Background Modeling Through Statistical Edge-Segment Distributions

Adin Ramirez Rivera, *Student Member, IEEE,* Mahbub Murshed, Jaemyun Kim, and Oksam Chae, *Member, IEEE*

***Abstract*—Background modeling is challenging due to background dynamism. Most background modeling methods fail in the presence of intensity changes, because the model cannot handle sudden changes. A solution to this problem is to use intensity-robust features. Despite the changes of an edge's shape and position among frames, edges are less sensitive than a pixel's intensity to illumination changes. Furthermore, background models in the presence of moving objects produce ghosts in the detected output, because high quality models require ideal backgrounds. In this paper, we propose a robust statistical edge-segment-based method for background modeling of non-ideal sequences. The proposed method learns the structure of the scene using the edges' behaviors through the use of kernel-density distributions. Moreover, it uses segment features to overcome the shape and position variations of the edges. Hence, the use of segments gives us local information of the scene, and that helps us to predict the objects and background precisely. Furthermore, we accumulate segments to build edge distributions, which allow us to perform unconstrained training and to overcome the ghost effect. In addition, the proposed method uses adaptive thresholding (in the segments) to detect the moving objects. Therefore, this approach increases the accuracy over previous methods, which use fixed thresholds.**

***Index Terms*—Background modeling, edge-segments, motion detection, object detection.**

I. Introduction

IN MOVING object detection methods which use a fixed camera, moving objects are detected by subtracting the background from the current image. The background model used in this process plays a critical role in determining the performance of the detection. For robust foreground detection, the background model should absorb the shape and illumination variation of the background, overcoming the presence of moving objects and noise. Many background modeling schemes have been proposed for a variety of moving object detection methods [1]–[5]. We classify the background modeling methods according to the feature used for moving object detection: pixel-based and edge-based methods.

The pixel-based methods represent the background with spectral features at each pixel, using one of several techniques [6]–[12] to create a model of each pixel. This approach can produce moving objects in the background model (background ghosts) that compromise the detection capabilities, because the background ghosts can cause false detected foreground. Although statistical techniques (e.g., Gaussians [12], mixture of Gaussians [10], and non-parametric models [6]) have been used to overcome the ghost effect, these methods are susceptible to sudden illumination variations. Thus, traditional pixel-based methods present two problems: multimodal distributions in dynamic environments, and sensitivity to illumination changes and noise.

On the other hand, edge-based—an edge is a feature that is less sensitive to intensity—one limitation of the pixel-based methods. But edges do transform and have position changes. Nevertheless, the use of edges allows this approach to use more robust and expressive models, because they work with fewer pixels than in pixel-based methods. We divide the edge-based methods according to their edge handling scheme: edge-pixel-based methods and edge-segment-based methods.

Several edge-pixel-based methods have been proposed [13]–[22]. Kim and Hwang [18] detected moving objects with an edge difference method using three-edge maps. Using edge-pixel-based frame difference, the authors computed the current moving edges and temporary moving edges; then true moving edges were extracted through an OR operation between them. Dailey *et al.* [14] computed two-edge maps from the edge difference image of three consecutive frames, pairwise. The moving edges were extracted through an AND operation between them. However, these two methods do not use a background model, limiting their detection capabilities. The lack of a background model does not allow them to detect slow moving objects. Furthermore, the absence of a model makes the methods sensitive to variations in shape of the moving objects and noise, and this spreads detection errors through subsequent frames. Also, representing and computing the edges in a pixel-based manner creates problems, e.g., scatter edges and false positives. Treating each edge's points individually neglects the contribution of its shape (i.e., orientation information and changes in that orientation) and neighborhood information.

A better solution for these problems are edge-segment-based methods, which treat each edge as a segment by considering

Manuscript received April 14, 2012; revised July 16, 2012 and September 26, 2012; accepted November 5, 2012. Date of publication January 30, 2013; date of current version July 31, 2013. This work was supported in part by the National Research Foundation of Korea under Grant 2012-0005523 funded by the Korean Government. This paper was recommended by Associate Editor R. C. Lancini.

A. R. Rivera, J. Kim, and O. Chae are with the Department of Computer Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do 446-701, Korea (e-mail: adin@khu.ac.kr; sense21c@khu.ac.kr; oschae@khu.ac.kr).

M. Murshed is with the School of Engineering and Computer Science, Independent University Bangladesh, Dhaka 1229, Bangladesh (e-mail: mmurshed@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2013.2242551

all pixels as a whole, rather than individually. Jain *et al*. [23] proposed a method that models the background based on a sub-pixel edge map, representing the position and orientation of the edge using a mixture of Gaussians. The objects are extracted when no match is found with the background model. However, Jain *et al.*'s method requires adjusting the number of Gaussians manually to avoid the incorporation of ghosts in the background model that are caused by moving objects. Hossain *et al*. [24] made use of an edge-segment-based approach and flexible matching to detect moving objects. A set of lists of edges holds the changes in the scene, and the object is detected from the elimination of edges on these lists. The edges are matched using a fixed threshold, treating all the search regions of the edges in the same way. Then, a watershed-based iterative algorithm is employed to segment the moving object region from the extracted moving edges. Nevertheless, this approach requires motion-free frames to create the background model. In fact, most of these methods apply unrealistic assumptions to model the background with motion-free sequences. Thus, they neglect the scene behavior.

Background modeling in edge-based methods has not been explored as much as pixel-based methods [25], leaving all the methods to use motion-free backgrounds and work with unreal assumptions. Most edge-based background modeling methods assume that a sequence free of moving objects is available to create the model. This assumption is not true in real scenarios, in which it is usually impossible to have a clear background, e.g., a parking lot, a crowded street, or freeways. Therefore, a method that can build a background model in the presence of moving objects is needed. Moreover, the current methods do not overcome the ghost effect caused by moving objects. In addition, the methods are sensitive to edges' shape and position changes. Despite the reality that edges present different changes and should be treated individually, existing edge-based methods use a fixed matching scheme to verify the edges.

In this paper, we present a novel statistical background modeling method based on edge-segments that overcomes illumination variation, moving objects, ghost effects, and edge problems. Hence, the proposed method models the structure in the scene by learning the edges' behavior, which is encoded in the background model as statistical distributions. In addition, we exploit the temporal information of the edges by analyzing the occurrence of the moving edges, and incorporating this into the model.

We propose a three-layer background model. It comprises the statistical map, the temporary edge-segment list, and the detected edge-segment list. Hence, the statistical map is a stable representation of the background learned, while the temporary list is a recent and changing version of the background. Moreover, the detected list represents the recent detected edge-segments that have appeared in the same position in consecutive frames, and that are candidates for becoming temporary edge-segments. In a nutshell, first, we create the frame statistical model, using a kernel-density distribution from the edge-segments. Then, the frame statistical distributions are accumulated using temporal information, and the accumulation is adaptively thresholded, allowing us to use non-ideal frames to learn the background. Furthermore, we detect the foreground by extracting the edge-segments of the current image to create a current edge-segment list. Then, we verify that list against the background model for edge-segments that are considered background; the ones that are not background will compose the detected edge-segment list. And the edge-segments from the detected edge-segment list that do not move after a certain number of frames are moved to the temporary edge-segment list. The moving edge-segments are the remaining edge-segments from the current edge-segment list that were not detected as background. Fig. 1 shows the overall view of the proposed method. Note that the use of statistics and adaptive thresholding helps us to overcome the ghost effect. Moreover, the use of edges makes our method robust against illumination, and treating edges as segments overcomes the edge instability problem that previous edge-based methods did not address.

The remainder of this paper is structured as follows. Section II describes the edge-based background modeling process and its challenges. Section III presents our statistical background modeling process. Experimental results appear in Section IV, where both quantitative and qualitative analyses are described. We demonstrate the validity of the method using several sequences with different types of backgrounds. The proposed method is compared with other edge-based methods. We present concluding remarks in Section V.

II. Edge-Based Background Modeling

Recent research [14], [18], [23], [24], [26] has focused on the use of edges as consistent features, due to their robustness against illumination and noise. Furthermore, the use of edges facilitates the pose detection and object identification problem, as they provide structural information of the objects. Nevertheless, edges have some problems: shape and position changes. If we consider a scene with moving background, such as waving trees (Fig. 2), we can appreciate, from the accumulation of several consecutive frames, the changes in the edges' position and shape [see Fig. 2(c)]. Moreover, each edge has different variations, e.g., in Fig. 2(c) the edges of the buildings exhibit little variation in comparison to the edges of the trees, and even edges from similar objects reveal diverse behavior, e.g., the tree in the middle manifests more motion than the trees in the upper corner of the picture. Therefore, a simple pixel-wise matching approach would fail most of the time, as shown in Fig. 3. Another problem arises when the moving object edges are close to the background edges, a case in which the moving edges are confused with the background. In addition, recent methods [18], [24], [26] verify all the edges based on fixed thresholds, and they utilize chamfer matching [27], which encumbers the task of distinguishing them from each other.

In contrast, our method copes with these problems well, thanks to edge-segment representation and the statistical approach. Consequently, to avoid edge shape and position changes, we learn the behavior of these edge-segments through a set of kernel-based statistical distributions. In addition, a score is assigned to each segment to verify whether it

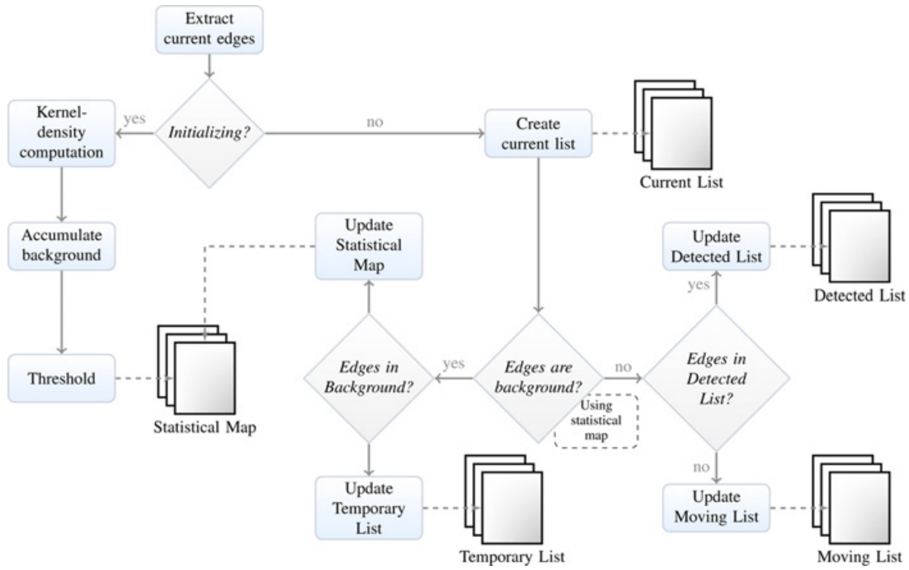


Fig. 1. Proposed method extracts the edges and builds the statistical map; then from the incoming edges (current list) the method updates the background model (i.e., statistical map, temporary list, and detected list), and extracts the foreground (moving list).

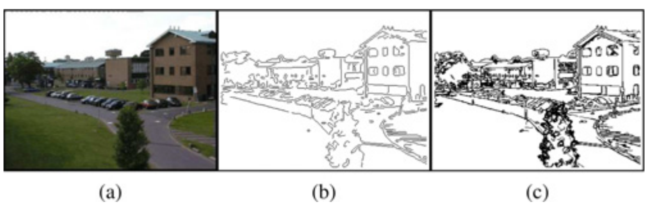


Fig. 2. Problems with edges: changes in shape and position. (a) Sample scene with waving trees. (b) Its edges. (c) We accumulate the edges of 100 frames, and that accumulation reveals the variation of the edges. The building's edges have small variations, while the trees' edges present high variations.

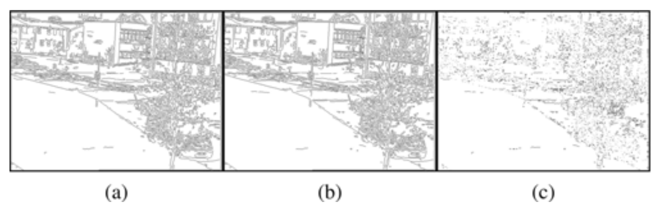


Fig. 3. Problems with pixel-wise edge subtraction. (a), (b) Edge maps of two consecutive frames. (c) Pixel-wise difference of the edge maps in (a) and (b) reveals that a simple approach is not suitable for background subtraction.

belongs to the background or to the foreground. Note that the use of segments introduces more information, which makes the score generation more reliable than the pixel-wise score approach for edges. And so we apply an adaptive threshold to each segment, which sets a different threshold automatically according to each segment's observed variations. Moreover, the segment's variations allow us to determine the position to split foreground edges that merge with the background. Thus, we increase the detection accuracy of our method.

Moreover, the initialization of the model is one problem that is ignored in many edge-based methods [14], [18], [23], [24], [26]. In a real-world scenario, objects appear and disappear from the scene constantly; thus, our approach can produce a background model in the presence of moving objects, without the restriction of clear backgrounds. This feature makes it suitable for real-world environments.

Nevertheless, one drawback of the edge-based methods is the frequent need to extract the foreground region from the detected edges. However, previous works have explored this issue. For example, one method uses a watershed algorithm to extract foreground regions using edges as seeds [24]. Another scheme uses a component connection algorithm, using the horizontal and vertical regions defined by the edges to segment the object region [18]. Furthermore, we can cast the segmentation-from-edges problem as an energy minimization problem and use a graph cut algorithm [28] using the edges as seeds. Thus, we consider the segmentation problem to be beyond the scope of this paper.

III. Proposed Method

Our statistical model attempts to predict the edge's behavior, i.e., its shape and position changes, and encode it into a set of parameters. Therefore, when a new edge comes to the scene, we test it against the previous observed edge's behavior, and determine whether it fits the previous edges or is a new one. Moreover, we use an adaptive comparison framework for the edges (i.e., the threshold for the matching score and the search window, which we infer from the distribution's characteristics) that increases the accuracy of the detection. In addition, the statistical model allows us to suppress the contribution of the moving objects from the training frames to generate the background model, leaving only the background edges contribution in the model. Fig. 4 shows an abstraction of the proposed method.

In Sections III-A to III-C, we present the background modeling initial training phase, and in Sections III-D and III-E, we detail the model update. Finally, we explain the foreground detection in Section III-F. This flow is similar to that shown in Fig. 1.

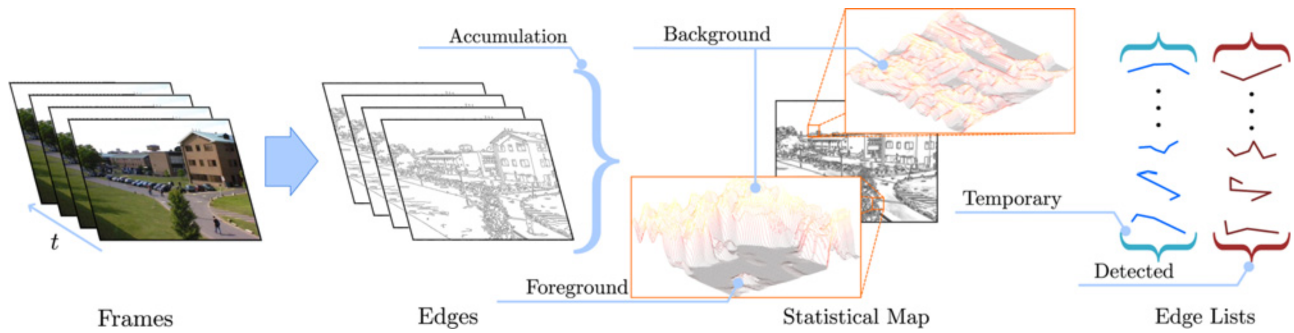


Fig. 4. Background model consists of statistical map and temporary and detected edge-segment lists. From the stream of frames, we extract the edge-segments, and accumulate them to build the statistical map. Then, we maintain two levels of background edges, as they present a static nature and they are incorporated into higher layers of the background model.

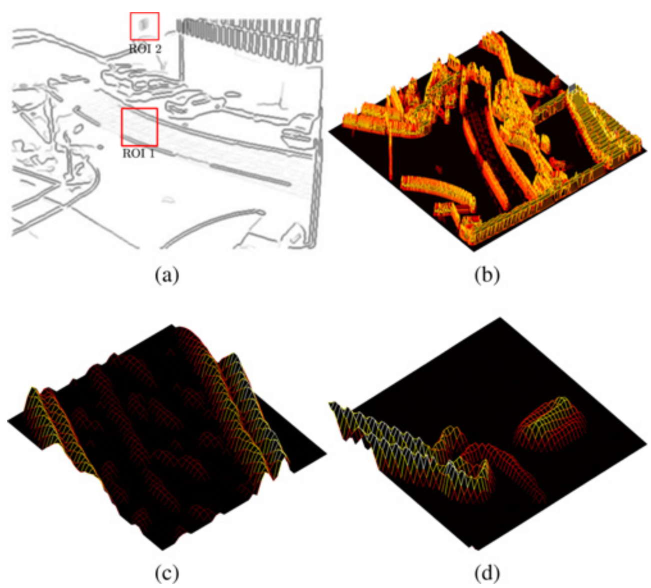


Fig. 5. Different characteristics of the statistical map distributions. (a) Statistical map of a scene with moving objects in it. (b) Its 3-D representation. (c) 3-D distributions of the ROI 1 in (a) that show the temporal variation: moving objects with small frequency, and background with high frequency. (d) Distributions of the ROI 2 in (a) that show the spatial variation: edges that move a lot with a wider distribution, and edges that do not move a lot with a narrow distribution.

A. Edge Segment

We model the edges, extracted by a Canny edge detector [29], as segments, that is, instead of treating the edges pixel-wise we treat each of them as a whole. We define an edge-segment E for a given frame as the set of pixels

$$E = \{p \mid p \in \mathcal{N}(q) \wedge q \in E\} \quad (1)$$

where p and q are the (x, y) positions of the edge-pixels, and $\mathcal{N}(q)$ is the 8-neighborhood of the pixel q. In other words, the set is defined as those edge-pixel's positions that are connected to (are in the neighborhood of) each other and to a seed edge-pixel, which we follow to construct the edge-segment. Thus, we scan the binary edge image until we find an edge-pixel. Then, we recursively follow the connected pixels that are the edge, add them to the segment, and mark them as visited—visited pixels are skipped and not further considered. We continue in this way, until there are no more edge or unvisited pixels. Then, we continue our scan process, searching for a new edge seed. In addition, we represent this set in a structure that allows us to traverse all the connected edge pixels in a fast manner, and we access the set from the ending points.

A stable (background) edge-segment will appear in nearby positions in consecutive frames (most of the time it will oscillate). Although it may have changes in its shape, over time it will define an area of movement. Thus, we model such behavior by constructing a statistical distribution of its frequency over the defined area (see Section III-B). Hence, to identify the same edge-segment through different frames, we use the distribution's area to differentiate among segments. Therefore, the segments that (in their majority) lie in the same area are considered to be part of the same edge. From now on, when discussing our method, we will use the terms segment and edge-segment interchangeably.

B. Statistical Modeling

To estimate the edges' behavior, first we extract the set of edge-segments, $\mathbb{E}^t$, from each frame t. Then, we create the statistical map SM that is the set of all the distributions, from a set of frames, by using

$$\mathcal{D}_{E^t} = \sum_{e \in E^t} K(e) \quad (2)$$

$$\mathrm{SM}(E) = \frac{1}{N} \sum_{t=t_0}^{t_f} \mathcal{D}_{E^t} \quad (3)$$

where E^t is an edge-segment from the set of segments $\mathbb{E}^t$ at frame t, $\mathcal{D}_{E^t}$ is the distribution of the segment E^t, e is a component of the segment E^t, $K(\cdot)$ is a kernel function estimator, and the number of frames N ranges from the initial frame t_0 to the final frame t_f. In our experiments, we use 200 frames to build the model ($N = t_f - t_0 = 200$). Note that E is a general segment for all the (temporal) segments that lie in the same region.

Consequently, we create a weighted distribution in each edge-segment by placing a kernel function estimator $K(\cdot)$ over each edge-pixel. If we observe the edges for a long enough period of time, its true behavior will appear. However, due to the small number of frames used for training, we use a Gaussian kernel, chosen to model the statistical behavior of

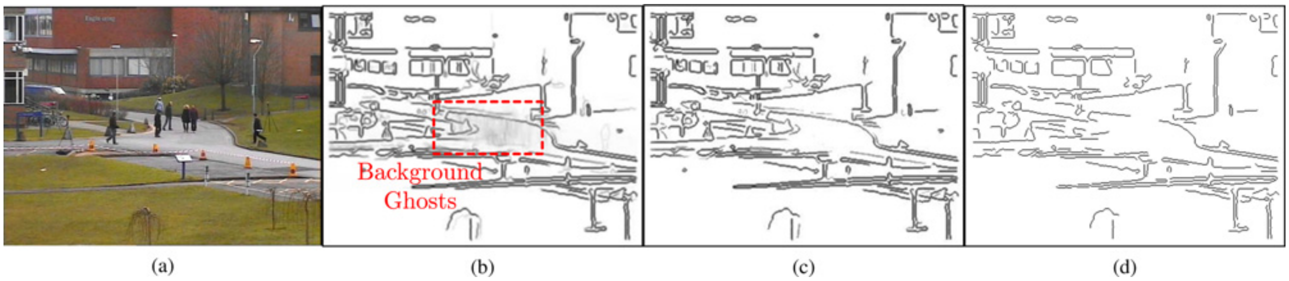


Fig. 6. Background creation process. (a) Sample frame from a scene. (b) Its statistical map with background ghosts. (c) Adaptively thresholded statistical map from (b), without background ghosts. (d) Extracted background edges from (c).

the edges. Moreover, it helps compensate for the small amount of data, since the edges are sparse and change so much. Thus, the kernel is given by

$$K(e) = \frac{1}{\sqrt{2\pi}h} \sum_{p \in \mathcal{N}(e)} \exp\left\{-\frac{(p-e)^2}{2h^2}\right\} \tag{4}$$

where h is the width of the kernel, and p is a pixel in the neighborhood $\mathcal{N}(e)$ of the pixel e. When the edges present no variation, the kernel will provide variation equal to its width. Consequently, we set the neighborhood size equal to the width of the kernel. Moreover, we approximate the kernel with a convolution mask to increase the performance. In our experiments, we noted that the use of a larger kernel width produced wider distributions, which tended to incorporate the moving objects into the background. However, the choice of whether to choose a wide or narrow kernel depends on the amount of data (number of frames N) and the dynamism of the background (i.e., the scene to be modeled). In general, we found that the use of 3×3 and 5×5 sizes produced good results—in our experiments, we used a kernel of 3×3.

C. Adaptive Threshold

The accumulation of edges among frames reveals the temporal (frequency) and spatial variation (motion) of the edges. Thanks to this knowledge, we can remove ghosts (temporal variation) and perform an adaptive threshold over the segments (spatial variation).

1) *Frequency Threshold:* As Fig. 5 shows, the incorporation of temporal information creates ghosts in the background model. But, the background and the foreground each have a distinctive frequency. For instance, the moving objects (which create edges that appear in different positions) create small peaks in the distribution, while the background edges have high values in the distribution (because they appear repeatedly in the same position), as shown in Fig. 5(c).

Therefore, we overcome the ghost problem by removing the distributions based on the segment's frequency (i.e., its number of appearances in a set of frames). Let $\mathcal{D}_E \in \mathrm{SM}$ be the distribution for the edge E. Then, we update the SM by removing the distributions such that

$$\mathrm{SM} = \mathrm{SM} - \{\mathcal{D}_E \mid \mathrm{avg}(\mathcal{D}_E) < T_f\} \tag{5}$$

where $\mathrm{avg}(\cdot)$ is the average operation over the distribution values, and T_f is the frequency threshold that differentiates background from foreground distributions. Thus, the updated statistical map, SM, contains only those distributions with an average frequency greater than a certain threshold. To set the threshold for the distributions of the moving objects [as Fig. 6(b) shows], we assume that the moving objects will have a minimum average speed v, in pixels per frame. Moreover, we know that the inverse of the speed, $1/v$, gives us the number of frames an object stays in the same place. Hence, we propose a threshold

$$T_f = \max\left[K(\cdot)\right]\frac{1}{v} \tag{6}$$

where $\max[K(\cdot)]$ is the maximum spatial value from the kernel function [i.e., the maximum value produced by (4)], and v is the minimum average speed of the moving objects. For our experiments, we assume that the objects will present a minimum average speed of $v = k_v/N$, where N is the number of frames used for learning, and $k_v = 2$. That is, an object will not be stopped in the same place in more than half of the total frames used for learning the model. Fig. 6(a) shows a sample frame of a sequence with moving people in it, and Fig. 6(b) shows the statistical map of that sequence, which presents ghosts of the moving people as other statistical approaches do. However, Fig. 6(c) shows how our proposed threshold removed the ghosts from the accumulated map, while preserving the background.

2) *Motion Threshold:* Fig. 5(d) illustrates the spatial variation for which edges with a lot of movement create wide spread distributions, while edges with little movement create sharp distributions. For example, edges with high movement can come from bushes or trees, while static edges come from buildings or roads. Therefore, the creation of *ad hoc* distributions for each edge allows us to define accurate search regions for the edge-matching process and an adaptive threshold for each edge, according to its characteristics, which improves the accuracy of our method.

In the matching process, we use the statistical map to compute the probability of an incoming segment being background. Hence, we need to adaptively set the comparison parameters for each segment. Moreover, when a moving object's segment merges with the background, we use the area of the distribution to split the moving segments (see Section III-F for more details). Therefore, we extract from each distribution the mean and the standard deviation to use them in this process.

We extract the peak (maximum spatio-temporal values) of each distribution in the statistical map; these peaks are the stable edges of the background [as Fig. 6(d) shows], obtained by

$$\mathcal{D}_E^{\max} = \{e \mid e \in \mathcal{D}_E \wedge e \in \max_{\mathcal{N}(e)}(e)\} \tag{7}$$

where $\mathcal{D}_E^{\max}$ is the set of the maximum components, e, of the distribution, and $\max_{\mathcal{N}(e)}(e)$ is the set of maximum values defined in the neighborhood of e, $\mathcal{N}(e)$. Then for each distribution in the statistical map, we compute its parameters: the mean, $\bar{\mu}_E$, and the variance, σ_E^2. Consequently, we calculate the mean of each segment's distribution, $\bar{\mu}_E$, using the maximum values of the distribution, as

$$\bar{\mu}_E = \frac{1}{M} \sum_{e \in \mathcal{D}_E^{\max}} e \tag{8}$$

where M is the number of elements of $\mathcal{D}_E^{\max}$, and e are the values of the peaks of the distribution. Then, for each frame, the sample mean of the incoming segment for that distribution, μ^t, is calculated by averaging the values indicated by that segment in the corresponding distribution. These sample means, μ_E^t, are used with the distribution mean, $\bar{\mu}_E$, to calculate the variance, σ_E^2, over N frames using

$$\sigma_E^2 = \frac{1}{N} \sum_{t=1}^{N} (\mu_E^t - \bar{\mu}_E)^2. \tag{9}$$

D. Statistical Map Update

Once the distributions are created, we update them by incorporating the detected segments, E^{t+1}, information using

$$\mathcal{D}'_{E^{t+1}} = \alpha \mathcal{D}_{E^t} + (1 - \alpha)\mathcal{D}_{E^{t+1}} \tag{10}$$

where $\mathcal{D}'_{E^{t+1}}$ is the final distribution for the segment E^{t+1}, $\mathcal{D}$ is the distribution for a segment defined in (2), and α is the mixture constant. Moreover, the new distribution $\mathcal{D}'$ replaces the old distribution in the statistical map. In our experiments, we set $\alpha = 1/(N-1)$ to incorporate the same weight to each segment in the distribution since its inception. Furthermore, we add distributions based on the learned edges (see Section III-E for more details).

E. Edge-Segment Lists: Temporary and Detected

The dynamic background tends to change constantly. Therefore, we need the background model to adapt after it is learned. To avoid rebuilding the entire background each time something in it changes, we incorporate two edge-segment lists. Hence, our background model comprises the statistical map, SM, that models the long term characteristics of the background, the temporary list, $\mathcal{L}_t$, that models the recently changing background, and the detected list, $\mathcal{L}_d$, that models the just-detected moving segments. The objective for the detected list is to act as a layer that learns the moving segments that are becoming background. When these segments have been detected in the same position for several frames, we consider them candidates for background, and we move them to the temporary list. Thus, the temporary list acts as another learning step, which avoids the incorporation of garbage information (flickering and moving edges) to the model.

Furthermore, we maintain a record of the frequency of each segment, E_f, in its structure. We update the frequency of the segments at each frame. If the segment is considered background (i.e., the segment matches the other edges in the corresponding list), its frequency increases; otherwise, it decreases. That is, the segment frequency, E_f, is updated by

$$E_f = \begin{cases} E_f + 1, & \text{if matched} \\ E_f - 1, & \text{otherwise.} \end{cases} \tag{11}$$

Moreover, when the frequency E_f is below or equal to zero, we drop the segment from its corresponding list, i.e., we forget the segment. Consequently, we update the lists through

$$\mathcal{L}_x = \mathcal{L}_x - \{E \mid E_f \leq 0\} \tag{12}$$

where $\mathcal{L}_x$ represents a list such that $x \in \{t, d\}$. Furthermore, given that these segments are still only candidates to become background, they are not consistent with the background behavior. Also, in case they become foreground again, we need to maintain them only for a certain period of time, before we forget them. To achieve this behavior, we set an upper bound to the frequency accumulation.

For the detected list, the upper bound, T_t, determines when the detected segment becomes a temporary segment. Therefore, when the frequency of a segment reaches the upper bound, T_t, we move the segment to the temporary list. And in case of the temporary list, the upper bound, T_b, determines when the temporary segment becomes background. Hence, when the frequency of a segment reaches the upper bound, T_b, we compute the average distribution of the segment and incorporate it to the statistical map. We create the distribution for the new background segment using (2), and then we add it to the statistical map. Consequently, we update the model by

$$\mathcal{L}'_d = \mathcal{L}_d - \{E \mid E \in \mathcal{L}_d \wedge E > T_t\} \tag{13}$$

$$\mathcal{L}'_t = \mathcal{L}_t \cup \{E \mid E \in \mathcal{L}_d \wedge E > T_t\} - \{E \mid E \in \mathcal{L}_t \wedge E_f > T_b\}, \tag{14}$$

$$\mathrm{SM}' = \mathrm{SM} \cup \{\mathcal{D}_E \mid E \in \mathcal{L}_t \wedge E_f > T_b\} \tag{15}$$

where the updated background model is $\{\mathcal{L}'_d, \mathcal{L}'_t, \mathrm{SM}'\}$. In other words, the learning threshold, T_t, represents the number of frames that a moving segment should appear in, before considering it background. And the background threshold, T_b, represents the number of frames that an edge should continue appearing as background before we incorporate it into the statistical map—in our experiments we set $T_t = N/10$ and $T_b = N/2$. Having the two additional levels in the background model makes the background representation more robust against flickering edges, as well as for moving objects that could stop for short periods of time.

F. Foreground Detection

To separate the foreground from the background, we use statistical and flexible matching on our three-layer background model. In our first layer, the statistical map allows us to incorporate the natural variations of edges (through adaptive thresholding) into the detection, which increases the accuracy of our method. We also use that statistical map to split the foreground edges when they merge with the background. Consequently, we split the edge based on the corresponding segment distribution. If a large part of the edge lies outside of the region defined by the corresponding segment distribution, we will split it into new edges using the intersection of the edge and the perimeter of the distribution as cutting points. The second layer comprises the segments of the temporary list, which holds the edges that are considered background due to their lack of movement in the sequence. This list is also updated to maintain the current background state. Finally, the third layer, comprised by the detected list, which acts as a learning list, maintains the moving edges detected over several frames.

In order to detect the foreground, we extract the current segments from the current frame (current list). Consequently, the background is eliminated by matching the current segments against the background model. First, the current segments are verified against the statistical map. The background segments are separated from the current frame segments by computing the statistical distance, SD, of each extracted segment, E, through

$$\mathrm{SD}(E) = \frac{1}{k} \sum_{e \in E} \mathcal{D}^E(e) \tag{16}$$

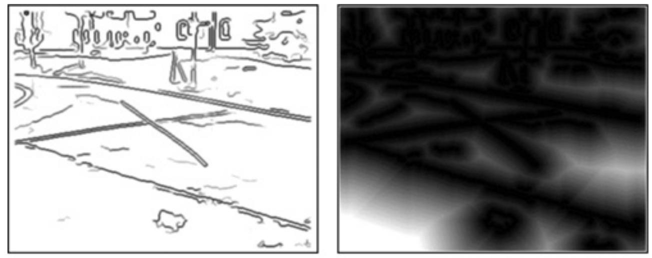
where the segment E has k pixels e, and $\mathcal{D}^E(e)$ is the value in the statistical map, SM, for the corresponding distribution of the edge, E, at the pixel, e's, location. If the statistical distance, SD, is close to the mean of the corresponding distribution, $\bar{\mu}_E - \sigma_E \leq \mathrm{SD}(E) \leq \bar{\mu}_E + \sigma_E$, then it is considered a background segment; otherwise, the segment, E, is tested in the next steps.

Then, the remaining edges are matched against the temporary and detected lists. To compute the degree of matching of the segments we create a chamfer 3/4 distance map [27] (CD) for each list. The segment distance in the distance map of each list is evaluated by normalized root mean square (RMS)

$$\mathrm{RMS}(E) = \frac{1}{3} \sqrt{\frac{1}{k} \sum_{e \in E} \mathrm{CD}(e)^2} \tag{17}$$

where k is the length of segment E, and $\mathrm{CD}(e)$ is the value at e's pixel position in the distance map. If the distance is within a threshold, the edge is matched to an edge from the list (we use a threshold of 2.5, i.e., the segment is on average two and a half pixels away from the original segment). First, we test the segments against the temporary list; if a match occurs the segment is considered background. Otherwise, it is tested against the detected list; if we match them with some edge in this list they are marked as background; otherwise, they are foreground. Note that the edges of these lists are potential background segments, and they do not have the complex movement that is encoded in the statistical map, as they appear for only a few consecutive frames. Therefore, a simple verification process is enough.

Previous works [24] matched different edges using a fixed threshold. However, this matching scheme does not consider the variation in the edges, which leads to inaccurate results.



(a) (b)

0	0	0	39	0	0	0
0	0	64	65	60	0	0
0	0	45	109	70	43	0
0	0	40	78	101	37	0
0	0	55	95	105	36	0
0	0	93	125	93	30	0
0	35	106	135	78	41	0
0	0	90	120	60	0	0

(c)

15	8	7	6	7	10	11
10	7	4	3	4	7	8
9	6	3	0	3	4	7
10	7	4	3	0	3	6
10	7	4	3	0	3	6
9	6	3	0	3	4	7
9	6	3	0	3	6	8
9	6	3	0	3	6	9

(d)

Fig. 7. Difference when using (a) statistical map and (b) chamfer distance map. (c), (d) Statistical and chamfer distance values from (a) and (b), with two superimposed edges (a variation or the true edge in orange, and a moving edge in red). Our adaptive threshold produces accurate matching, while the fixed threshold misses the edges.

The common score for fixed-threshold methods is the chamfer distance map. However, fixed-threshold methods have several limitations in distinguishing edges with different variations. For instance, let us consider a variation (in orange) of the original edge (in gray) shown in Fig. 7, and let us compare the matching scheme using a fixed-threshold (chamfer distance map) and a variable threshold (statistical map). In the chamfer distance map [Fig. 7(d)], its score is 1.37—through (17)—which is small enough, so we consider the variation to be the original edge. Meanwhile, the statistical map [Fig. 7(c)] gives it a score of 62.5—through (16). And given the true edge distribution parameters of 113±60, we can consider the orange edge to be the same original edge, too. However, if we consider a moving edge (in red) close to the original one, the chamfer map gives it a score of 2.59, so if using a fixed threshold it can easily be mistaken as being the same edge, while the statistical map gives it a score of 20.8, indicating it is not the same edge. This comparison shows that using the statistical map has advantages over fixed threshold methods, achieved by using adaptive thresholds that consider the edge's behavior. In contrast, the fixed threshold methods assume one type of edge variation over the entire image to set the thresholds, and this assumption reduces the accuracy of the method. Thus, the proposed method that mines the observed edges' behaviors and sets the thresholds accordingly, is able to produce superior accuracy.

IV. Results and Analysis

We evaluated the proposed method on video sequences from performance evaluation of tracking and surveillance (PETS).

Specifically, we used the PETS 2001 [30] and PETS 2009 [31] data sets. The test sequences from the PETS 2001 data set were captured at a campus parking lot by two cameras from different viewpoints. Meanwhile, the PETS 2009 data were captured at campus intersections from eight viewpoints. All the images in these two data sets present dynamic background and moving objects, as well as illumination changes. The first one presents moving objects with various sizes and speeds. The other presents crowds with people moving at various speeds. The PETS 2009 data set has more challenging sequences for modeling backgrounds because it lacks ideal frames [32]. We performed a two-fold evaluation: 1) a background-model robustness evaluation to test the modeling capabilities of the proposed method in the presence of moving objects, and 2) a detection evaluation to test the accuracy of the proposed method.

We compared our results with those obtained using other edge-based methods in order to provide a framework for meaningfully evaluating the propsed method's robustness and performance. We test all these methods using the PETS 2001 Data Set 3 (DS3) and Data Set 4 (DS4), and using the PETS 2009 Views 1, 5, and 6. We built the ground truth edges of these sequences by hand, using a previous ground truth [33] as a reference.

A. Evaluation

To evaluate the detected edges (either in the background model, in which detected edges stand for background model edges, or foreground detection, in which it stands for moving edges), we classify them into four categories: true positives (TP) are edges that were correctly detected, false positives (FP) are edges that were wrongly detected, true negatives (TN) are the image background pixels that were correctly detected as background, and false negatives (FN) are background pixels that were detected as edges. From these values we calculate precision, recall, true positive rate (TPR), and false positive rate (FPR), defined by

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{18}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{19}$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{20}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP}+\text{TN}}. \tag{21}$$

Precision is a measure of the percentage of moving edges that are truly moving. Recall is a measure of the percentage of moving edges detected. FPR is the percentage of background that is misclassified as moving edges, and TPR is the same as recall. Using TPR and FPR we create the ROC curve that shows the variation of the correctly classified detected edges with respect to the incorrectly classified background. Using precision and recall we create the PR curve that shows the variation of the correctly detected edges with respect to the detected edges. We present the comparative results of the methods using both ROC and PR curves.

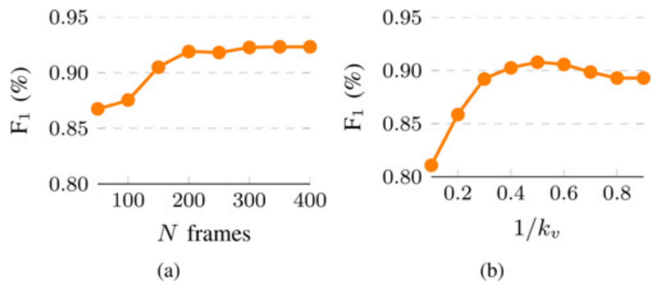


Fig. 8. Average F_1 measure of the proposed method while varying (a) number of frames (N) for training the background model, and (b) constant of the velocity of the object ($1/k_v$) for the frequency threshold (T_f).

B. Parameters Analysis

In this section, we present an analysis of the impact of the parameters of our proposed method; we test the parameters with the PETS 2001 database, DS4 sequence. The three main thresholds we use are: 1) number of frames for learning; 2) accumulation threshold, T_f; and 3) the thresholds of the lists: the temporal, T_t, and background, T_b, thresholds.

First, we need to establish the minimum number of frames used for training, as the amount of frames produces subtle changes in the general accuracy of the method. As shown in Fig. 8(a), the F_1 measure, given by

$$\mathrm{F}_1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{22}$$

of the method increases with the number of frames used to learn the background model. As expected, a higher number of frames produce a more stable background model. However, this rate of increase slows down after 200 frames. Although the accuracy increases for a larger amount of frames, the increment is not significant in comparison to the accuracy at 200 frames. Hence, in our experiments we used the smallest number of frames that yields a reasonable result: 200 frames.

Then, we analyzed the frequency threshold that determines which distributions are eliminated from the statistical map. Thus, when this threshold is too small we may leave spurious distributions in the background model, and when it is too high we may eliminate background distributions. Consequently, we need to find a balance between removing the moving distributions and maintaining the background ones. Note that (6) depends on the velocity of the moving objects, the number of frames, and the maximum value of the kernel. However, for a given training sequence, the number of frames and the maximum value of the kernel are constants, and we may re-write that equation as a function of the velocity alone. Thereby, we plot the F_1 measure of our proposed method while varying the constant k_v in the velocity. Given that increments in the velocity will increase the threshold T_f, higher values will re-move all the moving distributions, which eventually will boost the accuracy of the method. However, at the same time, when the threshold increases, we eliminate background distributions as well. Therefore, the overall F_1 measure decreases as we detect background edges as foreground, in addition to the miss-detected edges. Thus, we need to find the threshold value that maximizes the measure. As seen in Fig. 8(b), the best average

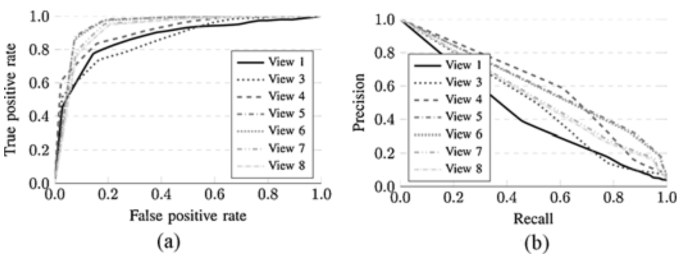


Fig. 9. Evaluation of the background model, using seven experiments on the PETS 2009 data set. (a) ROC of the background model. (b) Precision–recall curve of the background model.

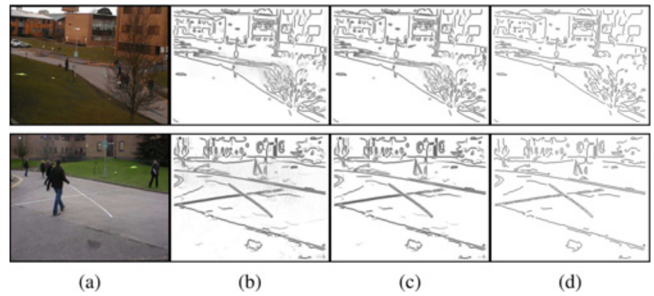


Fig. 10. First and second rows are the sequences Views 3 and 8 of the PETS 2009 data set. (a) Frame sample from the sequence. (b) Statistical map of (a). (c) Threshold statistical map of (b). (d) Background edge map of (a).

results are obtained when $k_v = 2$, as we remove most of the moving objects while maintaining the background. These results also support our assumption of having moving objects that will stop for at least half of the training sequence.

Finally, the thresholds of the list depend on the target application, since they represent learning rates for their respective lists. For example, if we are modeling a street with pedestrians that walk slowly and that stop for long periods of time before moving, we may want to use high values in these thresholds to slow down the incorporation of these moving objects (pedestrians) into the model. On the other hand, if we are modeling a highway with cars moving on it, we may want to use a faster learning rate, since the overall speed of the scene is higher. Although the two thresholds act in the same way, they indicate different stages in the learning process. The temporary threshold, T_t, indicates when to incorporate the dynamic changes in the scene into the background model. For example, edges from a moving car that has just stopped in a parking lot should be considered as temporary background edges after T_t frames. However, the system is sensitive to the selection of T_t threshold. Higher values of T_t cause the method to take longer to remove temporary background edges from the scene, whereas very low values of T_t do so much faster and may even eliminate edges from slowly moving objects. Our selection for that threshold $T_t = N/10$ means that edges from a moving object have to stay motionless for 20 frames to become temporary background. On the other hand, the background threshold, T_b, is the minimum number of frames for an edge-segment to appear in the same spatial position during the period N to be considered background. Consequently, edges that exceed this threshold are considered true background edges, and thus will update the statistical background map. Functionally, the background threshold, T_b, is similar to the frequency threshold, T_f, used for detecting true background edges from the training frames. Thus, we can assign a value within the range $N/1.5 \geq T_b \geq N/2.5$ [note that Fig. 8(b) shows the inverse of the velocity constant, and by computing T_f we can determine this range] that will not have any sensitivity for updating the statistical background map.

C. Background Model Evaluation

To evaluate the robustness of our background model, we perform seven experiments using the PETS 2009 (S2-L1 Time 12–34) data set, because it is challenging due to its lack of ideal backgrounds. In addition, we create a background ground truth for seven sequences in the PETS 2009, using the S0 sequence (ideal background), to measure the accuracy of the generated model. Moreover, we divide each sequence into seven parts to build a model for each one. The experimental sequences have three challenging characteristics: 1) people walking around who stop and start moving again; 2) standing people; and 3) trees. Moving people and trees have the challenge that they may be incorporated into the background model due to their slow movement, which in the detection produces ghosts. In addition, trees and foliage are difficult to model as background because they move randomly. Furthermore, a difficult task in background model creation occurs when the objects stop for a long period, and then start moving again, because they may be considered background rather than moving objects. Our algorithm is robust enough to overcome these problems, because it creates an accurate model by using edge statistics, the adaptive threshold mechanism, and several learning layers.

To evaluate the background model generation performance we create an ROC curve for the seven experiments. Fig. 9(a) shows the performance: all curves present an average area under the curve (AUC) of at least 91%. Moreover, views 3, 7, and 8 have AUCs of 91%, 93% and 92%. These results demonstrate the robustness of our method for dynamic backgrounds. The precision-and-recall curve in Fig. 9(b) also depicts the robustness of the method. Fig. 10 shows two sample sequences from the experiments, in which the background model produces stable background edges and avoids the ghost effect. For example, in the threshold accumulated map [Fig. 9(c)] no ghosts are present in the background model. Therefore, our method provides better performance than other modeling techniques that are not able to avoid ghost effects: our method avoids ghosts and extracts stable background edges [Fig. 10(d)].

D. Detection Evaluation

We evaluated the detection capabilities of our method against four other methods: those by Dailey *et al.* [14], Kim and Hwang [18], Hossain *et al.* [24], and Jain *et al.* [23]. We tested the algorithms using the PETS 2001 data set, Data Set 3 (Testing–Camera 1) and Data Set 4 (Training–Camera 1), and using the PETS 2009 data set, Views 1, 5 and 6. We chose these sequences for their challenging environments. All sequences present illumination changes and dynamic back-

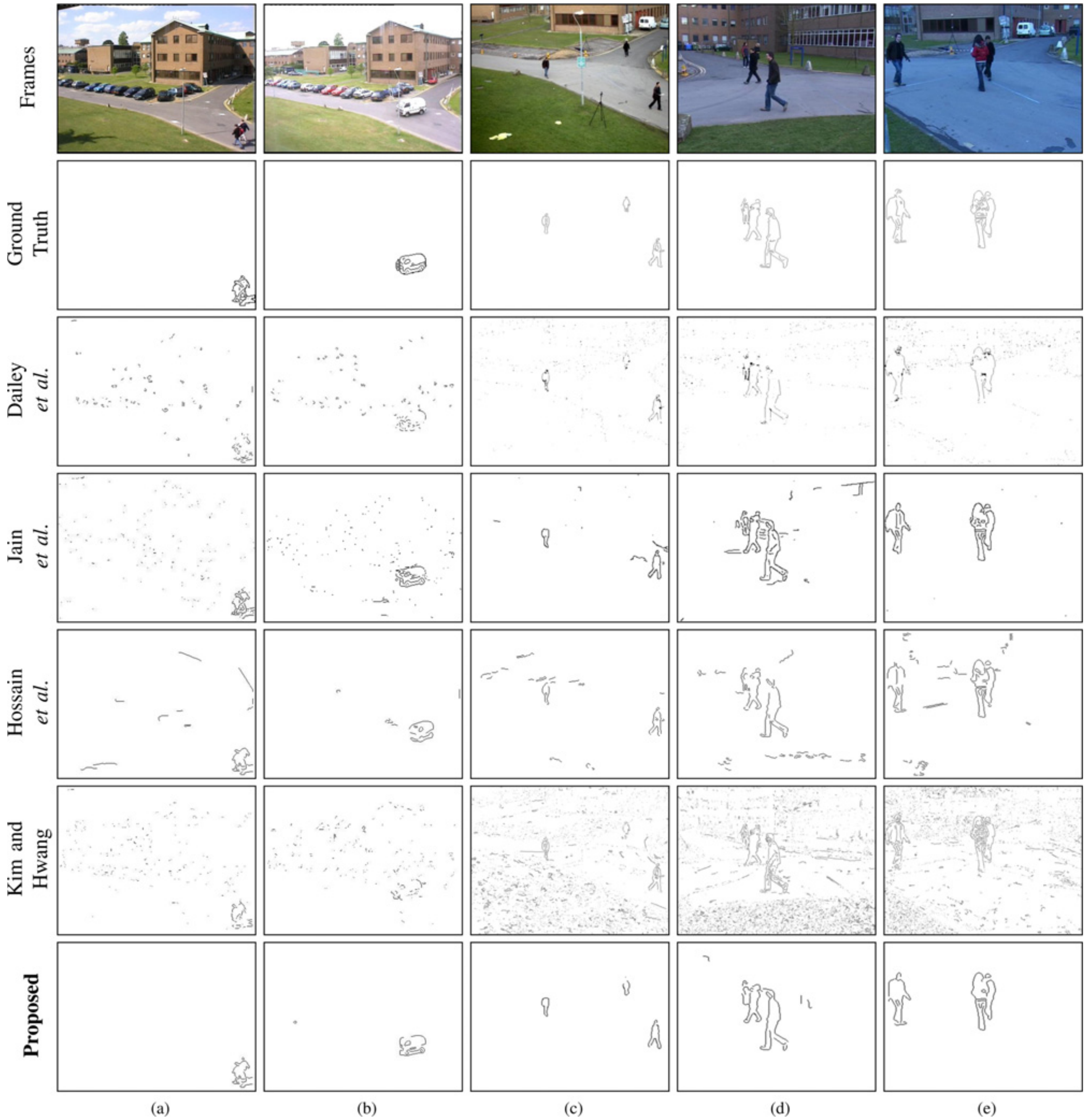


Fig. 11. Results from the methods evaluated. Each row represents one method. From the PETS 2001, (a) frame 1371 from sequence DS3, and (b) frame 1086 from sequence DS4. From the PETS 2009, (c) frame 401 from sequence View 1, (d) frame 268 from sequence View 5, and (e) frame 462 from sequence View 6.

grounds. For example, in some sequences people are walking around in the scene, and then stand in place several frames before restarting their movement. Data Set 3 (DS3) has clouds that move slowly in the scene and can be detected by mistake and creating illumination changes in the scene. Data Set 4 (DS4) has cars and people together in the scene. Moreover, the sequences from PETS 2009 have people moving in complex ways in the scene, as well as a dynamic background, which make these sequences very challenging.

In DS3, as illustrated in Fig. 11(a), the clouds present a challenge partially due to their slow motion and changes in shape. Jain *et al*.'s and Hossain *et al*.'s methods gave a false positive on the clouds edges, while the proposed algorithm overcame this problem due to its statistical background model and update and verification mechanisms. This sequence is difficult also because changes in the clouds produce changes in the illumination. Dailey *et al*.'s, Jain *et al*.'s, and Kim and Hwang's methods produced noisy edges because they miss-classified edges due to intensity and edge shape changes. The proposed method can overcome this problem because: 1) it incorporates the edge movement during background modeling by learning edge behavior, and 2) it performs *ad*

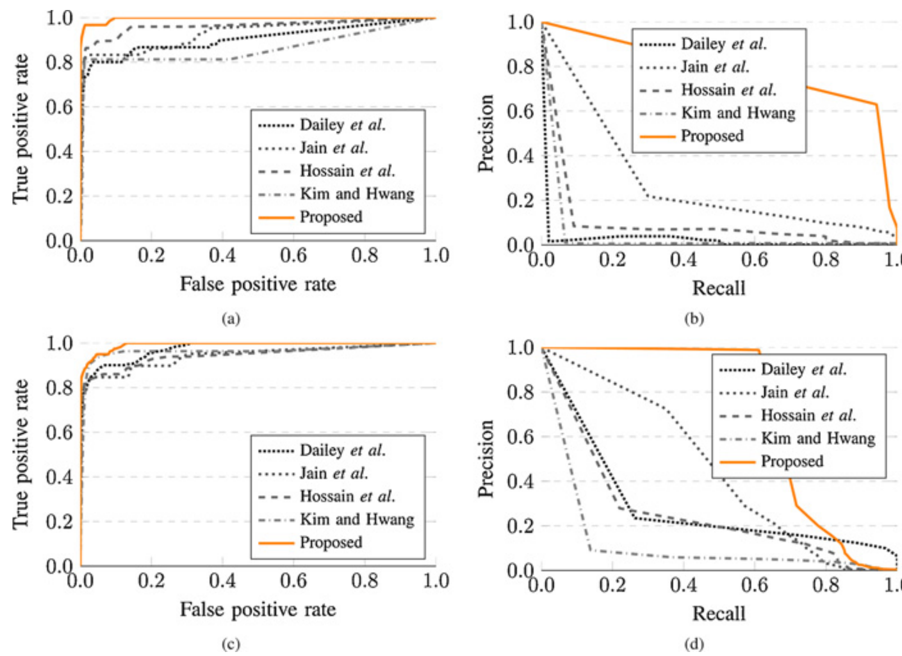


Fig. 12. Evaluation of the detection algorithms using PETS 2001 data set. (Left column) ROC and (right column) Precision–recall curves for the algorithms in (a), (b) DS3, and (c), (d) DS4.

hoc verification for each edge. Fig. 11(b) presents the intense illumination changes caused by the sun, visible in the car and building windows. All the other methods have problems dealing with the reflections in the windows, note that the same type of behavior occurs for other reflective surfaces, but our method is robust in these conditions too. When the reflection edges appear, our method builds distributions around them, and then detects them in posterior frames, even if they present shape and position changes.

Over the entire detection, we can appreciate how Dailey *et al.*'s and Kim and Hwang's methods produce scatter edges and spurious pixels as output, because they are edge-pixel-based methods. In contrast, Jain *et al.*'s, Hossain *et al.*'s, and the proposed methods produce solid edges that are better able to segment the moving objects. Nevertheless, due to their weak background model, they are not able to suppress all the false positives from the output. In addition, in the PETS 2009 sequences [Fig. 11(c)–(e)], the proposed method outperforms the other methods in modeling the dynamic background and segmenting the people walking around in the scene. Note that the background model absorbs small edges from the foreground, as those small edges are confused with background. However, the structural edges are maintained through the detection.

Moreover, we did a quantitative evaluation of the five methods, measuring the distance of the detected-edges to the ground-truth-edges. Fig. 12(a) shows that the proposed method has 95% AUC in the DS3. In addition, the proposed method is 9%, 7%, 4%, and 14% better than Dailey *et al.*'s, Jain *et al.*'s, Hossain *et al.*'s, and Kim and Hwang's methods, respectively. Furthermore, Fig. 12(b) shows the precision-and-recall curves of the five methods, which indicate the relationships among detected edges and expected edges. These curves show that the proposed method outperforms the comparison methods by 80% on average. Fig. 12(c) and (d) shows the ROC and precision-and-recall curves of the methods tested using DS4. With this sequence, all methods performed better, with an average ROC AUC of 96%. The proposed method outperforms them by an average of 3%. Nevertheless, in the precision-and-recall curve, we noticed that the older methods left a lot room for improvement. Here, the proposed method shows an average improvement of 60% over the other methods. Although when we release the constraints in the detection phase in order to detect more foreground, several noise edges are detected too, which decreases the precision of our method. However, this effect happens only when we try to detect all the small edges that are inside the foreground. Also, we tested all the methods with the PETS 2009 data set, and our proposed method outperformed the other methods, as shown by the ROC curves in Fig. 13(a), (c), and (e). In these ROC curves, the AUC of the proposed method is 97.54% on average, which outperforms the second best by 5.79%. Despite the high AUC in the ROC curves, the methods have low precision–recall curves, as shown in Fig. 13(b), (d), and (f). However, the proposed method outperforms the other methods, and on average, it produces better precision, with 4% more AUC than the second best.

In general, the methods we tested against were not built to work under such challenging conditions, e.g., illumination changes and dynamic backgrounds, so they show poor performance. They need stronger background models and detection algorithms to overcome the changing edges in dynamic environments. Our method is designed to overcome these challenges by using statistics and temporal information to model the environment, edges accumulation for learning, threshold and normalization processes to remove ghost effects,

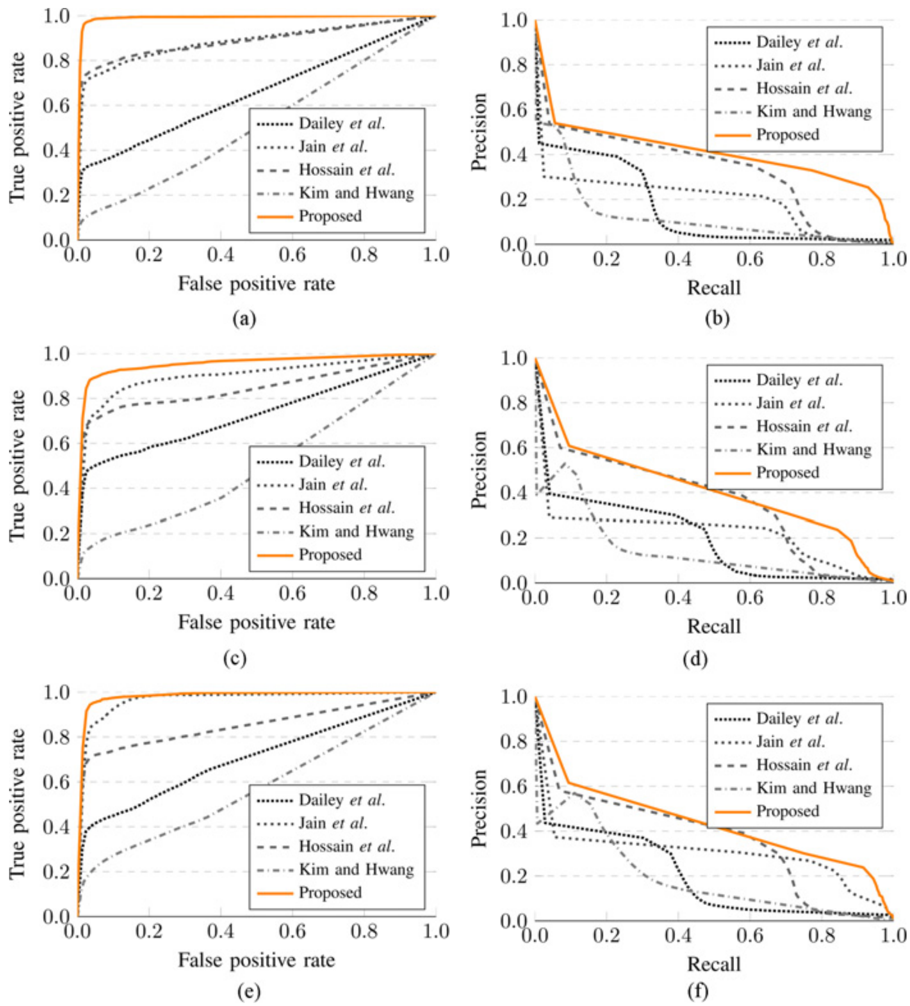


Fig. 13. Evaluation of the detection algorithms using PETS 2009 data set. (Left column) ROC and (right column) precision–recall curves for the algorithms in (a), (b) View 1, (c), (d) View 5, and (e), (f) View 6.

and an independent verification process for each edge to determine whether it is background or a moving object.

V. Conclusion

We presented a statistical edge-segment-based method to model the background and detect moving objects in dynamic environments. Our method modeled the background in the presence of moving objects without causing the ghost effects by using the accumulation of edges as segments, along with adaptive threshold operations. Previous methods were not able to perform as well in such challenging environments. The proposed method built statistical distributions for each edge-segment and used each edge's unique information, which resulted in a robust adaptive verification process.

Moreover, thanks to these features, we overcome the most common edge problems of shape and position changes. Furthermore, these mechanisms can be incorporated into other edge-based methods to extend their functionality and make them robust in dynamic environments. The proposed statistical map can be used to split foreground edges that merge with the background, increasing the detection accuracy. In addition, we explored the edge domain, which has not been researched as much as the pixel domain, for object detection. We found promising results that can be used in several applications, including surveillance in dynamic backgrounds, content-based video encoding, and pose recognition.

References

[1] T. Bouwmans, F. El Baf, and B. Vachon, "Statistical background modeling for foreground detection: A survey," in *Proc. HPRCV*, 2009, pp. IV-181–IV-199.

[2] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst., Man, Cybern C, Appl. Rev.*, vol. 34, no. 3, pp. 334–352, Aug. 2004.

[3] M. Piccardi, "Background subtraction techniques: A review," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, vol. 4. Oct. 2004, pp. 3099–3104.

[4] R. J. Radke, S. Andra, O. Al-kofahi, B. Roysam, S. Member, and S. Member, "Image change detection algorithms: A systematic survey," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 294–307, Mar. 2005.

[5] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.

[6] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. Eur. Conf. Comput. Vision*, 2000, pp. 751–767.

[7] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[8] L. Li and M. K. H. Leung, "Integrating intensity and texture differences for robust change detection," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 105–112, Feb. 2002.

[9] L. Li, W. Huang, I. Y. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[10] C. Stauffer, W. Eric, and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[11] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. 7th Int. Conf. Comput. Vision*, vol. 1. 1999, pp. 255–261.

[12] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[13] A. Behrad, S. Ahmad Motamedi, K. Madani, and M. Esnaashari, "A new algorithm for target tracking using fuzzy-edge-based feature matching and robust statistic," in *Proc. IEEE ICIP*, vol. 1. Sep. 2002, pp. I-577–I-580.

[14] D. J. Dailey, F. W. Cathey, and S. Pumrin, "An algorithm to estimate mean traffic speed using uncalibrated cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 98–107, Jun. 2000.

[15] G. Jiang, S. Wang, M. Yu, T. Choi, and Y. Kim, "New method of vision based vehicle detection and tracking in complicated background," in *Proc. TENCON IEEE Region 10 Conf.*, vol. 1. Nov. 2004, pp. 387–390.

[16] J.-N. Chi, C. Zhang, H. Zhang, Y. Liu, and Y.-T. Yan, "Approach of moving objects detection in active video surveillance," in *Proc. 48th IEEE Conf. Decision Control, Held Jointly With the 2009 28th Chin. Control Conf*, Dec. 2009, pp. 3130–3136.

[17] H. Kalviainen, "Motion detection using the randomised hough transform: Exploiting gradient information and detecting multiple moving objects," *Vision, Image Signal Process., IEE Proc.*, vol. 143, no. 6, pp. 361–369, Dec. 1996.

[18] C. Kim and J. Hwang, "Fast and automatic video object segmentation and tracking for content-based applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 2, pp. 122–129, Feb. 2002.

[19] Q.-y. Yang and X.-y. Gao, "Tracking on motion of small target based on edge detection," in *Proc. WRI World Congr. Comput. Sci. Inform. Eng.*, vol. 6. Mar. 2009, pp. 619–622.

[20] O. Yang, "Moving object detecting method in video," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 23, no. 1, pp. 18–20, Jan. 2008.

[21] Z. Wu, J. Bu, and C. Chen, "Detection and location of people in video streams by fusion of color, edge and motion information," in *Proc. IEEE ICIP*, vol. 3. Sep. 2002, pp. 449–452.

[22] L. Zhi-fang and Y. Zhisheng, "A real-time vision-based vehicle tracking and traffic surveillance," in *Proc. 8th ACIS Int. Conf. Software Eng. Artif. Intell. Networking Parallel/Distributed Comput.*, vol. 1. Jul. 2007, pp. 174–179.

[23] V. Jain, B. Kimia, and J. Mundy, "Background modeling based on subpixel edges," in *Proc. ICIP*, vol. 6. Sep. 2007, pp. VI-321–VI-324.

[24] M. J. Hossain, M. A. A. Dewan, and O. Chae, "Moving object detection for real time video surveillance: An edge based approach," *IEICE Trans. Commun.*, vol. 90-B, no. 12, pp. 3654–3664, 2007.

[25] D. Gutchess, M. Trajkovic, E. Cohen-Solal, D. Lyons, and A. K. Jain, "A background model initialization algorithm for video surveillance," in *Proc. IEEE ICCV*, pt. 1. Jul. 2001, pp. 733–740.

[26] M. Yokoyama and T. Poggio, "A contour-based moving object detection and tracking," in *Proc. 2nd Joint IEEE Int. Visual Surveillance Performance Evaluation Tracking Surveillance Workshop*, Oct. 2005, pp. 271–276.

[27] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 849–865, Nov. 1988.

[28] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[29] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.

[30] *Performance Evaluation of Tracking and Surveillance 2001*. (2010, Nov.) [Online]. Available: ftp://ftp.pets.rdg.ac.uk/pub/PETS2001/

[31] *Performance Evaluation of Tracking and Surveillance 2009*. (2010, Nov.) [Online]. Available: ftp://ftp.pets.rdg.ac.uk/pub/PETS2009/

[32] J. Ferryman and A. Shahrokni, "Pets2009: Dataset and challenge," in *Peoc. 12th Int. Workshop Performance Evaluation Tracking Surveillance*, Dec. 2009, pp. 1–6.

[33] Laboratory for Image and Media Understanding. (2012, Jul.) [Online]. Available: http://limu.ait.kyushu-u.ac.jp/dataset/en/index.html

Adin Ramirez Rivera (S'12) received the B.Sc. degree in computer engineering from San Carlos University, San Carlos, Guatemala, in 2009. He is currently pursuing the Ph.D. degree at the Department of Computer Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do, South Korea.

His current research interests include image enhancement, object detection, facial analysis, and pattern recognition.

Mahbub Murshed received the B.Sc. degree in computer science and engineering from Khulna University, Khulna, Bangladesh, in 2003, and the M.S.–Ph.D. combined degree in computer engineering from Kyung Hee University, Seoul, South Korea, in 2012.

Currently, he is an Assistant Professor with the School of Engineering and Computer Science, Independent University Bangladesh, Dhaka, Bangladesh. He is also with the Image Processing Laboratory, Kyung Hee University. His current research interests include image processing, computer vision, and pattern recognition.

Jaemyun Kim received the B.Sc. degree in computer engineering from Kyung Hee University, Yongin-si, Gyeonggi-do, South Korea, in 2010, where he is currently pursuing the Ph.D. degree at the Department of Computer Engineering.

His current research interests include object detection and tracking, and pattern recognition.

Oksam Chae (M'92) received the B.Sc. degree in electronics engineering from Inha University, Incheon, South Korea, in 1977, and the M.S. and Ph.D. degrees in electrical and computer engineering from Oklahoma State University, Stillwater, OK, USA, in 1982 and 1986, respectively.

From 1986 to 1988, he was a Research Engineer with the Texas Instruments Image Processing Laboratory, TX, USA. Since 1988, he has been a Professor with the Department of Computer Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do, South Korea. His current research interests include multimedia data processing environment, intrusion detection systems, sensor networks, and medical image processing in dentistry.

Dr. Chae is a member of SPIE, KES, and IEICE.