Automatic Detection of Events and Tracking of Moving Objects in Video Sequences

Mohammad Mahmood Otoom, Ph.D
Assistant Professor
Department of CSI,College of Science,
Al- Majmaah University, KSA

Khalid Nazim Abdul Sattar, Ph.D
Assistant Professor
Department of CSI,College of Science,
Al- Majmaah University, KSA

ABSTRACT
Digital video is being used widely in a variety of applications such as surveillance and security. Big amount of video in surveillance and security requires systems capable to process video automatically to detect events and track moving objects to alleviate the load on humans and enable preventive actions when events are detected [4]. our paper focuses to develop an intelligent visual surveillance system to replace the traditional passive video surveillance that is proving ineffective as the number of cameras exceeds the capability of human operators to monitor them, and it is able to track objects within a maximum solid angle speed which is measured at about 0.3 to 0.2 radian per second, further it also depends on the complexity of the system and the processor speed as well.

Keywords
Surveillance, background subtraction, cumulative distribution function (CDF), probability density function (PDF), Mixture of Gaussians (MoG).

1. INTRODUCTION
The traditional Video Surveillance systems based on human interaction are ineffective as the number of cameras exceeds the capability of human operators to monitor them. Therefore the active research topics in surveillance systems using real-time video attempts to detect and track some objects from sequences of images, and to describe how objects behave. In short, the goal of surveillance system is not only to have the cameras in place of security guards, but also to monitor all surveillance missions as automatically as possible [8].
Identifying moving objects from a video sequence is fundamental and critical task in surveillance systems, motion detection and tracking. A common approach to identify the moving objects is background subtraction, where each video frame is compared to another frame as a reference or compared to a previous frame. Pixels in the current frame that change significantly from the background are considered to be foreground pixels or moving objects, these foreground pixels are further processed for object localization and tracking.
In addition as there are many different background subtraction methods available with varying characteristics [7].

2. LITERATURE SURVEY
This section describes in detail the different methods, algorithms and the main principles that are used in object tracking and surveillance systems.

2.1 Histograms
Histograms are the basis for many image processing techniques, they provide tools for image statistics, as well as they provide image enhancement techniques. Histograms are preferred techniques because they are simple for software implementation and cheaper for hardware implementations.
The image with a low contrast has a narrow histogram, usually located in the middle of the gray scale, while the image that has a high contrast, its histogram covers a broad range of the gray scale [1].

2.1.1 Histogram Equalization
If we assume that a given image has a continuous range of levels from 0 to 1 and let p(r) the probability density function (PDF) of the these levels and T(r) is the cumulative distribution function (CDF) of the system. the transformation will be:

$$s = T(r) = \int_0^r p(w)dw \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

and this system will have a uniform PDF at the output $p_o(s)$, thus:

$$p_o(s) = \begin{cases} 1\,; for\ 0 \le s \le 1 \\ 0\,; otherwise \end{cases} \ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

This results in an image with a very high contrast and its intensity is distributed equally throughout the range and if we are working with discrete intensity levels, $p(r_j)$ which is the normalized histogram of the input image with the discrete intensity values (j=0, l, 2...L-1) ,and T (r_k) is the Histogram Equalization. The integration becomes summation with discrete values, and thus the transform function results in

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} (n_j/n) \ldots\ldots\ldots\ldots \quad (3)$$

where n_j is the number of pixels with the intensity level j and n the total number of pixels. s_k maps the levels into the range (0,1), so the cumulative distribution function which is normalized to (0, L-1) is:

$$c(r_k) = \sum_{j=0}^{k} n_j \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

And the general histogram equalization formula is:

$$h(r_k) = X_{min} + (X_{max} - X_{min}).\ s_k \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

where X_{min} and X_{max} are the minimum and maximum permissible luminance values., r_k is the input pixel value and $h(r_k)$ is the output pixel value after equalization. [1][3][5][6][11].

2.1.2 Full-scale Histogram Stretch
The full-scale histogram stretch, or the contrast stretch, is a simple technique that expands the image histogram to fill the entire available gray-scale range. If an image has a compressed histogram with maximum gray-level value B and minimum value A, this operation maps gray levels A and B in the original image to gray levels 0 and L - 1 in the transformed image. This can be expressed as follows:

$$g(n) = ((L-1)/(B-A)) * (f(n)-A) \ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Where $f(n)$ and $g(n)$ are the input value and the output value respectively[2].

2.2 Background Subtraction
Background subtraction and modeling techniques compare

incoming images to a learned background. In the case of background subtraction, an image subtraction followed by thresholding is performed to determine foreground pixels.

$I_{motion} = | I_{input} - I_{background} | > T$. . . . .(7)

where I_{motion} is the output motion image, I_{input} is the input image, $I_{background}$ is the background image and T is the motion threshold. Depending on the implementation, $I_{background}$ may be a fixed image initialized at start up, or may adapt to changes in the scene (i.e. lighting fluctuations).

There are many different background algorithms designed to perform the background subtraction and each algorithm has different advantages and limitations of its efficiency or processing time, and each algorithm has to fulfill some requirements such as robustness; it must be adaptable to environmental changes or it must be fast enough to assure the information being analyzed is still meaningful and it has to consume as little computer recourses as possible [4][8] [21].

The most common background subtraction algorithms are described as follows:

2.2.1 Frame Differencing

This method is the simplest form of background subtraction. The current frame's pixels are subtracted from the previous frame's pixels, and if the difference of pixel values is greater than a threshold T, the pixel is considered part of the foreground[13][16].

$FG=|I(x,y,t) - I(x,y,t-1) | > T$. . . . . (8)

This method does have two major advantages; the first advantage is the modest computational load. The second advantage is that the background model is high adaptive. Since the previous frame considered the background reference, this method can quickly adapt to changes in the background more than any other method. On the other hand, there are some flaws that the moving objects in the scene must be moving continuously, otherwise they will be recognized as background in subsequent frames [10][17].

2.2.2 Temporal Median and Mean Filtering

The information from previous frames is accumulated in order to get the average value for each pixel. This method creates a buffer of the last N frames and models the background as the median or mean of those frames, then (as with frame difference), the current frame's pixels are subtracted from the corresponding background's pixels then compare the difference of pixel values to a threshold value to determine the foreground pixels. Thus, in its simplest form, the background model is given by [12][20][22]

$B(x, y, t) = median \{ I(x,y,t-i) , i \in \{0, 1 . . . . n-1\}$. . . (9)

and in case of the mean background model, the background can be modeled by a running average:

$B(x, y, t) = \frac{1}{n}\sum_{i=0}^{n-1} I(x,y,t-i)$ . . . .(10)

Where I(x,y,t) is the instantaneous pixel value for the (x, y) pixel at time t. This can also be computed incrementally where $1/t$ is the learning rate:

$B(x, y, t) = (1-(1/t))\, B(x, y, t-1) + (1/t)\, I(x, y, t)$. . . (11)

This method is very adaptive but not as fast to adapt as the frame difference method and it is very robust and have good performance in most applications. However, this approach needs a lot of memory and computations to store and process several frames. .However, storing and processing frames at a rate lower than the frame rate can lower storage and processing requirements at the expense of a slower adapting background [9][24].

2.2.3 Temporal differencing

Temporal differencing is based on frame difference which attempts to detect moving regions by making use of the difference of consecutive frames (two or three) in a video sequence. This method is high adaptive to dynamic scene. On the other hand it is not sufficient to extract the entire shape of moving objects. (i.e. there may be holes left inside moving objects).

The difference between images is compared to adaptive threshold to get the foreground region, and the adaptive threshold T_d can be derived from image statistics.

In order to detect cases of slow motion or temporally stopped objects, a fixed weight is used to compute the temporal difference, and the pixel is considered part of the foreground if the difference for a given pixel is greater than a threshold T_d.

$FG = I_a(x,y,t)>T_d$. . . . .(13)

Where

$I_a(x, y, t) =(1-w)\, I_a(x, y, t-1) +w\, |I(x, y, t) - I(x, y, t-1)|$. . . . .(14)

and

$T_d = A^* mean\, (I_a(x, y, t -1))$ . . . . (15)

where w is a real number between 0 and 1 which describes the temporal range for difference images, A is a factor and $I_a(x, y, t -1)$ is initialized to an empty image[14][15][18].

2.2.4 Mixture of Gaussians (MoG)

The background model in MoG is modeled as a probability distribution function (PDF) for each pixel instead of a frame values, this PDF represents the sum of Gaussian functions:

$P(X_t) = \sum_{i=1}^{K} \omega_{i,t}\; \eta(X_t , \mu_{i,t} , \Sigma_{i,t})$ . . . . (16)

$\eta (X_t , \mu , \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}(|\Sigma|)^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu t)^T \Sigma^{-1}(X_t-\mu t)}$. . . .(17)

$\Sigma_{k,t} = \sigma_k^2 I$. . . . .(18)

where K is the number of distributions, $\omega_{i,t}$ is an estimate of the weight of the i^{th} Gaussian in the mixture at time t, $\mu_{i,t}$ is the mean value of the i^{th} Gaussian in the mixture at time t, $\Sigma_{i,t}$ is the covariance matrix of the i^{th} Gaussian in the mixture at time t, $\sigma_{i,t}$ is the standard deviation and η is a Gaussian probability density function. Typically, K ranges from three to five, depending on the available storage. For each input pixel X_t, the first step is to identify the component i' whose mean is closest to X_t. Component i' is declared as the matched component if:

$| X_t - \mu_{i',t-1}| \leq D. \sigma_{i,t-1}$. . . . .(19)

where D defines a small positive deviation threshold. The parameters of the matched component are then updated as follows :

$\omega_{i',t} = (1-\alpha)\, \omega_{i',t-1} + \alpha$. . . . . (20)

$\mu_{i',t} =(1-\rho)\, \mu_{i',t-1} +\rho\, X_t$. . . . .(21)

$(\sigma_{i',t})^2= (1-\rho)\, (\sigma_{i',t-1})^2+\rho(X_t - \mu_{i',t})^2$. . . . .(22)

where α is a user-defined learning rate with $0\leq \alpha\leq 1$. ρ is the learning rate for the parameters and can be approximated as follows:

$\rho \approx (\alpha / \omega_{i',t})$ . . . . (23)

If there are no matched component, the component that has the least weight is replaced by a new component with mean X_t , a large initial variance σ_o and a small weight ω_o. The rest of the components maintain the same means and variances, but lower their weights to achieve exponential decay:

$\omega_{i,t} = (1-\alpha)\, \omega_{i,t-1}$. . . . .(24)

Finally, all the weights are renormalized to sum up to one. To determine whether X_t is a foreground pixel, first, all components are ranked by their values of $\omega_{i,t} /\sigma_{i,t}$. Higher-rank components thus have low variances and high probabilities that are typical characteristics of background. If $i_1,i_2,....,i_k$ is the component order after sorting, the first M components which satisfy the following criterion are declared to be the background components:

$$\sum_{k=i_1}^{i_M} \omega_{k,t} \geq T \quad (25)$$

where T is the weight threshold. X_t is declared as a foreground pixel if X_t is within D times the standard deviation from the mean of any one of the background components. The formula (25) can be easily extended to handle color data. The complexity of computations and storage requirement of MoG is proportional in terms of the number of components K [14][23].

2.3 Object Tracking

The problem of generating an inference about the motion of an object is given as a sequence of images is called tracking. ln a tracking scenario, this problem is addressed by defining the following terms:

2.3.1 Object representation

An object can be defined as anything that is of interest for further analysis. For instance, cars, boats, people are a set of objects that a surveillance system wants to track. The object can be represented by their shapes or appearances; a single point for example can represent the object one wants to track. Objects can be represented by the centroid of the object's points, Primitive geometric shapes, Object silhouette and contour (the boundary of an object), Articulated shaped models, applying medial axis transform to the object silhouette, or histograms of Gaussian parameters [9][19][21].

2.3.2 Features selection for tracking

Object representation is a way to represent objects, but those are not parameters that one uses for tracking objects. Features for tracking are those that one uses to identify objects in future frames. For instance one can represent objects by its bounding box, which one can use for tracking. In general, the most desirable property of a visual feature is its uniqueness, which allows objects to be easily distinguished in the feature space. Some of those features are [9][19][21]:

- Color: this feature is a good choice for objects with a unique color. However, it is affected by light changes and the surface reflectance of the objects.
- Edges: this feature less sensitive to light changes compared to color features. However, object boundaries are usually different from one frame to another in a sequence, even in consecutive frames.
- Optical flow: information of the pixel displacements is contained in each optical flow vector.
- Texture: This feature represents variation of the intensity of a surface which quantities properties such as smoothness and regularity.

2.3.3 Object trajectory tracking

An object trajectory is locating the object position in every frame of the video. An object tracking method can also provide a complete region in the image that is occupied by the object at every time instant. Main object tracking methods are [9][19][21]:

- Point tracking: moving objects detected in consecutive frames are represented by associated points based on the previous object state.
- Kemel tracking: Kemel refers to the object shape and appearance. Object tracking is performed by computing the motion of the object shape in consecutive frames.
- Silhouette tracking: objects are tracked by estimating the object region in each frame.

3.METHODOLOGY

Our system is composed of three main stages, namely: Preprocessing, Motion Detection and Foreground Segmentation, Object Tracking, (see Figure 3.1). The Preprocessing stage runs some processing on the captured frames which makes the next stage easier to achieve. The Motion Detection and Foreground Segmentation stage receive the pre-processed information. It detects any motion using one of the described Background Subtraction methods; then it generates a binary image for further processing in Object Tracking which represents the last stage.

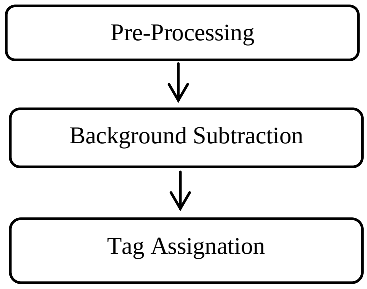


Fig 3.1: General Block Diagram

3.1 Pre-processing

The pre-processing is composed of four different steps (see Figure 3.2), where the input webcam streaming video is received to extract the region of importance in which the event is defined. It is then converted from the RGB color space to grayscale images for further processing and the histogram equalization is applied to improve the images contrast.

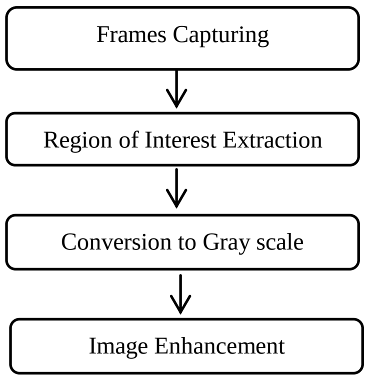


Fig 3.2: Pre-Processing Block Diagram

3.1.1 Frames capturing

Initially a video is displayed from a webcam for traditional monitoring. While displaying the video, a series of frames are captured automatically each a period of time depends on the required processing time. The default resolution of a displayed video from a webcam is 320 x 240 pixels, which is upscale by 2 to be 640 x 480 for better monitoring.

3.1.2 Region of Interest Extraction

It is recommended to determine the areas where events more likely take place in order to focus the detection analysis in the most suitable parts where the event existence probability is higher (i.e. a door). Using the region of interest in the event detection task will help to decrease the probability of false detections. It will be also useful to reduce the complexity of the system in future processing tasks because a part of the scene is required to be processed not the whole scene, then the decreased use of memory resources.

3.1.3 Conversion to Grayscale
This process done by converting from the RGB color space to the YCbCr color space in order to work only with the Y component, where Y is the luminance component that gives the average brightness of the image. And Grayscale images are related to the luminance component which is represented by Y component in the YCbCr color space. Therefore, Working only with one component instead of three, needs less memory resources.

3.1.4 Image Enhancement
In this step the histogram of the input image is calculated in array size of number of gray levels (L-1), then this histogram can be used in histogram stretching or in histogram equalization techniques to enhance the image contrast.
In Histogram Equalization the cumulative density function of the histogram is calculated in the same size histogram array, then each pixel's value in the output image is calculated using the general histogram equalization formula (5).
In Histogram stretching the minimum and the maximum level of the histogram are determined, then the histogram is mapped to gray levels 0 and L – 1 using the formula (6).

3.2 Background Subtraction
There are many different background algorithms designed to perform the background subtraction, each one with different advantages and limitations related to efficiency, quality, processing time and complexity.in view to the several background subtraction algorithms that are implemented, the temporal differencing and the Mean Filtering methods turned out to be the best option for most applications because it is fast, easy to implement, and handles noise relatively well.

3.2.1 Frame Differencing
The current frame is simply subtracted from the previous frame, and if the difference in values for a given pixel is greater than a threshold T, the pixel is considered part of the foreground region, the same code was used for frame differencing method it could be used for snapshot method which simply involves waiting until there has been no movement in the scene for a period of time, and then take a snapshot as the background frame instead of the previous frame.

3.2.2 Temporal Mean Filtering
The procedure to get the background model is as discussed, at first all pixels stored in a matrix (m x n), where *m* is the total number of pixels in the image and *n* is equal to the number of frames for training sequence, all brightness values for each pixel in each row are summed and divided by the number of frames for training sequences, or they are computed incrementally using the formula (11). The result is chosen to represent the initial background model for that pixel and so on.

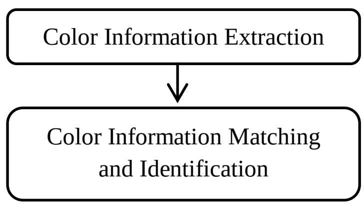
3.2.3 Temporal differencing
In this method, the temporal difference for each pixel is computed incrementally using the formula (14), then the result is compared to the adaptive threshold T_d to determine the foreground pixels. The threshold is computed by formula (15) and the factor is set to an integer value (i.e. 1, 2 or 3).

3.2.4 Mixture of Gaussians
As discussed in the literature review, we maintain a set of Gaussians values for each pixel of the background. Each Gaussian has a mean, variance, and weight, which are updated using formulas (20),(21) and (22) if criterion (19) is satisfied. Otherwise, maintain the same means and variances and update weights using formula (24), after that all components are ranked by their values of $\omega_{i,t}/\sigma_{i,t}$ and the first M components that satisfy criterion (25) are declared to be the background components, then each pixel is within D times the standard deviation from the mean of any one of the background components is declared as a foreground pixel. The tuned parameters in this method are the learning rate α, the weight threshold T and deviation threshold D.

3.3 Object Tracking
The binary image which has produced in background subtraction stage is used to calculate the moving object's centroid to track any object by its centroid, or to label groups of connected foreground's pixel (i.e. white pixels) in order to construct a region which represents a moving object, in order to track this object by its color information.

3.3.1 Object Tracking by the Centroid
This process can be thought of as calculating the center of the foreground pixels (i.e object pixels), by dividing the sum of foreground pixels' locations over the total of the foreground pixels; $C= (\Sigma_i r_i)/M$, *w*here M is the total of the foreground pixels, and r_i is the pixel location. The centroid is calculated twice, once for 'x' locations and once for 'y' locations.

3.4.2 Object Tracking by Color Information
The color information is extracted from the moving objects in the current frame to categorize matching color information between moving objects in the current frame and previous frames. Subsequently, a tag is assigned to the moving object in the current frame (see figure 3.3).
For each moving object in the current frame, color information is derived. This color information consists of 3 color channels, i.e. Red, Green and Blue (RGB), and the dominant color which has the greatest value is used to compare and identify matching color information between the moving objects in the current frame and all moving objects in the previous frames, this is repeated for every moving object in the current frame and for each comparison made, the processor computes a respective comparison score.
Once the average comparison score of the moving object in the current frame is computed, the processor assigns tags to the moving objects in the current frame with either a new tag or an old tag depending on the average comparison score computed for the moving object in the current frame and all moving objects in the previous frame. If a moving object in the previous frames is tagged as N, and the moving object in the current frame has an average comparison score that is higher than a predetermined threshold of the moving object in the previous frames, then the moving object in the current frame will be assigned with the same tag N. Otherwise, the moving object in the current frame will be assigned with a new tag.



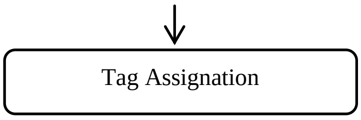


Fig 3.3: Object Tracking Using Color Information block diagram

4. EXPERIMENTAL RESULTS

This section deals with the various experimental results that we have obtained by using the different evaluation methods as discussed below:

4.1 Histograms Methods

For the system, contrast enhancement using Histograms methods is desired because it facilitates the differentiation between the object and the background. Consider the following Figure 4.1 that depicts a person's moving hand, note how after the contrast enhancement, the person's hand and the background tend to be mostly white and mostly black which makes the object easier to recognize.

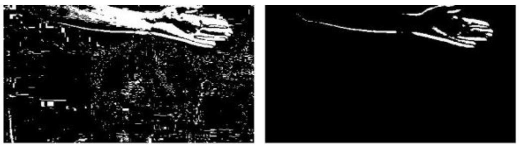


Fig 4.1: Foreground Segmentation without/with contrast enhancement respectively.

4.2 Background Subtraction Methods

We started with the snapshot method which is analyzed as a low complexity approach. It is fast and easy to implement, but there is no guarantee that there will be a period when the entire image is stationary, so this method is difficult to be implemented for a long period as well as it cannot be implemented outdoor because of lighting conditions and it is also possible that some elements of background may actually move continuously, such as trees moving in a breeze. The results show a tradeoff between the processing speed and complexity and it also shows limitations and advantages of each method

4.2.1 Frame Differencing

The frame differencing method is analyzed as a low complexity approach. It is fast and easy to implement, and it can adapt to changes in the background faster than any other method. However, it is dependent on continuous motion and sensitive to noise, and as objects move, their homogeneous interiors do not result in changing image intensities over short time periods. Therefore, motion can only be detected at the boundaries, (see Figure 4.2(a)).

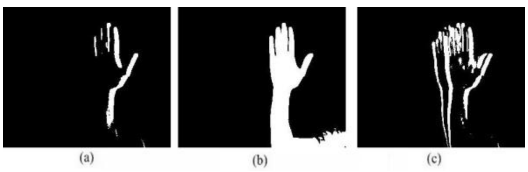


Fig 4.2: (a) Frame differencing (b) Mean Filtering (c) Temporal Differencing

4.2.2 Mean Filtering

The mean filtering method is analyzed as a medium complexity approach. It is easy to implement and is more robust than the Frame Difference method. It offers performance near what we can achieve with higher-complexity methods, therefore is less sensitive to noise, but it consumes a lot of memory resources as it is necessary to store several frames. However, this method also determines the most pixels of moving objects which makes it more suitable for foreground segmentation, (see Figure 4.2(b)).

4.2.3 Temporal Differencing

The temporal difference is analyzed as a medium complexity approach, it is similar to frame difference but it is less dependent on continuous motion and the adaptive threshold T_d can restrain the noise very well, but it creates ghost regions in the binary image, (see Figure 4.2(c). All the previous methods, if the scene is not static, they will very sensitive to any movement and is difficult to differentiate the true and false movement.

4.2.4 Mixture of Gaussians

In Mixture of Gaussians a different thresholds is selected for each pixel, and these thresholds are adapting by time. In this method also objects are allowed to become part of the background without destroying the existing background model, and it provides fast recovery. However, this method is analyzed as a high complexity approach; its implementation is more difficult because it has to deal with the parameter optimization of its five variables that generates significant impact on the performance of the algorithm. Although this method is more elegant, it consumes a significantly larger amount of computer resources and time than the former; (see Table 1), and it cannot deal with sudden lighting changes.

Table 1. The average time taken to process 100 frames for each background subtraction method

Background Subtraction Method	Time of processing 100 frames
Frame Differencing	6 to 7 seconds
Mean Filtering	7 to 8 seconds
Temporal Differencing	7 to 8 seconds
Mixture of Gaussians	20 to 21 seconds

4.2.5 Threshold Value in Frame Differencing

The frame differencing algorithm has the threshold parameter which was changed for different values (threshold=1, 5 and 15) according to the observations showed in Figure 4.3, higher values reduce the existence of foreground pixels while lower values lead to the presence of noise; A threshold value of 5 was considered to be the most suitable to reduce the noise and have enough foreground pixels.

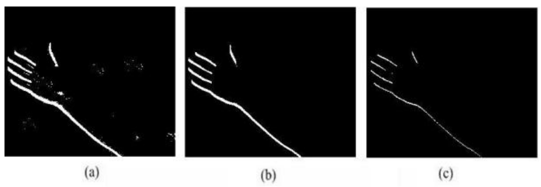


Fig 4.3: Threshold comparison for Frame Difference method, (a) T=1 (b) T=5 (c) T=15

4.2.6 Threshold Value in Temporal Mean Filtering

The Temporal Mean Filter algorithm also has the threshold parameter which was changed for different values (threshold=1, 5 and 15) according to the observations showed in Figure 4.4, higher values reduce the existence of foreground pixels while lower values lead to the presence of noise, 5 was the most suitable value to reduce the noise and have enough foreground pixels.

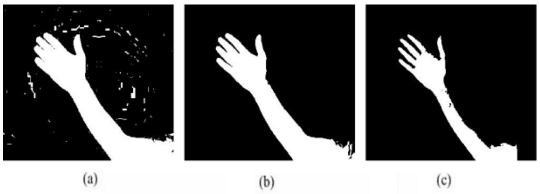


Fig 4.4: Threshold comparison for Temporal Mean Filtering method, (a) T=1 (b) T=5 (c) T=15

4.2.7 Adaptive Threshold Value in Temporal Differencing

In this method, we set T_d as the three times of mean value of $I_a(x, y, t -1)$ and $w = 0.5$ for all the results. Figure 4.5 below shows the results of temporal differencing method under a simulation environment which has a static background.

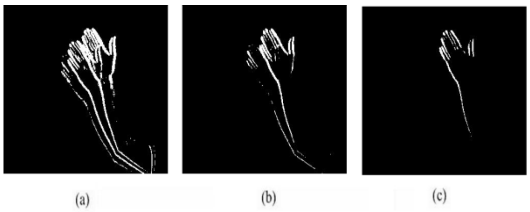


Fig 4.5: Threshold comparison for Temporal Differencing method. $T_d = A \times mean(I_a(x, y, t -1))$ where (a) A=1(b) A=3 (c) A=5

The Temporal Differencing algorithm also has the threshold parameter which was changed for different values (threshold=1*mean(I_a(x, y, t -1)), 3*mean(I_a(x, y, t -1)) and 5*mean(I_a(x, y, t -1)). according to the observations showed in Figure 4.5, higher values reduce the existence of foreground pixels while lower values lead to the presence of ghost regions in the image, 3*mean (I_a(x, y, t -1) was the most suitable value to reduce the ghost regions and have enough foreground pixels.

4.3 Object Tracking Methods

The Object Tracking by centroid method calculates the centroid of all white pixels in the binary image, so it is capable to track only one object, while the existing of more than one object in the scene leads to mistaken tracking.

In this stage, the results is shown by measuring the maximum solid angle speed in radian per second that this system can still be able to track, this is done by dividing the horizontal object speed by object's distance from the camera, (see Table 2 and Table 3).

Table 2. the maximum solid angle speed in object tracking by calculating the centroid

Distance from the camera	Horizontal Distance	Time	Max. Horizontal Speed	Max. solid angle speed
4.5 m	3 m	2.5 s	1.2 m/s	0.26 rad/s
3m	3 m	3 s	1 m/s	0.33 rad/s
2.5m	3 m	3.5 s	0.86 m/s	0.34 rad/s

Table 3. the maximum solid angle speed in object tracking using color information

Distance from the camera	Horizontal Distance	Time	Max. Horizontal Speed	Max. solid angle speed
4.5 m	3 m	4 s	0.75 m/s	0.17 rad/s
3m	3 m	4.5 s	0.66 m/s	0.15 rad/s
2.5m	3 m	5.5 s	0.55 m/s	0.22 rad/s

From these results we can see that the maximum solid angle speed is about 0.3 rad/sec in object tracking by calculating the centroid method and it is about 0.2 rad/sec in object tracking using color information, where the system can still be able to track.

5. CONCLUSION

Our work has presented a working system for detection and tracking of objects in real-time video sequences, and this proposed system is capable of:

- Motion detection and Tracking moving objects in video sequences.
- Implementing several different background subtraction algorithms.
- Choosing a region of interest in a video sequence.
- Handling noise due to video conditions using some image enhancement techniques.

On the other hand, the system is subject to the following limitations:

- A single, static camera setting.
- Processing time and memory capacity limits affect the video size and the amount of information that can be extracted.
- Limited number of tracked objects present at a given time in the video. Due to the limitation in memory and analysis time.
- The system needs a minimum of contrast between the object and the background in order to work.
- The system needs for tracking an object to be moving continuously across the scene.

6. FUTURE WORK

Possible areas for future work related to this paper include:

- In real time based systems, Video shall constitute as a good early alarm device for security applications.
- If a system with several cameras is used, the system could constitute a trigger event that will direct cameras that are closer to the object to track and focus on the object in order to extract more valuable information.
- More data can be incorporated in the tracking using occlusion handling algorithm.
- A wide variety of videos with varying conditions and objects could improve the overall evaluation of the system.

7. REFERENCES

[1] Gonzalez, R.C. & Woods, R.E. 2007. *Digital Image Processing*. third edition. Prentice Hall.

[2] Bovik, A. 2005. *Handbook of Image and Video Processing.* Second edition. Canada, Academic Press.

[3] Singha, K. Kapoorb, R. 2014. Image enhancement via Median-Mean Based Sub-Image-Clipped Histogram Equalization. Optik - International Journal for Light and Electron Optics, September.

[4] Singh, B. Singh, D. Singh, G. Sharma, N. 2014. Motion detection for video surveillance. Signal Propagation and Computer Technology (ICSPCT) International Conference, 12-13 July.

[5] Khan, M., Khan, E. Abbasi, Z.A. 2014. Segment selective dynamic histogram equalization for brightness preserving contrast enhancement of images. Optik - International Journal for Light and Electron Optics, February.

[6] Singha, K. Kapoorb, R. 2014. Image enhancement using Exposure based Sub Image Histogram Equalization. Pattern Recognition Letters, 15 January.

[7] Al-Berry, N. Salem, A. Hussein, S. Tolba, F. 2013. Motion Detection using Wavelet-enhanced Accumulative Frame Differencing .Computer Engineering & Systems (ICCES) International Conference, November.

[8] Catherine. 2013. Implementation of background subtraction algorithm for motion detection. New Media Studies (CoNMedia) Conference, November.

[9] Kamaraj, M. Balakrishnan. 2013. An improved motion detection and tracking of active blob for video surveillance. Computing, Communications and Networking Technologies (ICCCNT) Fourth International Conference, July.

[10] Wirayuda, B. Laksitowening, A. Sthevanie, F. Rismala, R. 2013. Development methods for hybrid motion detection (frame difference-automatic threshold). Information and Communication Technology (ICoICT) International Conference, March.

[11] Celik, T. 2012. Two-dimensional histogram equalization and contrast enhancement. Pattern Recognition, October.

[12] Huang, S. Cheng, F. 2012. Motion detection with pyramid structure of background model for intelligent surveillance systems. Engineering Applications of Artificial Intelligence, October.

[13] Bang, J. Kim, D. Eom, H. 2012. Motion Object and Regional Detection Method Using Block-Based Background Difference Video Frames, Embedded and Real-Time Computing Systems and Applications (RTCSA) IEEE 18th International Conference, August.

[14] Cheng, F. Ruan, S. 2012. Accurate Motion Detection Using a Self-Adaptive Background Matching Framework. Intelligent Transportation Systems IEEE Transactions, June.

[15] Huang, S. 2011. An Advanced Motion Detection Algorithm With Video Quality Analysis for Video Surveillance Systems. Circuits and Systems for Video Technology IEEE Transactions, January.

[16] Zhao, Y. Liu, J. 2010. An improved method for human motion detection and application. Image and Signal Processing (CISP) 3rd International Congress, October.

[17] Fang, L. & Meng, Z. 2009. Smart Motion Detection Surveillance System. IEEE International Conference on Education Technology and Computer, 17-20 April.

[18] Lu, N., Wang, J., Wu, Q.H. & Yang, L. 2008. An Improved Motion Detection Method for Real-Time Surveillance. IAENG International Journal of Computer Science, 19 February.

[19] Yilmaz, A. Javed, O. & Shah, M. 2006. Object Tracking: A Survey. ACM computing surveys, December.

[20] Piccardi, M. 2004. Background subtraction techniques: a review. IEEE International Conference on Systems, Man and Cybernetics, 10-13 Oct.

[21] Hu, W. Tan, T. Wang, L. & Maybank, S. 2004. Survey on Visual Surveillance of Object Motion and Behaviors. IEEE Transactions on systems, man and cybernetics, August.

[22] Cheung, S.S. & Kamath, C. 2004. Robust techniques for background subtraction in urban traffic video. SPIE Electronic Imaging In Proceedings of Video Communications and Image Processing, January.

[23] Stauffer, C. & Grimson, W.E.L. 1999. Adaptive background mixture models for real-time tracking. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 23-25 June.

[24] Friedman, N. & Russell, S. 1997. Image segmentation in video sequences: a probabilistic approach. Computer Science Division, University of California, Berkeley, Thirteenth Conference on Uncertainty in Artificial Intelligence.