

Detection and Tracking System of Moving Objects Based on MATLAB

Habib Mohammed Hussien
Electrical and Electronics Technology Department
Federal TVET Institute
Addis Ababa, Ethiopia

Abstract: Moving Object detection and tracking are receiving a growing attention with the emergence of surveillance systems. Video surveillance has been in used in the monitor security sensitive areas (such as banks, department stores, highways, crowded public places and borders, and etc.). In this thesis, video surveillance system with moving object detection and tracking capabilities is presented. This thesis is committed to the problems of defining and developing the basic building blocks of video surveillance system. The video surveillance system requires fast, reliable and robust algorithms for moving object detection and tracking. The system can process both color and gray images from a stationary camera. It can handle object detection in indoor or outdoor environment and under changing illumination conditions. This paper presents detection and tracking system of moving objects based on matlab. It is described for segmenting moving objects from the scene. The proposed system is capable of adapting to dynamic scene, removing shadow, and distinguishing left/removed objects both in indoor and outdoor. The proposed technique combines simple frame difference (FD), simple adaptive background subtraction (BS), and accurate Gaussian modeling to benefit from the high detection accuracy of Mixture of Gaussian solution (MoG) in outdoor scenes while reducing the computations. Thus, making it faster and more suitable for real time surveillance applications. This study used IFD (Inter-Frame Differencing algorithm) and bounding box method to track the objects.

Keywords: Video Surveillance, Moving Object Detection, Background Subtraction, Moving Object Tracking, Frame differencing, Mixture of Gaussian model

1. INTRODUCTION

As surveillance systems are becoming more popular, robust detection and tracking techniques are needed to determine moving objects. Moving object detection and tracking in the military, intelligence monitoring, human machine interface, virtual reality, motion analysis and many other fields have Wide application prospects in science and engineering. It has important research value, which attracting more and more researchers at home and abroad. The current detection and tracking system is mainly using two techniques: First, the use of radar technology for tracking; while another is based on image processing technology to achieve target tracking. Our paper is based on image processing to achieve the objectives of detection and tracking system. Image sequence of moving target tracking, is detected in each

frame of the goal of the various independence movements, or the Users interested in the movement areas (such as the human body, vehicles, etc.), and extract the object's position information, to receive the various objectives Trajectory. The making of video surveillance system best requires fast, reliable and robust algorithm for moving object detection and tracking system. Identifying moving objects from a video sequence is a fundamental and critical task in many computer-vision applications. A common approach is to perform background subtraction, which identifies moving objects from the portion of a video frame that differs significantly from a background model. There are many challenges in developing a good object detection algorithm. First, it must be robust against changes in illumination. Second, it should avoid detecting non-stationary background objects such as swinging leaves, rain, snow, and shadow cast by moving objects. Finally, its internal background model should react quickly to changes in background such as starting and stopping of vehicles. After identifying moving object in a given scene, the next step in video analysis is tracking. Tracking can be defined as the creation of temporal correspondence among detected objects from frame to frame. This procedure provides temporal identification of the segmented regions and generates cohesive information about the objects in the monitored area such as trajectory, speed or direction. In everyday life, humans visually keep detecting and tracking a multitude of objects with certain objectives in mind. Examples are orientation in the environment, recognizing persons in the surroundings, locating, recognizing, monitoring and handling of objects. Although much has been learned from biological and human vision for image processing and analysis, the markedly different tasks and capabilities of the function modules have to be kept in mind. It is not possible to simply transfer or copy the human biological vision to machine vision system. However, the basic functionality of the human vision system can guide us in the task of how these principles can be transferred to technical vision system such as motion analysis. The analysis of motion gives access to the dynamics of processes. Motion analysis is generally a very complex problem. The true motion of objects can only be inferred from their motion after the 3-D reconstruction of the scene. The increasing use of video sensors, with Pan-Tilt and Zoom capabilities or mounted on moving

platforms in surveillance and autonomous driving vehicle. The outputs of these algorithms can be used both for providing the human operator with high level data to help him to make the decisions more accurately in a shorter time and for offline indexing and searching stored video data effectively. The advances in the development of these algorithms would lead to breakthroughs in applications that use visual surveillance. Papers inspired me to develop the proposed systems algorithm are [1-5,10,15,22,23,24,25,26,27,28]

This paper presents a detection and tracking Detection and Tracking systems of moving objects for monitoring outdoor and indoor scenes with gradual illumination changes and swinging tree branches. The technique combines frame differencing (FD), Background subtraction (BS) and Mixture of Gaussian Model (MoG) to get a robust detection results by focusing the attention on the most probable foreground pixels. And we used IFD (inter-frame differencing tracking algorithm) to track the detected objects.

2. RELATED WORKS

There are several background modeling schemes ranging from simple techniques keeping single background state estimates and providing acceptable accuracy for simple applications, all the way to complicated techniques with full density estimates; thus providing better accuracy at the expense of increased memory and complexity. Background subtraction is particularly a commonly used technique for motion segmentation in static scenes [6].

FD is the simplest technique with the background being the previous frame. This method has low memory requirements, $O(1)$, high speed, and high adaptability; but suffers from the aperture problem. IVSS [3] uses FD to generate "hypotheses" about the objects, then verifies them by extracting the features using Gabor filter, and classifies objects using support vector machine.

Collins et al. developed a hybrid method that combines three-frame differencing with an adaptive background subtraction model for their VSAM project [3]. The hybrid algorithm successfully segments moving regions in video without the defects of temporal differencing and background subtraction.

The W4 [4] system uses a statistical background model where each pixel is represented with its minimum (M) and maximum (N) intensity values and maximum intensity difference (D) between any consecutive frames observed during initial training period where the scene contains no moving objects.

Stauffer and Grimson [5] presented a novel adaptive online background mixture model that can robustly deal with lighting changes, repetitive motions, clutter, introducing or removing objects from the scene and slowly moving objects

3. PROBLEM STATEMENT

Problems concerning about this system can be classified into two categories; motion detection and motion tracking. Motion detection involves verifying the presence of a moving object in image sequences based on the object's temporal change and possibly locating it precisely for tracking or

recognition purpose. Whereas motion tracking is an iterative process of determining the trajectory of a moving object during a video sequence, by monitoring the object's spatial and temporal changes, including its presence, position, size, shape, etc. This is done by solving the temporal correspondence problem, i.e. the problem of matching the target region in successive frames of a sequence of images taken at closely-space time intervals. These two processes are closely related because motion tracking usually starts with detecting moving objects, while detecting an object repeatedly in subsequent image sequence is often necessary to help and verify tracking. Given a live feed frames sequences from a fixed camera, detecting all the foreground objects and estimate the trajectory of the object of interest moving in the scene. There are many challenges in developing a good object detection algorithm. First, it must be robust against changes in illumination. Second, it should avoid detecting non-stationary background objects such as swinging leaves, rain, snow, and shadow cast by moving objects. Finally, its internal background model should react quickly to changes in background such as starting and stopping of moving objects. In video surveillance in addition to detection problems there are also problems in tracking objects. One of the problems of tracking is that dynamic objects move in patterns that are highly non-linear. Another important problem is simultaneously tracking multiple moving objects, correspondence, partial overlapping and occlusions. The proposed system must solve these problems correctly.

4. METHODOLOGY

Each application that benefit from video processing has different needs, thus requires different treatment. However, they have something in common: moving objects. Thus, detecting regions that correspond to moving objects such as people and vehicles in video is the first basic step of almost every vision system and video processing as shown in the fig. 1 below.

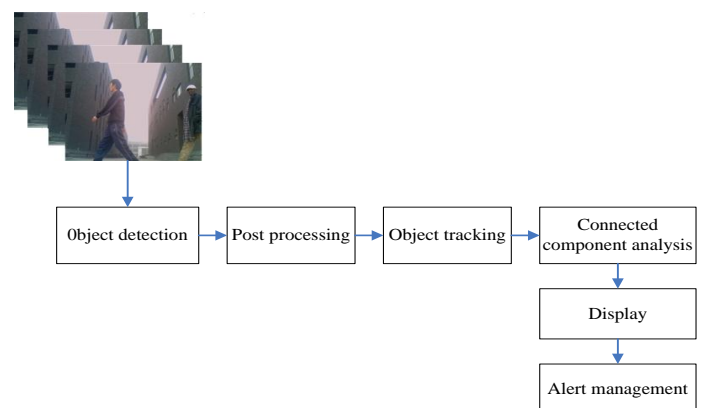


Fig. 1 A Generic Framework for Video Processing Algorithm

An overview of the entire moving object detection and tracking system is shown in Fig.2 The technique combines Frame differencing, adaptive background subtraction and MoG to reduce the computations of MoG and improve its accuracy by focusing the attention on the most probable

foreground pixels. This proposed system able to detect and track moving object in a given scene. The system uses **stationary camera** to acquire video frames from outside world. The system starts by feeding video frames from a static camera that supposed to monitor a given site. The system can work both indoor as well as outdoor. **The first step** of our approach is **identifying foreground objects from stationary background.** Here we use **hybrid adaptive scheme algorithm,** to identify the foreground object. **Next step** is **post processing to remove noises** that cannot handled by proposed system. **After** applying post processing, **connected component analysis** will follow and group the connected regions in the foreground map **to extract individual object's features** such as bounding box, area, center of mass etc. **The final step** of this system is **tracking.** The tracking algorithm is **inter-frame differencing (IFD).** This algorithm uses object features to track objects from frame to frame.

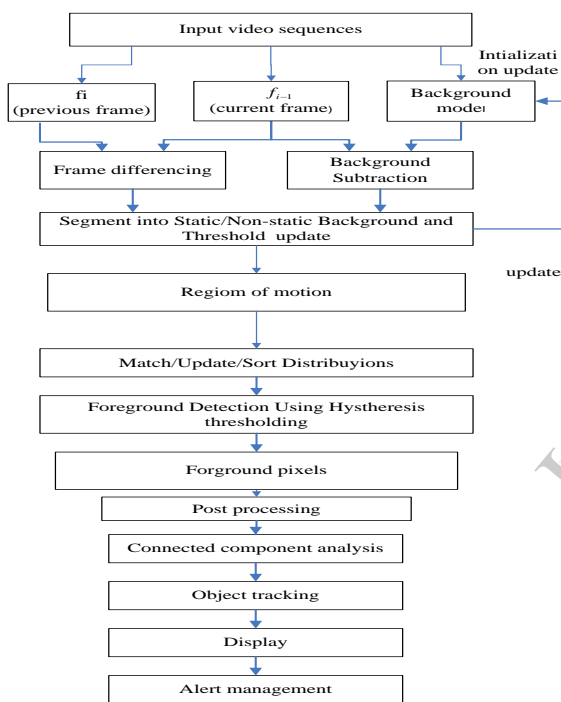


Fig. 2 Block Diagram for Detection and Tracking System

4.1. MOVING OBJECT DETECTION

Distinguishing foreground objects from the stationary background is both a significant and difficult research problem. The first step of almost all visual surveillance systems is detecting foreground objects. This both creates a focus of attention for higher processing levels such as tracking and **reduces computation time considerably since only pixels belonging to foreground objects need to be dealt with.** Short and long term dynamic scene changes such as repetitive motions (e. g. waiving tree leaves), light reflections, shadows, camera noise and sudden illumination variations make the reliable and fast object detection difficult. So that the proposed system solved those problems.

4.1.1. FOREGROUND DETECTION

Foreground detection plays a very important role in a video content analysis system. It is a foundation for various post-

processing modules such as object tracking, recognition, and counting. Many approaches are proposed on this topic based on the background module and procedure used to maintain the model. In the past, the computational power of the processor limited the complexity of Foreground detection implementation.

4.1.2. THRESHOLDING

Thresholding operation is an operation **used to convert a gray scale image to binary image.** A binary image consists of 2 colours, whether it's black ('0') or white ('1'). A suitable threshold value must be selected in order to separate the object from the background. The equation for the thresholding is given by Equation (1). Fig.3 illustrates an example of thresholding operation of sub-image.

$$f(x, y) = \begin{cases} 1 & \text{'255' } f(x, y) > T_h \\ 0 & \text{'0' } f(x, y) < T_h \end{cases} \quad (1)$$

Where T = Threshold value

1	8	15	80
200	3	20	168
4	250	10	40
60	70	100	4

T=30

0	0	0	1
1	0	0	1
0	1	0	1
1	1	1	0

Fig. 3 Illustration of Thresholding Operation

In other words, the thresholded image $g(x, y)$ is defined as

$$ga(x, y) = \begin{cases} 1 & \text{if } fr(x, y) \geq thresh \\ 0 & \text{if } fr(x, y) \leq thresh \end{cases} \quad (2)$$

4.1.3. ADAPTIVE BACKGROUND SUBTRACTION MODEL

Our implementation of background subtraction algorithm is partially inspired by the study presented in [3] and **works on grayscale video imagery from a static camera.** Our background subtraction method **initializes a reference background with the first few frames of video input.** Then it **subtracts the intensity value of each pixel in the current image from the corresponding value in the reference background image.** Let $B_g(x, y)$

be the corresponding background intensity value for pixel position

(x, y) estimated over time from video images $I_o(x, y)$ through $I_{t-1}(x, y)$. As the generic background subtraction scheme suggests, a pixel at position (x, y) in the current video image belongs to foreground if it satisfies

$$|I_t(x, y) - B_g(x, y)| > T_h \quad (3)$$

This algorithm (Equation (3)) provides the most complete features data. **Moving objects in the scene are detected by the difference between the current frames and the current background model.**

The algorithm of **adaptive Background subtraction detection** is described below.

- fi : A pixel in a current frame, where i is the frame index.
- μ : A pixel of the background model (fi and m are located at the same location).
- di : Absolute difference between fi and m .
- bi : F mask - 0: background. 0xff: foreground

- T: Threshold
- a: Learning rate of the background.

$$1. di = |fi - \mu|$$

2. If $di > T$, fi belongs to the foreground; otherwise, it belongs to the background.

Although these are usually detected, they leave behind 'holes' where the newly exposed background imagery differs from the known background model. While the background model eventually adapts to these holes. They generate false alarms for a short period of time. In this algorithm, a pixel in the current frame is considered as foreground or part of the region of motion if the absolute difference between its current values and its background values is larger than thresh.

$$|I(x, y) - Bg(x, y)| > thresh$$

4.1.4. TEMPORAL DIFFERENCING

Temporal differencing makes use of the pixel-wise difference between two or three consecutive frames in video imagery to extract moving regions. It is a highly adaptive approach to dynamic scene changes; however, it fails in extracting all relevant pixels of a foreground object especially when the object has uniform texture or moves slowly. When a foreground object stops moving, temporal differencing method fails in detecting a change between consecutive frames and loses the object. Special supportive algorithms are required to detect stopped objects. We implemented a two-frame temporal differencing method in our system. Let $I_t(x, y)$ represents the gray-level intensity value at pixel position (x) and at time instance n of video image sequence $I(x, y)$ which is in the range [0, 255]. The two-frame temporal differencing scheme suggests that a pixel is moving if it satisfies the following (Equation (6)).

$$|I_t(x, y) - I_{t-1}(x, y)| > I_{th} \quad (5)$$

The algorithm of temporal differencing based foreground detection is described below.

- fi : A pixel in a current frame, where I is the frame index.
- $fi-1$: A pixel in a previous frame (fi and $fi-1$ are located at the same location.)
- dI : Absolute difference of fi and $fi-1$.
- bI : F mask - 0: background. 0xff: foreground
- T: Threshold value

$$1. dI = |fi - fi-1|$$

2. If $dI > T$, fi belongs to the foreground; otherwise, it belongs to the background

4.1.5. PROPOSED SYSTEM (HYBRID ADAPTIVE ESCHEME ALGORITHM)

We have developed a hybrid adaptive scheme algorithm for detection of moving object as you see from fig.4 by combining temporal differencing technique with background subtraction technique. As discussed above, frames differencing algorithm cannot extract all entire shape or interior object information. On the other hand background subtraction has a capability of extracting entire shape and all interior pixels information of the moving object. The drawback of background, means generating false alarm, and

unable to detect a stopped target object start to move, can be fully overcome by frame differencing. By combining the advantages of these algorithms we can get algorithm that is robust to illumination change, high computational efficiency or accuracy and highly adaptive to dynamic changes.

Frame differencing algorithm

- $fi-1$: A pixel in a previous frame, where i is the frame index.
- fi : A pixel in a current frame
- Background initialization -----Bg = fr0
- di : A pixel-wise absolute difference between fi and $fi-1$.
- $di+1$: A pixel-wise absolute difference between fi and $fi+1$.
- bi : foreground mask - 0: background. 0xff: foreground.
- T: Threshold

$$1. di = |fi - fi-1| \text{ and } di+1 = |fi - Bg|$$

2. If $di > T$ & $di+1 > T$, fi belongs to the foreground; otherwise, it belongs to the background.

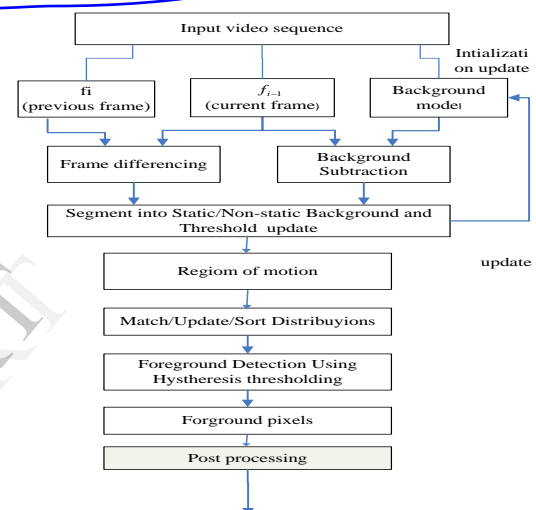


Fig. 4 Proposed technique (Hybrid adaptive scheme algorithm) block diagram

I. COMPUTING REGION OF MOTION

The first step is to compute the region of motion. FD (Frame differencing) is applied to get the boundaries of the portion of the scene that is moving. A pixel $I_t(x, y)$ is classified as foreground if the difference between $I_t(x, y)$ and its predecessors at time $t-1$ is larger than a threshold I_{th} as shown in Equation(6).

$$I_t(x, y) - I_{t-1}(x, y) > I_{th} \quad (6)$$

comparisons for all the pixels in the frame, we update only the parameters of the portion of the scene where an object is suspected and sort the corresponding distributions. Since objects are usually very small compared to the whole frame, this leads to huge reductions in computations. In other words, the proposed technique will be beneficial in two folds: Even though simple FD is less immune to noise than 3FD, any noise introduced will be corrected in the later detection stage. FD is very adaptive to dynamic changes and requires keeping one previous frame only. However, FD cannot detect all interior object information and fails to detect any if the object stops moving at a certain frame. Thus an additional background

subtraction technique is required to form the region of motion. A simple background is kept and selectively updated at each frame. A pixel in the current frame is considered part of the region of motion if the absolute difference between its current value and its background value is larger than I_{Th} :

$$|I_t(x, y) - B_t(x, y)| > I_{Th} \quad (7)$$

Initially, the background is the first frame, assuming there are no objects. I_{Th} may be initialized as shown in Equation (8)

$$I_{Th}(x, y) = m(x, y) + c * s(x, y) \quad (8)$$

where $m(x, y)$ and $s(x, y)$ are the mean and standard deviation of a local area, which is supposed to be small enough to preserve local details but large enough to suppress noise, and c defines how much of the total print object boundary is taken as a part of the given object. The whole image could also be taken as one area since the threshold will be corrected for each pixel in the training phase anyway. At each new frame, $B(x, y)$ and $I_{Th}(x, y)$ are then updated for nonmoving pixels:

$$B_{i,t}(x, y) = \alpha_1 B_t(x, y) + (1 - \alpha_1) I_t(x, y) \quad (9)$$

$$I_{Th,i,t}(x, y) = \alpha_1 I_{Th_t}(x, y) + (1 - \alpha_1) (5 * |I_t(x, y) - B_t(x, y)|) \quad (10)$$

Where α_1 is a time constant specifying the rate of adaptation. The final motion region contains interesting moving objects or uninteresting swinging of trees or even both. Eventually, only interesting moving objects should be left. So, the next step is to cluster/fill the objects in the motion area and then perform selective MoG to get the interesting objects out of the whole motion area.

II. SELECTIVE MATCH AND UPDATE

Each pixel is modeled as a mixture of K Gaussians [9]:

$$f(I_t = u) = \sum_{i=1}^K \omega_{i,t} \eta(u; \mu_{i,t}, \sigma_{i,t}) \quad (11)$$

Where $\eta(u; \mu_{i,t}, \sigma_{i,t})$ is the i^{th} Gaussian distribution also called component, $\omega_{i,t}$ is the weight or probability of each Gaussian, and K is the number of distributions, which ranges from 3 to 5. Usually, 3 Gaussian distributions are kept per pixel; at least one distribution is needed to represent foreground objects and two distributions to represent multimodal backgrounds. Increasing the number of distributions improves the performance to a certain extent at the expense of increasing memory requirements and computations. Even though K may be go up to 7, not much improvement is obtained. The threshold T is a value between 0 and 1. If T is very small, then most of the distributions will be classified as foreground and consequently one distribution at most will correspond to the background, which means that we will not be able to handle multimodal backgrounds. If T is very large, then most of the distributions will be classified as background; thus objects will quickly become part of the background. A trade off would be to choose T around 0.6. This value may vary depending on the type of the scene, whether it is a busy scene with lots of moving objects or just few objects. The next step is to compare the current pixel to these background distributions and classify it as background if it matches any of the background distributions, otherwise as a foreground pixel. Instead of using simple matching, Power and Schooners suggested using hysteresis thresholding when looking for a

match [15]. The idea is to have two thresholds, T_{low} and T_{high} . If the difference between the current pixel value and the distribution mean is less than T_{low} , the pixel is strongly classified as background. If the difference between the current pixel value and the distribution mean is larger than T_{high} the pixel is classified as foreground. Otherwise, the pixel is a foreground candidate or a weak candidate. Above $K = 5$. For each new frame, the pixels inside the motion area are checked and the corresponding parameters are updated. Given a pixel I_t in the motion area; we check the corresponding distributions for the distribution that I_t best fits or is most likely to belong to. The best match is defined as the distribution whose mean is not just the closest to I_t but also close enough to be considered alike. Let $d_{k,t}$ be the distance($I_t, \mu_{k,t}$) for $k=1:K$,

$$math = \{(\mu_{k,t}, \sigma_{k,t})d_{k,t} \quad \text{and} \quad d_{k,t} = \min[d_{1,t}, \dots, d_{K,t}]\} \quad (12)$$

For grayscale images, λ is usually 2.5, which accounts for almost 98.76% of the values from that distribution [14]. If there is a match, the corresponding parameters will be updated as

$$\mu_{i,t} = (1 - \rho)\mu_{i,t-1} + \rho I_t \quad (13)$$

$$\sigma^2_{i,t} = (1 - \rho)\sigma^2 + \rho(I_t - \mu_{i,t})^2 \quad (14)$$

Where $\rho = \alpha / \omega_{i,t}$

The approximation in(10) is faster and more logical than the one $\alpha=0.9$, $v=0.09$, $\alpha=0.005$, $T=0.7$ (non busy scene), $T=2$, $T=3.1$ init low high used in [9] for winner-takes all scenarios [15]. Note that distance the proposed technique outperforms the other techniques and is able computations may be modified to avoid square root operations. All to handle scenes with waving trees and light changes. The holes in weights are then updated as:

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha M_{i,t} \quad (15)$$

$$M_{i,t} = \begin{cases} 1 & \text{if matched} \\ 0 & \text{if nonmatched} \end{cases} \quad (16)$$

Otherwise, the component with the least weight is replaced by a Foreground Pixels correctly identified by algorithm new component with mean I_t , large variance and small weight and Precision(16) maintain the means and variances of the other components, but lower Total Fore ground detected Pixels by algorithm their weights according to (16).

III. FOREGROUND DETECTION USING HYSTERESIS THRESHOLDING

All the components are then sorted by their values of ω_i/σ_i , with the higher ranked components being classified as background. This is because high values correspond to bigger weight values and smaller variances, which means more prominent components. The first B distributions that verify (17) are chosen as background distributions:

$$B = \arg \min_b (\sum_{k=0}^n \omega_k > T) \quad (17)$$

The threshold T is a value between 0 and 1. If T is very small, the most of the distributions will be classified as foreground and consequently one distribution at most will correspond to the background, this means that we will not be

able to handle multimodal backgrounds. If T is very large, then most of the distribution will be classified as background; thus objects will quickly become part of the background. A trade off would be to choose T around 0.6. This value may vary depends on the type of the scene, whether it is a busy scene with lots of moving objects or just few objects.

The next step is to compare the current pixel to this background distribution and classify it as background if it matches any of the background distributions, otherwise as a foreground pixel. Instead of using simple matching, powers and schooners suggested using hysteresis thresholding when looking for a match [15]. The idea is to have two thresholds and T_{high} . If the difference between the current pixel value

and the distribution mean is larger than T_{high} , the pixel is classified as foreground. Otherwise, the pixel is a foreground candidate or weak candidate as shown in (14). Additional connected component check procedure is needed to classify the candidate pixels as foreground or background. If a candidate pixel is found to be 8-connected to a foreground pixel, it becomes a foreground pixel. Afterwards, morphological operators are applied to form the final foreground mask.

$$d_i = |f_r - f_{r-1}| \text{ or } d_{i+1} = |f_r - B_g| \quad \text{Or for both.} \quad (18)$$

As in equation (18) explained In this proposed algorithm or hybrid adaptive scheme algorithm we will have three kinds of foreground pixels

Foreground pixels belongs to both frame differencing and adaptive background subtraction. Foreground pixels belongs to either frame differencing or adaptive background subtraction. Foreground pixels that belong to only either to frame differencing or adaptive background subtraction. As we know the main problem with frame differencing is that pixels interior to an object with uniform intensity are not included in the set of moving object pixels. Interior pixels can be filled by applying adaptive background subtraction to extract all of the moving pixels. Let $Bgt(x, y)$ represent the current background intensity value at pixel (x, y) , then pixels that are significantly different from the background model $Bgt(x, y)$

$$bn(x, y) = \{(x, y) : |I_r(x, y) - B_g(x, y)| > thresh\} \quad (19)$$

$(X, y) \in$ moving pixels. Equation(19) will overcome the drawback of frame differencing. Then for moving and non moving object pixels the background and the threshold can be update as follows in Equation (20) and (21).

(FG=foreground pixels, BG=Non-moving object pixels)

$$B_{g_{i+1}}(x, y) = \begin{cases} \alpha * B_g(x, y) + (1 - \alpha) * I_r(x, y), & (x, y) \in BG \\ B_g(x, y), & (x, y) \in FG \end{cases} \quad (20)$$

$$thresh = \begin{cases} thresh & \text{for } (x, y) \in FG \\ \alpha * thresh + (1 - \alpha)(\gamma * (f_r - diff(x, y))) & \text{for } (x, y) \in BG \end{cases} \quad (21)$$

4.2. POST PROCESSING

The outputs of foreground region detection algorithms we explained in previous three sections generally contain noise and therefore are not appropriate for further processing without special post-processing. In the post-processing step the following basic tasks must be done before doing any

further operation so as to improve the efficiency of foreground detection mechanism. The next step is noise removal.

4.2.1. NOISE REMOVAL

The algorithms that are applied to detect moving object produce the expected foreground. However, it is highly expected to observe some noises that cannot be handled by background model. This noise affects the output of many calculation stages during the processing of a frame and overall mask becomes inaccurate due to noise. In order to get improved results, noise removal is a crucial step. Morphological operation, dilation and erosion, are applied to the foreground pixel map in order to remove noise caused by camera noise, reflectance noise, background colored object noise, and any noises that cannot handled by background model [18],[19].

4.2.2. MORPHOLOGICAL OPERATIONS

Morphological operations, erosion and dilation[20], are applied to the foreground pixel map in order to remove noise that is caused by the first three of the items listed above. Our aim in applying these operations is removing noisy foreground pixels that do not correspond to actual foreground regions.

4.3. CONNECTED COMPONENT ANALYSIS

A set of pixels in an image which are all *connected* to each other is called a *connected component*. Finding all connected components in an image and marking each of them with a distinctive label is called connected component labeling. After detecting foreground regions and applying post-processing operations to remove noise and shadow regions, the filtered foreground pixels are grouped into connected regions (blobs) and labeled by using a two-level connected component labeling algorithm presented in [18]. After finding individual blobs that correspond to objects, the bounding boxes of these regions are calculated.

4.4. MOVING OBJECT TRACKING

Tracking is generating the trajectory of an object over time by locating its position in every frame of the video. The aim of object tracking is to establish a correspondence between objects or object parts in consecutive frames and to extract temporal information about objects such as trajectory, posture, speed and direction. The tracking method we have developed is inspired by the study presented in [21]. Our approach shown in fig.5 makes use of the object features such as size, center of mass, bounding box etc. which are extracted in previous steps to establish a matching between objects in consecutive frames. Obtaining the correct track information is crucial for subsequent actions, such as event modeling and activity recognition. Once we get the filtered foreground pixels, in the next step, connected regions are found by using a connected component labeling algorithm and objects' bounding rectangles are calculated. The labeled regions may contain near but disjoint regions due to defects in foreground segmentation process. Hence, it is experimentally found to be effective to merge those overlapping isolated regions. Also,

some relatively small regions caused by environmental noise are eliminated in the region-level post-processing step.

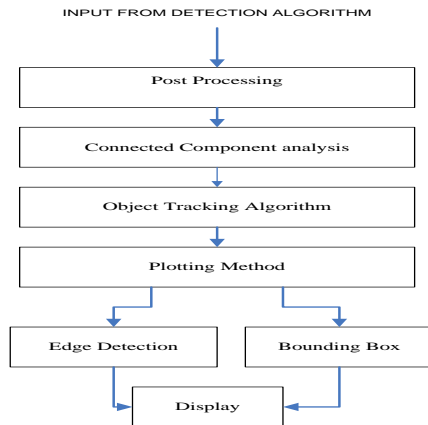


Fig. 5 Block Diagram for Object tracking system

The moving object detection and, tracking steps are dependent on each other. Thus, the tracking system would deliver inappropriate results, if one of the previous steps does not achieve good performance. An important condition in an object tracking algorithm is that the motion pixels of the moving objects in the images are segmented as accurately as possible.

4.5. MOVING A OBJECT TRACKING BASED ON REGION

This method identifies and tracks a blob token or a bounding box, which are calculated for connected components of moving objects in 2D space. The method relies on properties of these blobs such as size, color, shape, velocity, or centroid. A benefit of this method is that it is time efficient, and it works well for small numbers of moving objects. For example, [40] presents a method for blob tracking. Kalman filters are used to estimate pedestrian parameters. Region splitting and merging are allowed. Partial overlapping and occlusion is corrected by defining a pedestrian model. From the above Object tracking system flow chart, we can see that we apply morphological filters based on combinations of dilation and erosion to reduce the influence of noise, followed by a connected component analysis for labeling each moving object region. Very small regions are discarded. At this stage we calculate the following feature for each moving object region: bounding rectangle: the smallest rectangle that contains the object region. We keep record of the coordinate of the upper left position and the lower right position, what also provides size information (width and height of each rectangle). The basic principle is, after detecting foreground regions and applying post-processing operations to remove noise and shadow regions, the filtered foreground pixels are grouped in to connected regions (blobs) and labeled by connected component labeling algorithm. After finding individual blobs that correspond to objects, the bounding boxes of these regions are calculated.

4.5.1. IFD (INTER- FRAME DIFFERENCING)

The external memory is also limited that we cannot store multiple frames for processing, as we did in the matlab. This

makes impossible many complicated algorithms. From these considerations, we decide to implement a simple, direct, but effective algorithm, which here we refer to as IFD algorithm. It simply finds the difference between the current frame and a reference frame and labels the non-stationary pixels. The mean and standard deviation of the non-stationary pixels are calculated and they are used as an indication of the position and size of the foreground object. This procedure is repeated for each frame, and the tracking output is displayed on the monitor as a rectangular box.

4.6. EVENT MODELING

This thesis presents a system for adaptive moving object detection and tracking applications. The system has been designed to monitor outdoor as well as indoor environments. The final task of the architecture is to automatically provide alarms when specific events of interest are detected

4.7. GRAPHICAL USER INTERFACE DESIGN

We used Matlab software to develop the GUI shown in Fig.6. The GUI was designed to facilitate interactive system operation. GUI can be used to setup the program, launch it, stop it and display results. During setup stage the operator is promoted to choose motion detection and tracking algorithm. Whenever the start/stop toggle button is pressed the system will be launched and the selected program will be called to perform the calculations until the start/stop button is pressed again which will terminate the calculation and return control to GUI. Results can be viewed as detection and tracking consequently.

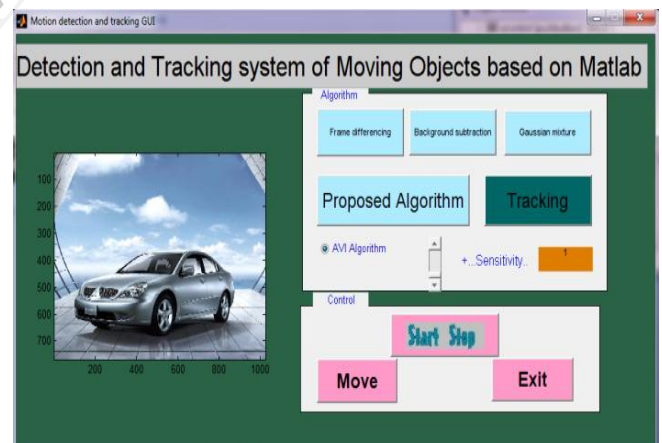


Fig. 6 GUI Layout Design

4.8. EXPERIMENTAL RESULTS

This section demonstrates some of the tested image sequences that are able to highlight the effectiveness of the proposed detection system. These experimental results are obtained using the proposed detection and tracking algorithm that has been discussed above. Fig.7 shows the difference between different algorithms and fig.8 shows the better result of the proposed detection system.

4.8.1. DIFFERENT ALGORITHMS DETECTION RESULTS



Fig. 7 Results from different algorithm, ((a1), (a2), and (a3)) are background images, ((b1), (b2), and (b3)) are feed video inputs from camera, (c1), (c2), and (c3) are detection results from frame differencing, ((d1), (d2) and (d3)) are detection results from background subtraction, ((e1), (e2) and (e3)) are detection results from proposed system (hybrid adaptive scheme)

4.9. PROPOSED DETECTION SYSTEM DETECTION RESULTS



Fig. 8 Proposed detection system detection results

As shown in fig.7, detection results form single object and multiple objects video feed, (a1)-(a3) are background image (b1)-(b3) are feed video sequences from static camera and (e1)-(e3) are results from proposed detection system, hybrid adaptive scheme algorithm. From the detection result we can see that, the algorithm determine the legitimate region(s) as well as it extract all information of moving object(s).fig 9 shows the proposed tracking algorithm results and we got the result we need.

4.10. PROPOSED TRACKING ALGORITHM RESULTS



Fig. 9 Results from proposed tracking algorithms

4.11. CONCLUSIONS AND FUTURE WORKS

Generally, this project is to develop an algorithm for moving object detection and tracking system. This algorithm is successfully implemented using Matlab integrated development environment. As a result, the algorithm is able to detect and track a moving object that is moving, as long as the targeted object emerged fully within the camera view range. The input for this project is video sequences which were captured via USB camera or built-in integrated webcam. The first step of the algorithm is to sample the video sequence into static frames. This train of frames is originally in red-green-blue (RGB). To ease computation, then RGB frames are converted to a suitable format (binary). Each frame is then being put into the filtering process, adaptive scheme algorithm, thresholding, post processing and finally, tracking process done.

The main objective of this project is to develop an algorithm that is able to detect and track moving objects. In this thesis we presented a set of methods and tools for moving object detection and tracking system. We implemented three different object detection algorithms and compared their detection quality and time-performance. Our thesis (detection and tracking system of moving objects based on matlab) includes the following two main building blocks: Moving Object Detection and Object Tracking. Moving Object detection segments the moving targets from the background and it is the crucial first step in video surveillance. In this thesis we implemented and tested temporal differencing object detection algorithms, background subtraction, and Hybrid adaptive scheme algorithms, and compared their detection quality and time performance.

The proposed technique combines simple frame difference (FD), simple adaptive background subtraction (BS), and accurate Gaussian modeling to benefit from the high detection accuracy of Mixture of Gaussian solution (MoG) in outdoor scenes while reducing the computations required, thus, making it faster and more suitable for surveillance applications. Furthermore, it gives the most promising results in terms of detection quality, speed and computational complexity to be used in surveillance system

with stationary cameras. The tracking is based on Inter-frame differencing and bounding box method. Every system has its own limitation and needs improvements.

The proposed system is not perfect in all direction. As some future works, shadow removal and sudden illumination changes process can be achieved with more robust detection algorithms which improves object detection and tracking. From tracking point of view, improvements in partial overlapping, multiple object tracking and occlusions problems can be achieved with more robust algorithm and finally we can get more robust and accurate system for video surveillance.

BIBLIOGRAPHY

- [1] A. J. Lipton, H. Fujiyoshi, and R.S. Patil. Moving target classification and tracking from real-time video. In *Proc. of Workshop Applications of Computer Vision*, pages 129–136, 1998.
- [2] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, March 2003.
- [3] R. T. Collins et al. A system for video surveillance and monitoring: VSAM final report. Technical report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May 2000.
- [4] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: A real time system for detecting and tracking people. In *Computer Vision and Pattern Recognition*, pages 962–967, 1998.
- [5] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 2: 246–252, 1999.
- [6] A. M. McIvor. Background subtraction techniques. In *Proc. of Image and Video Computing*, Auckland, New Zealand, 2000.
- [7] M. Xu and T. Ellis. Colour-Invariant Motion Detection under Fast Illumination Changes, chapter 8, pages 101–111. *Video-Based Surveillance Systems*. Kluwer Academic Publishers, Boston, 2002.
- [8] S.J. McKenna, S. Jabri, Z. Duric, and H. Wechsler. Tracking interacting people. In *Proc. of International Conference on Automatic Face and Gesture Recognition*, pages 348–353, 2000.
- [10] T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proc. of IEEE Frame Rate Workshop*, pages 1–19, Kerkira, Greece, 1999.
- [11] H.T. Chen, H.H. Lin, and T.L. Liu. Multi-object tracking using dynamical graphs matching. In *Proc. Of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 210–217, 2001.
- [12] I. Haritaoglu. A Real Time System for Detection and Tracking of People and Recognizing Their Activities. PhD thesis, University of Maryland at College Park, 1998.
- [12] X. Zhou, R. T. Collins, T. Kanade, and P. Metes. A master-slave system to acquire biometric imagery of humans at distance. In *First ACM SIGMM International Workshop on Video Surveillance*, pages 113–120. ACM Press, 2003.
- [14] J. Owens and A. Hunter. A fast model-free morphology-based object tracking algorithm. In *Proc. of British Machine Vision Conference*, pages 767–776, Cardiff, UK, September 2002.
- [15] A. Amer. Voting-based simultaneous tracking of multiple video objects. In *Proc. SPIE Int. Symposium on Electronic Imaging*, pages 500–511, Santa Clara, USA, January 2003.
- [16] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: A real time system for detecting and tracking people. In *Computer Vision and Pattern Recognition*, pages 962–967, 1998.
- [17] Y. Ivanov, C. Stauffer, A. Bobick, and W.E.L. Grimson. Video surveillance of interactions. In *International Workshop on Visual Surveillance*, pages 82–89, Fort Collins, Colorado, June 1999.
- [18] C. R. Wren, A. Azarbayejani, T. J. Darrell, and A. P. Pentland. *Pfinder: Real-time tracking of the human body*. IEEE *Pattern Recognition and Machine*
- [19] I. Haritaoglu, D. Harwood, and L. S. Davis: W4: Who? When? Where? What? A real-time system for detecting and tracking people. In *Proc. 3rd Face and Gesture Recognition Conf.*, pages 222–227, 1998.
- [20] F. Heijden. *Image Based Measurement Systems: Object Recognition and Parameter Estimation*. Wiley, January 1996.
- [21] J. Owens and A. Hunter. A fast model-free morphology-based object tracking algorithm. In *Proc. of British Machine Vision Conference*, pages 767–776, Cardiff, UK, September 2002.
- [22] Zelnio, Edmund G.; Garber, Frederick D Algorithms for Synthetic Aperture Radar Imagery XIV. *Proceedings of the SPIE*, Volume 6568, pp. 65680U (2007).
- [23] J. Heikkila and O. Silven. A real-time system for monitoring of cyclists and pedestrians. In *Proc. of Second IEEE Workshop on Visual Surveillance*, pages 74–81, Fort Collins, Colorado, June 1999.
- [24] R. Rosales and S. Sclaroff. Improved tracking of multiple humans with trajectory prediction and occlusion modeling. In *Proc. of IEEE CVPR Workshop on the Interpretation of Visual Motion*, Santa Barbara, CA, 1998.
- [25] R. Rosales and S. Sclaroff. Improved tracking of multiple humans with trajectory prediction and occlusion modeling. In *Proc. of IEEE CVPR Workshop on the Interpretation of Visual Motion*, Santa Barbara, CA, 1998.
- [26] R.T. Collins, A.J. Lipton, and T. Kanade, “A system for video surveillance and monitoring,” *Proceedings of the American Nuclear Society (ANS) Eighth International Topical Meeting on Robotics and Remote Systems*, April, 1999.
- [27] I. Haritaoglu, D. Harwood, L.S. Davis, “W4: real-time surveillance of people and their activities,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp.809–830, August 2000.
- [28] C. Stauffer and W.E. Grimson, “Adaptive background mixture models for real time tracking,” *IEEE Proc. CVPR*, pp.246–252, June 1999.

ACKNOWLEDGEMENTS

First and for most, I want thank ALLAH (s.w.t), for giving me the source of power, knowledge and strength to finish the project and dissertation for completion of my master degree final project.

I am eager to express my most sincere thankfulness to some invaluable people who made this work possible. My gratitude and appreciation go to Prof. Lili, for her always helpful suggestions in discussion and for her support and guidance in all faiths of this work .Her energy and great vision have inspired me many times; her knowledge and understanding of science as a whole have made a difficult subject simple, and turned the complicated one into uncomplicated. Also, I would like to express my sincere gratitude to my friends in TUTE.

Finally my biggest gratitude is to my family members, my Mother, Father, brothers, sisters and my beloved wife Ferdos Seid, for all their endless love, encouragement and support to this journey.