Real-Time Target Detection and Tracking: A Comparative In-depth Review of Strategies

Javed Iqbal[1], Mustafa Pasha[1], Riaz-un-Nabi[2], Hamza Khan[3], Jamshed Iqbal[1]

[1]Department of Electrical Engineering, COMSATS Institute of Information Technology, Islamabad, Pakistan
[2]Department of Electrical Engineering, NED University of Engineering and Technology, Karachi, Pakistan
[3]Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Genova, Italy
Email: jamshed.iqbal@comsats.edu.pk

Abstract: This survey reports the approaches for object detection and successful target tracking based on image acquisition for real time applications. Moving target detection and its tracking is an important research topic in computer vision and robotics. Complexity of algorithms for this purpose ranges from simple, single and stationary targets tracking to complex, multiple and moving targets handling. Their applications range from basic object detection and pick & place task using a simple robotic arm to complex surveillance using Unmanned Air Vehicles (UAVs). Depending upon the application scenario, various algorithms have been proposed. The objective of this research is to present a comprehensive survey on reported multipurpose algorithms for target real time detection and tracking. Particular consideration has been paid to the underlying norms of each algorithm and its ability to tackle various situations. The paper, besides introducing relevant algorithms and key terminologies, presents a comparative analysis of various associated techniques. It is anticipated that the present review will provide a very useful framework for researchers working in the area of computer vision. The review, in addition to helping them in choosing a particular algorithm, will offer guidance to tailor an existing technique for a specific problem in hand.
[Javed Iqbal, Mustafa Pasha, Riaz-un-Nabi, Hamza Khan, Jamshed Iqbal**. Real-Time Target Detection and Tracking: A Comparative In-depth Review of Strategies.** *Life Sci J* 2013; 10(3): 804-813]. http://www.lifescience site.com 121

Keywords: Object detection, Target tracking, Image processing, Computer vision, Robot control

1. Introduction

To detect a moving target and its tracking is a challenging and essential task in video surveillance systems and is point of focus since late Sixties. Scientific literature reports numerous techniques and algorithms for analysis of moving objects. This analysis has two main parts i.e. detection and tracking. In order to analyze motion of a body, the first step is to detect the object reliably and efficiently. Based on this detection, the object is then tracked. The hierarchal flow-diagram is shown in Figure 1.

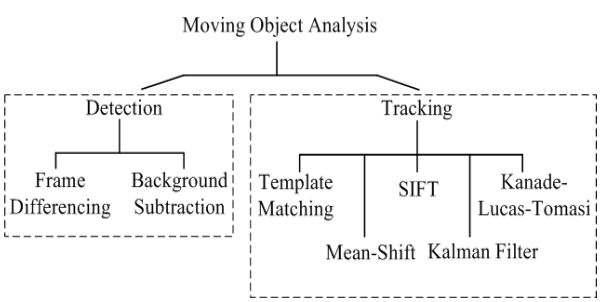


Figure 1. Hierarchal flow-diagram of target handling algorithms.

The remaining paper is outlined below: Section 2 presents state-of-the-art of algorithms for object detection while Section 3 reviews the tracking algorithms in detail. A comparative analysis of reported algorithms is highlighted in Section 4. Finally, Section 5 comments on conclusion.

2. Detection Algorithms

To efficiently and accurately detect moving objects, two basic methods are well-known (a) Frame differencing algorithm [1] (b) Background subtraction method [2-4]. These are detailed below:

Frame Differencing Algorithm: Frame Differencing (FD) algorithm is widely used to detect moving targets due to its efficient results. FD algorithm is dependent on reference frame and current frame, threshold and changes in illumination. It gives centroid (center coordinates) of moving object and area of moving object as output. According to this algorithm, in case of recorded video as well as real time video, two consecutive frames are taken and then compared with each other within certain threshold. Objects which have changed their position with respect to previous frame are thus treated as moving objects.

The difference between reference frame and current frame, which is input to FD algorithm, is an indicator to detect the moving object. After the detection, the resultant image is reprocessed to clarify the detected object. The reprocessing involves filtering and morphological operations.

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 property.

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

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 values less 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 be performing morphological operations e.g. dilation, corrosion etc. results in the object detection. Figure 2 illustrates the results of FD algorithm.

(a) (b)

Figure 2. FD algorithm: (a) Results of frame subtraction (b) Clear and noise-free image showing the detected objects [1]

Background Subtraction Algorithm: Background Subtraction (BS) algorithm is very efficient technique especially in case of static background and static camera. BS algorithm finds numerous applications in motion analysis, Human Computer Interaction (HCI) and intelligent control systems. The objective is to separate the desired moving object from background in video and to use it for advanced operations like segmentation, recognition and tracking. The algorithm is based on dynamic background modeling and dynamic thresholding. BS technique is very sensitive to changes in external environment like effect of light and other background clutters and noises.

The algorithm uses the technique of frame differencing algorithm for detection of moving objects. The previous frame with certain dynamically modeled features acts as a background for the current frame. Taking difference of the two frames indicates the moving objects. Morphological operations are then applied to refine the image and finally to pin point the detected target. Dynamic background modeling is used in BS algorithm to overcome the effect of variations in background. The ideal condition is when the background is static meaning that camera is operating in static conditions. To further improve the accuracy of the results of object detection, dynamic thresholding is used.

The reference background is modeled by taking static background frames and certain normalization parameters on pixel by pixel basis. These parameters including E_i, s_i, a_i and b_i are calculated for each pixel. E_i is the color value anticipated for a particular i^{th} pixel, s_i is the corresponding standard deviation for that color value, a_i shows the disparity of the brightness distortion and b_i is variation of the chromaticity distortion of the pixel. For efficient detection background model is updated by using the following equation,

$$B_{k+1}(x,y) = \beta B_k(x,y) + (1-\beta)F_k(x,y) \qquad (3)$$

Where β is update coefficient and is usually equal to 0.004. The pixel grey scale value of the pixel in the current frame is represented by $F_k(x,y)$. Values of current frame and next frame for the background is symbolized by $B_k(x,y)$ and $B_{k+1}(x,y)$ respectively. Reported results are shown in Figure 3.

(a) (b) (c)

Figure 3. BS algorithm:(a) Original frame (b) Results of subtraction results (c)Image showing the detected objects [5]

3. Tracking Algorithms

After successful detection, tracking of moving object is an important task in motion analysis. Many algorithms to efficiently track moving bodies have been reported in literature. These algorithms include

Template matching [1,6], Mean-Shift [7-9], Scale Invariant Feature Transform [10,11], Kanade-Lucas-Tomasi tracker [12-14], Kalman Filter [15,16] etc. Each algorithm has its own capabilities, specification and limitations.

Template Matching Algorithm: Template Matching (TM) is a simple but efficient algorithm for object tracking. Detection can also be achieved using this algorithm but the performance is poor. The algorithm can be implemented in different ways i.e. mathematical correlation, edge detection, texture analysis etc. Simple variant of TM algorithm takes sequences of frames as input and searches the desired specified object on the basis of maximum correlation. In the first frame, a template of the desired object is selected and it is correlated with the second frame. In case of maximum correlation, a message of object matched is displayed. Otherwise, the next frame is traversed.

TM algorithm works like a tracker module. It can be used to track still objects as well as moving objects. The algorithm can recognize and detect the object of interest irrespective of motion or static condition of other objects. The template specified is known as target template, which is selected based on some previous information. This information may be a predefined template or can be selected on the basis of centroid or mean coordinates of desired object or results of some other detection algorithm. The target template is matched in next frames. Using mathematical correlation, if it matches at some instant, then the template is updated with current matched template. It is necessary to convert the image into bi-level or grey level format for to compute the correlation as given by (4).

$$\mathrm{cor} = \frac{\sum_{i=0}^{N-1}(x_i - \bar{x}).(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N-1}(x_i - \bar{x})^2.\sum_{i=0}^{N-1}(y_i - \bar{y})^2}} \qquad (4)$$

Where x_i is the grey level in the image of the template, $\bar{x}$ is the average value of grey level in the template (target) image, similarly y_i represents the portion in source image and $\bar{y}$ shows the average value of grey level in the source image, N corresponds to the total no. of pixels in the template section image (N = rows * columns). cor is the resultant correlation obtained, having value in range $-1<\mathrm{cor}<+1$. The larger value of cor corresponds to the stronger association between the two images.

In a large source image, the matching process moves the target template to all possible positions since in this regard pixel-by-pixel matching is done. A numerical catalog implying the fact that template suits the particular graphic in this position will be computed. The trajectory of the tracked object can then be plotted. Results of TM algorithm are shown in Figure 4.

(a) (b)

Figure 4. TM algorithm: (a) The target template to be tracked (b) The tracked target [1]

TM algorithm when used in real time applications suffer from certain limitations including:

1. Target template should be entirely located in the source image.
2. Partial template matching cannot be accomplished due to boundaries.
3. Variation and scaling cause pitiable matches

Mean-Shift Algorithm: Mean-shift algorithm, widely used for moving object tracking, is based on statistical calculations of color in the target model and the candidate model. Bhattacharyya coefficients are used to measure the similarity [9]. The new position of the target is calculated on the basis of similarity function. This algorithm is a very useful tool for probability density estimation, mode seeking, clustering and tracking.

The algorithm iteratively shifts a data point to the average value of points in its neighborhood. It uses the color 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 $\|x\|^2$ and is given by

$$K(x) = k\left(\|x\|^2\right) \qquad (5)$$

Where k is called as the profile of K, which is nonnegative and non-increasing. Commonly used kernels are: Flat kernel, Gaussian kernel and Epanechnikov kernel. The simple mean (m) at point x with kernel K(x) is calculated as

$$m(x) = \frac{\sum_{i=1}^{n} K(x-x_i)x_i}{\sum_{i=1}^{n} K(x-x_i)} \qquad (6)$$

The difference calculated by the term m(x)-x is known as the mean shift. The mean shift tracking based on color histogram is divided into three basic steps. In the first step Probability Density Function (PDF) of the target model q_u and the candidate model p_u is calculated on the basis of color histogram and kernel density estimation as given by (7) and (8) respectively.

$$q_u(x) = C_q \sum_{i=1}^{n} k\left(\left\|\frac{x_i}{h_q}\right\|^2\right) \delta(b(x_i) - u) \qquad (7)$$
$$u = 1,2,\ldots,m$$

$$p_u(y) = C_P \sum_{i=1}^{N_p} k\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right) \delta(b(y_i) - u) \qquad (8)$$
$$u = 1,2,\ldots,m$$

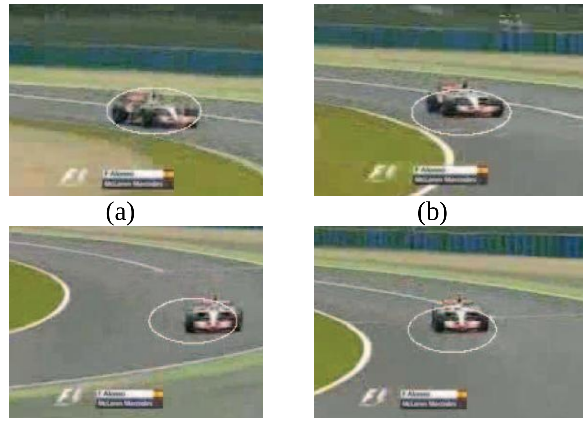
Where $b(x_i)$ and $b(y_i)$ represent the histogram index function of pixels. h_p and h_q are radius of the candidate and target model kernels respectively and δ is the Kronecker delta function. Thus, if $b(x_i)=u$ the PDF will have certain value otherwise kernel will contribute zero value to q_u and p_u. C_q and C_p are normalization constants. 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.

$$\rho(y) = \rho[p(y), q] = \frac{1}{2}\sum_{u=1}^{m} \sqrt{p_u(y) q_u} \qquad (9)$$

Let y denotes the target location of its current position with the corresponding color probability $\{p_u(y)\}$, $p_u(y) > 0$ for $u = 1, \ldots, m$. Let z denotes the estimated new target location near y where change in color probability does not occur rapidly. Finally, the new position 'z' of the object in the candidate model is calculated in the neighborhood 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 as

$$z = \frac{\sum_{i=1}^{N_p} y_i w_i g\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)}{\sum_{i=1}^{N_p} w_i g\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)} \qquad (10)$$

Where $g(x) = -k'(x)$. The results obtained are shown in Figure 5.



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

Figure 5. Mean-shift algorithm:(a-d) Progressive tracking of the moving target [7]

Algorithm Description:

1. Initialize the first frame, the size of the frame is WxH with W being the width and H is the height of the frame.
2. Assume that the size of the bounding rectangle of the current object is $W_0 x H_0$.
3. Calculate the color histogram or PDF of the target template q_u .
4. Read the next frame image and calculate the candidate model PDF p_u.
5. Compute the similarity measure i.e. Bhattacharyya coefficient $\rho(p(y), q)$ of the target model.
6. Compute the new target location.
7. Calculate the PDF of the candidate model subject to the new position and compute respective similarity measure, $\rho(p(z), q)$.
8. If $\rho(p(z), q) < \rho(p(y), q)$ then $z = \frac{1}{2}(y + z)$
9. If $\|z - y\| < \varepsilon$ i.e. is small enough, then stop and goto step 4.

else set $y \leftarrow z$ and goto step 4.

Scale Invariant Feature Transform Algorithm: Today's video applications in research areas like robotics, surveillance and computer vision essentially demand tracking of moving objects. Scale Invariant Feature Transform (SIFT) is based on feature extraction and matching. The algorithm extracts features of the object that are invariant to changes in the object like illumination and orientation and continuously detects and matches the features.

The feature points in the target model are extracted and then matched with the feature points in the candidate model. SIFT also computes location of the features points and matches it with the location of features points in the candidate model on the basis of Euclidian distance. SIFT involves following major computations for generation of the image-feature set:

1. Detecting the extrema (scale-space).
2. Localizing the keypoint.
3. Assigning the orientation.
4. Formulating the description of the keypoint.

Let $L(x, y, \sigma_L)$ denotes the image scale-space, which is obtained as a result of convolution of Gaussian $G(x, y, \sigma_L)$ with an image $I(x, y)$. So

$$L(x, y, \sigma_L) = G(x, y, \sigma_L) * I(x, y) \tag{11}$$

Where * is the convolution operator and $G(x, y, \sigma_L)$ is the 2D Gaussian kernel and is given as

$$G(x, y, \sigma_L) = \frac{1}{2\pi\sigma_L^2} e^{\frac{-(x^2+y^2)}{2\sigma_L^2}} \tag{12}$$

The difference of Gaussians separated by a factor k can be used to compute the stable keypoint locations. k holds a scalar multiplicative constant value.

$$D(x, y, \sigma_L) = \left(G(x, y, k\sigma_L) - G(x, y, \sigma_L)\right) * I(x, y) \tag{13}$$

$$D(x, y, \sigma_L) = L(x, y, k\sigma_L) - L(x, y, \sigma_L) \tag{14}$$

This difference indicates an approximate value of the normalized scale Laplacian of Gaussian. D can be linked to $\sigma_L^2 \nabla^2 G$ by

$$\sigma^2 \nabla^2 G = \frac{\partial G}{\partial \sigma_L} \approx \frac{G(x, y, k\sigma_L) - G(x, y, \sigma_L)}{k\sigma - \sigma} \tag{15}$$

Therefore

$$G(x, y, k\sigma_L) - G(x, y, \sigma_L) \approx (k - 1)\, \sigma_L^2 \nabla^2 G \tag{16}$$

(16) depicts that subtraction of two Gaussian functions has scales which differ by a constant value. It incorporates the normalization scale needed for σ_L^2 scale-invariant Laplacian. After extracting the feature points and matching with initial template in the current frame by using SIFT algorithm, the centroid of area surrounded by these feature points is computed using (17)

$$M = \frac{\sum m_i r_i}{\sum m_i} \tag{17}$$

Where m_i is the pixel value at i^{th} position and r_i is corresponding pixel coordinate.

SIFT algorithm achieves tracking of stable multi-objects by taking rectangular windows around the object of interest in both the current frame as well as reference frame. The basic idea is to determine the feature points and exclude those, whose locations are mismatched. SIFT generates the keypoints. A database holds their local features. From the next consecutive frame, keypoints are obtained in a same way. The matching keypoints are chosen based on Euclidian distance. These keypoints form the basis for tracking of multiple objects. SIFT algorithm is detailed as follows:

1. Load a video in the program.
2. Detect the moving object.
3. Select the window which bounds the object in the reference frame.
4. Apply SIFT algorithm to extract the feature points.
5. Perform step 2-4 on the current frame.
6. Match the descriptor features.
7. Select location matched feature points.
8. Track the object.

Object can be detected using frame differencing algorithm or Otsu’s method. Applying SIFT algorithm, key points are extracted in the reference frame and in the current frame as

$$[\mathrm{im}, \mathrm{des}, \mathrm{loc}] = \mathrm{SIFT}(\mathrm{image}) \tag{18}$$

Data ‘im’ includes pixel values of the test image, ‘des’ shows descriptor vectors matrix and ‘loc’ has orientation values, location and scale. The extended feature points have both components; object and background. The keypoints related to background are removed using

$$p_{keypoint}(i, j) = \begin{cases} \text{participate} & \text{if}(i, j) \in \text{object area} \\ \text{not} & \text{if}(i, j) \in \text{background area} \end{cases} \tag{19}$$

Keypoint matching is then performed on the basis of distance comparison. The steps to achieve matching are as follows:

1. Using dot product, the distances between m keypoints of the reference frame and n keypoints of the (candidate) frame that is the current frameare calculated from the descriptor vectors. Distance D_{ij} between i^{th} keypoint descriptor vector des_{Ri} in reference frame and j^{th} keypoint descriptor vector des_{Cj} in current frame is

$$D_{ij} = \cos^{-1}(des_{Ri} \cdot des_{Cj})$$

2. Distances d_{ij} are sorted in such a way that distance ratios among the distances of the nearest neighbor to that of the second-nearest neighbor are determined.

$$\mathrm{distRatio} = \frac{\text{the closest distance}}{\text{the second closest distance}} \tag{20}$$

3. A threshold value is set to reject the matched portion i.e. 0.8, the distance greater than 0.8 is rejected amongst which 90% rejection is of the false matches and only the 5% of the correct portion is also rejected.

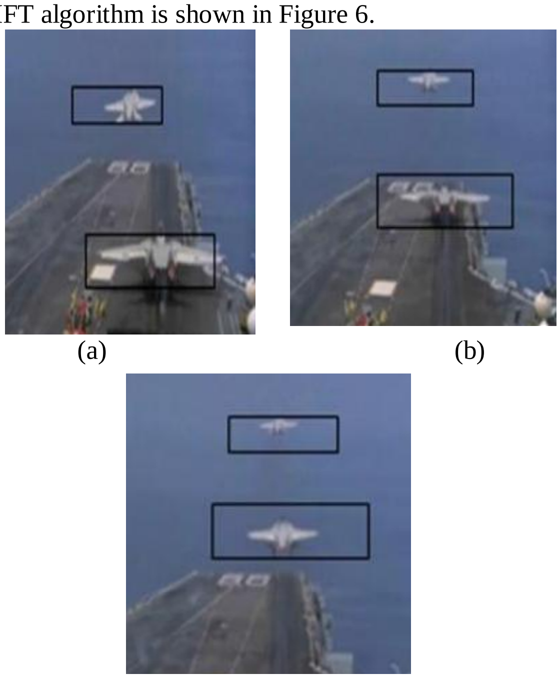
$$\text{match} = \begin{cases} \text{accept if distRatio} \le 0.8 \\ \text{reject if distRatio} > 0.8 \end{cases} \qquad (21)$$

The keypoints thus obtained as a result of matching are stored in a database, which is used for tracking. Among the matched feature points, only location-matched keypoints are selected. To find these type of keypoints, difference in the location d_{RC} between the both windows is determined for that particular candidate. If the location difference d_{RC} is less than or equal to a certain value set as threshold d_{th}, then the candidates are designated as a location-matched keypoints, otherwise the feature points are discarded. Difference can be calculated as

$$d_{RC} = \sqrt{(x_C - x_R)^2 + (y_C - y_R)^2} \qquad (22)$$

$$\text{location} - \text{matched} = \begin{cases} \text{yes} & \text{if } d_{RC} \le d_{th} \\ \text{no} & \text{otherwise} \end{cases} \qquad (23)$$

Where (x_R, y_R) is the position coordinate of a particular candidate in a frame known as reference frame, on the other hand accordingly (x_C, y_C) is the position coordinate of the particular candidate in a frame known as current frame. The value of threshold d_{th} for better results is usually varied from 1-6. Considering a typical value d_{th}=3, results of applying SIFT algorithm is shown in Figure 6.



(a) (b) (c)

Figure 6. SIFT algorithm: (a-c) Results of features matched and location matched keypoints [11].

Kanade-Lucas -Tomasi Algorithm: Kanade-Lucas-Tomasi (KLT) algorithm is based on features which are tracked frame to frame. Moving objects can be tracked by correlating the features because the features are distributed all over the image. The features belonging to moving objects will be clustered and assigned to the corresponding moving object motion layer. The first step is the detection of moving objects and construction of motion layer. Due to maintaining the layers of all moving objects permits tracking of the objects in motion. To maintain the total number of feature same some new detected features along the tracking process are added against the lost features. Successive frames side by side are used as control points for indexing a frame to its previous frames. In order to separate background features, region division method is preferred. KLT algorithm can be used after a detection algorithm. Followed by successful detection, KLT tracker function extracts the feature points, uses these for tracking purpose on frame to frame basis and update that feature points.

Kalman Filter: Rudolf Kalman realized a powerful filtering technique for tracking spacecraft trajectory at NASA. The proposed algorithm, later on, became popular in other areas of computer vision and image processing. The filter is very useful in the estimation of target position while tracking moving objects. Kalman filtering is based on a set of mathematical formulations. These equations computationally solve the problem of sequential systems. The filter has many distinguishing characteristics including ability to estimate previous, current and predicted future states. The process of estimation works fine even in worst-case scenario when the exact characteristics of the modeled system are not known. (24-25) present state space representation of the target motion.

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \qquad (24)$$

$$y_k = Hx_k + v_k \qquad (25)$$

Where the state vector is represented by x=[x y $\acute{x}$ $\acute{y}$]^T. y represents the output of the system, w and v shows the noise impact (as white noise) for the process noises and the measurement noises respectively. A and H are termed as transition matrix and observation matrix respectively and are given by (26).

$$A = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad (26)$$

Kalman filter equations are structured into two categories, i.e. time update equations and measurement update equations. The time update equations projects the current state at time k to the next state at time k+1 to have 'a priori' estimate for the time step at k+1. The measurement update equations correct the estimated state by replacing it with actual measurements obtained as feedback. An improved posteriori estimate is obtained by incorporating new measurements in the time update equations. Though implementation of Kalman filter is a bit complicated but the results obtained (Figure 7) are comparatively good. The filter can also be used for real time multiple object tracking.

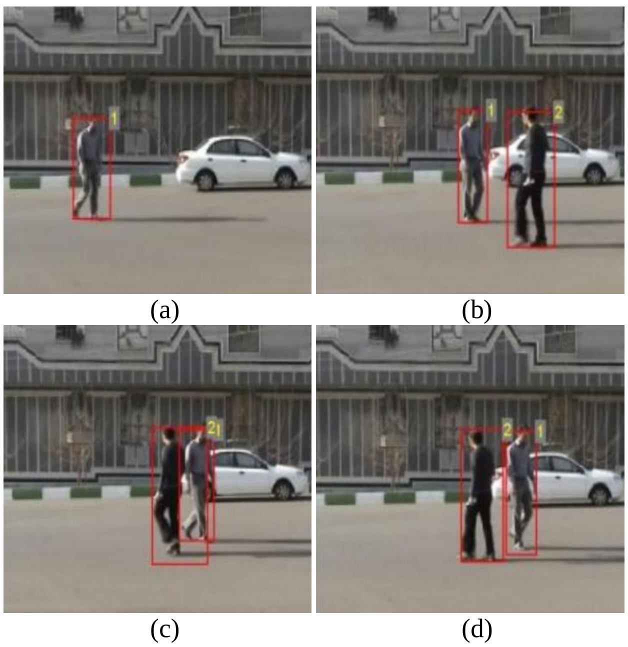


Figure 7. Kalman filter: (a-d) Frame wise tracking results [15]

Affine Transformation: The most common problem in image processing and computer vision applications is localization of object's boundaries and its tracking in consecutive video frames. Affine transformation [29] plays an important role in tracking of moving objects especially on a smaller scale. It is based on a set of linear equations. After successful detection, a vector of the affine parameters for each template of the detected object is computed using this transformation. These parameters are inherited and updated to track the object in consecutive frames.

Affine transformation has the advantage that it can estimate future properties and position of the object on the basis of relevant present and past information. Additionally, this method can be useful on systems with speedy objects where the traditional trackers fail. Linear affine transformations strongly suffer from changes in image properties like translation, rotation, and/or scaling of the target. This transformation finds a potential application to track cells in microscopic imagery [30] when combined with particle filtering.

4. Comparison of Algorithms

In this section, different detection and tracking algorithms are compared on the basis of their properties, efficiency, specifications and limitations. Table 1 presents a comparative overview of the algorithms discussed in this paper.

5. Conclusion

Moving target analysis involves its detection and tracking. To accurately and efficiently detect moving targets two basic algorithms are more popular i.e. (1) Frame differencing algorithm and (2) Background subtraction algorithm. These algorithms have their own usage, importance, specifications, level of complexity and limitations. Frame differencing algorithm has good detection results but is very sensitive to variations in light intensity. Background subtraction algorithm performs best under static camera conditions. However, it is very sensitive to background changes.

Similarly, many algorithms have been proposed to accurately track moving targets. These include: (1) Template Matching (2) Mean-Shift Tracking algorithm (3) SIFT Tracking technique (4) KLT tracker (5) Affine Transformation and (6) Kalman Filtering. Template matching uses mathematical correlation between the target template and the source image. TM has good tracking results in case the target template is completely present in the source image. Mean-Shift achieves tracking using kernel function and Bhattacharyya coefficients to calculate similarity measure and have good tracking results. The algorithm can also be used for detection of slow speed moving objects. SIFT finds out the features that are invariant to scaling, rotations and light conditions but is very difficult to implement specially in real time. KLT tracker is dependent upon optical flow field of the image but is very sensitive to noise and suffers from computational complexity. Finally, the paper presents a quick comparison chart of the algorithms discussed.

Acknowledgements:

This project has been funded by National Information & Communication Technology (ICT) R&D Fund under agreement no. NICTRFD/ NGIRI/ 2012-13/Corsp/3 Sr. 3. We would like to acknowledge and thank the funding body.

Corresponding Author:
Dr. Jamshed Iqbal,
Head – Robotics & Control Research Group,

COMSATS Institute of Information Technology (CIIT), Islamabad, Pakistan.
E-mail: jamshed.iqbal@comsats.edu.pk.

References

[1] Prabhakar N, Vaithiyanathan V, Sharma AP, Singh A, Singhal P. Object tracking using Frame differencing and template matching. Research Journal of Applied Sciences, Engineering and Technology 2012; 4(24): 5497-5501.
[2] Pande RP, Mishra ND, Gulhane S, Joshi A. 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] Niu L, Jiang N. A moving objects detection algorithm based on improved background subtraction. In: Proceedings of 8thInternational Conference on Intelligent Systems Design and Applications 2008, vol. 3, pp. 604-607.
[4] Zhang L, Liang Y. Motion human detection based on background subtraction. In: Proceedings of 2ndInternational Workshop on Education Technology and Computer Science 2010, pp. 284-287.
[5] Madhavi BSM, Rao MVG. A fast and reliable motion human detection and tracking based on background subtraction. IOSR Journal of Electronics and Communication Engineering 2012;1(1):29-35.
[6] B JS, Song TL. Image tracking algorithm using template matching and PSNF-m. International Journal of Control, Automation, and Systems 2008;6(3):413-423.
[7] Du K, Ju Y, Jin Y, Li G, Qian S, Li Y. MeanShift tracking algorithm with adaptive block color histogram. In: Proceedings of 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet) 2012, pp. 2692-2695.
[8] Fukunaga K, Hostetler L. The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Tran. IT 1975; vol. 21:32-40.
[9] Wen ZQ, Cai ZX. Mean shift algorithm and its application in tracking of objects. In: Proceedings of 5th International Conference on Machine Learning and Cybernetics, Dalian 2006, pp. 4024-4028.
[10] Lowe DG. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 2004;60(2):91-110.
[11] Yang WB, Fang B, Tang YY, Shang ZW, Li DH. SIFT features based object tracking with discrete wavelet transform. In: Proceedings of International Conference on Wavelet Analysis and Pattern Recognition 2009, pp. 380-385.
[12] Wrioutius M, Ramel JY, Vincent N. Selection of points for on-line signature comparison. In: Proceedings of 9th International Workshop on Frontiers in Handwriting Recognition 2004, pp. 503-508.
[13] Baker S, Matthews I. Lucas-Kanade 20 years on: A unifying framework. International Journal of Computer Vision 2004;56(3):221-255.
[14] Svoboda T, Kanade–Lucas–Tomasi Tracking, Research Report, Center for Machine Perception, Czech Technical University, Prague, 2007. available online at http://cmp.felk.cvut.cz [accessed on 26th June 2013].
[15] Land DS, Lang SW, Choi H. Human body tracking with structural Kalman filter. Pattern Recognition 2002; 35(10):2041-2049.
[16] Simon D. Kalman filtering. Embedded Systems Programming 2001;14(6):72-79.
[17] Kodjo AMD, Jinhua Y. Real-time moving object tracking in video. In: Proceedings of International Conference on Optoelectronics and Microelectronics (ICOM) 2012, pp. 580-584.
[18] Gupta K, Kulkarni AV. Implementation of an automated single camera object tracking system using frame differencing and dynamic template matching. Advances in Computer and Information Sciences and Engineering, Springer 2008:245-250.
[19] Roichman E, Solomon Y, Moshe Y. Real-Time pedestrian detection and tracking. In: Proceedings of 3rd European DSP Education and Research Symposium (EDERS) 2008, pp. 281-288.
[20] Kuralkar PP, Gaikwad VT. Human object tracking using background subtraction and shadow removal techniques. International Journal of Advanced Research in Computer Science and Software Engineering 2012;2(3):294-297.
[21] Yuan D, Wei W, Yi L, Guo Z. Enhanced mean shift tracking algorithm based on evolutive asymmetric kernel. In: Proceedings of International Conference on Multimedia Technology (ICMT) 2011, pp. 5394 – 5398.
[22] Gorry B, Chen Z, Hammond K, Wallace A, Michaelson G. Using mean-shift tracking algorithms for real-time tracking of moving images on an autonomous vehicle testbed platform. In: Proceedings of World Academy of Science, Engineering and Technology 2007, vol. 25, pp 356-361.
[23] Ha SW, Moon YH. Multiple object tracking using SIFT features and location matching. International Journal of Smart Home 2011;5(4):17-26.
[24] Angelov P, Gude C, Tehran SP, Ivanov T. ARTOT: Autonomous real-Time object detection and tracking by a moving camera. In: Proceedings of 6thIEEE International Conference Intelligent

Systems (IS), 2012, pp. 446 – 452.

[25] Huang CM. Real-time object detection and tracking on a moving camera platform. In: proceedings of ICROS-SICE International Joint Conference, Japan 2009, pp. 717-722.

[26] Han B, Paulson C, Lu T, Wu D, Li J. Tracking of multiple objects under partial occlusion. Automatic Target Recognition XIX. In: Proceedings of the SPIE, 2009, vol. 7335.

[27] Mirabi M, Javadi S. People tracking in outdoor environment using Kalman filter. In: proceedings of 3rdInternational Conference on Intelligent Systems Modelling and Simulation 2012, pp. 303-307.

[28] Torkaman B, Farrokhi M. Real-time visual tracking of a moving object using pan and tilt platform: A Kalman filter approach. In: proceedings of 20th Iranian Conference on Electrical Engineering (ICEE) 2012, pp. 928-933.

[29] Tzouveli P, Avrithis Y, Kolllias S. Fast Video Object Tracking using Affine Invariant Normalization. In: proceedings of 3rd IFIP Conference on Artificial Intelligence Applications & Innovations, Athens, Greece, June 2006.

[30] Cui J, Ray N, Acton ST, Lin Z. An affine transformation invariance approach to cell tracking. Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society. 2008; 32(7), pp. 554-465.

[31] Murshed M, Dewan M.A.A, Chae O. Moving object tracking - a parametric edge tracking approach. In: proceedings of 12th International Conference on Computers and Information Technology, ICCIT, 2009, pp. 471 – 476.

7/25/2013

Table 1. Comparison of target detection and tracking algorithms

No.	Technique/ Algorithm	Detection	Tracking	Dependencies	Constraints/ Limitations	Performance	Complexity*	Ref**	Ref***
1	Frame Differencing (FD)	√	√	Reference and current frame, Threshold, Illumination	Multiple objects, Shadow effects	Better detection, Poor tracking and gives center position of object	Simple	[1]	[17, 18]
2	Background Subtraction (BS)	√	X	Dynamic background modeling, Thresholding	Static camera, Change in external environment	Detection efficient, Calculates object position	Medium	[2-4]	[5,19,20]
3	Template Matching (TM)	X	√	Threshold, Correlation of templates	Changes in illumination and shape rapidly, Shadow effects	Tracking efficient, Plots objects' trajectory	Simple	[1,6]	[18]
4	Mean-shift	√	√	Color information, Similarity measure, Kernel function	Poor detection result, No spatial information, Illumination changes	Tracking efficient, Position and trajectory of the object	Complex	[7-9]	[21,22]
5	Scale Invariant Feature Transform (SIFT)	X	√	Feature points (local maxima and minima etc.), Euclidian distance	Background keypoints, Shadow effects	Efficient tracking. Invariant to environmental changes	Complex	[10,11]	[23,24]
6	Kande-Lucas-Tomasi (KLT)	X	√	Feature points extraction, Image flow field	Sensitive to noise, Lengthy calculations	Efficient tracking under specific conditions	Complex	[12-14]	[25,26]
7	Kalman Filtering	X	√	Estimation of past, present and future	Sensitive to rapid changes in environment, Lengthy calculations	Good tracking under non ideal conditions	Complex	[15,16]	[27,28]
8	Affine Transform	X	√	Estimation of future on the basis of present and past	Suffers from rapid changes in object properties	Tracking efficient, new position and properties estimation	Complex	[29]	[30,31]

* Complexity w.r.t. implementation is reported

** Reference where algorithm definition can be seen

*** Reference where algorithm implementation details can be found