is calculated on the basis of colour histogram and kernel density estimation as given by (4) and (5) respectively.

$$q_u(x) = C_q \sum_{i=1}^{n} k\left(\left\|\frac{x_i}{h_q}\right\|^2\right) \delta(b(x_i) - u) \tag{4}$$

$$u = 1,2,\dots,m$$

$$p_u(y) = C_p \sum_{i=1}^{N_p} k\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right) \delta(b(y_i) - u) \tag{5}$$

$$u = 1,2,\dots,m$$

where $b(x_i)$ and $b(y_i)$ represent the histogram index function of pixels. h_p and h_q are radii of the candidate and target model kernels respectively and δ is the Kronecker delta function (6).

$$\delta(a) = \begin{cases} 1 & if\ a = 0 \\ 0 & otherwise \end{cases} \tag{6}$$

Thus, if $b(x_i) = u$, the PDF will have some value otherwise kernel will contribute zero value to q_u and p_u. C_q and C_p are normalization constants and have values given by (7) and (8) respectively.

$$C_q = \left[\sum_{i=1}^{n} k\left(\left\|\frac{x_i}{h_q}\right\|^2\right)\right]^{-1} \tag{7}$$

$$C_p = \left[\sum_{i=1}^{N_p} k\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)\right]^{-1} \tag{8}$$

In the second step, similarity between the target model and the candidate model is calculated. For this purpose Bhattacharyya coefficient ρ is calculated. i.e.

$$\rho(y) = \rho[p(y), q] = \sum_{u=1}^{m} \sqrt{p_u(y) q_u} \tag{9}$$

Let *y* denotes the target location of its current position with the corresponding color probability $\{p_u(y)\}$, $p_u(y) > 0$ for u = 1, . . . , m. Let *z* denotes the estimated new target location near *y* where change in color probability does not occur rapidly. Using Taylor expansion, (9) can be reduced to

$$\rho[p(z), q] \approx \frac{1}{2}\sum_{u=1}^{m} \sqrt{p_u(y) q_u} + \frac{1}{2}\sum_{i=1}^{N_p} w_i k\left(\left\|\frac{z - y_i}{h_p}\right\|^2\right) \tag{10}$$

where

$$w_i = \sum_{u=1}^{m} \delta(b(y_i) - u) \sqrt{\frac{q_u}{p_u(y)}} \tag{11}$$

Finally, the new position 'z' of the object in the candidate model is calculated in the neighbourhood of the previous target position y_i with the help of similarity measure and the object is tracked in the candidate model. The new position *z* is calculated using (12).

$$z = \frac{\sum_{i=1}^{N_p} y_i w_i g\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)}{\sum_{i=1}^{N_p} w_i g\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)} \quad (12)$$

where $g(x) = -k'(x)$. Figure 3 presents the flowchart of MS algorithm.

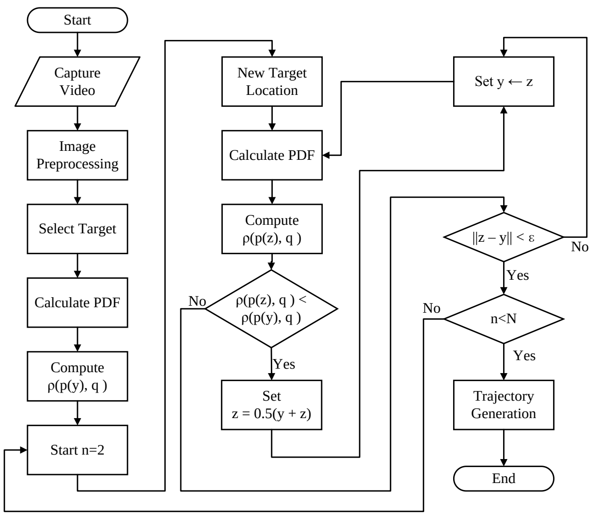


Figure 3. Flowchart of MS algorithm

4. Custom-designed Platforms

To validate the presented approaches for moving object analysis, a simulation environment has been realized followed by the design and development of hardware platform.

Simulation Environment: In simulation environment, a recorded video or live streaming captured from a camera forms the input of detection and tracking algorithms running in MATLAB to detect moving object and to find its location. The object position is found in the form of coordinates (x, y) which is then converted into horizontal and vertical angles of the target motion with respect to the reference point. The calculated angles are then communicated to Proteus through virtual serial port. The microcontroller circuit in Proteus generates the Pulse Width Modulated (PWM) signals depending on the angles provided by the algorithm in MATLAB through virtual serial port. PWM is applied to motors models (in Proteus) to show motion of the pan-tilt system in the direction of the target.

Hardware Platform: The custom-developed platform consists of an on-board camera, a control board, a pan-tilt system, a laser pointer light source and an assembly. A H.264 compatible camera of 15 Mega-pixels resolution (Logitech C920) mounted on a stand has been used to monitor the environment. Live video is captured and processed to detect moving object using the algorithms detailed in Section 3. After successful detection, the object has been tracked by finding its coordinates. The trajectory of the tracked object is plotted. In order to lock the moving object, pan-tilt system is used on which a laser light is mounted that points to the moving object. Horizontal and vertical angles are calculated w.r.t the centre of the image frame. Calculated angles are serially communicated to the developed control board. This board is centered on a AVR microcontroller (ATMega 32) and associated circuitry. The µC generates PWM signals corresponding to the angles obtained. These PWM then finally drive servomotors (Hitech HS422) of pan-tilt mechanism. The block diagram of the developed hardware platform is shown in Figure 4.

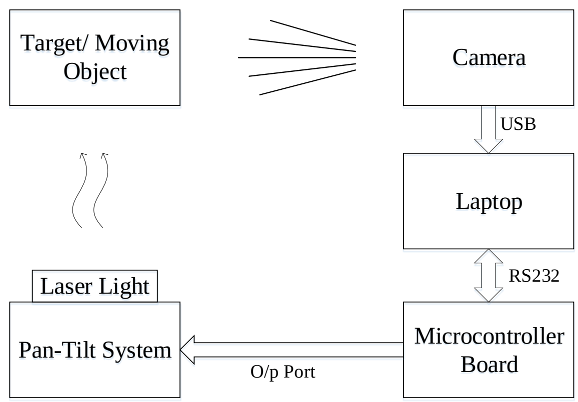


Figure 4. Custom-developed hardware platform

A wooden assembly to house various components of the hardware has been designed and fabricated. The fabricated hardware is presented in Figure 5.

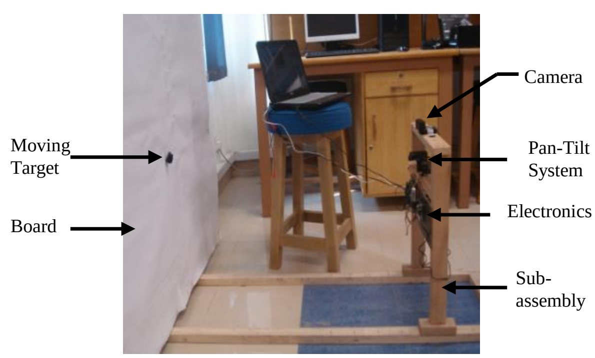


Figure 5. Fabricated hardware prototype

Tracking algorithm gives position of the desired moving object as output. Position obtained is then compared with reference position (centre of the frame) to compute angles. Two angles are interest here i.e. horizontal angle and vertical angle. Horizontal angle shows motion of the Pan motor and vertical angle shows motion of the Tilt motor. Target is locked based on motion of both motors. Trigonometric representation to compute these angles is shown in Figure 6 where (x_0, y_0) is the centre of the frame, (x_1, y_1) is the centre position of the tracked object, D is distance between centre of the frame and centre of the camera, d_x and d_y are distances from reference position to the centre of object along x-axis and y-axis respectively. T_1 and T_2 are horizontal and vertical angles respectively and are calculated using (13) and (14).

$$T_1 = \arctan\left(\frac{d_x}{D}\right) \quad (13)$$

$$T_2 = \arctan\left(\frac{d_y}{D}\right) \quad (14)$$

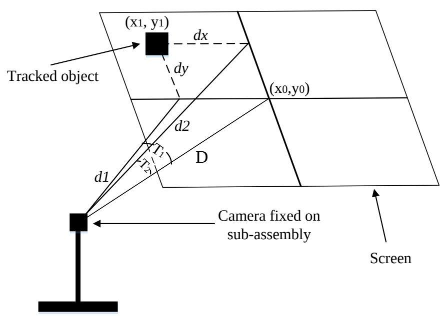


Figure 6. Computation of pan-tilt angles

Let T_x and T_y are previous horizontal and vertical angles respectively. The motion angles (T_{mx} and T_{my}) are calculated using (15) and (16).

$$T_{mx} = T_1 - T_x \quad (15)$$

$$T_{my} = T_2 - T_y \quad (16)$$

where T_{mx} and T_{my} are horizontal and vertical motors angles respectively. These angles are passed to the control board through serial interface that drives the pan-tilt system. The control board generates PWM signals for servomotors motion control. PWM signals are generated using Output Compare Register (OCR) values. The calibration equation for the OCR values is obtained using (17) based on the computed motion angles.

$$val_{OCR} = 2.434 * T_m + 95.978 \quad (17)$$

where T_m is horizontal or vertical motion angle as given by (15) or (16) respectively.

5. Results

Experimental results have been collected from the recorded video as well as from live streaming.

Simulation Results: Using recorded video samples, detection and tracking performance of both algorithms (FD and MS) has been investigated in Simulation. Figure 7 illustrates results of FD algorithm where (a) shows a frame acquired from the original video, (b) indicates the tracked object in the selected frame while (c) represents the path followed by the tracked object in Cartesion coordinates, (d) shows the distance of tracked object from center position of the frame and finally (e) shows the angular position of the tracked object with respect to the center of the frame. Results obtained by applying FD algorithm demonstrate its efficiency and effectiveness to detect the moving object.

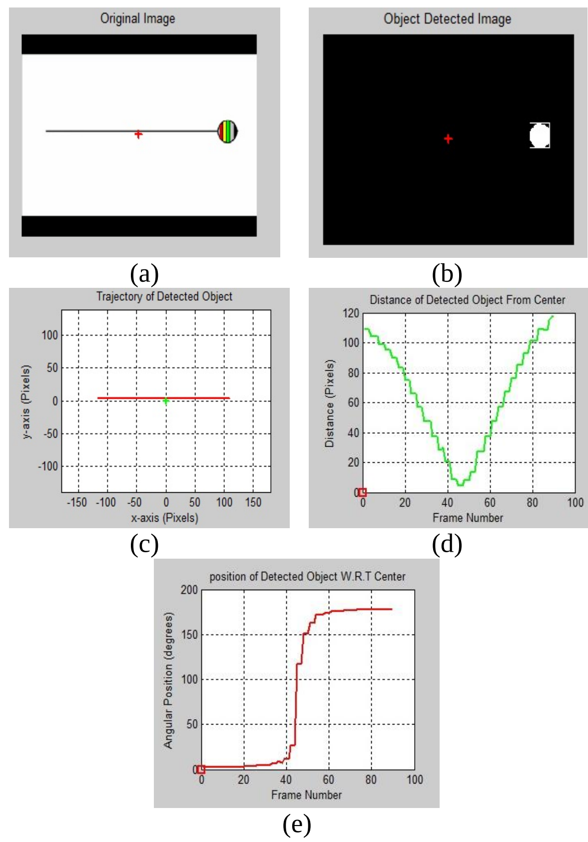


Figure 7(a-e). Simulated results of FD algorithm from a recorded video

Figure. 8 (a-e) presents simulated results of detecting and tracking a moving object in case of MS algorithm. As can be seen in (c) that the algorithm has poor detection performance capacity.

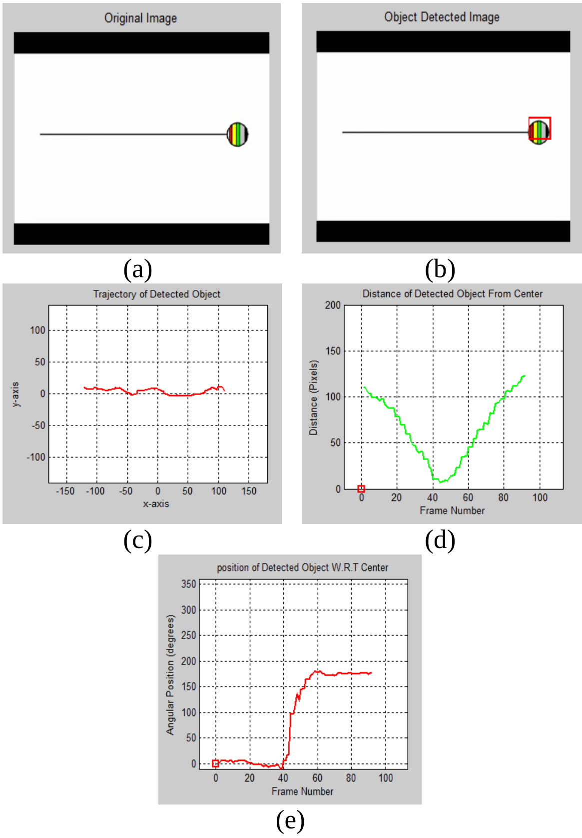
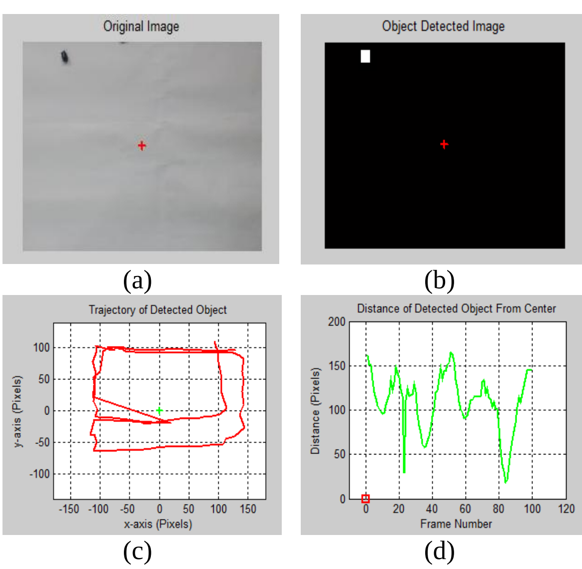


Figure 8(a-e). Simulated results of MS algorithm from a recorded video

Real Time Results From Prototype: Finally, experimental trials have been conducted using the custom-developed hardware platform. A spherical ball has been moved randomly in camera Field of View (FoV). The motion of the ball is detected and tracked in real time. Figure 9 and 10 show corresponding results of FD and MS algorithms.



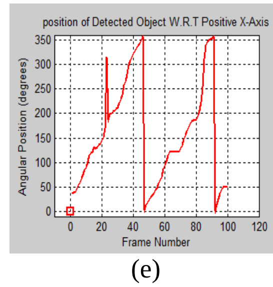


Figure 9(a-e). Results of FD algorithm from a real-time video

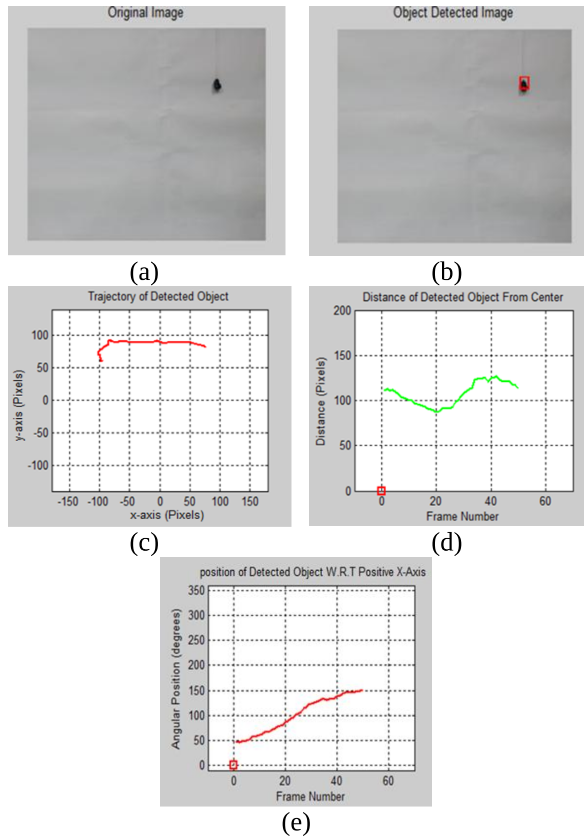
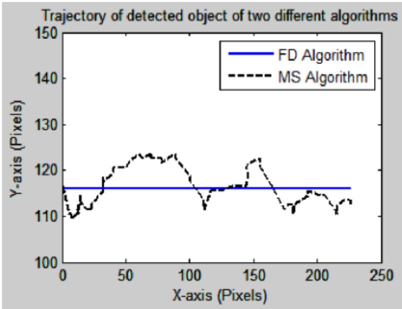
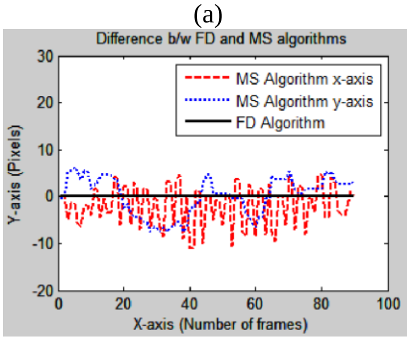


Figure 10(a-e). Results of MS algorithm from a real-time video

Figure 11 presents comparative results of FD and MS algorithms. Performance to detect a moving object in case of both algorithms is highlighted in (a) while (b) shows the difference between MS algorithms trajectory coordinates (x,y) and corresponding coordinates in FD algorithm. Results show that MS is less accurate for detection purposes.



(a)



(b)

Figure 11. Comparison between FD and MS algorithms

6. Conclusion

A complete paradigm for analyzing moving objects in real time autonomous mode has been presented. FD and MS algorithms detect and track the moving target while developed control system locks it. Results from simulation environment and hardware prototype demonstrated that FD algorithm performs more better detection as compared to MS while for object tracking, the later outstands. This consequently dictates that combining good features from each algorithm results in an optimized object analysis strategy. The presented platform opens avenues of research in computer vision and image processing. Further desirable features of such a target analysis platform include object recognition and identification. Sophisticated image processing algorithms based on correlation can be incorporated in the presented platform. Moreover, the research can be integrated in various domains of robotics [11-26] with minor modifications.

Acknowledgements:

This research was funded by National Information & Communication Technology (ICT) R&D Fund under agreement no. NICTRFD/NGIRI/2012-13/Corsp/3 Sr. 3. Special thanks to our colleagues Mr. Said Wali and Mr. Asim Ul Haq for their kind help in hardware prototyping. Moreover, we would like to acknowledge the management of "Modelling and Simulation Lab" for providing us an environment to conduct research.

Corresponding Author:
Dr. Jamshed Iqbal
Assistant Professor,
Head – Robotics & Control Research Group,
COMSATS Institute of Information Technology (CIIT), Islamabad, Pakistan
E-mail: jamshed.iqbal@comsats.edu.pk

References

[1] N. Prabhakar, V. Vaithiyanathan, A.P. Sharma, A. Singh, P. Singhal. Object tracking using frame differencing and template matching. Research Journal of Applied Sciences, Engineering and Technology, 2012;4(24):5497-5501.

[2] R.P. Pande, N.D. Mishra, S. Gulhane, A. Joshi. Detection of moving object with the help of motion detection alarm system in video surveillance. Journal of Signal and Image Processing, 2012;3(3):118-121.

[3] A. Talukder, S. Goldberg, L. Matthies, A. Ansar. Real-time detection of moving objects in a dynamic scene from moving robotic vehicles. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2003, vol. 2, pp. 1308- 1313.

[4] J. S. Bae, T. L. Song. Image Tracking algorithm using template matching and PSNF-m. International Journal of Control, Automation, and Systems, 2008;6(3):413-423.

[5] K. Du, Y. Ju, Y. Jin, G. Li, S. Qian, Y. Li. Mean shift tracking algorithm with adaptive block color histogram. In: Proceedings of 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), 2012, pp. 2692-2695.

[6] W. B. Yang, B. Fang, Y. Y. Tang, Z. W. Shang, D. H. Li. SIFT features based object tracking with discrete wavelet transform. Proceedings of International Conference on Wavelet Analysis and Pattern Recognition, 2009, pp. 380-385.

[7] S. Baker, I. Matthews. Lucas-Kanade 20 years on: A unifying framework. International Journal of Computer Vision, 2004;56(3):221-255.

[8] I. Leichter, M. Lindenbaum, E. Rivlin. Tracking by affine kernel transformations using color and boundary cues. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009;31(1):164-171.

[9] D. S. Land, S. W. Lang, H. Choi. Human Body tracking with structural kalman filter. Pattern Recognition, 2002;35(10):2041-2049.

[10] J. Iqbal, M. Pasha, S. Riaz, H. Khan, J. Iqbal. Real-time target detection and tracking: A comparative in-depth review of strategies. Life Sci J, ISSN: 1097-8135, 2013;10(3): 804-813.

[11] J. Iqbal, S. Heikkila, A. Halme. Tether tracking and control of ROSA robotic rover. In: Proceedings of 10th IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV), 2008, Vietnam, pp. 689 – 693.
[12] A.A. Khan, S. Riaz, J. Iqbal. Surface estimation of a pedestrian walk for outdoor use of power wheelchair based robot. Life Sci J, ISSN: 1097-8135, 2013;10(3): 1697-1704.
[13] A. Meddahi , K. Baizid, A. Yousnadj, J. Iqbal. API based graphical simulation of robotized sites. In: Proceedings of 14th IASTED International Conference on Robotics and Applications (RA) 2009, Massachusetts US, pp. 485-492.
[14] M. F. Khan, R.ul. Islam, J. Iqbal. Control strategies for robotic manipulators. In: Proceedings of IEEE International Conference on Robotics and Artificial Intelligence (ICRAI) 2012, Islamabad, Pakistan, pp. 26-33.
[15] J. Iqbal, N.G. Tsagarakis, A.E. Fiorilla, D.G. Caldwell. A portable rehabilitation device for the hand. In: Proceedings of 32nd annual IEEE international conference of Engineering in Medicine and Biology Society (EMBS), 2010, Buenos Aires, Argentina, pp. 3694-3697.
[16] J. Iqbal, R.U. Islam, A.H. Khan. Modeling and analysis of a 6 DOF robotic arm manipulator. Canadian Journal on Electrical and Electronics Engineering (CJEEE), ISSN: 1923-0540, 2012; 3(6):300-306.
[17] J. Iqbal, N.G. Tsagarakis, A.E. Fiorilla, D.G. Caldwell. Design requirements of a hand exoskeleton robotic device. In: Proceedings of 14th IASTED International Conference on Robotics and Applications (RA), 2009, Massachusetts US, pp. 44-51.
[18] R. ul Islam, J. Iqbal, S. Manzoor, A. Khalid, S. Khan. An autonomous image-guided robotic system simulating industrial applications. In: Proceedings of 7th IEEE International Conference on System of Systems Engineering (SoSE), 2012, Italy, pp. 344-349.
[19] J. Iqbal, N.G. Tsagarakis, D.G. Caldwell. Design of a wearable direct-driven optimized hand exoskeleton device. In: Proceedings of 4th International Conference on Advances in Computer-Human Interactions (ACHI), 2011, France, pp. 142-146.
[20] J. Iqbal, N.G. Tsagarakis, D.G. Caldwell. A human hand compatible optimised exoskeleton system. In: Proceedings of IEEE international conference on RObotics and BIOmimetics (ROBIO), 2010, China, pp. 685-690.
[21] K. Naveed, J. Iqbal, H.U. Rahman. Brain controlled human robot interface. In: Proceedings of IEEE International Conference on Robotics and Artificial Intelligence (ICRAI), 2012, Islamabad, Pakistan, pp. 55-60.
[22] J. Iqbal, O.Ahmad, A. Malik. HEXOSYS II – Towards realization of light mass robotics for the hand. In: Proceedings of 15th IEEE International Multitopic Conference (INMIC), 2011, Pakistan, pp. 115-119.
[23] M.M. Azeem, J. Iqbal, P. Toivanen, A. Samad. Emotions in robots. Emerging Trends and Applications in Information Communication Technologies, Communications in Computer and Information Science (CCIS), Springer-Verlag Berlin Heidelberg, 2012, Vol. 281, ISBN 978-3-642-28961-3, pp. 144–153.
[24] J. Iqbal, N.G. Tsagarakis, D.G. Caldwell. A multi-DOF robotic exoskeleton interface for hand motion assistance. In: Proceedings of 33rd annual IEEE international conference of Engineering in Medicine and Biology Society (EMBS), 2011, Boston, US, pp. 1575-1678.
[25] J. Iqbal, N.G. Tsagarakis, D.G. Caldwell. Design optimization of a hand exoskeleton rehabilitation device. In: Proceedings of RSS workshop on understanding the human hand for advancing robotic manipulation 2009, Seattle US, pp. 44-45.
[26] K. Baizid, R. Chellali, A. Yousnadj, A. Meddahi, J. Iqbal, T. Bentaleb. Modelling of robotized site and simulation of robots optimum placement and orientation zone. In: Proceedings of 21st IASTED International Conference on Modelling and Simulation (MS), 2010, Canada, pp. 9-16.

12/6/2013