Fast Object Detection Method for Visual Surveillance

Hyo-Kak Kim, Suryanto, Dae-Hwan Kim, Dongni Zhang and Sung-Jea Ko
Department of Electronics Engineering, Korea University, Seoul, Korea
Email: hkkim81@korea.ac.kr

Abstract: Most of the algorithms developed for object detection employ a background-subtraction technique which requires heavy computation. In this paper, we present a fast background-subtraction technique which can be readily applied to many existing object detection algorithms. The proposed technique consists of three parts: persistent background-subtraction, background-subtraction with nearby searching, and skipped background-subtraction. Experimental results show that the proposed technique can detect the moving objects effectively without any degradation of reliability.

1. Introduction

Detection of moving objects is the first step of video surveillance, which is followed by object classification, tracking, and behavior interpretation. Hence, the object detection itself should have low computational complexity to allow the following video analysis to be carried out in real time.

Many techniques have been developed to detect moving objects [1]-[10]. These techniques can be classified into two categories: the pixel-based and the region-based methods. In the simplest approach of the pixel-based methods, the difference between current and background frames is achieved using a threshold. In this method, the threshold is selected based on the pixel value distribution, which can be modeled by Gaussian distribution.

Even though the pixel-based methods require less computation, they are sensitive to illumination change and noise. A more robust region-based method has been proposed by Nagel et al. [3]. The algorithm defines a likelihood ratio to check whether or not the intensity distributions of moving object and background regions are identical. Since this algorithm compares the texture features instead of the intensity value, it is robust to shadow and illumination changes. However, if the texture features of moving object and background are similar, they cannot be discriminated. To solve this problem, Li et al. [4] introduced a hybrid scheme that combines both the pixel-based and region-based methods. The performance of this hybrid approach is satisfactory, whereas the computational requirement is high.

In general, object detection in outdoor environment requires more difficult tasks since the background is inherently non-stationary. In the non-stationary background, one pixel location can have more than one pixel value, each representing different object in the background. In this case, the background can be modeled by Mixture of Gaussian (MOG) [5][6][7], Kernel Density Estimation (KDE) [8][9], and Codebook (CB) [10]. Even though the MOG and the KDE algorithms have been widely used to model the non-stationary background, they suffer from low sensitive detection, difficult background adaptation, and high memory requirement. In order to overcome these problems, the CB algorithm constructs a highly compressed background model. However, it is too complex for real time realization. Therefore, there is an emerging need for a fast technique to reduce the computation load.

In order to address the computational problem, in this paper, we propose a fast algorithm which can be easily integrated with existing object detection algorithms.

2. Proposed Fast Algorithm

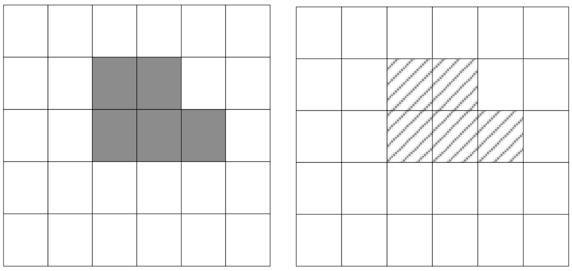
Generally, video data for surveillance system is recorded in special environment. Hence, video frames and moving objects in the video surveillance system have different properties as follows:

1. Temporal persistence: if a pixel is classified as an object in a previous frame, it is highly probable that the pixel is still an object in a current frame.
2. Spatial compactness: the foreground area is much smaller than the background area.
3. Spatial translation: background scene can be translated due to the incomplete alignment of input images or due to a small camera shake.

Based on these three properties, we propose a fast technique which can be applied to a background-subtraction step of existing object detection algorithms.

2. 1 Persistent Background-Subtraction

Generally, moving objects appear in a few frames and occupy only a small area of a frame. Hence, applying the background-subtraction to every pixel to be classified as a background is a waste of computational power. If the object regions are predicted, we can minimize the waste of computational power. According to the property 1 (temporal persistence of moving objects), object regions in a preceding frame are expected to be classified as object regions in a following frame, too. Thus, we propose to exploit temporal persistence of a moving object in successive frames. Let the shaded squares be the pixels of moving objects in Fig. 1 (a). The corresponding pixels in the next frame are evaluated if these pixels are parts of moving objects as shown in Fig. 1 (b).

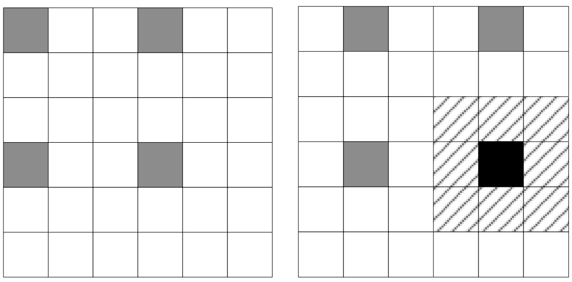


(n-1) frame (a) (n) frame (b)

Fig. 1. An example of persistent background-subtraction. A square indicates a pixel.

2. 2 Skipped Background-Subtraction

Using the persistent background-subtraction, we can detect the moving objects without the waste of computation. But moving objects which appear in the current frame for the first time cannot be detected with the persistent background-subtraction. Thus, in order to minimize the waste of computation while maintaining high detection rate, we propose to apply the background-subtraction at only every N pixel location instead of every pixel based on the property 2 (spatial compactness of moving objects). And, if once a pixel is classified as a moving object, its neighboring skipped pixels are tested because an object is composed of more than a pixel. Fig. 2 shows an example of the proposed skipped background-subtraction technique. In Fig. 2 (a), shaded squares indicate the pixels to be checked and white squares show the pixels to be skipped. In Fig. 2 (b), the background-subtraction is applied to the slashed squares regardless of skipped background-subtraction since a black square is detected as an object pixel.



(n-1) frame (a) (n) frame (b)

Fig. 2. An example of skipped background-subtraction. A square indicates a pixel.

2. 3 Background-Subtraction with Nearby Searching

In pixel-based algorithms, each pixel location has its own background model that is used for the background-subtraction. Thus, it is essential that the background model and the input scene are strictly aligned. However, same background can be imaged into nearby pixel location, especially in case of the video sequence with high resolution. And small camera shake causes the misalignment, too. Therefore, in this paper, we propose to compare the pixel value not only to the background model of the co-located pixel but also to the background models of nearby pixels. If any background model is matched to the current pixel, the pixel is classified as a foreground. Fig. 3 shows comparing directions of each comparing pixel.

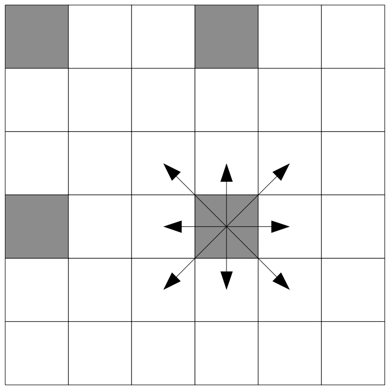


Fig. 3. Nearyby comparing directions.

3. Experimental Results

We implemented the proposed fast technique to the CB algorithm [10] and the KDE algorithm [8]. The experiment is performed on a 3.20 GHz Intel Pentium 4 processor with 1 GB RAM. And the video sequences are compressed using Xvid codec with 320x240 frame size.

Fig. 3 shows the experimental results in the CB and the KDE algorithms. We can see that the proposed algorithm does not degrade the detection accuracy.

Table 1 shows the average processing time of a frame in two cases: frames with objects and frames without objects. In Table 1, it is seen that the proposed fast technique reduces the computation time of the conventional algorithms in the both cases.

We also evaluated the effect of the background-subtraction with nearby searching technique. In Table 2 and Table 3, we can see that the proposed nearby searching technique slightly increases the average processing time in 'School' sequence because nearby searching technique requires more computation. However, in case of the 'Tree' sequence, the average processing time is reduced since the nearby searching technique minimizes the false detection which results in additional computation.

This paper proposed a fast background-subtraction technique that improves the computational speed of the existing object detection algorithms. Experimental results show that the proposed algorithm can reduce the processing time of the convention object detection algorithms while remaining detection quality.

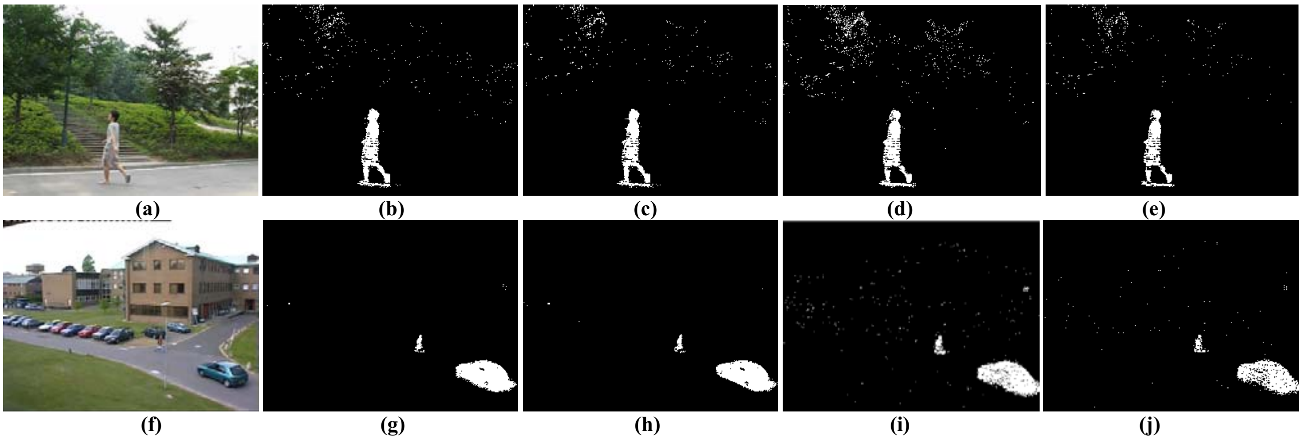


Fig. 3. Result of background-subtraction. (a) ‘ Tree’ sequence (b) result of the CB in ‘ Tree’ sequence (c) result of the CB with the proposed algorithm (d) result of the KDE (e) result of the CB with the proposed algorithm (f) ‘ School’ sequence (g) result of the CB (h) result of the CB with proposed algorithm (i) result of the KDE (j) result of the KDE with the proposed algorithm

Table 1. Average processing time (ms)

Sequence	Algorithm	No Object	Objects
Tree	CB	17.24	20.40
	Proposed in CB	3.58	6.56
	KDE	15.16	16.44
	Proposed in KDE	5.77	11.33
School	CB	15.76	18.40
	Proposed in CB	2.82	6.56
	KDE	8.40	10.00
	Proposed in KDE	5.16	9.73

Table 2. Average processing time (ms) in the CB algorithm

Sequences	CB	Fast w/o Nearby	Fast w/ Nearby
Tree	18.80	5.14	4.51
School	16.74	3.74	4.35

4. Conclusion

This paper proposed a fast background-subtraction technique that improves the computational speed of the existing object detection algorithms. Experimental results show that the proposed algorithm can reduce the processing time of the convention object detection algorithms while remaining detection quality.

5. Acknowledgment

This research was supported by Seoul Future Contents Convergence (SFCC) Cluster established by Seoul R&BD Program.

References

[1] C.R. Wren, A. Azarbayejani, T. Darrell, and A. Petland, “Pfinder: Real-Time Tracking of the Human Body,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 780-785, 1997.

[2] W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee, “Using Adaptive Tracking to Classify and Monitor Activities in a Site,” in *Proc. Int. Conf. on Computer Vision*, pp. 274-279, 1998.

[3] H. Nagel, “Formation of an object concept by analysis of systematic time variation in the optically perceptible environment,” *Comput. Graph. Image Process*, vol. 7, no. 10, pp. 149-194, 1978.

[4] L. Li and M.K.H. Leung, “Integrating Intensity and Texture Differences for Robust Change Detection,” *IEEE Trans. Image Processing*, vol. 111, no. 2, pp. 105-112, 2002.

[5] N. Friedman and S. Russell, “Image Segmentation in Video Sequences: A Probabilistic Approach,” in *Proc. Annual Conf. on Uncertainty in Articial Intelligence*, pp. 175-181, 1997.

[6] W.E.L. Grimson, C. Stauer, and R. Romano, “Using Adaptive Tracking to Classify and Monitor Activities in a

Site," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 22-29, 1998.

[7] W.E.L. Grimson and C. Stauer, "Adaptive Background Mixture Models for Real-Time Tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 252, 1999.

[8] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric Model for Background Subtraction," *Lecture Notes In Computer Science*, vol. 1834, pp. 751-767, 2000.

[9] Y. Sheikh and M. Shah "Bayesian Modeling of Dynamic Scenes for Object Detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol 27 no. 11, pp. 1778-1792, 2005.

[10] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time Foreground-Background Segmentation using Codebook Model," *Real-time Imaging*, vol. 11, no. 3, pp. 167-256, 2005.