2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)

Unattended Object Detection based on Edge-Segment Distributions

Jaemyun Kim
Kyung Hee University
South Korea
sense21c@khu.ac.kr

Adín Ramírez Rivera
Universidad Diego Portales
Chile
adin.ramirez@mail.udp.cl

Byungyong Ryu
Kyung Hee University
South Korea
read100nm@khu.ac.kr

Kiok Ahn
Kyung Hee University
South Korea
kiokahn@khu.ac.kr

Oksam Chae
Kyung Hee University
South Korea
oschae@khu.ac.kr

Abstract

Unattended object detection is an important task in surveillance. Thus, we propose a new method to detect unattended object by modeling the objects as newly learned temporal background. We use edge-segments to model the structural changes in the scene. Specifically, we construct distributions of these edge-segments to analyze the scene, and to segment its different components: background, foreground, and the interesting new objects. Additionally, we propose a clustering algorithm to recover the unattended objects from a set of edges based on the assumption that spatially close edges come from the same object. Our experiments on several datasets validate our proposed method.

1. Introduction

These days, people suffer unpredictable threats, such as terrorism. Especially, explosive and chemical attacks using unattended objects are occurring repeatedly in public areas, such as airports, train or subway stations, bus terminals, etc. Many studies about prevention or prediction of threats have been in progress for people safety [3, 14, 17]. The importance of video-based surveillance systems is increasing everyday. However, it generates huge amounts of data. So, an intelligent surveillance system (such as moving object detection, dynamic background change adaptation, or static scene elimination) for unattended objects detection and analysis is required.

Foreground detection for a fixed-camera environment has been studied for long time. In here, several methods apply background subtraction. Basically, background-subtraction-based foreground detection requires a model of reliable background for accurate detection, as consecutive images have different characteristics, such as illumination changes, background movements, and even dynamic situations. According to the way of absorbing these characteristics on the background model, we classify the background modeling methods into two groups: pixel-based and edge-based.

Pixel-based methods model the background using the information directly from the pixels in the frame in a dense manner. These methods are robust to changes in position and orientation of the background environment. However, they have problems separating foreground from background if the color of foreground is similar to the background [9]. Besides, they are sensitive to color changes too, such as illumination change by light sources or shadows [15]. To overcome these problems, a background model updating mechanism is required for every incoming frame. In spite of this effort, pixel-based methods leave a ghost effect when moving objects have sudden speed changes [16]. By applying statistical techniques, previous researchers suppressed the ghost effect. Moreover, pixel-based methods [15] have problems dealing with multi-modal distributions in dynamic environments with illumination and noise changes.

On the other hand, edge-based methods generate background models by using value of edge-pixel (i.e., level of intensity difference) to represent the background variation. These methods are less sensitive than pixel-based methods to illumination changes. Furthermore, edge-based methods do not leave a ghost effect [5, 6]. However, edge-based methods have a critical problem that is edge-distortion, such as position, shape, and orientation variation from a static object in sequential frames. Consequently, edges may not be consistent from consecutive frames. Therefore, these methods are not useful in a simple subtraction-based object detection scheme. Moreover, existing edge-based methods [6] have many false alarms, because they use a simple edge dif-

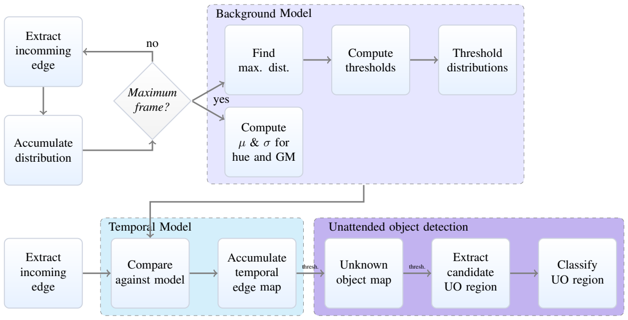

Figure 1: Flow chart for proposed method.

ferencing method. To solve this problem, edge-segment-based methods [7, 8, 10, 11] model background using the connected edges instead.

In this paper, we present an edge-segment-based unattended object detection algorithm to highlight suspicious objects that people dropped or left alone, such as bags or boxes. Our proposed method models background in the presence of moving objects in the training sequence, and then uses the background model to segment the image sequence into the incoming objects and foreground. We train the background model from a set of frames from which we extract the edge behavior and build edge-segment distributions. Additionally, we extract the color and gradient magnitude of the edges to prevent over elimination of foreground when they lie over background distributions. The temporal model accumulates the foreground edge distributions extracted from the incoming frames. Moreover, we detect unattended object by checking the distributions of the accumulated temporal model.

2. Background Modeling

We propose an unattended object detection method in which the unattended objects appear as a new background (i.e., they are in a static position after some time). This detection requires to isolate the detected objects without any foreground that may be present (such as persons or other moving objects) from the incoming frame. Our method is based on a foreground detection method [7], which creates edge-segment distributions from a training sequence as a background and incoming frames. The system adds color and gradient information to the background to disambiguate foreground edges that are confused with background. Additionally, we create an unknown object map from the temporal model to group the edges and detect the unattended objects—Fig. 1 shows the whole process.

2.1. Thin-Edge Extraction

We apply a thin edge extraction, instead of the common Canny edge extraction [2], because Canny may lose edges which have low gradient magnitude. Our thin edge extraction is similar to Canny algorithm, with the difference that the hysteresis step is not used. So our thin edge extraction requires only one threshold. We set a low value for thresholding, because our statistical background model can control infrequent noisy edges. We calculate gradient magnitude and orientation by computing Sobel [4] from an image. Then, a non-maximum suppression [4] with threshold th_{thin} is applied to generate a thin-edge map $\mathbb{E}_b^t$ for the frame t.

2.2. Background Modeling

In order to model the background we perform five different steps. (*Step 1*) We extract edges from each training frame t and generate the thin-edge map $\mathbb{E}_b^t$ (as described above). (*Step 2*) Then, we create edge-distributions by accumulating the edges in an accumulator, ACC, and build a statistical map, SM, by smoothing it. Formally, we define these two maps by

$$\text{ACC}(p) = \sum_{t=t_0}^{t_f} \mathbb{E}_b^t(p), \tag{1}$$

$$\text{SM} = \text{ACC} * G, \tag{2}$$

where p is a pixel position, t_0 and t_f are initial and final training frames, respectively, $\mathbb{E}_b^t$ is a binary thin-edge map for the training frame t, and G is a normalized Gaussian kernel function. (*Step 3*) We designed our background model to tolerate moving objects in the training set. As moving objects leave a small trail in the SM, our method suppresses this small statistical edge's behaviors (caused by the foreground objects, i.e., moving objects have wider shape and lower-accumulated value distribution in the SM in comparison to static background) by computing a threshold T, such that

$$T = \frac{\max\{\text{SM}\}}{v}, \tag{3}$$

where $\max\{\text{SM}\}$ is the maximum value in SM, v is the average minimum speed for moving objects in pixels per frames. Then, we define the Statistical Distribution Map, SDM, by

$$\text{SDM}(p) = \begin{cases} \text{SM}(p) & \text{if SM}(p) > T, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

which contains only fixed background.

(*Step 4*) We generate the optimal background model by trimming the distributions in the previously computed SDM. Therefore, we create a support region, SR, by quadratic approximation [12] on each distribution in the SDM. First, we extract maximum peak segment from each distribution in the SDM by using the Multi-Directional Non-Maximum Suppression [13].

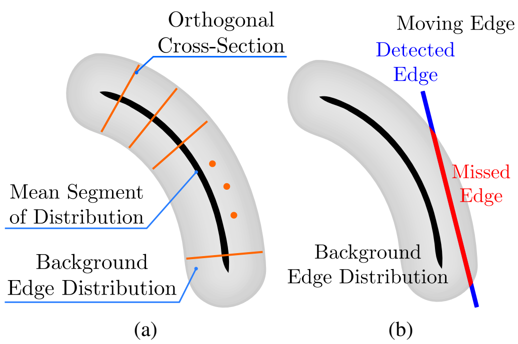

Figure 2: (a) Illustration of orthogonal cross-sections from a mean segment of a distribution. (b) Illustration of missed moving edge by over-elimination.

Second, we extract several orthogonal cross points from the maximum-peak segment on the distribution [as shown in Fig. 2(a)] to approximate the distribution by a quadratic model of the form

$$y = a_0 + a_1 x + a_2 x^2 + \epsilon, \tag{5}$$

where $x = \{x_1, x_2, \ldots, x_n\}$ are the position values within the cross-section, $y = \{y_1, y_2, \ldots, y_n\}$ are the respective accumulation values, and ϵ is an unobserved random error with mean zero conditional on a scalar variable x. Consequently, we find the coefficients a_0, a_1, and a_2 with the minimum error ϵ using a least-squares approach [12]. Furthermore, we define the cutting point to prune the distributions as the intercept with the x axis (i.e., $y = 0$) by

$$p_{\text{cut}} = \begin{cases} -\frac{a_1 \pm \sqrt{a_1^2 - 4a_0 a_2}}{2a_2} & \text{if } a_1^2 - 4a_0 a_2 \geq 0, \\ 0 & \text{otherwise}, \end{cases} \tag{6}$$

and we define the optimal support region, SR, by

$$\text{SR}(p) = \begin{cases} \text{SDM}(p) & \text{if } dist(p, \bar{p}) \leq p_{\text{cut}}, \\ 0 & \text{otherwise}, \end{cases} \tag{7}$$

where p is a pixel position from the distribution with cut point p_{cut}, $\bar{p}$ is the maximum peak location in the distribution, and $dist(\cdot, \cdot)$ measures the Euclidean distance between the arguments.

(*Step 5*) Our edge-segment based background model has a problem when the foreground detection if foreground edges lie in the background distribution. To solve it, we add hue and gradient magnitude information to the regions that represent the background. For each pixel in the distribution, we create a set of Gaussians that model the hue (H) and gradient magnitude (GM). This method will improve result of moving edge detection. Hence, we build two Gaussians, G_{H} and G_{GM}, and each Gaussian is defined by

$$\text{G}_x = \{\mu_x, \sigma_x\}, \tag{8}$$

where μ_x is a mean, σ_x is a standard deviation of the Gaussian, and $x \in \{\text{H}, \text{GM}\}$. And the mean and the standard deviation are defined by

$$\mu_x = \frac{1}{N} \sum_t^N v_x^t, \tag{9}$$

$$\sigma_x^2 = \frac{1}{N} \sum_t^N (\mu_x - v_x^t), \tag{10}$$

where v_x^t is a pixel value of $x \in \{\text{H}, \text{GM}\}$ at frame t, and N is a number of frames.

3. Foreground Detection

We classify the moving edge as foreground by using the background subtraction method. In this detection step, we extract a binary edge map $\mathbb{E}_b^{\text{new}}$ from incoming frame using the same thin-edge extraction method. Then, we classify the edges from $\mathbb{E}_b^{\text{new}}$ as moving edges when they do not lie in an SR distribution (not candidate background edges). This mechanism can quickly extract moving edges from the incoming frame. However, it has a limitation as we mentioned above, we may lose moving edges, as shown in Fig. 2(b). Thus, we compare the edges that lie in an SR distribution using its hue and gradient magnitude. The decision of whether to recover an edge pixel r is as follows:

$$r_x(p) = \begin{cases} 1 & \text{if Cond}_H(p) \wedge \text{Cond}_{GM}(p), \\ 0 & \text{otherwise}, \end{cases} \tag{11}$$

$$\text{Cond}_x(p) = \begin{cases} true & \text{if } |\mu_x(p) - x(p)| \leq c\sigma_x(p), \\ false & \text{otherwise}, \end{cases} \tag{12}$$

where p is a pixel location, $\mu_x(p)$ is the mean value at the position p, $\sigma_x(p)$ is the standard deviation at the position p, $x(p)$ is the value of hue or gradient magnitude at the position p from the current incoming frame, and c is a constant to match the Gaussian distribution (in our experiments we used $c = 1$). Therefore, if $r_x(p)$ is 1 then we classify an edge pixel at the position p as background, otherwise as foreground.

4. Unattended Object Detection

The unattended object appears as a new object in the background scene, but it will have no motion. In order to decide whether we are seeing an unattended object region from the unknown object regions, we applied the following assumptions: 1) People drop unattended object, 2) unattended object has no motion itself, 3) unattended object has fixed shape, and 4) size of unattended object is smaller than a person. Using these assumptions, we convert the temporal edge weight map to rectangular regions. This regions form

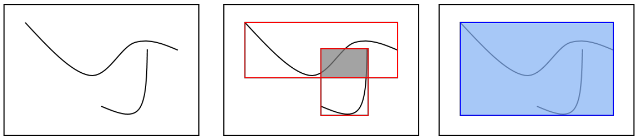

Figure 3: Edge's rectangular boundary and region merging. The red boxes represent the boundary of each edge, and the gray box represents intersected region. And the blue box represents the result of merging the two previous regions.

our unknown map to decide whether candidate regions are unattended object regions or new revealed background regions.

4.1. Temporal Edge Weight Map

We define an unattended object as foreground that has no movement for several frames. For example, a person brought a bag and he drop that bag in the scene. Then, we start suspecting from that bag. Thus, to detect unattended objects, we accumulate moving edges into a temporal edge weight map, TEWM, and exploit the spatial persistence of edges in sequential frames (using a similar mechanism to that proposed by Ramírez Rivera et al. [11]). Therefore, we implement a learning-forgetting approach, such that, if a moving edge appears, we increase the value of the corresponding pixel location, otherwise decrease the value of corresponding pixel location. Formally, the TEWM is defined by

$$\mathrm{TEWM}_t(p) = \begin{cases} \mathrm{TEWM}_{t-1}(p) + T_{\text{inc}} & \text{if edge appears,} \\ \mathrm{TEWM}_{t-1}(p) - T_{\text{dec}} & \text{otherwise,} \end{cases} \tag{13}$$

where p is the pixel location, and T_{inc} and T_{dec} are predefined learning and forgetting values. And to allow a tolerance to small change of edge from noise, we use Gaussian smoothing in this edge map too. Furthermore, we set an upper bound, T_{max}, to the learning weight for an effective elimination of unattended object when that object moves out. Thus, we define

$$\mathrm{TEWM}_t^{\text{bound}}(p) = \min(\mathrm{TEWM}_t(p), T_{\text{max}}), \tag{14}$$

where TEWM_t is the temporal edge weight map for the frame t we are analyzing.

4.2. Unknown Map

We proposed to detect unattended object from every incoming frames by analyzing the accumulated moving edges on TEWM. Unfortunately, moving edges which do not have motion can be classified to unattended object or new appeared background. So, before classifying, we define unknown object by rectangular region. This rectangular-region-based strategy can be a simple method to convert weighted edges to region based representation. In this mechanism, we set a threshold T_{unknown} for separating static temporal edges as unknown edges, UE, by

$$\mathrm{UE}_t(p) = \begin{cases} 1 & \mathrm{TEWM}_t^{\text{bound}}(p) > T_{\text{unknown}} \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

where p is pixel location at frame t, and $\mathrm{TEWM}_t^{\text{bound}}$ is defined as in Eq. (14).

After separating unknown edges, we apply a two step unknown-object region generation. First, we calculate the boundary of each unknown edge that represents its respective rectangular region, i.e., top, left, right, and bottom. The object's edges may be fragmented into several edges. So, we assume that if the boundaries of each edge are overlapped then we merge those edges to represent a region, as shown in Fig. 3. Finally, we have non-overlapped rectangular regions for representing unknown objects.

For every unknown region, we accumulate unknown object regions into a unknown map, UM, for each consecutive frame. Unlike of TEWM accumulation, the UM use rectangular region based accumulation. Hence, the value of a region on UM increases if the region from previous frame intersects T_{inc} with an unknown object from current frame. Consequently, we merge those regions, otherwise we decrease the value of that region by T_{dec}. And, when a new region appears we set the region value to T_{init}.

4.3. Unattended Region Classification

We extract the unknown-object regions by using TEWM and UM. We classify an unknown region from the UM as an object when the accumulation in that region reaches a threshold T_{decision}. In the case of a new background, the region grows isolated. On the contrary, the unattended object's regions grow closer together. Therefore, we set a search window t_{search} for each candidate region to check other unknown regions occurrence. If an unknown region is inside the search window then that candidate region is classified as the unattended object region, otherwise is classified as new appeared background.

4.4. Post-processing

We check the moving edge's frequency, in a frame-wise basis, to remove noise that may influence our decisions. As some edges have been corrupted by noise or small background edge's motion, we performed a size-based refinement step to eliminate those false positive regions. In the detection result, we check the size of each region from detection result. Then, if the size region is too small, we remove that region from the results. Finally, our detection mechanism detects unattended object stably.

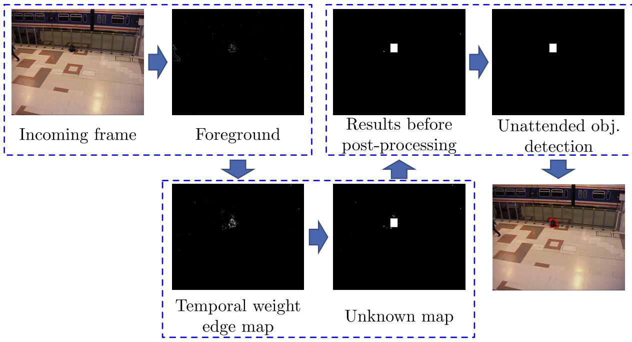

Figure 4: Example of the full processing method for the sequence S1 frame 2126.

Scenario	S1	S2	S4	S5	S6	S7
T_{loiter}	10	32	34	18	24	12
T_{drop}	3.6	2	1.6	4	2.4	3.6

Table 1: Parameters settings for the proposed method in PETS 2006 database [1].

Scenario	Precision	Recall	F-Measure
S1	1	0.803	0.891
S2	0.502	0.375	0.429
S4	1	0.618	0.764
S5	1	0.692	0.818
S6	1	0.563	0.720
S7	1	0.885	0.939
Average	**0.917**	**0.656**	**0.760**

Table 2: Evaluation of the proposed method in six scenarios of the PETS 2006 database [1].

5. Experimental Results

We tested the proposed method on video sequences from Performance Evaluation of Tracking and Surveillance 2006 (PETS 2006) [1]. Specifically, we tested several scenarios to verify our proposed method by using camera 3 for each dataset—except scenario S3 (it does not have unattended objects in whole sequence). The scenarios of the test sets have a person appear in the scene and loiters few seconds in a public station. Then, he puts a bag in the scene and moves out from the scene or switches with another person. So, the bag stays long time alone. Each dataset supports 25 fps and a resolution is 720×576.

5.1. Parameter setting

For reliable-background generation, we define several parameters. In our experiments, we assume a moving object speed of five pixel per frame, $v = 5$. We set the parameters as following, to perform the accumulation and detection,

$$T_{\text{inc}} = 1, \tag{16}$$

$$T_{\text{dec}} = 1, \tag{17}$$

$$T_{\text{max}} = 2 \times T_{\text{unknown}}, \tag{18}$$

$$T_{\text{unknown}} = t_{\text{loiter}} \times \text{fps}, \tag{19}$$

$$T_{\text{init}} = \frac{T_{\text{decision}}}{2}, \tag{20}$$

$$T_{\text{decision}} = t_{\text{drop}} \times \text{fps}, \tag{21}$$

$$t_{\text{search}} = 7, \tag{22}$$

where t_{loiter} is the time (in seconds) that a person loiter before dropping an object, such as a bag, and t_{drop} is time (in seconds) of a person dropping an object. For each frame, if an edge or an object region occurs at the same position then we increase the accumulation by T_{inc}, otherwise decrease accumulation value by T_{dec}. T_{max} limits over-accumulation during unknown object separation and T_{search} sets a size of search window to classify an unattended object region or a background region from an unknown region. We need to set two values for t_{loiter} and t_{drop} that depend on the scenario occurring and the value of threshold T_{search} depends on a size of a person who brings an unattended object. So, in these experiments we set T_{unknown} (which means that a temporal edge moves to the unknown region), t_{loiter} (which means that we will consider the edge as a candidate region if it has been in the scene) as Table 1. The parameters are decided to minimize false alarm.

5.2. Results

We show the entire process for a given frame in Fig. 4. We show how the temporal weight edge map looks after several frames. Then, the edges are transformed into unknown regions in the unknown map. After that we transform the weights into binary regions. These regions are post processed to remove the noisy detections and enhance the resultant object. Finally, we show the detected region masked in the original frame.

We evaluate Precision, Recall, and F-Measure of the proposed method by calculating frames of object detection as shown in Table 2. In each scenario, we focused on minimizing the false alarm to reduce useless information. Also, recall will increase if the length of the datasets is longer. A reason of moving background objects, the evaluation of S2 is not good.

In Fig. 5, we show the results for three frames in each sequence. In general, the scenarios S1 to S7, with exception of S2, have good detection rates and with stable regions. In some results, we detected sub-parts of the objects, but we still recovered it. Scenario S2 has many false alarm due background movement (different parts of the background move and stay close to the object, and for long periods of time). Nevertheless, we can detect the objects too. Also, the object detection is stable when people occludes the object,

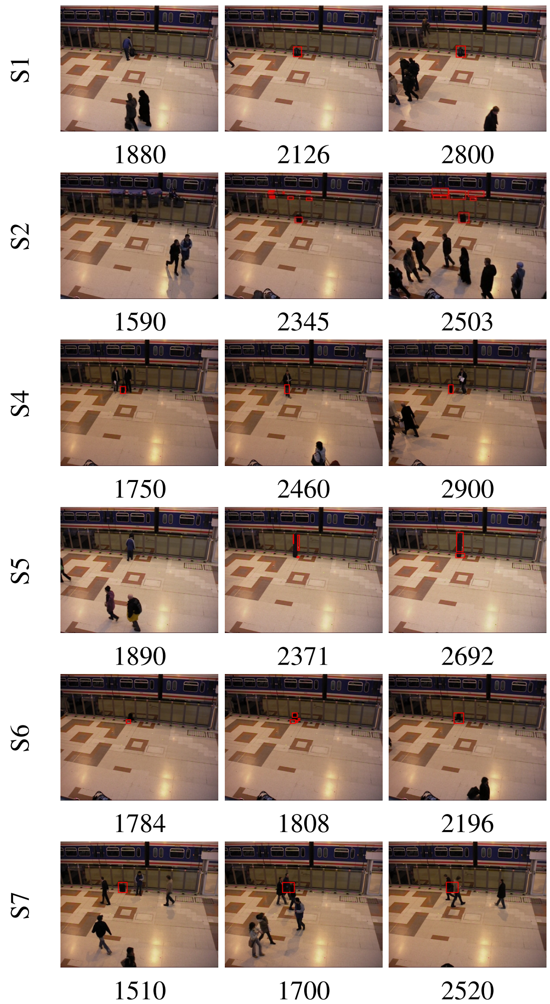

Figure 5: Examples of detection on several frames on each of the different sequences.

even in scenario S7 which has 16% of occlusion.

6. Conclusions

In this paper we presented an unattended object detection algorithm based on edge-segment distributions. The unattended objects were detected by modeling them as new background, in a temporal model. This model is created as new background that arises from incoming objects that become static. We proposed a grouping algorithm to cluster the detected edges, and infer the object position. Our experiments showed that our proposed method is reliable, and less sensitive to objects occlusion and noise.

References

[1] Performance evaluation of tracking and surveillance 2006. ftp://ftp.cs.rdg.ac.uk/pub/PETS2006/, Aug. 2013.

[2] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1987.

[3] M. Elhamod and M. Levine. Automated real-time detection of potentiall suspicious behavior in public transport areas. *Intelligent Transportation Systems, IEEE Transactions on*, 14:688–699, 2013.

[4] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001.

[5] M. Hossain, M. Dewan, and O. Chae. Moving object detection for real time video surveillance: An edge based approach. *IEICE Transactions*, 2007.

[6] C. Kim and J. Hwang. Fast and automatic video object segmentation and tracking for content-based applications. *IEEE Trans. Circuits Syst. Video Technol.*, 2002.

[7] J. Kim, M. Murshed, A. Ramirez Rivera, and O. Chae. Background modelling using edge-segment distributions. *International Journal of Advanced Robotic Systems*, 10, 2013.

[8] J. Kim, A. Ramirez Rivera, G. Song, B. Ryu, and O. Chae. Edge-segment-based background modeling: Non-parametric online background update. In *Advanced Video and Signal-Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, 2013.

[9] J. McHugh, J. Konrad, V. Saligrama, and P. Jodoin. Foreground-adaptive background subtraction. *IEEE Signal Process. Lett.*, May 2009.

[10] A. Ramirez Rivera, M. Murshed, and O. Chae. Object detection through edge behavior modeling. In *AVSS*, 2011.

[11] A. Ramirez Rivera, M. Murshed, J. Kim, and O. Chae. Background modeling through statistical edge-segment distributions. *IEEE Trans. Circuits Syst. Video Technol.*, 2013.

[12] K. Smith. On the standard deviations of adjusted and interpolated values of an observed polynomial function and its constants and the guidance they give towards a proper choice of the distribution of the observations. *Biometrika*, 1918.

[13] C. Sun and P. Vallotton. Fast linear feature detection using multiple directional non-maximum suppression. In *ICPR*, 2006.

[14] M. Takai. Detection of suspicious activity and estimate of risk from human behavior shot by surveillance camera. In *Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on*, 2010.

[15] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 2006.

[16] W. Zhang, Q. Wu, and H. Yin. Moving vehicles detection based on adaptive motion histogram. *Digital Signal Processing*, 2010.

[17] T. T. Zin, P. Tin, H. Hama, and T. Toriu. Unattended object intelligent analyzer for consumer video surveillance. *IEEE Trans. Consumer Electronics*, 52:549–557, 2011.