IJCTA, 9(10), 2016, pp. 4677-4682

Background Subtraction Approach Under Motion Detection Techniques in Video Surveillance

***Mr. R. Rajesh **Capt. Dr. S. Santhosh Baboo**

Abstract : Moving object detection is the basic step for further analysis of video. It handles segmentation of moving objects from stationary background objects. This not only creates a focus of attention for higher level processing but also decreases computation time considerably. Commonly used techniques for object detection are background subtraction, statistical models, temporal differencing and optical flow. Due to dynamic environmental conditions such as illumination changes, shadows and waving tree branches in the wind object segmentation is a difficult and significant problem that needs to be handled well for a robust visual surveillance system.

Object classification step categorizes detected objects into predefined classes such as human, vehicle, animal, clutter, etc. It is necessary to distinguish objects from each other in order to track and analyze their actions reliably. Currently, there are two major approaches towards moving object classification, which are shape-based and motion-based methods [1].

Keywords : NFN, NBN, Pixels

1. INTRODUCTION

The overview of this real time video object detection, classification and tracking system is shown in Figure 1.1. The proposed system is able to distinguish transitory and stopped foreground objects from static background objects in dynamic scenes; detect and distinguish left and removed objects; classify detected objects into different groups such as human, human group and vehicle; track objects and generate trajectory information even in multi-occlusion cases and detect fire in video imagery.

This system is assumed to work real time as a part of a video-based surveillance system. The computational complexity and even the constant factors of the algorithms used are important for real time performance. Hence, the decisions on selecting the computer vision algorithms for various problems are affected by their computational run time performance as well as quality. Furthermore, the system's use is limited only to stationary cameras and video inputs from Pan/Tilt/Zoom cameras where the view frustum may change arbitrarily are not supported.

The system is initialized by feeding video imagery from a static camera monitoring a site. Most of the methods are able to work on both color and monochrome video imagery. The first step of this approach is distinguishing foreground objects from stationary background. To achieve this, combination of adaptive background subtraction and low-level image post-processing methods to create a foreground pixel map at every frame are used. Then grouping the connected regions in the foreground map to extract individual object features such as bounding box, area, center of mass and color histogram.

* Research Scholar, P.G and Research Department of Computer Science, D. G. Vaishnav College, Arumbakkam, Chennai, Tamil Nadu – 600106, India. *E-Mail: dgvcrajeshphd@gmail.com,*

** Associate Professor, P.G and Research Department of Computer Science, D. G. Vaishnav College, Arumbakkam, Chennai, Tamil Nadu – 600106, India. *E-Mail: santhos1968@gmail.com*

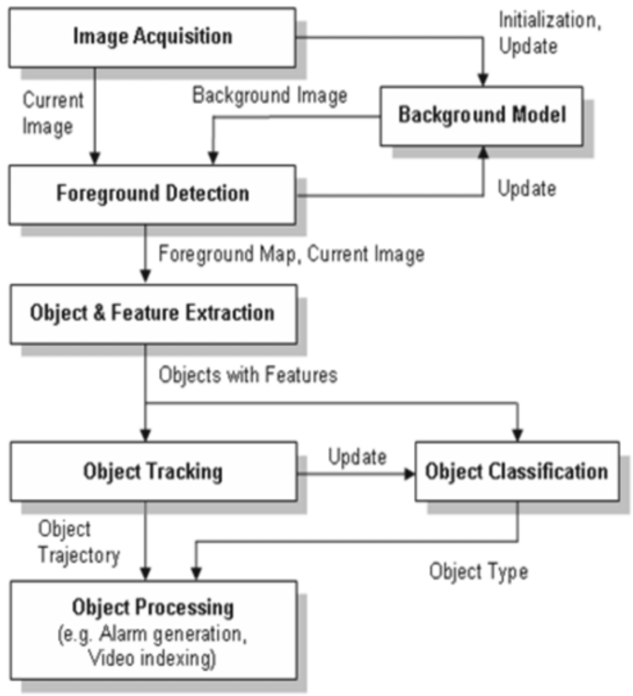

Fig. 1. The system block diagram.

The novel object classification algorithm makes use of the foreground pixel map belonging to each individual connected region to create a silhouette for the object. The silhouette and center of mass of an object are used to generate a distance signal. This signal is scaled, normalized and compared with pre-labeled signals in a template database to decide on the type of the object. The output of the tracking step is used to attain temporal consistency in the classification step.

The object tracking algorithm utilizes extracted object features together with a correspondence matching scheme to track objects from frame to frame. The color histogram of an object produced in previous step is used to match the correspondences of objects after an occlusion event. The output of the tracking step is object trajectory information which is used to calculate direction and speed of the objects in the scene.

After gathering information on objects' features such as type, trajectory, size and speed various high level processing can be applied on these data. A possible use is real-time alarm generation by pre-defining event predicates such as "A human moving in direction d at speed more than s causes alarm a1." or "A vehicle staying at location l more than t seconds causes alarm a2.". Another opportunity may make use of the produced video object data is to create an index on stored video data for offline smart search. Both alarm generation and video indexing are critical requirements of a visual surveillance system to increase response time to forensic events.

2.1. OBJECT DETECTION

Distinguishing foreground objects from the stationary background is both a significant and difficult research problem. Almost the visual surveillance systems' entire first step is detecting foreground objects. This both creates a focus of attention for higher processing levels such as tracking, classification and behavior understanding and reduces computation time considerably since only pixels belonging to foreground objects need to be dealt with. Short and long term dynamic scene changes such as repetitive motions (*e.g*. waiving tree leaves), light reflectance, shadows, camera noise and sudden illumination variations make reliable and fast object detection difficult. Hence, it is important to pay necessary attention to object detection step to have reliable, robust and fast visual surveillance system.

The system diagram of our object detection method is shown in Figure 2. Our method depends on a six stage process to extract objects with their features in video imagery. The first step is the background scene initialization.

In order to evaluate the quality of different background scene models for object detection and to compare run-time performance, by implementing three of these models which are adaptive background subtraction, temporal frame differencing and adaptive online Gaussian mixture model. The background scene related parts of the system is isolated and its coupling with other modules is kept minimum to let the whole detection system to work flexibly with any one of the background models.

Next step in the detection method is detecting the foreground pixels by using the background model and the current image from video. This pixel-level detection process is dependent on the background model in use and it is used to update the background model to adapt to dynamic scene changes. Also, due to camera noise or environmental effects the detected foreground pixel map contains noise. Pixel-level post-processing operations are performed to remove noise in the foreground pixels.

Once extracting the filtered foreground pixels, in the next step, connected regions are found by using a connected component labeling algorithm and objects' bounding rectangles are calculated. The labeled regions may contain near but disjoint regions due to defects in foreground segmentation process. Hence, it is experimentally found to be effective to merge those overlapping isolated regions.

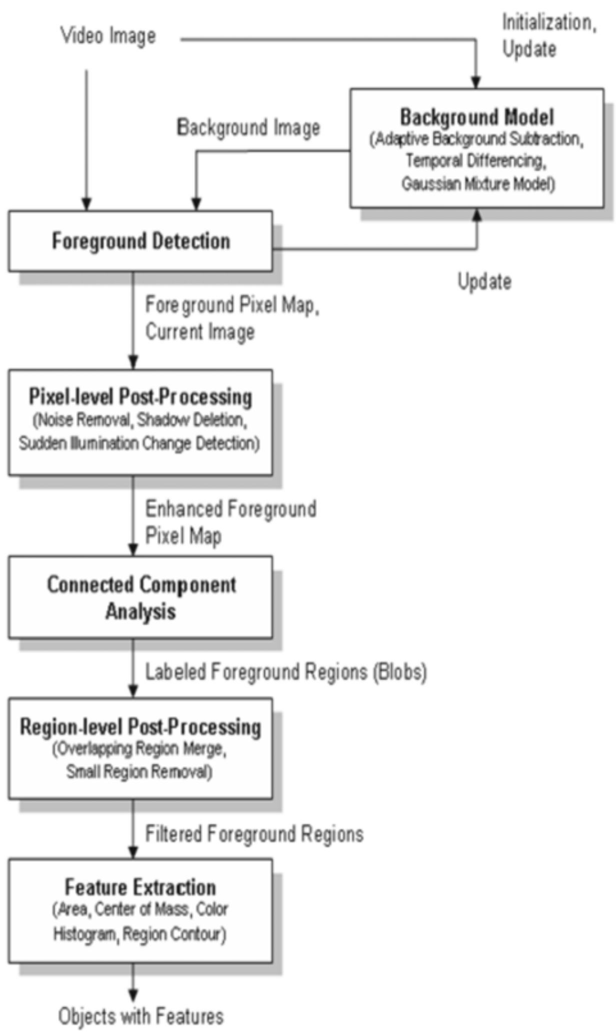

Fig. 2. The object detection system diagram.

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

In the final step of the detection process, a number of object features are extracted from current image by using the foreground pixel map. These features are the area, center of mass and color histogram of the regions corresponding to objects.

2.1.1. Foreground Detection

The system is of a combination of a background model and low-level image post-processing methods to create a foreground pixel map and extract object features at every video frame. Background models generally have two distinct stages in their process: initialization and update.

2.1.2. Pixel Level Post-Processing

The outputs of foreground region detection algorithms generally contain noise and therefore are not appropriate for further processing without special post-processing. There are various factors that cause the noise in foreground detection such as:

- **Camera noise :** This is the noise caused by the camera's image acquisition components. The intensity of a pixel that corresponds to an edge between two different colored objects in the scene may be set to one of the object's color in one frame and to the other's color in the next frame.
- **Reflectance noise :** When a source of light, for instance sun, moves it makes some parts in the background scene to reflect light. This phenomenon makes the foreground detection algorithms fail and detect reflectance as foreground regions.
- **Background colored object noise :** Some parts of the objects may have the same color as the reference background behind them. This resemblance causes some of the algorithms to detect the corresponding pixels as non-foreground and objects to be segmented inaccurately.
- **Shadows and sudden illumination change :** Shadows cast on objects are detected as foreground by most of the detection algorithms. Also, sudden illumination changes (e.g. turning on lights in a monitored room) makes the algorithms fail to detect actual foreground objects accurately.

Morphological operations, erosion and dilation[2], are applied to the foreground pixel map in order to remove noise that is caused by the first three of the items listed above. Our aim in applying these operations is removing noisy foreground pixels that do not correspond to actual foreground regions (let us name them non-foreground noise, shortly NFN) and to remove the noisy background pixels (non-background noise, shortly NBN) near and inside object regions that are actually foreground pixels. Erosion, as its name implies, erodes one-unit thick boundary pixels of foreground regions. Dilation is the reverse of erosion and expands the foreground region boundaries with one-unit thick pixels. The subtle point in applying these morphological filters is deciding on the order and amounts of these operations. The order of these operations affects the quality and the amount affects both the quality and the computational complexity of noise removal.

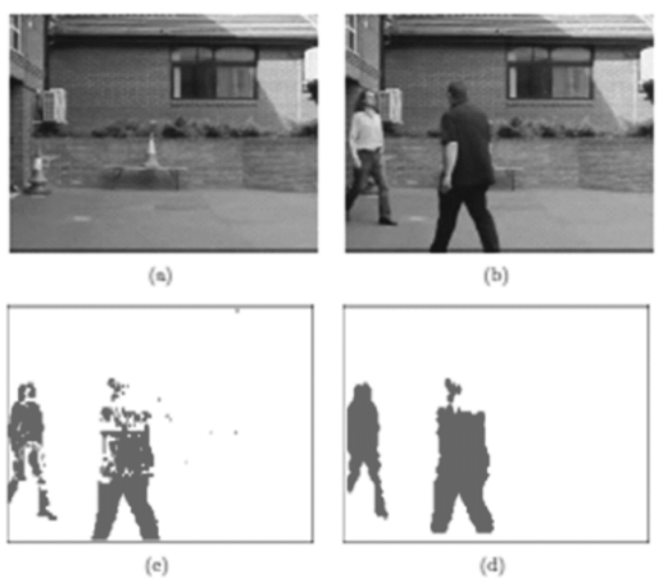

Fig. 3. Pixel level noise removal sample. (*a*) Estimated background image (*b*) Current image (*c*) Detected foreground regions before noise removal (*d*) Foreground regions after noise removal

For instance, if on applying dilation followed by erosion it cannot get rid of one-pixel thick isolated noise regions (NFN) since the dilation operation would expand their boundaries with one pixel and the erosion will remove these extra pixels leaving the original noisy pixels. On the other hand, this order would successfully eliminate some of the non-background noise inside object regions.

In case on applying these operations in reverse order, which is erosion followed by dilation, we would eliminate (NFN) regions but this time we would not be able to close holes inside objects (NBN).

After experimenting with different combinations of these operations, we have come up with the following sequence: two-levels of dilation followed by three levels of erosion and finally one-level of dilation. The first dilation operation removes the holes (NBN) in foreground objects that are detected as background and expands the regions' boundaries. In the next step, three-levels of erosion removes the extra pixels on the region boundaries generated by the previous step and removes isolated noisy regions (NFN). The last step, one level of dilation, is used to compensate the one-level extra effect of erosion. Figure 3 shows sample foreground regions before and after noise removal together with original image.

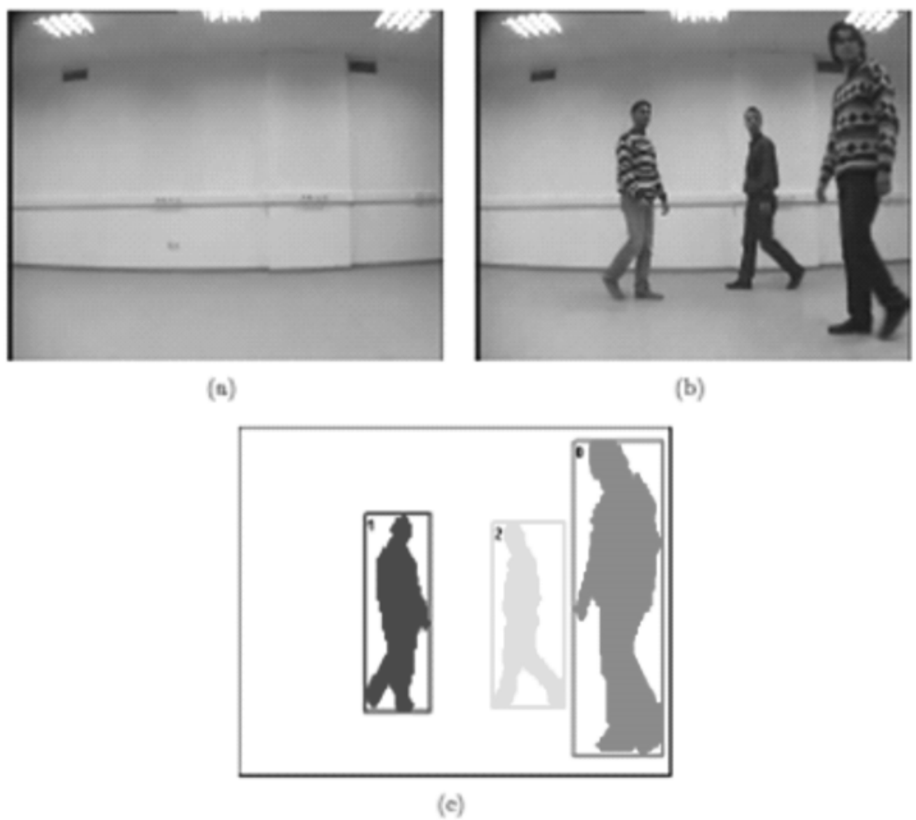

Fig. 4. Connected component labeling sample. (*a*) Estimated background (*b*) Current image (*c*) Filtered foreground pixels and connected and labeled regions with bounding boxes

Note that the resolution of actual image (320 × 240) is different than the one used for foreground detection (160 × 120). Removal of shadow regions and detecting and adapting to sudden illumination changes require more advanced methods which are explained in the next section.

3.1.3. Detecting Connected Regions

After detecting foreground regions and applying post-processing operations to remove noise and shadow regions, the filtered foreground pixels are grouped into connected regions (blobs) and labeled by using a two-level connected component labeling algorithm presented in [2]. After finding individual blobs that correspond to objects, the bounding boxes of these regions are calculated. Figure 4 shows sample foreground regions before and after region connecting, labeling and boxing.

3.1.4. Region Level Post-Processing

Even after removing pixel-level noise, some artificial small regions remain due to inaccurate object segmentation. In order to eliminate this type of regions, the average region size (α) in terms of pixels is calculated for each frame and regions that have smaller sizes than a fraction (γ) of the average region size (Size (region) < $\alpha * \gamma$)are deleted from the foreground pixel map.

Also, due to segmentation errors, some parts of the objects are found as disconnected from the main body. In order to correct this defect, the bounding boxes of regions that are close to each other are merged together and the region labels are adjusted.

2.1.5. Extracting Object Features

Once have segmented regions we extract features of the corresponding objects from the current image. These features are size (S), center of mass (Cm), color histogram (Hc) and silhouette contour of the object's blob. Calculating the size of the object is trivial and we just count the number of foreground pixels that are contained in the bounding box of the object.

3. CONCLUSION

This work presents novel algorithms for open-sea visual maritime surveillance using a highly non-stationary camera. The camera installed on a buoy is a subject to rapid erratic motion. The proposed algorithm detects, localizes, and tracks ships in the field of view of the camera and outputs images of the found targets. The experiments, conducted on a large dataset of video data obtained from a prototype of a buoy-based surveillance system, show good results.

Specifically, the algorithm detects and tracks correctly of up to 88% of ships. In the context of ship detection, a new horizon detection scheme was developed for a complex maritime domain that provides accuracy of horizon localization of 98%, and detection of horizon images with the rate of 99%. To the best of our knowledge this is the first work that focuses on low-quality image data from highly non-stationary camera. The developed algorithms are fast and are well suited for low-powered autonomous systems deployed for long periods of time.

4. REFERENCES

1. L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. Pattern Recognition, 36(3):585–601, March 2003.
2. F. Heijden. Image Based Measurement Systems: Object Recognition and Parameter Estimation. Wiley, January 1996.
3. A. Amer. Voting-based simultaneous tracking of multiple video objects. In Proc. SPIE Int. Symposium on Electronic Imaging, pages 500–511, Santa Clara, USA, January 2003.
4. R. Bodor, B. Jackson, and N. Papanikolopoulos. Vision-based human tracking and activity recognition. In Proc. of the 11th Mediterranean Conf. on Control and Automation, June 2003.
5. H.Ramoser, T.Schlgl, M.Winter, and H.Bischof. Shape-based detection of humans for video surveillance. In Proc. of IEEE Int. Conf. on Image Processing, Barcelona, Spain, 2003.