

# Automatic Detection, Extraction and Recognition of Moving Objects

Adnan Khashman

**Abstract**—The security of restricted areas such as borders or buffer zones is of utmost importance; in particular with the worldwide increase of military conflicts, illegal immigrants, and terrorism over the past decade. Monitoring such areas rely currently on technology and man power, however automatic monitoring has been advancing in order to avoid potential human errors that can be caused by different reasons. This paper introduces an automatic moving object detection, extraction and recognition system (aMODERs), which uses image processing to detect and extract moving objects within a restricted area, and a neural network to recognize the extracted object. The proposed system monitors movement by humans, animals or vehicles across a secured zone. Experimental results indicate that (aMODERs) provides a simple, efficient and fast solution to the problem of detecting, extracting and recognizing moving objects within one system.

**Keywords**— Moving object detection, extraction and recognition, neural network classification, security systems.

## I. INTRODUCTION

**M**OVING object detection is a basic and important problem in video analysis and vision applications. Automatic recognition systems for still and moving objects can be invalid in security applications, such as monitoring border areas, buffer zones and restricted areas. A simple recognition system would comprise a camera fixed high above the monitored zone, where images of the zone are captured and consequently processed. Processing the captured images can be in three phases, namely, detection of a moving object, extraction of the object and finally recognition of the object.

Optical flow and background subtraction have been used for detecting moving objects in image sequences [1]-[4]. For example, adaptive optical flow for person tracking [2] is dependent on being able to locate a person accurately across a series of frames. Optical flow can be used to segment a moving object from a scene, provided the expected velocity of the moving object is known; but successful detection also relies on being able to segment the background. Other works such as moving objects segmentation using optical flow

estimation [3] presented a method for the segmentation of moving objects, where a powerful variation method using active contours for computing the optical flow is used. However, the high computational time to extract the optical flow and the lack of discrimination of the foreground from the background, make this method unsuitable for real time processing. On the other hand, background subtraction detects moving objects by subtracting estimated background models from images. This method is sensitive to illumination changes and small movement in the background, e.g. leaves of trees [4]. Another problem of background subtraction is that it requires a long time for estimating the background models. It usually takes several seconds for background model estimation because the speed of illumination changes and the small movement in the background are very slow.

Much research work into moving object recognition has been previously presented. In [5], a semantic video object generation and temporal tracking technique for providing content-based video representation and indexing were proposed. In [6], a Gramian determinant-based method was proposed to detect moving objects between multiple images and to detect changes between color images or any type of multi-spectral images. In [7], a moving object detection method was proposed using global motion estimation and edge information, where the final objects are extracted by combining the contours and moving regions from motion detection. In [8], a method based on background subtraction was proposed to recognize abnormal human behavior in public areas using segmentation of moving objects in real-time from images acquired by a fixed color video camera. In [9], a system for moving object detection and shadow extermination by building an adaptive background model was described. In [10], a method for detecting poorly visible moving objects in bad weather was proposed, and was based on measuring a variation on a cross correlation between a short accumulated histogram and along accumulated histogram of detected objects. In [11], a method was proposed to extract contours of moving objects mainly by combining gradient information extracting with three-frame differencing and connectivity-testing-based noise reduction. In [12], a binary image filtering method based on discrete information entropy was proposed for moving object detection in image sequences. In [13], a method for detecting and extracting moving objects from a moving stereo camera was described, where camera motions were calculated from three-dimensional optical flow, and the moving object was detected

Manuscript received Febr. 8, 2008; Revised version received April 4, 2008.

A. Khashman is with the Intelligent Systems Research Group (ISRG), Department of Electrical & Electronic Engineering, Near East University, Nicosia, Cyprus (corresponding author, phone: +90-392-223-6464; fax: +90-392-223-6622; e-mail: amk@neu.edu.tr).

using images corrected in accordance with the camera motion. In [14], a statistical method was used to obtain background model, which was updated in real time to adapt to illumination changes and scene changes.

More recent research work on moving object detection has also emerged. In [15] a Belief Propagation approach for moving object detection using a 3D Markov Random Field (MRF) model was described. This work suggested that moving objects were detected and tracked successfully while handling appearance change, shape change, varied moving speed/direction, scale change and occlusion/clutter. In [16], extensions to a method for moving object detection [17] were proposed. In [18], an algorithm based on frame difference and Canny edge detection was presented for moving object detection. In [19], a method for moving object detection in foggy days using a background maintenance algorithm based on the Unscented Kalman Filter (UKF) was described. In [20], an intelligent sensor network was developed for object detection, classification and recognition, by utilizing wireless sensors as the first layer to detect coordinates of moving objects in a secured area. In [21], a low-complexity moving object detection algorithm by exploring the ideas of uncertainty analysis and spatiotemporal activity clustering was suggested. In [22], a background subtraction algorithm for detecting moving objects from a static background scene, based on brightness distortion and chromaticity distortion in (RGB) color space, was described. In [23], an approach to automatically locate multiple objects in home videos, by taking into account of how and when to initialize objects was proposed.

Other research works, aiming at providing solutions to the problem of object extraction, have also emerged. In [24], a tracking initialization method that combines a rough extraction of moving objects and a refined segmentation of their contours was proposed. In [25], a moving object extraction method based on an active contour model, called Snake, and utilizing a splitting mechanism was proposed. In [26], a method of clustering video frame pixels was suggested for moving object extraction. In [27], an algorithm based on inter-frame change detection and Canny edge detection in wavelet domain was proposed. In [28], an algorithm based on clustering and the use of a self organizing network was proposed.

More recently, in [29] a moving object extraction scheme using contour-feature and motion vector-based projection for (MPEG) stream was proposed. In [30], a technique that extracts the shape of moving objects by combining snakes and watershed algorithm was described. In [31], a motion estimation method to detect and extract a moving object from a video sequence taken by a hand-held camera was proposed.

These methods have shown success in detecting moving objects or in extracting the objects. Few works have addressed both tasks within one system, where a moving object is detected and then extracted automatically. However, the setback of such works is their complexity and computational costs. Furthermore, a robust simple automatic system that

performs detection, extraction and recognition of moving objects is much preferred.

The aim of the work presented within this paper is to develop such an automatic system for the detection, extraction and recognition of moving objects within a restricted zone. The developed system is named (aMODERs) which stands for automatic Moving Object Detection, Extraction and Recognition system.

Moving object detection is achieved by comparing two subsequent still images from a surveillance camera, captured every two seconds. The difference between the pixel values is calculated and the output is obtained using a deterministic rule. Further processing is also carried out in order to provide a clearer image of the detected object and its surrounding by improving the result image contrast using background compensation.

The detected object extraction method is based on a set of comparison deterministic rules, which find pixel variations within the image, eliminate phantom objects, and output the relevant object. Further processing of the extracted object image is also applied by squaring, framing and scaling the image to a pre-defined size in preparation for using the image in the third and final phase of (aMODERs), which is the recognition of the object using a neural network.

Object recognition is achieved using a supervised neural network based on the back propagation learning algorithm. The neural network is trained using images of potential objects that may cross a restricted zone, such as humans, animals and vehicles. The images in the database were obtained while assuming a restricted monitored zone with the monitoring camera fixed at a high position to prevent potential system vandalism. The experiments in this work consider the detection and extraction of the following potential objects [32]: humans (male, female), animals (goat), and vehicles (car, jeep, loader, motorbike).

## II. MOVING OBJECT DETECTION

In order to detect a movement within a secured area, a surveillance camera is positioned to monitor the area. The detection of a moving object within the monitored area is the first phase in (aMODERs). An image-processing scheme to detect moving objects in real time is a key technology for the automatic surveillance. This section presents the method which was developed for this purpose.

The movement detection uses a simple but efficient method of comparing image pixel values in subsequent still frames captured every two seconds from the surveillance camera. Two still images are required to detect any movement. The captured image size is 256x256 pixels. The first image, which is called "reference" image, represents the reference pixel values for comparison purpose, and the second image, which is called the "input" image, contains the moving object.

The two images are compared and the differences in pixel values are determined. If the input image pixel values are not equal to the reference image pixel values, then the input image

pixel values are thresholded and saved in a third image, which is called output image, with a black or white background. The choice of the output image background (black or white) is determined by comparing the average pixel value of the “difference” between the result output image and the reference and input images. If the “difference” average pixel value is smaller than a certain threshold value, then the output image background will be white (pixel value is 255); otherwise, the background will be black (pixel value is 0).

After tracking the moving object motion, the previous input image will now be used as a reference image, and a third image is captured and is called now the input image. This process is repeated with the images being captured every two seconds, where the same comparison method is applied. If there is a difference between the reference and input images, then an output image is created. The obtained output image contains an object that will be extracted (in the second phase) and recognized (in the third phase). Fig. 1 shows the process of moving object detection.

The reference and input images are compared by taking their difference where the output of this comparison  $D_{RI}(x,y)$  is determined by the following rule:

$$D_{RI}(x,y) = \begin{cases} 0, & \text{if } R(x,y) = I(x,y) \\ I(x,y), & \text{if } R(x,y) \neq I(x,y) \end{cases} \quad (1)$$

where  $R(x,y)$  and  $I(x,y)$  are the reference and input images, respectively at image coordinates  $(x,y)$ .

The obtained “difference” matrix is then thresholded in order to determine the background color of the third image;

output image  $O(x,y)$ , which contains the moving object. To threshold the difference matrix, we first calculate its average pixel value using:

$$D_{average} = \frac{\sum_{x=1}^{n_x} \sum_{y=1}^{n_y} D_{RI}(x,y)}{n_x \cdot n_y} \quad (2)$$

where  $n_x$  and  $n_y$  are, respectively, the numbers of difference matrix pixels in the  $x$  and  $y$  directions.

The background color ( $B$ ) of the output image  $O(x,y)$ , which contains the moving object, will be then set to black (gray level “0”) or white (gray level “255”) as follows:

$$B = \begin{cases} 0, & \text{if } D_{average} > 126 \\ 255, & \text{if } D_{average} \leq 126 \end{cases} \quad (3)$$

The reason for this background compensation is to provide a clear and high contrast image of the moving object in preparation for the next phases of (aMODERs). The result image after thresholding and background compensation is output image  $O(x,y)$  which is defined as:

$$O(x,y) = \begin{cases} B, & \text{if } R(x,y) = I(x,y) \\ I(x,y), & \text{if } R(x,y) \neq I(x,y) \end{cases} \quad (4)$$

To demonstrate image comparison and thresholding in this phase, assume that we have two 6x6 grayscale images; firstly, a reference image  $R(x,y)$  and secondly an input image  $I(x,y)$  represented in the following matrices:

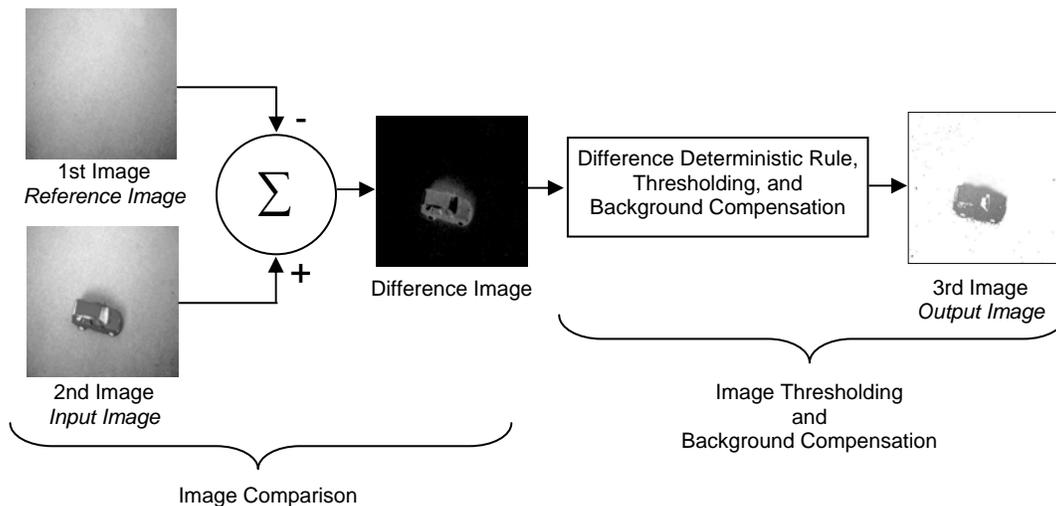


Fig. 1 Moving object detection phase

$$R(x, y) = \begin{bmatrix} 21 & 63 & 70 & 101 & 142 & 66 \\ 36 & 59 & \overline{78} & \overline{55} & \overline{39} & 74 \\ 105 & 84 & \overline{58} & 47 & \overline{35} & 69 \\ 145 & 188 & \overline{99} & \overline{125} & \overline{86} & 121 \\ 14 & 21 & \overline{27} & 19 & 240 & 213 \\ 46 & 37 & 40 & 11 & 178 & 198 \end{bmatrix}$$

$$I(x, y) = \begin{bmatrix} 21 & 63 & 70 & 101 & 142 & 66 \\ 36 & 59 & \overline{26} & \overline{0} & \overline{92} & 74 \\ 105 & 84 & \overline{24} & 241 & \overline{14} & 69 \\ 145 & 188 & \overline{19} & \overline{51} & \overline{77} & 121 \\ 14 & 21 & \overline{27} & 19 & 240 & 213 \\ 46 & 37 & 40 & 11 & 178 & 198 \end{bmatrix}$$

The output image  $O(x, y)$  will be:

$$O(x, y) = \begin{bmatrix} 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & \overline{26} & \overline{0} & \overline{92} & 255 \\ 255 & 255 & \overline{24} & 241 & \overline{14} & 255 \\ 255 & 255 & \overline{19} & \overline{51} & \overline{77} & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

The next phase in (aMODERs) is the detected moving object extraction.

### III. DETECTED OBJECT EXTRACTION

The second phase in (aMODERs) is the extraction of a detected object. In this phase the output image  $O(x, y)$  which was obtained at the end of the first phase is used. A simple but efficient way of extracting the object is to use horizontal and vertical scanning of the output image. In a two-dimensional matrix an object could be addressed by finding its vertical and horizontal coordinates. The width and height of the extracted object image can be found by using its starting coordinates and ending coordinates.

#### A. Extraction Method

The implementation of this extraction method has two steps. Firstly, the output image matrix is scanned horizontally, starting at the first x-direction coordinate value, the pixel values in the corresponding column are summed up. The x-direction coordinate value is incremented by one and total pixel value in the next column is calculated. This process is repeated until the last value in the x-direction. As result the total pixel values of each column is calculated. Each total value is compared to a certain threshold value in order to determine the x-coordinate where an object starts or ends within the image. Secondly, the horizontal scanning method is repeated vertically, thus calculating the total pixel value in each row, and then apply thresholding to determine the y-coordinate where an object starts or ends within the image.

The background of the image that contains the object is uniform as it has already been set to white or black at the end of the first phase in (aMODERs). The mathematical description of the object extraction method is summarized in the following subsections.

#### B. Horizontal Scanning

Starting at image coordinates  $(x=0, y=0)$  until coordinate  $(x=n_x, y=n_y)$ , the total pixel value of the column is defined as:

$$TotalY_x = \sum_{y=0}^{n_y} O(x, y) \quad (5)$$

This total value is compared to a threshold value  $\theta_x$  which is defined as:

$$\theta_x = n_y \cdot B \quad (6)$$

where  $n_y$  is the maximum number of vertical pixels ( $n_y = 256$  in this work), and  $B$  is the gray level value of the image background as defined in equation (3). For white ( $B=255$ ), whereas for black ( $B=0$ ).

The comparison deterministic rules are defined as follows:

$$\text{For } B = 255, \quad \begin{array}{l} \text{if } TotalY_x \leq (\theta_x - C) \text{ then } S_x = x \\ \text{if } TotalY_x > (\theta_x - C) \text{ then } E_x = x \end{array}$$

$$\text{For } B = 0, \quad \begin{array}{l} \text{if } TotalY_x > (\theta_x + C) \text{ then } S_x = x \\ \text{if } TotalY_x \leq (\theta_x + C) \text{ then } E_x = x \end{array}$$

where  $S_x$  and  $E_x$  are respectively the starting and ending x-coordinates of the object.  $C$  is a correction value that is added to eliminate unwanted pixels which could be due to noise. It should be noted that the correction value should not be set high, as it would result in the loss of relevant object pixels, on the other hand if the correction value is set low, then unwanted objects of noise may be obtained; in this work ( $C=2025$ ).

The application of this algorithm continues until the column at the last x-coordinate pixel is accounted for. If there is another object within the image, its x-coordinate positions are determined similarly to the first object.

#### C. Vertical Scanning

Starting also at image coordinates  $(x=0, y=0)$  until coordinate  $(x=n_x, y=n_y)$ , the total pixel value of the row is defined as:

$$TotalX_y = \sum_{x=0}^{n_x} O(x, y) \quad (7)$$

This total value is compared to a threshold value  $\theta_y$  defined as:

$$\theta_y = n_x \cdot B \quad (8)$$

where  $n_x$  is the maximum number of horizontal pixels ( $n_x = 256$  in this work), and  $B$  is the grey level value of the image background as defined in equation (3).

The comparison deterministic rules are defined as follows:

$$\text{For } B = 255, \quad \begin{aligned} &\text{if } TotalX_y \leq (\theta_y - C) \text{ then } S_y = y \\ &\text{if } TotalX_y > (\theta_y - C) \text{ then } E_y = y \end{aligned}$$

$$\text{For } B = 0, \quad \begin{aligned} &\text{if } TotalX_y > (\theta_y + C) \text{ then } S_y = y \\ &\text{if } TotalX_y \leq (\theta_y + C) \text{ then } E_y = y \end{aligned}$$

where  $S_y$  and  $E_y$  are respectively the starting and ending y-coordinates of the object, and  $C$  is the correction value. The size of the extracted object image is  $(E_x - S_x) \cdot (E_y - S_y)$ . The application of this algorithm continues until the row at the last y-coordinate pixel is accounted for. If there is another object within the image, its y-coordinate positions are determined similarly to the first object.

To demonstrate the extraction method, the output image, containing the moving object, which was obtained at the end of the detection phase, is shown in Fig. 2. This image appears to have two objects; in fact it is one object (vehicle) that appears in the output image at two different locations within the image. The first object appearance, which we call phantom object, occurs two seconds before the second appearance. We are interested in extracting the clear object which is the second appearance. The proposed extractor can efficiently extract both objects together or separately.

Once the start and end coordinates of the object(s) are determined using the proposed extraction method, the width and height of the matrix as defined by those coordinates are noted and the matrix is saved. The result is a smaller image containing only the required object. It is also possible to add one pixel (white) to each side of each row and column in order to frame the extracted object.

#### D. Extracted Image Squaring and Framing

In real life applications, the extracted object image may not be a square image, and its size will not be of a standard size.

This may create a problem for further processing of the extracted image, more notably in the object recognition phase. The proposed (aMODERs) uses a neural network to classify the extracted objects in its third and final phase. The neural network will have a defined structure with a fixed number of input layer neurons. These input neurons receive pixel values from the extracted object image. Therefore, further processing of the extracted object image is required, and it involves simple squaring of the extracted image and resizing it to assure the neural network input layer neurons receive a uniform set of pixel input values in a real life application. For this work, a standard extracted object image size of 100x100 pixels is selected, and shall be used as the input image size in the third phase of (aMODERs), which is neural network classification.

A simple algorithm is used to square the image by comparing its width and height. If the width is greater than the height, then the height is set equal to the width and vice versa. The difference between width and height (*dif*) is saved for later use when centralizing the object within the square image. This squaring rule is defined as:

$$\text{Squaring: } \begin{cases} \text{If } height > width, \text{ then } width = height \\ \text{If } width > height, \text{ then } height = width \end{cases}$$

The extracted object in the new square image is then centralized within image by shifting the object horizontally (if  $width > height$ ) or vertically (if  $height > width$ ). Here, the difference between height and width (*dif*) is used to determine the amount of shifting required to obtain a centralized square image  $M_c(x,y)$  from a square image  $M_s(x,y)$  as follows:

$$M_c(x,y) = \begin{cases} M_s\left(x + \frac{dif}{2}, y\right), & \text{if } height > width, \\ & dif = height - width \\ M_s\left(x, y + \frac{dif}{2}\right), & \text{if } width > height, \\ & dif = width - height \end{cases} \quad (9)$$

The extracted image is now square and the object is in the centre of the image. In order to clearly outline the object

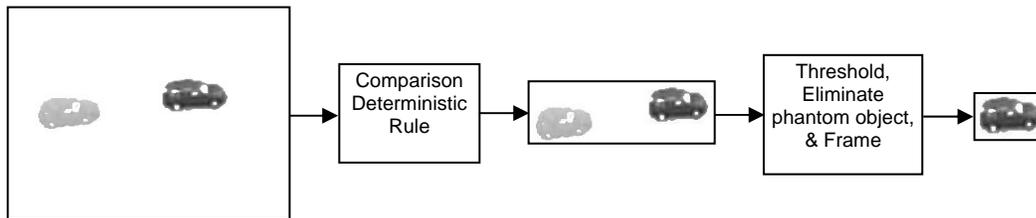


Fig. 2 Object extraction phase

image, a frame can be added around the image. This is a simple addition of one-pixel rows and columns to all four sides of the image. The pixel value should be equal to the background grayscale value. The image is then resized to standard size of 100x100 pixels in preparation for the third phase which is the object recognition by a neural network.

#### IV. OBJECT RECOGNITION

This is the third and final phase in (aMODERs). Here the detected and extracted moving object is recognized by a trained supervised neural network that is based on the back propagation learning algorithm. This algorithm is chosen due to its implementation simplicity and efficiency in pattern classification. This section describes the image database, the neural network input pattern preparation, and the implementation of the neural network.

##### A. Image Database

Meaningful training of a neural network is vital if we are to generalize it successfully. Two aspects in training are noted here: firstly the number of images used and, secondly, the sufficiency of the extracted patterns from these images. This section will briefly describe the objects considered in this work; bearing in mind the assumed scenario for this application which is monitoring a secured area with a surveillance camera being positioned higher than the ground level. The assumed secured area is a border controlled area or a buffer zone between two countries, thus the choice of objects in this application.

The considered objects in this work are classified into three groups: humans, animals, and vehicles. Images of objects from each group will be used for training and later on generalizing (or testing) the neural network. The objects in each group were as follows: human (Female and Male), animal (Goat), vehicle (Car, Jeep, Motorbike, and Loader). Fig. 3 shows example images of these seven objects [32].

In order to provide a more objective simulation of a real-life application, different orientations of all objects are considered, since their movement directions can not be restricted or anticipated. For this purpose, the directions {1,2,3,..., 8} shown in Fig. 4 are used as the movement directions of these objects for neural network training. Images of each moving object are obtained in six directions; excluding two directions which are {3 and 7}, since the object movement will be towards or away from the camera.

The movement of an object in these two directions will not be detected by (aMODERs), which is not considered as a setback, since our proposed system aims to detect movement across a restricted zone. Testing the neural network, after training, uses object images at random directions; for example direction {1.5} in Fig. 4.

Training the neural network uses six images of each object at directions {1,2,4,5,6,8} in animal and vehicle groups, and three images of each object in human group (3 Male images at directions {1,2,4}, and 3 Female images at directions {5,6,8}), thus resulting in 36 images for training the neural network. Testing the generalized neural network uses eight random directions images of each object in all three groups, thus resulting in 56 images for testing the generalized neural network.

##### B. Input Pattern Feature Extraction

The image database that is used for this application has a total of 92 images (36 training and 56 testing images). Each image is in grayscale and of size 100x100 pixels. The presentation of these images to the neural network on a pixel-per-node basis is computationally expensive; therefore, pattern averaging is used to provide input feature vectors with smaller size. These feature vectors will be presented to the neural network during both training and testing.

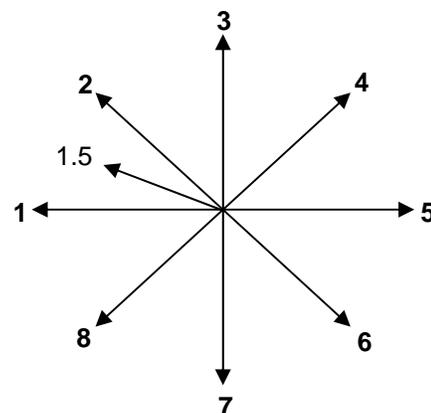


Fig. 4 Moving object directions

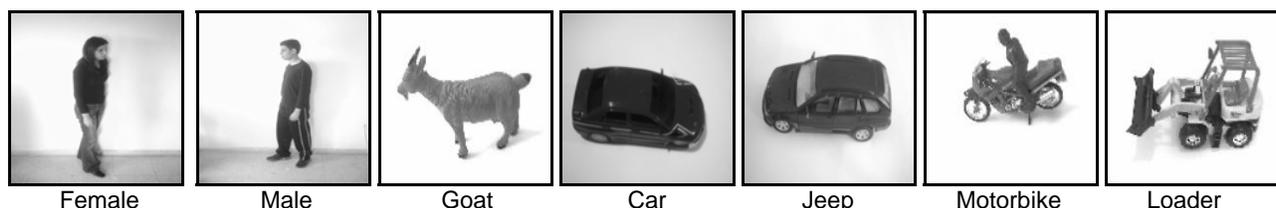


Fig. 3 Example images of the considered objects

Pattern averaging is a simple but efficient method that creates "fuzzy" patterns as compared to multiple "crisp" patterns, which still provides the neural network with meaningful learning while reducing computational expense [33]. Pattern averaging is defined as:

$$PatAv_i = \frac{1}{S_k S_l} \sum_{l=1}^{S_l} \sum_{k=1}^{S_k} p_i(k,l) \quad (10)$$

where  $k$  and  $l$  are segment coordinates in the  $x$  and  $y$  directions respectively,  $i$  is the segment number,  $S_k$  and  $S_l$  are segment width and height respectively,  $P_i(k,l)$  is pixel value at coordinates  $k$  and  $l$  in segment  $i$ ,  $PatAv_i$  is the average value of pattern in segment  $i$  that is presented to neural network input layer neuron  $i$ . The number of segments in each image (of size  $X.Y$  pixels;  $X = Y = 100$ ) containing an object, as well as the number of neurons in the input layer is  $i$  where  $i = (1, 2, 3, \dots, n)$ , and

$$n = \left( \frac{X}{S_k} \right) \left( \frac{Y}{S_l} \right) \quad (11)$$

Segment size of  $10 \times 10$  pixels ( $S_k = S_l = 10$ ) has been used and average values representing the image were obtained, thus resulting in 100 average values in total ( $n = 100$ ) that were used as the input to the neural network for both training and testing.

### C. Neural Network Arbitration

The neural network consists of an input layer with 100 neurons receiving the input feature vector, one hidden layer with 200 neurons which was determined after many experiments involving the adjustment of the number of hidden

neurons, and an output layer with six neurons corresponding to six object classifications using binary output data representation.

A recognition system "sensitivity" feature was also developed as part of the neural network classification of extracted objects. Two levels of tolerance, namely Low (minimum 75% resemblance) or High (minimum 65% resemblance) can be used depending on the required level of accuracy.

During the learning phase, the number of hidden layer neurons, the learning coefficient, and the momentum rate were adjusted in various experiments in order to achieve the required minimum error value of 0.009 which was considered as sufficient for this application. Fig. 5 shows the topology of this neural network.

The neural network learnt and converged after 9300 iterations and within 164.6 seconds, whereas the running time for the generalized neural network after training and using one forward pass was 0.012 seconds. These results were obtained using a 1.8 GHz PC with 1 GB of RAM, Windows XP OS and Borland C++ compiler. Table I lists the final parameters of the successfully trained neural network.

TABLE I  
 FINAL NEURAL NETWORK SIMULATION PARAMETERS

Parameter	Final value
Input Layer neurons	100
Hidden layer neurons	200
Output layer neurons	6
Learning coefficient	0.006
Momentum rate	0.35
Minimum error	0.009
Iterations	9300
Training time (seconds)	164.6
Run time (seconds)	0.012

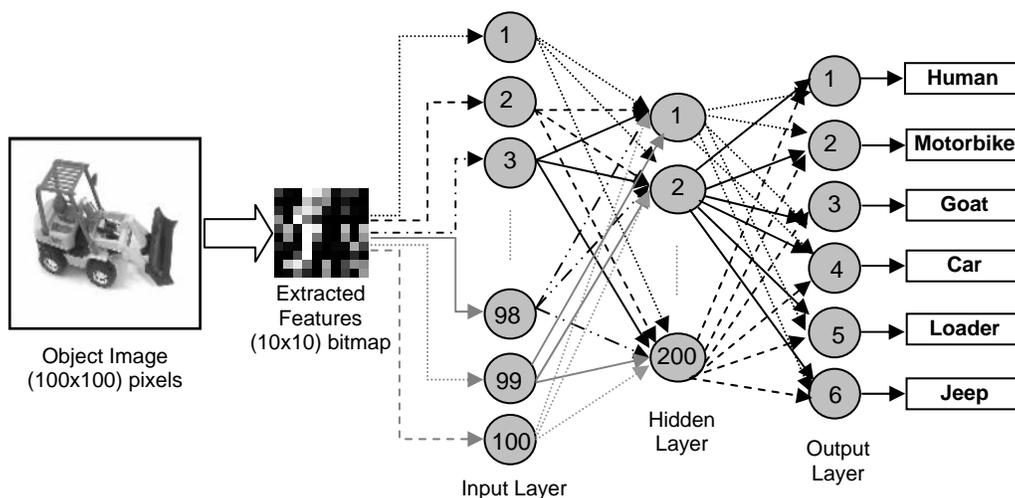


Fig. 5 Object recognition phase in (aMODERs)

TABLE II  
OBJECT RECOGNITION RESULTS

Object Classification	Tolerance Level	Training Set Recognition	Testing Set Recognition	Combined Set Recognition
Human (Female & Male)	Low	6/6 (100%)	15/16 (93.8%)	21/22 (95.5%)
	High	6/6 (100%)	15/16 (93.8%)	21/22 (95.5%)
Motorbike	Low	6/6 (100%)	4/8 (50%)	10/14 (71.4%)
	High	6/6 (100%)	8/8 (100%)	14/14 (100%)
Goat	Low	6/6 (100%)	6/8 (75%)	12/14 (85.7%)
	High	6/6 (100%)	8/8 (100%)	14/14 (100%)
Car	Low	6/6 (100%)	5/8 (62.5%)	11/14 (78.6%)
	High	6/6 (100%)	6/8 (75%)	12/14 (85.7%)
Loader	Low	6/6 (100%)	4/8 (50%)	10/14 (71.4%)
	High	6/6 (100%)	7/8 (87.5%)	13/14 (92.9%)
Jeep	Low	6/6 (100%)	5/8 (62.5%)	11/14 (78.6%)
	High	6/6 (100%)	7/8 (87.5%)	13/14 (92.9%)
All Objects	Low	36/36 (100%)	39/56 (70%)	75/92 (81.5%)
	High	36/36 (100%)	51/56 (91.1%)	87/92 (94.6%)

The implementation results of the trained neural network were as follows: using the training image set (36 images) at both tolerance levels yielded 100% recognition as would be expected. The recognition system implementation using the testing image set (56 images that were not previously exposed to the neural network) yielded different correct object recognition according to the tolerance level. At Low tolerance level 39 objects were correctly recognized, thus achieving 70% correct identification rate, whereas, at High tolerance level 51 objects were correctly recognized, thus achieving 91.1% correct identification rate. Combining the results using testing images (56) and training images (36), yields an overall correct identification rate of 81.5% with Low tolerance, and 94.6% with High tolerance. Table II shows the extracted objects recognition results.

## V. CONCLUSION

In this paper, an automatic moving object detection, extraction, and recognition system (aMODERs) was presented. The proposed system receives still images captured every two seconds from a surveillance camera, which monitors a restricted zone, such as international border crossings, buffer zone in conflict areas, or any monitored area where movement across that area is to be detected.

The implementation of (aMODERs) has three phases: firstly, moving object detection which is achieved using image pixel value difference, a deterministic rule to determine the moving object, and background compensation. Secondly, the extraction of the detected object, which is achieved by using a set of deterministic rules to find pixel variations within the image of the detected object(s) and eliminating phantom objects that may have been obtained in the first phase. This second phase has also further processing of the extracted object image, such as squaring, framing and scaling the image to a predefined size in preparation for the next phase. Finally, the third phase is the recognition of the extracted object using a supervised neural network based on the simple but efficient back propagation learning algorithm. Here the neural network

is trained using images of potential objects that may cross the monitored restricted zone. Once trained, the neural classifier is able to recognize similar objects regardless of their scale and motion orientation of direction across the secured area.

The potential objects which have been considered in this work are classified into three groups: human (female, male), animals (goat), and vehicles (motorbike, car, loader, jeep). The use of (aMODERs) in a real life application is not limited to these objects, because the neural network classifier can be quickly and easily trained to recognize as many different objects as required.

The implementation of (aMODERs) in this work has been successful with an overall correct identification rate of 94.6% which was achieved in 0.012 seconds. The proposed system provides solutions to the problem of monitoring secured areas; namely, the detection of movement across the area, the extraction of the moving object, and the recognition of the object. Future work will focus on further experiments with multiple objects in a larger area, in addition to increasing the number of potential objects.

## REFERENCES

- [1] J. H. Duncan, and T. C. Chou, "Temporal edges: The detection of motion and the computation of optical flow," in *Proc. IEEE 2nd Int. Conf. Computer Vision*, Florida, USA, Dec. 1988, pp. 374-382.
- [2] S. Denman, V. Chandran, and S. Sridharan, "Adaptive Optical Flow for Person Tracking," in *Proc. Digital Image Computing: Techniques and Applications*, Cairns, Australia, Dec. 2005, pp. 44-50.
- [3] F. Ranchin, and F. Dibos, "Moving objects segmentation using optical flow estimation," in *Proc. Workshop Mathematics and Image Analysis*, Paris, France, Sep. 2004.
- [4] L. Li, W. Huang, I.Y.H. Gu, and Q. Tian, "Foreground object detection in changing background based on color co-occurrence statistics," in *Proc. IEEE 6th Workshop Applications of Computer Vision*, Florida, USA, Dec. 2002, pp. 269-274.
- [5] J. Fan, Y. Ji, and L. Wu, "Automatic Moving Object Extraction toward Content-Based Video Representation and Indexing," *J. Visual Communication and Image Representation*, vol. 12, no. 3, pp. 306-347, 2001.
- [6] E. Durucan, and T. Ebrahimi "Moving object detection between multiple and color images," in *Proc. IEEE Conf. Advanced Video and Signal Based Surveillance*, Florida, USA, July 2003, pp. 243 - 251.
- [7] R. Jinchang, P. Astheimer, and D. D. Feng, "Real-time moving object detection under complex background," in *Proc. IEEE 3rd Int. Symp.*

- Image and Signal Processing and Analysis*, Rome, Italy, Sep. 2003, 2, pp. 662 – 667.
- [8] D. Duque, H. Santos, and P. Cortez, "Moving object detection unaffected by cast shadows, highlights and ghosts," in *Proc. IEEE Int. Conf. Image Processing*, Genoa, Italy, vol. 3, no. 3, Sep. 2005, pp. 413-16.
- [9] W. Huawei, B. Jianhua, L. Cuihua, and W. Qiyang, "A system for moving object detection and shadow extermination," in *Proc. IEEE Int. Workshop VLSI Design and Video Technology*, Suzho, China, May 2005, pp.117-120.
- [10] K. Onoguchi, "Moving Object Detection Using a Cross Correlation between a Short Accumulated Histogram and a Long Accumulated Histogram," in *Proc. IEEE 18th Int. Conf. Pattern Recognition*, Hong Kong, vol. 4, Aug. 2006, pp. 896-899.
- [11] M. Zhao, J. Zhao, S. Zhao, and Y. Wang, "A Novel Method for Moving Object Detection in Intelligent Video Surveillance Systems," in *Proc. IEEE Int. Conf. Computational Intelligence and Security*, Guangzhou, China, vol. 2, Nov. 2006, pp. 1797-1800.
- [12] J. Zuo, C. Zhao, Q. Pan, and W. Lian, "A Novel Binary Image Filtering Algorithm Based on Information Entropy," in *Proc. IEEE 6th World Congress on Intelligent Control and Automation*, Dalian, China, vol. 2, June 2006, pp. 10375-10379.
- [13] Y. Mito, M. Morimoto, and K. Fujii, "An Object Detection and Extraction Method using Stereo Camera," in *Proc. World Automation Congress*, Budapest, Hungary, July 2006, pp. 1-6.
- [14] H. W. Lin, S. Q. Yang, Z. J. Xia, and C. Y. Kang, "A Moving Objects Detection Approach for Smart Sensor," in *Proc. IEEE 5th Int. Conf. Machine Learning and Cybernetics*, Dalian, China, Aug. 2006, pp. 3751-3754.
- [15] Z. Yin, and R. Collins, "Belief Propagation in a 3D Spatio-temporal MRF for Moving Object Detection," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, Minneapolis, USA, June 2007, pp. 1-8.
- [16] L. Maddalena, and A. Petrosino, "Moving Object Detection for Real-Time Applications," in *Proc. IEEE 14th Int. Conf. Image Analysis and Processing*, Modena, Italy, Sep. 2007, pp. 542-547.
- [17] H. Fujiyoshi, and T. Kanade, "Layered detection for multiple overlapping objects," *IEICE Trans. Inf. & Syst.*, E87-D, no. 12, pp. 2821-2827, 2004.
- [18] C. Zhan, X. Duan, S. Xu, Z. Song, and M. Luo, "An Improved Moving Object Detection Algorithm Based on Frame Difference and Edge Detection," in *Proc. IEEE 4th Int. Conf. Image and Graphics*, Chengdu, China, Aug. 2007, pp. 519-523.
- [19] G. Chen, H. Zhou, and J. Yan, "A Novel Method for Moving Object Detection in Foggy Day," in *Proc. IEEE 8th ACIS Int. Conf. Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, Qingdao, China, vol. 2, Aug. 2007, pp. 53-58.
- [20] F. Y. Shih, Y. T. Wu, C. F. Chuang, J. L. Chen, H. F. Lu, and Y. C. Chang, "An Intelligent Sensor Network for Object Detection, Classification and Recognition," *J. of Information Science and Engineering*, vol. 23, pp. 1955-1969, 2007.
- [21] Y. C. Chung, and Z. He, "Low-Complexity and Reliable Moving Objects Detection and Tracking for Aerial Video Surveillance with Small UAVS," in *Proc. IEEE Int. Symp. Circuits and Systems*, New Orleans, USA, May 2007, pp.2670-2673.
- [22] R. Zhang, S. Zhang, and S. Yu, "Moving Objects Detection Method Based on Brightness Distortion and Chromaticity Distortion," *IEEE Trans. Consumer Electronics*, vol. 53, no. 3, pp. 1177-1185, 2007.
- [23] Z. Pan, and C. W. Ngo, "Moving-Object Detection, Association, and Selection in Home Videos," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 268-279, 2007.
- [24] J. Huart, G. Foret, and P. Bertolino, "Moving object extraction with a localized pyramid," in *Proc. IEEE 17th Int. Conf. Pattern Recognition*, Cambridge, UK, vol. 4, Aug. 2004, pp. 783-786.
- [25] M. Haseyama, and Y. Yokoyama, "Moving object extraction using a shape-constraint-based splitting active contour model," in *Proc. IEEE Int. Conf. Image Processing*, Genoa, Italy, vol. 3, no. 3, Sep. 2005, pp. 1260-1263.
- [26] M. Chen, G. Ma, and S. Kee, "Pixels Classification for Moving Object Extraction," in *Proc. IEEE 7th Workshop on Motion and Video Computing*, Colorado, USA, vol. 2, Jan. 2005, pp. 44-49.
- [27] X. Zhang, R. Zhao, and Z. Ma, "A new method for moving object extraction automatically," in *Proc. IEEE Int. Symp. Communications and Information Technology*, Beijing, China, vol. 2, Oct. 2005, pp.1460-1463.
- [28] D. Vasquez, and T. Fraichard, "A Novel Self Organizing Network to Perform Fast Moving Object Extraction from Video Streams," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Beijing, China, Oct. 2006, pp. 4857-4862.
- [29] Q. Zhou, G. Yang, W. Chen, and Z. Zhang, "A Fast and Accurate Moving Object Extraction Scheme in the MPEG Compressed Domain," in *Proc. IEEE 4th Int. Conf. Image and Graphics*, Chengdu, China, Aug. 2007, pp. 592-597.
- [30] K. Imamura, M. Hiraoka, and H. Hashimoto, "Watershed algorithm for moving object extraction considering energy minimization by snakes," in *Proc. IEEE Conf. Advanced Video and Signal Based Surveillance*, London, UK, Sep. 2007, pp. 534-539.
- [31] G. Zhang, J. Jia, W. Xiong, T. T. Wong, P. A. Heng, and H. Bao, "Moving Object Extraction with a Hand-held Camera," in *Proc. IEEE 11th Int. Conf. Computer Vision*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1-8.
- [32] H. Bareke, "Intelligent Moving Object Recognition Systems" M.Sc. thesis, Dept. Elect. & Elec. Eng., Near East Univ., Nicosia, Cyprus, 2006.
- [33] Khashman, A.: 'Intelligent Global Face Recognition', in Delac, K., and Grgic, M. (Ed.): 'Face Recognition' (Ch. 11, ITECH Education and Publishing, Vienna, Austria, 2007), pp. 219-234.



**Adnan Khashman** received the B.Eng. degree in electronic and communication engineering from University of Birmingham, England, UK, in 1991, and the M.S and Ph.D. degrees in electronic engineering from University of Nottingham, England, UK, in 1992 and 1997.

During 1998-2001 he was an Assistant Professor and the Chairman of Computer Engineering Department, Near East University, Nicosia, Cyprus. Since 2001 he is an Associate Professor and Chairman of Electrical and Electronic Engineering Department at the same university. Since 2007 he is also the Vice-Dean of Engineering Faculty and the Head of the Intelligent Systems Research Group (ISR) at the same university.

Dr. Khashman is a senior member of IEEE, and his current research interests include medical image processing, intelligent systems, and neural networks and their engineering applications. In these fields, he authored or co-authored about 60 scientific papers. Contact him at: ISR, Department of Electrical and Electronic Engineering, Near East University, Nicosia, Cyprus; amk@neu.edu.tr.