Background Modeling For Tracking Object Movement

Yue Feng
School of Electrical and Computer Engineering
RMIT University, Australia.
yue.feng@innovonics.com.au

Alan L. Harvey
School of Electrical and Computer Engineering
RMIT University, Australia.

Andrew Jennings
School of Electrical and Computer Engineering
RMIT University, Australia.

Abstract

A background modeling method for tracking object movement in a cluttered scene is proposed in this paper. The emphasis of the system is on the analysis of background information gathered solely from an image sequence. Two main processes are implemented in this system. The first process uses a novel technique to extract an initial background model from an image sequence. The second process updates the background model, when the background situation changes at a later time. Extensive knowledge of segmentation and edge detection is used in the modeling. It is planned to implement the algorithm on a personal computer to produce a real time working system. The image capture rate was 10 images/second. The image sequence may contain the moving objects during the background extract in stage.

Keywords Image processing, background modeling, edge detection, difference image

INTRODUCTION

This paper presents a novel technique for a low-cost PC based read-time visual modeling system; called a background modeling system, for simultaneously tracking movement object, and monitoring their activities in monochromatic video. This system also constructs dynamic background models to isolate object movement. Background modeling system has been designed to work with monochromatic stationary video sources, either visible or infrared.

The remainder of the paper is organized as follows. Section 2 describes the segmentation that employs an adaptive double window modified trimmed mean (DW-MTM) filter for filtering the input image in our real time system and a local neighborhood edge operator in a low level image processing. In this section, we modified a novel cluster analysis technique to extract some interesting points to labeling object movement and issued background model. Section 3 describes the background model updating. Section 4, experimental results are reported and summarized a complete algorithm. Discussions and conclusions are provided in section 5.

A simple and commonly used background modeling method involves subtracting each new image from a model of the background scene and thresholding the resulting difference image to determine foreground pixels. The pixel intensity of a completely stationary background can be reasonably modeled with a normal distribution, and it can adapt to slow changes in the scene by recursively updating the model. However, those approaches have difficulty in modeling backgrounds in a complex and very varied environment such as some lighting changes. In this case, more than one process may be observed over time at a single pixel. In [1], a mixture of three normal distributions was used to model pixel values for traffic surveillance applications model road, shadow, and vehicle. In [2], pixel intensity is modeled by a mixture of K Gaussian distributions (typically, K is three to five). [3] uses a nonparametric background model by estimating the probability of observing pixel intensity values based on a sample intensity values for each pixel. [4] uses a model of background variation that is a bimodal distribution constructed from order statistics of background values during a training period. They apply to obtain the background model even if there are moving foreground objects in the field of view, such as walking people, moving cars, etc. Our background modeling method developed in our laboratory creates a background scene model which obtains a non-movement object binary image when there are moving foreground objects or if there are not moving foreground objects in the field of view.

The major features of background modeling are as follows:

- Leaning initial background model when the system starts up.
- Updating background model.
- Issue a background model for the future tracking.

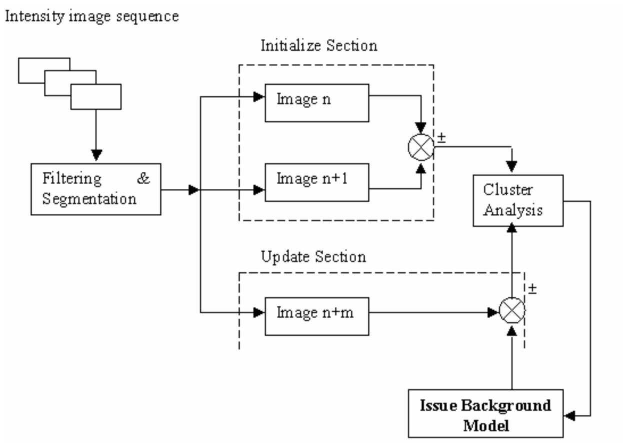

Figure 1 System architecture of background modeling

The block diagram in Figure 1 shows the system architecture for background modeling. In the first stage, a sequence of raw intensity images is first fed to a segmentation process to binaries and subtracts two input image to a binary image. In second stage, we perform a cluster analysis to obtain some interesting points from the binary image. Then, we issue a background model from the result of the cluster analysis.
The Image used in this research was taken from a vertical direction movement in the cluttered monitoring area. The image was digitized from a video camera with an initial pixel resolution of 768 × 576 pixels. The resolution was reduced to 384 × 288 pixels for processing. At this resolution, 384 × 288 pixels were equivalent to about 3 × 2.3 square meter on the monitoring area. The test image sequence (figure 2) was captured and digitized at the rate of 10 images / per second – the fastest rate possible with the computing equipment available for this work.

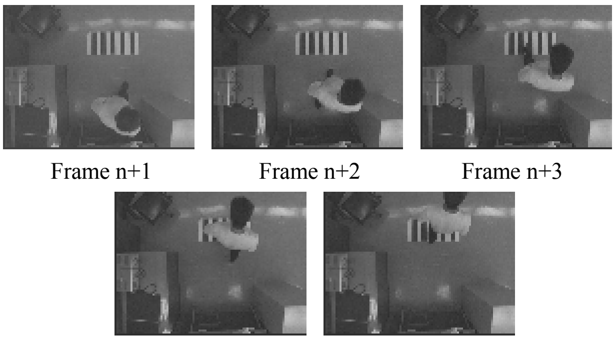

Figure 2 a single movement object sequence

Learning Initial Background Model

In a real time system, its noise comes from anywhere. The noise overcomes the difficulties of using the median estimator to estimate the local mean in our image processing. A new local mean is then computed using only pixels within a small gray-level range about the median. This effectively removes outliners in the calculation of the mean estimate, hence improving the overall performance of the mean filter in the presence of outliners such as salt and pepper noise. In this section, we describe the sequence image filtering as first process. An adaptive double window modified trimmed mean (DW-MTM) filter is employed for the first step that filters the input image in our real time system. The adaptive DW-MTM filter algorithm is described as follows.
Given a pixel located at x, y within the image, a median filter (MED [g (x, y)]) is computed within an n × n local region surrounding the location x, y. The g (x, y) is the noise-corrupted image. The median value computed from this filter is used to estimate the mean value of the n × n local area. Next, a larger-size window surrounding the pixel at location x, y of size q × q is used to calculate the mean value. In computing the mean value in the q × q window, only pixels within the gray-level range of

$$\mathrm{MED}\,[g(x, y)] - c \text{ to } \mathrm{MED}\,[g(x, y)] + c$$

are used, eliminating any outliners from the mean calculation. The output of the DW-MTM filter is the q × q mean filter. The value of c is chosen as a function of the noise standard deviation as

$$c = K \bullet \sigma_n$$

Where σ_n is the variance of the noise. Typical values of K range from 1.5 to 2.5. This range for K is based on the assumption that for Gaussian noise statistics the peak-to-peak gray-level variations will be in the range of $\pm 2\sigma_n$ 95% of the time and any values outside this range are more than likely outliers. For K=0, the DW-MTM filter to a n × n median filter, and for K very large, the DW-MTM reduces to a q × q mean filter. Hence, as K decreases, the filter does a better job of filtering impulsive noise, but a poor job of filtering uniform and Gaussian-type noise. In our system, we assume n=3 and q=5.
After the input image is filtered, we state that edge detection is part of a process called segmentation—the identification of regions within an image for second step processing. There are many varieties of edges; they may be classified into three major classes: a line-edge has a zero order discontinuity, a step-edge has a first order discontinuity, and a roof-edge has a second order discontinuity. It is compared with the Marr-Hildreth edge detection involve large windows and often needs floating-point calculations to maintain accuracy [5]. Canny edge detector that extracts not only step edges but also ridge and roof edges [6] and Shen-Castan edge detector. [7.] We are not interested in the direction of the edge but only in its presence, we should use direction-invariant edge detectors. On other way, we need to minimize the process time of edge detection. Here, we use a local neighborhood edge operator which is direction-invariant is the Laplacian 3 × 3 edge detector for this application. For the result refer to Figure 3.

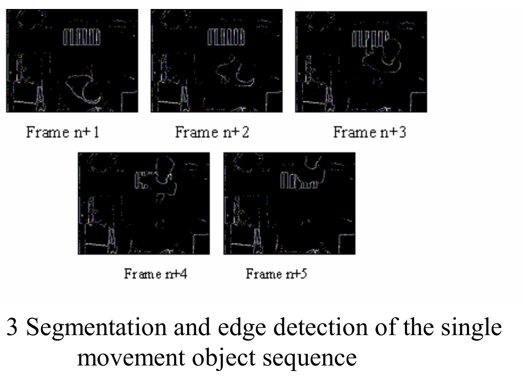

Figure 3 Segmentation and edge detection of the single movement object sequence

We use two consecutive frames to retrain the background model. After the sequence starts, we subtract two consecutive frames to obtain a result of a binary image (difference image) as figure 4 that shows to subtract two consecutive frames from the figure 3.

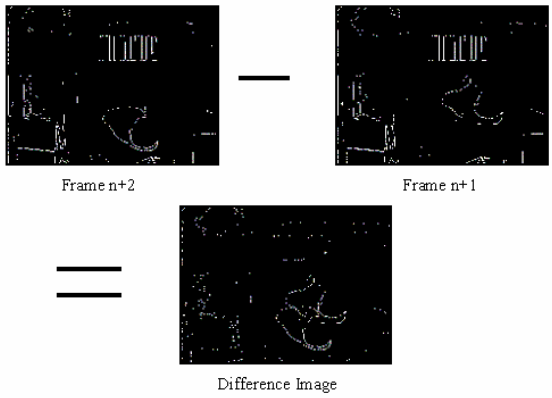

Figure 4 Subtract two consecutive frames from the single object movement sequence

The cluster analysis technique is used to extract some interesting points for labeling the object movement and some noise from the result image. In the result image, there is the object movement and some noise. For this noise, due to a great many factors such as light intensity, type of camera and lens, motion, temperature, atmospheric effects, dust, and others, it is very unlikely that two pixels that correspond to precisely the same gray level in the scene will have the same gray level in the image. Noise is a random effect, and is characterizable statistically only. The result of noise on the image is to produce a random variation in level from pixel to pixel, and so the smooth lines and ramps of the ideal edges are never encountered in real images. Subtracting the current image from the sequence image also produces the noise. As the result image is a binary image, the white pixels show the movement object or noise. There are some white pixels that merge together from pixel to pixel. We can state that the result image can response the noise only has a small number pixel merge as it comes from two consecutive frames. We can assume a threshold value to reject these noise from pixel to pixel in one gray-level at clustering process. This threshold value is called Noise_Threshold in number of pixels. The detail processing is as following.

Almost always, when information about an object or region class is available, a pattern recognition method is used to find out some interest points for the moving object. For this application, we investigate some related techniques: statistical pattern recognition, neural nets, syntactic pattern recognition, recognition as graph matching, optimization techniques in recognition, and fuzzy systems. They not only require considerable time but are also beyond our requirements. A cluster analysis technique is modified for this application as follows.

In the image analysis, clustering can be used to find groups of pixels with similar gray levels, colors, or local textures, in order to discover the various regions in the image. Clustering is the process of counting and labeling of objects within an image. Clusters consist of pixel groupings that are related to one another by a predetermined criterion. This criterion on measure can be defined as a distance between clusters or a similarity measure such as a pixel value, or it may be a complex set of identifiers and constraints that define membership in a cluster. The cluster may ultimately be mapped into a binary image as a unique object.

Formally, if p is a value of pixel, the cluster is defined as

$$C_i(p) = p \qquad (1)$$

Where i is an integer (0, 1, 2, ….) and cluster center point is defined as an interest point to response this cluster.

Here, our approach is to specify a connected in pixel that the clusters should be. In the binary image, the clustering algorithm processes each pixel and when one is found that is nonzero, it becomes part of the first cluster and is marked. The next nonzero pixel $f(x,y)$ found is tested to see if it is connected to one previous pixel of outer corners of its half_8_adjacent [$f(x-1,y-1)$, $f(x,y-1)$, $f(x+1,y-1)$, $f(x-1,y)$, $f(x-1,y+1)$]as shown in Figure 5. If it is, it is marked as a member of the first cluster and the search continues. If it is not, it becomes the first member of the second cluster and is marked accordingly. This process continues until all pixels have been evaluated.

$f(x-1,y-1)$	$f(x,y-1)$	$f(x+1,y-1)$
$f(x-1,y)$	$f(x,y)$	
$f(x-1,y+1)$		

Figure 5 Outer corners of a pixel's half_8_adjacent neighbors

If we set the Noise_Threshold = 4 (pixels) and maximum value of pixel number =11, we can obtain that $C_3(p)$ is a valid cluster and $C_1(p)$ &$C_2(p)$ are noise value with the equation (2),.

In order to minimize the processing time, we need to define a maximum value of pixel number for the cluster. Then we get one point to response these valid clusters as an interest point in the object as following relate equation.

$$\text{Noise\_Threshold} < \text{valid cluster } C(p) < \text{maximum value of pixel number} \qquad (2)$$

The example shows a graphic of a set of a set of clusters that have been labeled by grayscale value. After clustering, we found out there are three clusters $C_1(p)$, $C_2(p)$, and $C_3(p)$ as shown in Figure 6. From the equation (1), we known that $C_1(p) = 4$, $C_2(p) = 2$, and $C_3(p) = 10$.

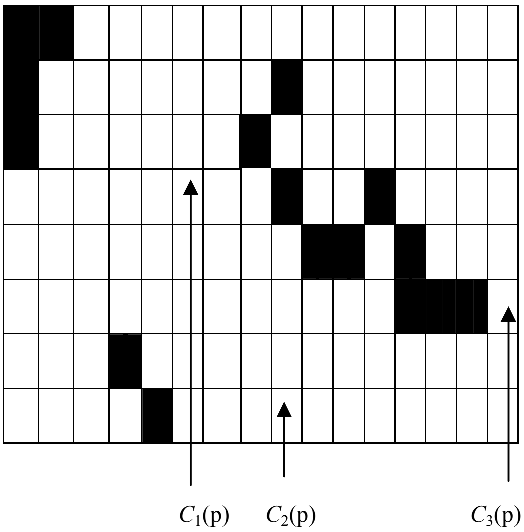

Figure 6 A graphic of a set of clusters

At the completion of the function, a 2D data array contains pixel values that reflect their membership in the clusters found that meet the criteria of being pixel connected apart. Cluster of pixels that are not connected with other cluster in outer corners of its 8_adjacent will be partitioned into pieces as they are found in the image. The difference binary image (after segmentation) is scanned from the upper left corner origin in column order. If a zero value gray-level is defined for object in the difference binary image (image [row][col]), we have that a chart of the function is shown in Figure 7.

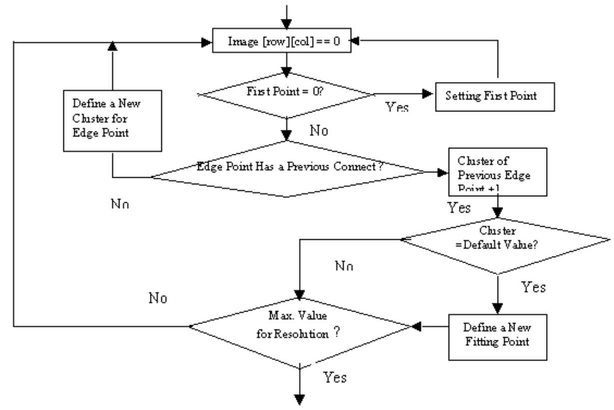

Figure 7 A chart of cluster analysis function

During the cluster analysis, a valid cluster result is generated with some points to interrupt the movement object. These points are called movement object points. The valid cluster result is generated with some points in response to background noise. These points are called non-movement object points such as a value (3, 263) at figure 8 that shows a real time result of the cluster analysis. It was produced by clustering the difference image (figure 4).

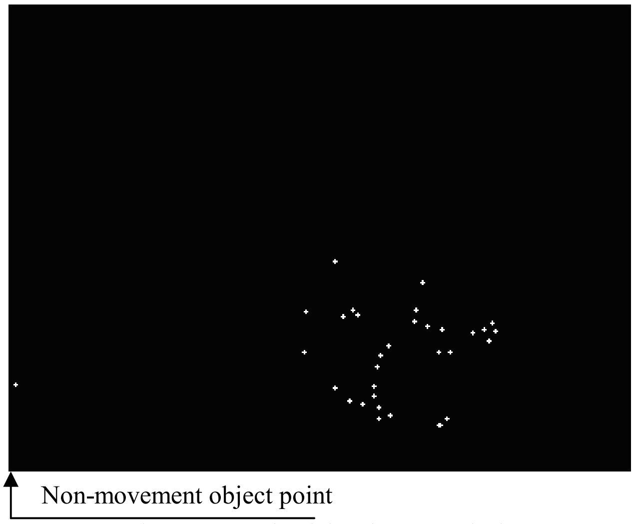

Figure 8 Result of the cluster analysis

Issue Background Model

The cluster result is generated with some points to interrupt the object movement and other points to response the background noise or without any point. If this result is zero there is not any point, we state one of both images is its background model for currently sequence. If this result is not zero such as figure 8, we state there is some object movement at the scene. The background model is issued from one of two consecutive frames binary images remove the movement object pixels. For the background modeling, we are interested in the outline points of the moving object as the figure 8. Removing all the moving object pixels that are enclosed from its outline points creates the binary background model. Figure 9 show a result of background model that remove the movement object of frame n+1 on figure 4.

Figure 9 A background model for the single moving object sequence

Updating background Model

The background model cannot be expected to stay the same for long periods of time. There could be illumination changes, such as the sun being blocked by clouds causing changes in brightness, or physical changes, such as a deposited object. As the cluster analysis is applied to each frame for tracking object movement at our object tracking system. From the figure 8, we know the non-movement point that is out of the movement object. When we keep the background model to apply to it sequence, a cluster analysis result show as figure 11 at the frame n+7 of single movement object sequence as following figure 10. From the figure 11, the points of non-movement object are 6.

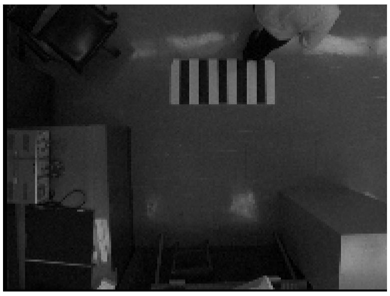

Figure 10 Frame n+7 of single movement object sequence

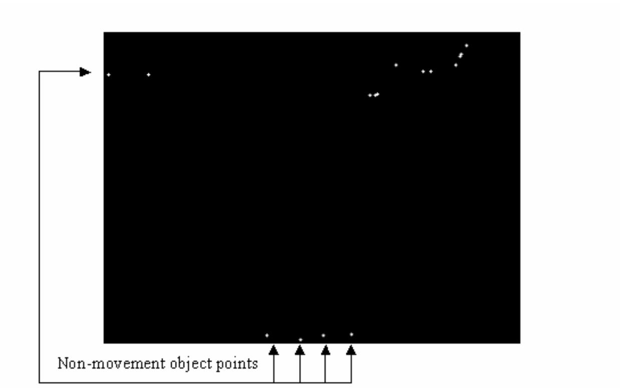

Figure 11 Cluster analysis result for frame n+7 of single movement object sequence

When there are more non-movement object points than a limit value, we need to update the background model. If this limit value is assumed set to 6 points, an update background processing will occur at frame n+7 of single movement object sequence. This update background-processing algorithm is summarized as follows:

- Get a cluster analysis result image by subtracting next frame.
- If this result is zero there is not any non-movement object points and movement object points, we state one of both frames is the background model for current sequence.
- If this result is non-zero, its background model for currently sequence is that remove all of movement object pixels that are enclosed from its outline points.

Experiments

The complete algorithm for our background modeling method is summarized in section 4.1. To evaluate our algorithm, two experiments are presented. The first experiments (section 4.2) demonstrate the performance of our background modeling method in the case of a multi-movement object environment. The second experiments (section 4.3) show the performances of our background modeling method to apply to our object tracking system.

The complete algorithm using the background modeling system is as follows:

- Capture image on and filter the sequence image,
- Edges detect two current consecutive frames and subtract them,
- Issue an initial background model from a result of the cluster analysis,
- If non-movement object points are over a limit value, update processing occurs for the background model.

When we apply this background model method to a multi-movement object sequence as figure 12, a result of background model is shown on figure 15. We subtract two consecutive frames as figure 13 at our background modeling. A result of the cluster analysis on multi-object movement sense is shown on figure 14. We are also success using the current background model to tracking the multi-movement object at this multi-movement object sequence. The figure 16 shows a cluster analysis result of the frame m+10 with same background model. From this result, we can state that the background model can be expected to stay the same for next tracking; there is not any non-movement point at this frame.

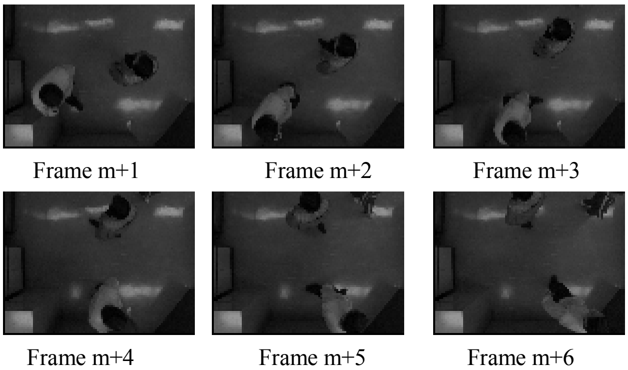

Figure 12 A sequence for multi-object movement

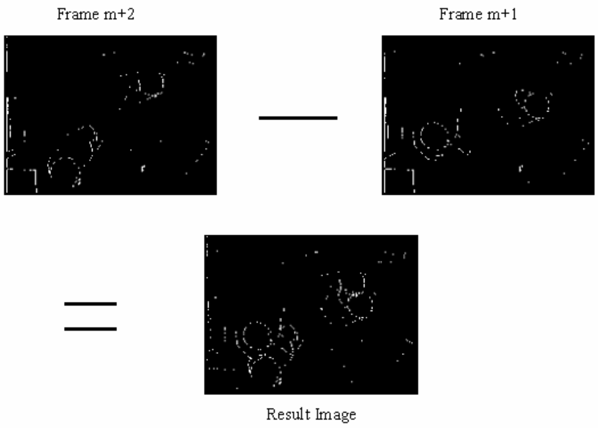

Figure 13 Subtract two consecutive images

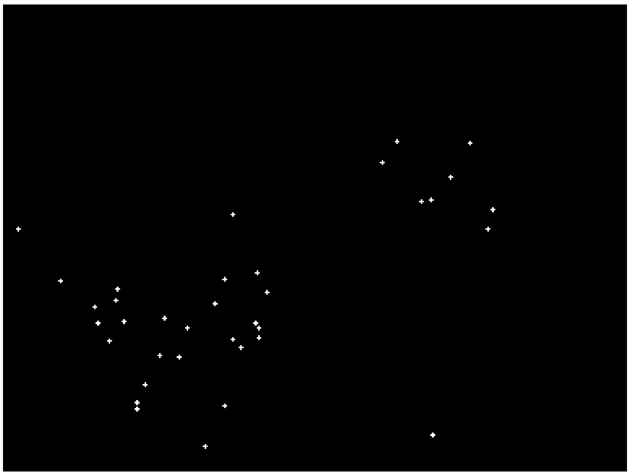

Figure 14 A result of the cluster analysis on multi-object movement sense

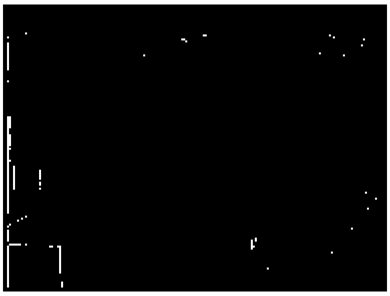

Figure 15 A background model for multi-movement object sequence

Figure 16 (a) Frame m+10

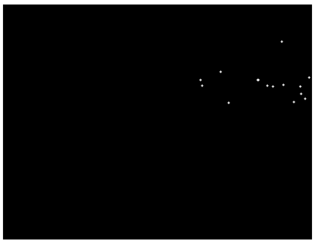

Figure 16 (b) A result of the cluster analysis for frame m+10

Figure 16 A result of the cluster analysis for frame m+10 with the background model on multi-object movement sense

The background model method was success fully implemented on our tracking system as in figure 16. The object tracking result shows the single movement object from frame n+1 to frame n+7 of the single movement object sequence (Figure 2). There are four windows to display an original image, movement object detection, movement objects tracking and tracking result.

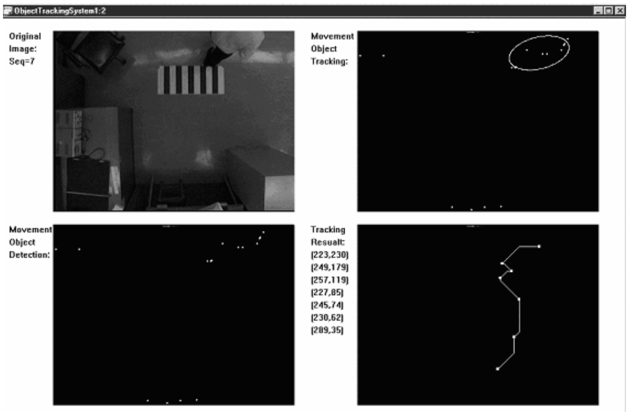

Figure 16 a movement object tracking system

Conclusion

A real-time computer vision system able to model a stationary object background or a movement object background in cluttered environments has been presented. The proposed system is based on the modeling of the structure of the scene. The quality of the detection is improved when the background is highly textured.

Therefore in our future works we will use this modeling method in our object tracking system for tracking rigid and non-rigid movement object. It will be used for to tracking multiple non-rigid movement objects in a cluttered scene.

Reference

[1] N. Friedman and S. Russell, “Image Segmentation in Video Sequences: A Probabilistic Approach,” Uncertainty in Artificial Intelligence, 1997.

[2] E. Grimson, C. Stauffer, R. Romano, and L. Lee, “Using Adaptive Tracking to Classify and Monitoring Activities in a Site,” Proc. Computer Vision and Pattern Recognition Conf., pp. 22-29, 1998.

[3]A. Elgammal, D. Harwood, and L. Davis, “Non-Parametric Model for Background Subtraction,” Proc. IEEE Frame Rate Workshop, 1999.

[4]I. Haritaoglu, D. Harwood, and L. S. Davis, “W4: Real-Time Surveillance of People and Their Activities,” PAMI, August 2000.

[5] Marr, D. and E. Hildreth, “ Theory of Edge Detection” Proceedings of the Royal Society of London, Series B. Vol. 207.187-217, 1980.

[6]Canny, J., “A Computational Approach to Edge Detection” IEEE, PAMI, VOL8, 679-698, JUNE 1986.

[7] A. Elgammal, D. Harwood, and L. Davis, “Non-Parametric Model for Background Subtraction,” Proc. IEEE Frame Rate Workshop, 1999.