Recognizing Objects by Detecting Multiple Moving Parts

Azzam Sleit[1], Wesam Almobaideen[1], Mohamad Qatawneh[1], Shatha Al-Asir[2], Oraib Al-Megdadi[3]

Department of Computer Science, King Abdulla II School for Information Technology
University of Jordan, P.O. Box 13898, Amman 11942, Jordan

1 {azzam.sleit, wesmoba, mohd.qat}@ju.edu.jo, 2 sh.alasir@gmail.com,
3 orayb_miqdadi@yahoo.com

ABSTRACT: Detecting objects in a given video stream is an important step to understand their types, movements, and activities. Existing object detection algorithms suffer from their inability to detect the components constituting a particular object that may result in classifying such components as standalone objects. Such instances may happen particularly when the colors of some components of the object have colors, which are close to the background. In this paper, we propose a technique to detect such objects by analyzing multiple images for the same object and observing the motion of various components of the object. [The Journal of American Science. 2008;4(4):38-49]. (ISSN: 1545-1003).

1. INTRODUCTION

Computer vision research has increasingly focused on recognizing objects. Their main purpose is to track objects especially human and vehicles in order to detect abnormal or desired behaviors. Object detection techniques can be classified into two categories [1, 2]. The first category requires a preprocessing phase, which subtracts the current image from a reference frame in order to isolate moving objects. After detecting and subtraction the background classification of the object performed by applying two approaches: the first is a codebook approach, and the second involves tracking of the object based on color histograms, motion and size of the foreground blob. False alarms due to static oscillatory motions are also detected and removed. Infrared (IR) cameras can be used instead of standard camera for detecting moving objects. The cameras in [3] are mounted close to each other and observe the same scene from a similar viewpoint. Actually, the approach is not human specific, but will detect any moving object. Background subtraction is performed independently by using Gaussian probability distribution to model each background pixel. The detected foreground from the two cameras is registered using a hierarchical genetic algorithm, and the two registered silhouettes are then fused together into the final estimate. On the other hand, IR images can be fused with images from a regular camera [4], Humans display a characteristic signature in IR images due to their skin temperature, but these images typically have low contrast. They can be fused with images from a standard camera to obtain superior detection results. In addition, a shape-based approach can be utilized to determine the object types as human, animal or vehicle based on approximate matching of the object's shape boundary contour to a polygon [5].

The second category detects moving objects by directly observing temporal difference between images to separate the foreground and background using a statistical model to classify human depending on the fact that the relative positions of various body parts are common to all humans, although the pixel values may vary because of the clothes [6]. The technique uses a structure known as the distance map, which is built by taking an image of a human and breaking it into MxN blocks then using distance maps for a large database of human and non-human images. A statistical model is built for distance maps of each type, which consists of the average and covariance matrix for each block. Then, the image is segmented into smaller objects based on threshold of luminance and/or color components of the image. Real-time detecting human system implemented in [7] by combining Viola's face detection framework and the HOG feature pool. The system keeps the discriminate power of Histograms of Oriented Gradients (HOG) features and the real-time properties of Viola's face detection framework. Besides human, the detection framework can be used to detect other objects such as vehicles.

Detect passengers onboard public transport vehicles method proposed in [8] by detecting the passenger head first using the curvature profile of the human head as a cue, followed by applying the geometric blur features which are consistent to affine distortion of the image to keep track of the movement of the head within the vehicle. The profile of moving heads with respect to each other within a length of time can be used as indicative features to detect the advent of suspicious behavior of the passengers. The two categories differ in speed, memory requirements and accuracy [9]. Most algorithms may fail to detect some objects or separate one object to multiple ones because of false background estimation [10, 11].

In this article, we introduce a new approach, which is a hybrid of both categories. The proposed method recognizes human objects in spite of undetected components by inspecting multiple frames in a video stream and calculating the average speed of the detected components. Consequently, the dimensions of the human object in the image can be determined.

2. OBJECT DETECTION METHODS

Background estimation and subtraction play gigantic role in object detection and may affect the quality of the detection outcome. The big question is what if some parts were undetected from the video frames during the background subtraction process?
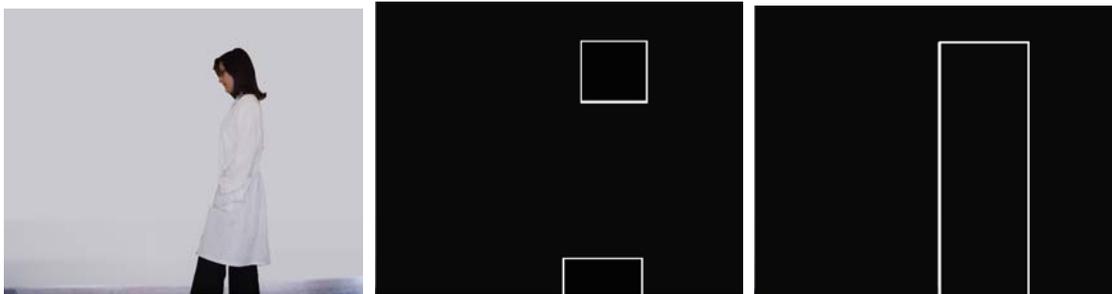

Figure 1: a) An image for a walking human. b) Object detection for the human c) Dimensions of human using average component speed.

Figure 1(a) shows a human dressed in white, walking in front of a white wall. Clearly, applying classical object detection algorithms on such an image leads to undetected parts of the human object as shown in figure 1(b), which may lead to false decision in recognizing the identity of the object. By studying multiple frames, it is quite reasonable to assume that if both rectangles in figure 1(b) move with the same average speed, that the dimensions of the human object be as shown in figure 1(c).

Figure 2 displays a flow chart for the proposed technique. Clearly, considering multiple frames, all components, which lie in the same vertical position and move at the same average speed, may be combined as one object. The proposed approach is also capable of recognizing multiple humans walking at the same speed by calculating their speed, position and direction as shown in figure 3. The components of each object are in the same vertical position and move at approximately the same speed.
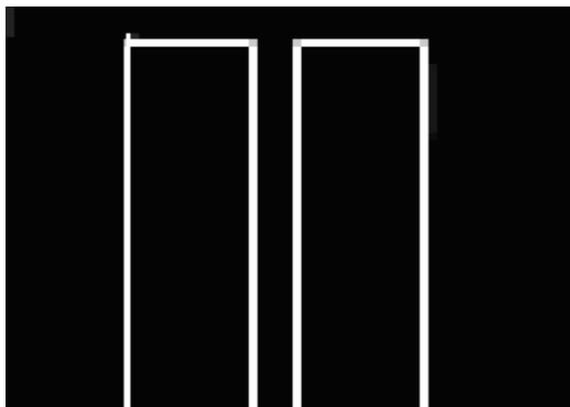
Figure 2: Enhancement flowchart
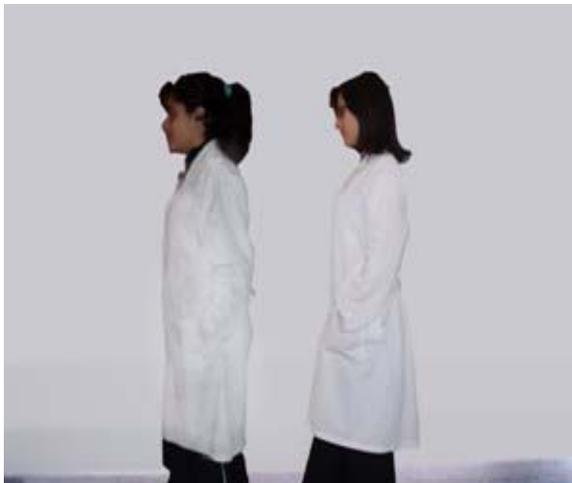
(a)
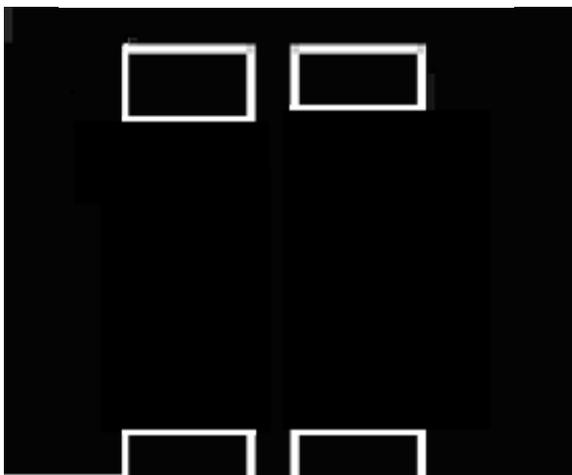
(b)

Figure 3: (a) Two walking humans
(b) Object detection for the human
(c) Dimensions of human using the average
component speed.

3. ANALYSIS AND EVALUATION

The proposed approach can be used to enhance existing background subtraction [12,13], RGB-difference and absolute difference algorithms [9]. Detecting components of objects using the previous steps and finding their minimum bounding rectangles, (i.e. MBR's) are prerequisites for the proposed enhancement. The absolute difference of two images is performed in a single pass by taking two images P1 and P2 of equal size as input and producing a third image Q of the same size whose pixel values are simply those of the first image minus the corresponding pixel values from the second image as describe in equation 1.

$$Q(i,j) = P1(i,j) - P2(i,j) \qquad ..... \ (1)$$

The absolute difference of two images can be achieved using the MATLAB function *imabsdiff*. The RGB-difference can be computed using the algorithm in figure 4 where the pixel value is divided into three components.

RGB-Difference (A, B)

```
> create a 2-D array called BW.
For i = 1 to row
    For j = 1 to columns
        Diff-Red = A [i, j, 1] – B[i, j, 1] ; Diff-Green = A [i, j, 2] – B[i, j, 2] ; Diff-Blue = A [i, j, 3] – B[i, j, 3] ;
        > row1 represent red-value
        > row2 represent green-value
        > row3 represent blue-value

     Difference = Diff-Red + Diff-Green + Diff-Blue;

                if (Difference <= th)
                >th: threshold value

                {    BW [i, j, 1] = 255;   BW [i, j, 2] = 255; BW [i, j, 3] = 255;
                }
                else
                {    BW [i, j, 1] = 0; BW [i, j, 2] = 0; BW [i, j, 3] = 0;
                }
        end
end
return BW;
```


Figure 4: RGB-Difference

Combining image subtraction and object detection techniques can be utilized to examine the proposed enhancement. Figure 5 explains the first method, where absolute image subtraction is performed with respect to the reference frame.

Method 1

>*All functions are built-in MATLAB*
>*functions [12, 13]*
A: Reference Frame
>*A: contains only the background with*
>*no objects*
B: Frames Sequence
> B: *contains the background with /*
>*without objects*
Resize A, B
> *for Memory Utilization*
Image Subtractions
>*imabsdiff function*
Find threshold value
>*graythresh function*
Convert the image to binary image
>*im2bw function*
Dilation
>*imdilate function*
Filling holes
>*imfill function*
Smoothing
>*imerode function*
Remove all object containing fewer than 50 pixels
>*bwareaopen function*
Finding objects and drawing MBR
> *bwlabel , regionprops, boundaries*
functions
> *BoundingBox property*

Figure 5: Method 1
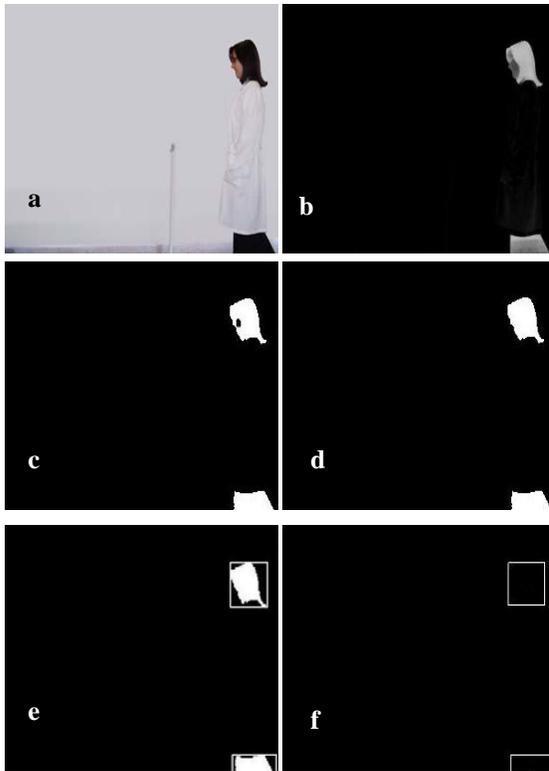
Figure 6: Applying method 1 to the image in figure 3 (a) original image (b) subtracted image (c) binary image (d) smoothing and hole filling (e)(f) minimum bounding rectangle

Figure 6 displays the outcome of applying method 1 on the image in figure 3. The subtracted image is converted into binary and holes are removed through dilation filling. Small objects are totally ignored. Method 2 is displayed in figure 7 with edge detection being employed. The outcome of method 2 with respect to the image in figure 3 is traced in figure 8.

Method 2

>*All functions are built-in MATLAB*
>*functions [12, 13]*

A: Reference Frame
>A: *contains only the background with*
>*no objects*

B: Frames Sequence
> B: *contains the background with /*
>*without objects*

Resize A, B
> for Memory Utilization

Image Subtractions
>imabsdiff function

Drawing the edges
> edge function

Dilation
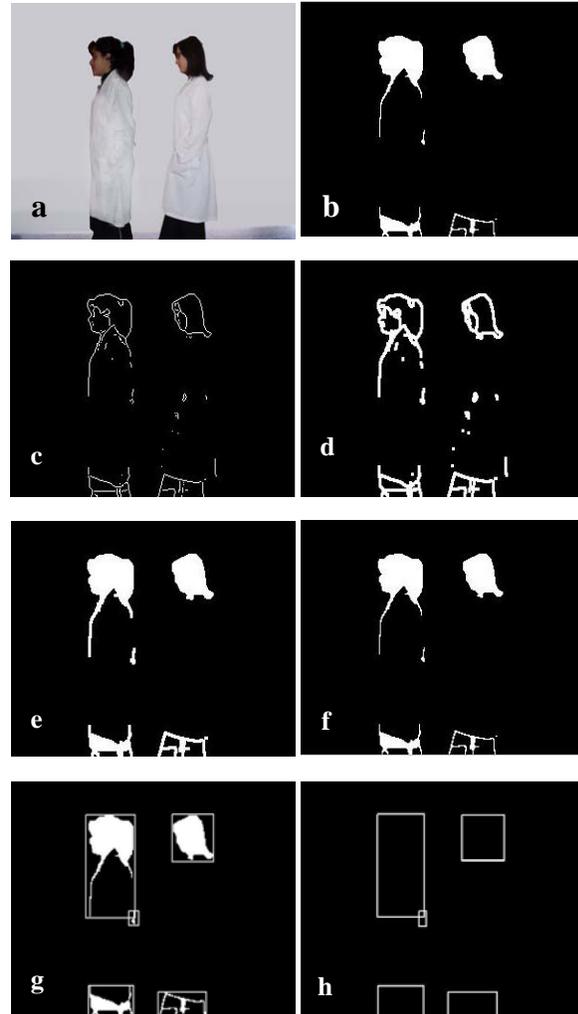>imdilate function

Filling holes
>imfill function

Remove all object containing fewer than 50 pixels
>bwareaopen function

Smoothing
>imerode function

Finding objects and drawing MBR
> bwlabel , regionprops, boundaries

functions
> BoundingBox property

Figure 7: Method 2

Figure 8: applying method 2
(a) original image (b) subtracted images
 (c) edge detection   (d) dilation   (e) holes filling
(f) smoothing (g)(h) minimum bounding rectangle

Method 3 utilizes the RGB-difference with respect to the reference frame as explained in figure 9. A trace for the method is displayed in figure 10.

Method 3

A: Reference Frame
>A: *contains only the background with*
>*no objects*

B: Frames Sequence
> B: *contains the background with /*
>*without objects*

Resize A, B
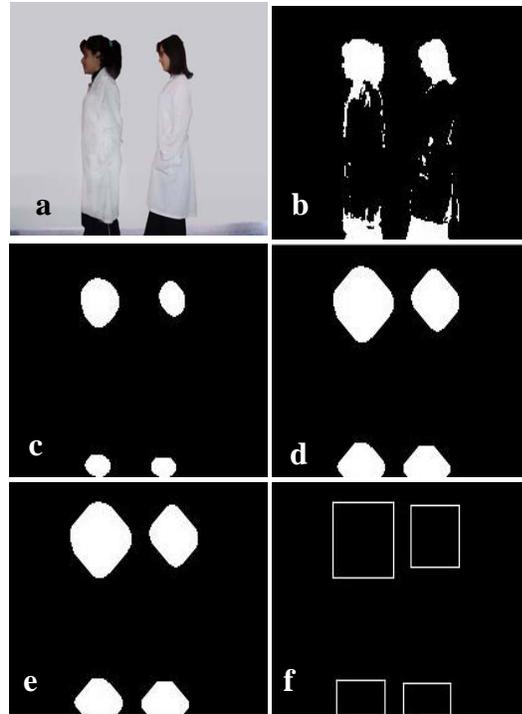> for Memory Utilization

Diff =RGB Difference (A, B)

Smoothing (Diff)

Dilation (Diff)

Filling Holes (diff)

Drawing MBR

Figure 9: Method 3

Figure 10: Applying method 3
(a) original image   (b) RGB-Difference
(c) smoothing   (d) dilation   (e) holes
filling   (f) minimum bounding rectangle

The enhancement proposed in this article connects components of objects based on their spatial position in the image and their average speed. Figure 11 presents the pseudo code of the enhancement.

Enhancement Method

>*this enhancement can be applied to any human* >*detection algorithm (considering human* >*movement is horizontal.*
>*after detecting MBRs*
For number of frames
        >*in our evaluation =3*
For 1 to number of objects in every frame
        >*take center of each MBR Centroid*
        >*property and compare the X-values*
If (difference (X1,X2) ≥ th)
{
        > *th : in our evaluation =15 , if yes may be they belong to the same object*
    If (average speed is equal)
      {
        If (same direction movement)
        They are the same object
        }
}
Else
    They are different objects.
> Average speed is equal: objects are moving in >the same speed ratio (according to the Y- axis)
> Same direction movement: if objects position >is increasing or decreasing.

Figure 11: Enhancement Method

	Frame1	Frame2	Frame3	Frame4	Frame5	Frame6	Frame7
Original images							
Intermediate step (method 1)							
Final Outcome (method 1)							
Applying Enhancement on method 1							
Intermediate step (method 2)							
Final Outcome (method 2)							
Applying Enhancement on method 2							
Intermediate step (method 3)							
Final Outcome (method 3)							
Applying Enhancement on method 3							

Figure 12: Object detection using three methods along with the proposed enhancement for single object frames.

	Frame1	Frame2	Frame3	Frame4	Frame5	Frame6
Original images						
Intermediate step (method 1)						
Final Outcome (method 1)						
Applying Enhancement on method 1						
Intermediate step (method 2)						
Final Outcome (method 2)						
Applying Enhancement on method 2						
Intermediate step (method 3)						
Final Outcome (method 3)						
Applying Enhancement on method 3						

Figure 13: Object detection using the three methods along with the proposed enhancement for single two walking humans with different speed**.**

In order to study the impact of the proposed enhancement, we executed the three methods along with the enhancement on the sequence of frames shown in figure 12. Clearly, the three methods fail to detect the human object in all the seven frames. However, the enhancement in combination with each of the methods correctly generated the MBRs in the fourth frames. Similarly, we analyzed another sequence of six frames with two humans as shown in figure 13. Obviously, none of the three methods successfully detected the two walking humans. Method 1 and 3 counted two to four objects in each frame while method 2 detected

two to six objects in each frame. In combination with the proposed enhancement, method 1 and 2 successfully detected both humans in frame 6 while method 3 failed.

4. CONCLUSION

This paper discusses object detection techniques. It proposes an enhancement based on the average component speed in order to integrate the ingredients of a specific object. To enable object detection, the proposed technique considers a stream of frames and examines the speed of various components appearing in the frames. Combining this approach with classical image processing techniques, demonstrates clear superiority.

REFERENCES

1. Benezeth, Y., Emile, B., Rosenberger, C. 2007, Comparative Study on Foreground Detection Algorithms for Human Detection, ICIG 2007, Fourth International Conference on Image and Graphics, pp. 661- 666.
2. Jos V., "Human Body Detection Methods", 2005. Industrial Design Technical University Eindhoven.
3. Ju, H., Bhanu, B., 2003. Detecting moving humans using color and infrared video, IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, 30:228–233.
4. Lijun, J., Feng, T., Lim, E., Shiqian, W., Susu, Y., Zhongkang, L., and Lijun, X., 2004. Perceptual-based fusion of IR and visual images for human detection, International Symposium on Intelligent Multimedia, Video and Speech Processing, 514–517.
5. Lee, P., Zhan, A., and Schoenberger, R., 2004. Shape-based human intrusion detection, SPIE International Symposium on Defense and Security, Visual Information Processing XIII, 5438:81–91.
6. Akira, U., Nobuji, T., 2002. Human detection using geometrical pixel value structure, Fifth IEEE International Conference on Automatic Face and Gesture Recognition, 39-45.
7. Hui-Xing, J., Yu-Jin, Z., 2007. Fast Human Detection by Boosting Histograms of Oriented Gradients, Fourth International Conference on Image and Graphics, 53-60.
8. Boon, C., Mihai, L., Tele, T., 2007. Detection and Monitoring of Passengers on a Bus by Video Surveillance, 14th International Conference on Image Analysis and Processing, IEEE computer society, 52-59.
9. Piccardi, M., 2004. Background subtraction techniques: a review, IEEE International Conference on Systems, Man and Cybernetics, 3099 – 3104.
10. Rafael C., Richard, E., 2008. "Digital Image Processing," third Edition, Pearson Prentice Hall.
11. Pratt, W., "Digital Image Processing: PIKS Inside," 2006. Third Edition, John Wiley & Sons.
12. www.mathwork.com
13. Gonzalez, R., Woods, R., Steven L., 2008, Eddins, "Digital Image Processing using MATLAB", Pearson Prentice Hall.

2/1/2008