



Modified Approach for Object Detection in Video Sequences

Rajni Nema¹, Dr. A.K.Saxena²

Shri Ram College of Engineering and Management Banmore^{1,2}
Gwalior INDIA

Abstract: Object detection in video sequence processes is the basic and starting steps for more complex processes, such as video context analysis and multimedia indexing. A simple method of object detection for static camera movies is proposed. In this paper, First, Canny Edge operator applies on video sequence. It detects the video objects by edge detection. Then some morphological processes give the object in video sequence. This paper proposed a modified approach for object detection in video sequence. The software is developed using MATLAB R2010a.

I. Introduction

Video object segmentation, detection and tracking processes are the basic, starting steps for more complex processes, such as video context analysis and multimedia indexing. Object tracking in videos can be defined as the process of segmenting an object of interest from a sequence of video scenes. This process should keep track of its motion, orientation, occlusion and etc. in order to extract useful context information, which will be used on higher-level processes [1].

The procedure of moving object tracking is to decide whether there exist objects moving into video and to position the target basically and recognize it. A video sequence is made of basically a series of still images at a very small interval time between each capture. As the video sequence consists of frame sequences which have certain temporal continuity, the detection for moving objects in video is conducted in a way that frame sequences are extracted from the video sequence according to a definite cycle [2].

Therefore, moving object detecting has something similar to object detection in still images. Only moving object detecting is more relying on the motion characteristics of objects, i.e. the continuity of time, which is the difference between moving objects and object detection in still images. The need of real-time object detection for video surveillance has spawned a huge amount of our daily life, especially in some domains where it has received.

In the literature, the previous works concentrated mainly on moving-object D&T in videos. One can find bunch of methods dedicated to generic-object D&T in video processing like Background Subtraction (BS) [5, 6], Mean-Shift (MS) and/or Continuously Adaptive Mean-Shift (CMS) [7-9], Optical Flow (OF) [10, 11], Active Contour Models (i.e. Snakes) [12, 13] and etc. Template matching is an essential-object D&T method, but it is simpler than others, and is generally based on matching a given template as an object in giving a frame.

In this paper, we propose an efficient algorithm for detecting a moving object using canny edge operator and some morphological process. Section 2 focused on the methodology. Section 3 focused on canny edge detection method and morphological process. Section 4 provides comparable results, and finally the conclusion is drawn in section 5.

II. Methodology

In the present algorithm, we assume that the background is stationary for the video clips considered. The architecture and modeling of the proposed algorithm are shown in Figure 1.

Input

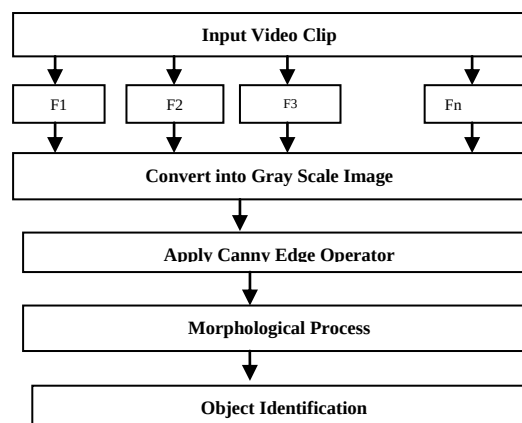


Figure 1. Architecture for Object Identification

In the first step we collect video clips and generate the frames.

In the second step we convert each frame into the grayscale image. It is necessary for third step that is edge detection of the image using the canny operator. Edge detection is one of the most frequently used techniques in digital image processing. The boundaries of object surfaces in a scene often lead to oriented localized changes in intensity of an image, called edges [3]. Image Edge detection significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. Since edge detection is at the forefront of image processing for object detection.

The result of canny edge algorithms uses for morphological process, where the difference of two successive video frames indicates the motion between them, the resulted non-black image zones representing the moving regions. In this step provides the object.

III. Canny Edge Detection Algorithm

The Canny edge detection algorithm is known to many as the optimal edge detector. Canny's intentions had been to enhance the many edge detectors already out at the time he started his work. He was very successful in achieving his goal and his ideas and methods can be found in his paper, "A Computational Approach to Edge Detection"[14]. In his paper, he followed a list of criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be no responses to non-edges. The second criterion is that the edge points are well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first two were not substantial enough to completely eliminate the possibility of multiple responses to an edge. Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (nonmaximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2 [15].

Step 1: - In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width is increased.

Step 2: - After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. The Sobel operator [6] uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). They are shown below:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The magnitude, or edge strength, of the gradient is then approximated using the form

$$|G| = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

Step 3: - The direction of the edge is computed using the gradient in the x and y directions. However, an error will be generated when sum x is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If G_y has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The method for finding the edge direction is just:

$$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right)$$

Step 4: - Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5x5 image are aligned as follows:

```

x x x x x
x x x x x
x x a x x
x x x x x
x x x x x
    
```

Then, it can be seen by looking at pixel "a", there are only four possible directions when describing the surrounding pixels - 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), or 135 degrees (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. If the orientation angle is found to be 3 degrees, make it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions. Therefore, any edge direction falling within the yellow range (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the green range (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the blue range (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the red range (112.5 to 157.5 degrees) is set to 135 degrees.

Step 5: - After the edge directions are known, non-maximum suppression now has to be applied. Non-maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image.

Step 6: - Finally, hysteresis [11] is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels. If you think of following an edge, you need a gradient of T2 to start but you don't stop till you hit a gradient below T1.

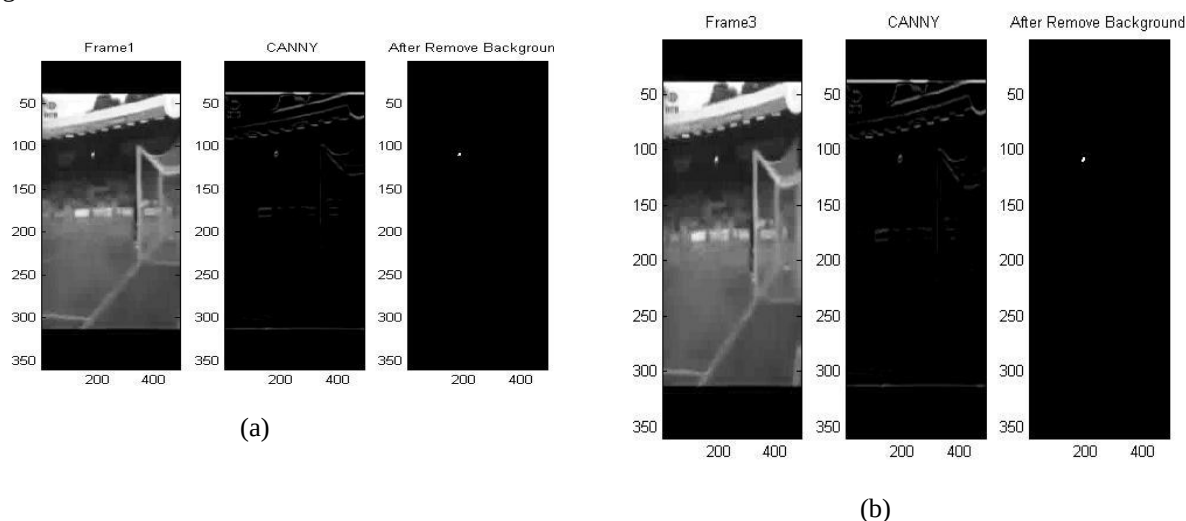
IV. Morphological Process

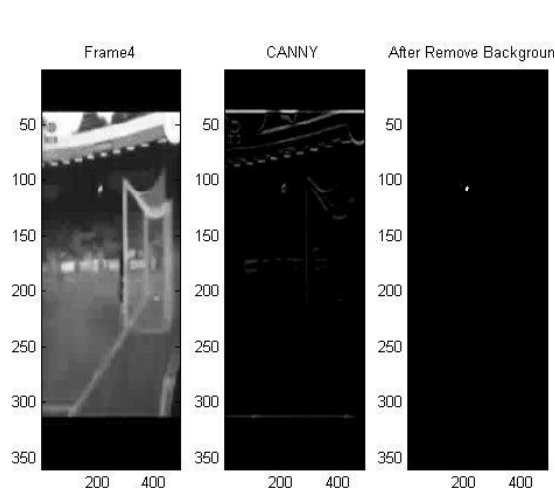
Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors [4]

The difference of two successive video frames indicates the motion between them, the resulted non-black image zones representing the moving regions.

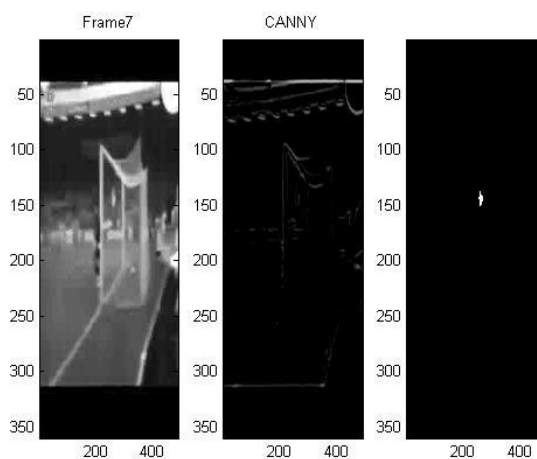
V. Result

We apply this proposed work on two video clips. Figure2 (a, b, c, d, e, f) show the object (football) location is changed in frames.





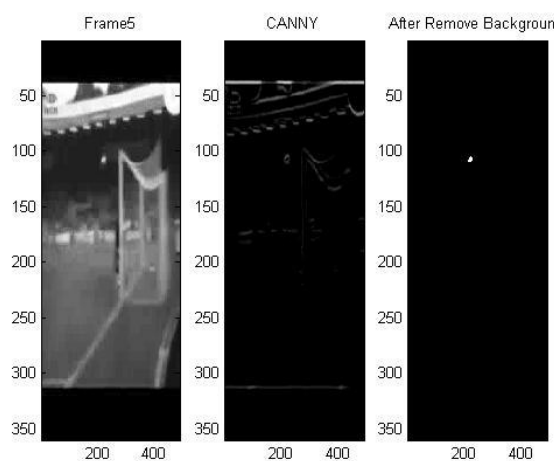
(c)



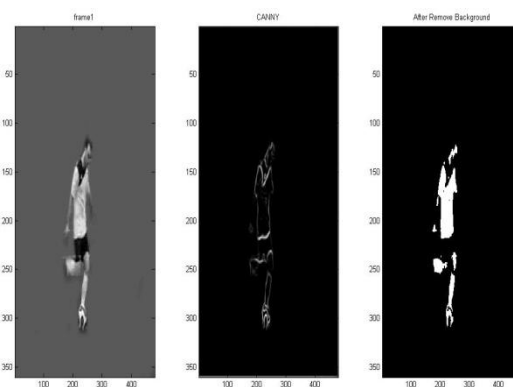
(f)

Figure 2: Results of object detection in video sequence

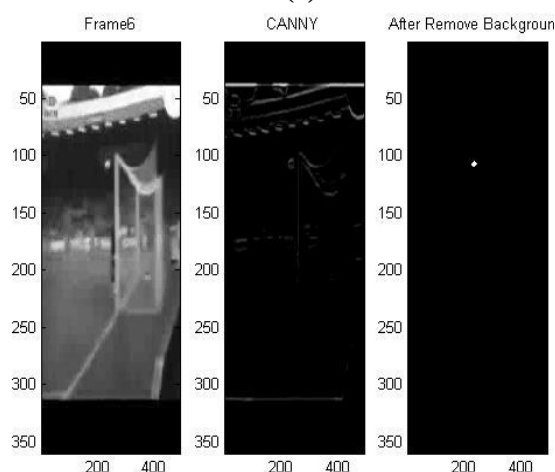
Figure3 (a, b, c, d, e, f) shows the object (football player) movement change in frames.



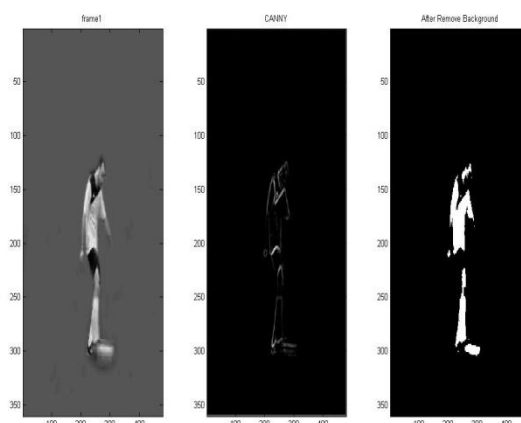
(d)



(a)



(e)



(b)

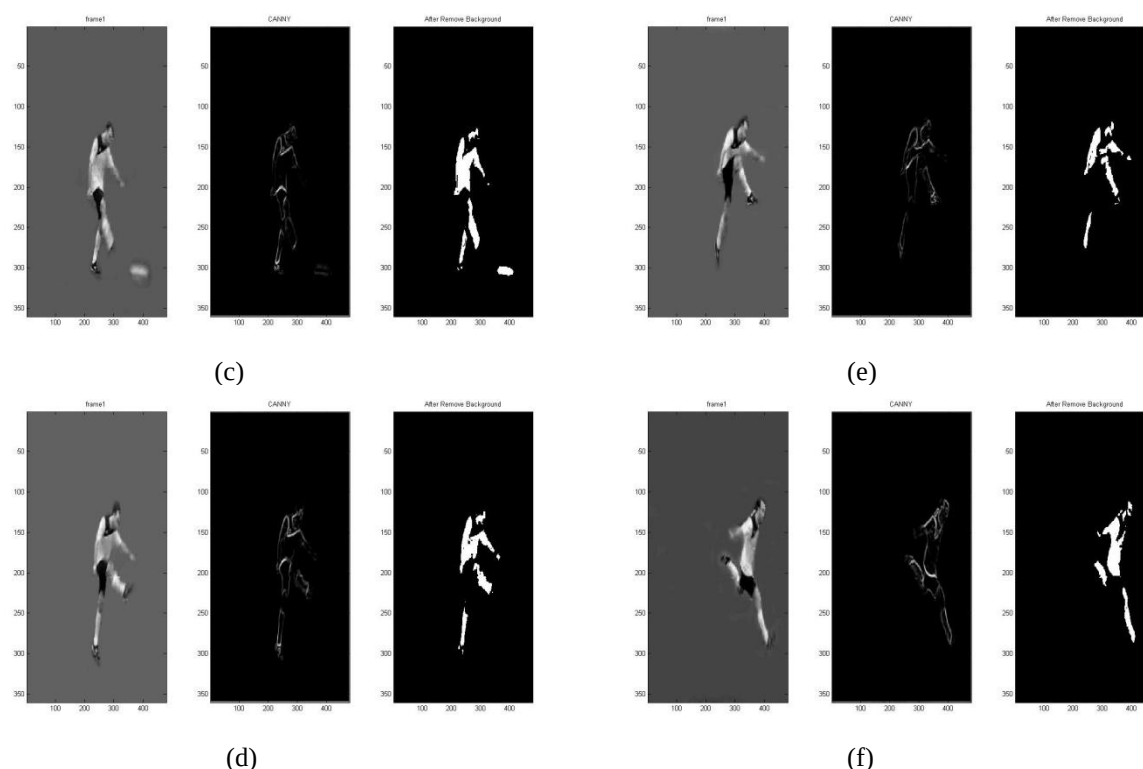


Figure 3: Results of object detection in video sequence

VI. CONCLUSIONS

In this paper, we propose an efficient algorithm for detecting a moving object using a canny edge detector and morphology process. It can be seen from the analysis and examples that the computer language Matlab has the characteristics of simple programming, easy operation and high processing rate, etc. when used in series of processing of moving object detecting algorithm. Initially, we convert the frame image to edge frames then frame difference between two consecutive input frames give the location of the moving object. Then, each incoming frame is compared to the previous frame. Finally, the resulted non-black image zones representing the moving regions. As a result, the implementation of the algorithm becomes very fast.

References

- [1] Bahadir KARASULU, "Review and evaluation of well- Known methods for moving object detection and tracking in videos", *JOURNAL OF AERONAUTICS AND SPACE TECHNOLOGIE*, JULY 2010 VOLUME 4 NUMBER 4 (11-22)
- [2] Kauleshwar Prasad, Richa Sharma and Deepika Wadhvani, "A Review on Object Detection in Video processing" *International Journal of u- and e- Service, Science and Technology* Vol. 5, No. 4, December, 2012
- [3] Senthilkumaran.N and R.Rajesh, "Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches", School of Computer Science and Engineering, Bharathiar University, Coimbatore -641 046, India
- [4] R. C. Gonzalez and R. E. Woods. "Digital Image Processing". 2nd ed. Prentice Hall, 2002.
- [5] Carmona, E. J., Martínez-Cantos, J., and Mira, J., "A new video segmentation method of moving objects based on blob-level knowledge", *Pattern Recognition Letters*, Vol. 29, No. 3, pp. 272-285, 2008.
- [6] Kim, J.B., Kim, H.J., "Efficient region-based motion segmentation for a video monitoring system", *Pattern Recognition Letters*, Vol. 24, No. 1-3, pp. 113-128, 2003.
- [7] Bradski, G. "Computer Vision Face Tracking For Use in a Perceptual User Interface", In *Intel Technology Journal*, (http://developer.intel.com/technology/itj/q21998/articles/art_2.htm, (Q2 1998).
- [8] François, A. R. J., "CAMSHIFT Tracker Design Experiments with Intel OpenCV and SAI", *IRIS Technical Report IRIS-04-423*, University of Southern California, Los Angeles, 2004.
- [9] Comaniciu, D., Meer, P., "Mean Shift Analysis and Applications", *IEEE International Conference Computer Vision (ICCV'99)*, Kerkyra, Greece, pp. 1197-1203, 1999.
- [10] Jodoin, P.M., Mignotte, M., "Optical-flow based on an edge-avoidance procedure", *Computer Vision and Image Understanding*, Vol. 113, No. 4, pp. 511-531, 2009.
- [11] Pauwels, K., Van Hulle, M. M., "Optic flow from unstable sequences through local velocity constancy maximization", *Image and Vision Computing, (The 17th British Machine Vision Conference (BMVC 2006))*, Vol. 27, No. 5, pp. 579-587, 2009.
- [12] Kass, M., Witkin, A., and Terzopoulos, D., "Snakes: active contour models", *International Journal of Computer Vision*, Vol. 1, No. 4, pp. 321-331, 1988.
- [13] Dagher, I., Tom, K. E., "WaterBalloons: A hybrid watershed Balloon Snake segmentation", *Image and Vision Computing*, Vol. 26, pp. 905-912, doi:10.1016/j.imavis.2007.10.010, 2008.
- [14] J. F. Canny. "A computational approach to edge detection". *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 6, pp. 679-697, 1986
- [15] Maini Raman and Agrawal Himanshu, "Study and Comparison of Various Image Edge Detection Techniques", Punjabi University, Patiala-147002(Punjab), India.