International Journal of Innovative Computing, Information and Control
Volume **8**, Number **7(B)**, July **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 4943–4957

STATISTICAL BINARY EDGE FREQUENCY ACCUMULATION MODEL FOR MOVING OBJECT DETECTION

Mahbub Murshed, Adin Ramirez, Jaemyun Kim and Oksam Chae

Department of Computer Engineering
Kyung Hee University
Yougin-si, Gyeonggi-do 446-701, South Korea
mmurshed@gmail.com

Received April 2011; revised September 2011

Abstract. *We propose an edge segment based statistical background modeling algorithm and a moving edge detection framework for the detection of moving objects using a static camera. Traditional pixel based methods create difficulties to update the background model. They also bring out ghosts while a sudden change occurs in the background. Although edge based methods are robust to illumination variation and noise, existing edge-pixel based methods suffer from scattered moving edge pixels since they cannot utilize edge shape information. Moreover, traditional edge-segment based methods treat every edge segment equally creating edge mismatch due to non stationary background. This paper presents an edge-segment based statistical approach to modeling the background by using ordinary training images that may even contain moving objects. The proposed method relies on background edge segment matching; thus it does not leave any ghost behind. Moreover, the proposed method uses a statistical model for every background edge segment individually that makes the approach robust to handle camera movement as well as adapt to background motion (moving tree branches). Experiments with natural image sequences show that our method can detect moving edges efficiently under the above mentioned difficulties.*

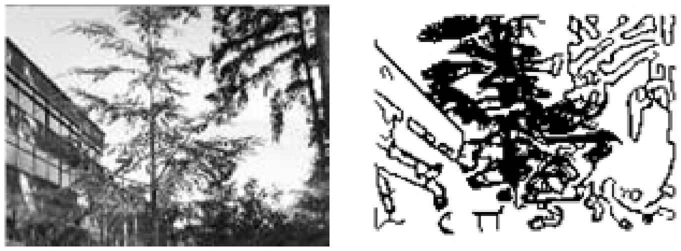
Keywords: Background modeling, Statistical distribution map, Moving edge segment, Edge segment matching

1. **Introduction.** Detection of moving objects has been an important research topic for the past few decades having prevalent applications in a variety of disciplines. A simple but popular method for moving object detection is the background subtraction. Here moving objects are obtained from the difference image, made from the difference between the current frame and the background model. Existing approach to background model initialization assumes that a sequence of motion free frames is available prior to building the background model [1]. In case of a surveillance area, a busy street or in a public place, it is very difficult to collect training background frames without any moving objects in it. Background modeling becomes more challenging while there are illumination variations and noise in the background. This becomes worse in the outdoor environment due to weather condition, reflectance, motion in the background (waving tree branch), and unintentional camera motion. Moreover, color, pattern or shape of a background sometimes resemble a moving object [2]. Thus, to adapt to this changing environment, the background model needs to be updated in every frame with an adaptation rate. Existing methods do not consider object's motion for the selection of optimal update rate but rather they set a common rate for updating every background pixel. Thus the pixel intensity based moving object detection methods leave ghosts (especially when a sudden change occurs for slowly moving objects) behind them [3]. Additionally, to segment the

moving area, they need to set a threshold over the difference image. Choosing optimum threshold value is application dependent and very difficult to achieve. Thus, some other methods [4, 5, 6, 7] have come up with spatial features like edge, corner and contour of the boundary. to improve the performance of the background model. Sugandi et al. [8] detected moving object by encoding texture information of the background. To avoid comparing pixel intensity directly, they have created PISC image by encoding edge texture using a 5×5 pixel mask. Moving objects are determined by utilizing the correlation value between background and foreground PISC code. Edge is more robust than intensity in the illumination changing situation [1]. Edge extraction can significantly reduce data access rate and discard less useful information by keeping important structural property of an image [7]. Thus, processing edge image is much faster than processing intensity image. These features attract many researchers [1, 5, 6, 9, 10, 11] to work with edge information for various application domain including object detection, tracking, change detection, recognition, etc. There are extensive surveys on various moving object detection methods along with a variety of background modeling techniques [12, 13, 14].

Existing background modeling algorithms model every background pixel individually. Once the color likelihood of the input frame is computed, the pixels that deviate from the background model are labeled as the foreground pixels. Modeling every background pixel is difficult since intensity feature is very prone to illumination change. Thus, Musa et al. [15] used histograms to cluster object motions from sequence image. However, multiple colors can be observed at certain locations due to the repetitive object motion, shadows, noise or reflectance from other objects. Traditional method utilizes temporal differencing or optic flow based method. The temporal differencing method utilizes two or more consecutive frames to extract moving regions [16]. Optical flow uses the characteristics of flow vectors of the moving objects over time [17]. If the background contents are not visible for a long time, all these methods will fail to generate accurate background model. These methods are vulnerable and prone to false detection, if the temporal changes are generated by noise or illumination change due to weather condition [18]. Edge features are thus useful tools for modeling the environment under these limitations. However, existing edge based methods [5, 6, 11] use edge differencing. They treat every edge pixel individually and thus they suffer from random noise. Pixel by pixel matching of edge points is not suitable due to higher computational cost. Additionally, edges extracted from each frames do not always show consistency within frames. Background edges show shape and size variation within frames. Moreover, the variations for different edges are not the same. Without considering this variation from the environment, detectors output cannot be reliable. Figure 1 describes the necessity to incorporate statistical information for background modeling. As is shown in Figure 1, edges in reference image change their position due to illumination variation, noise, and reflectance and for the movement in the air. Figure 1(a) shows a sample background reference image. Figure 1(b) is made from the superimposition of fifty reference edge images. It is clear that edges change their position and thus the edges in the superimposed edge image have thick lines. This thickness of the line indicates the movement statistics for that corresponding reference edge segment. Here we also see that the movement statistics for different reference edge segments is different. Therefore, traditional methods would fail in such a situation.

In our method, we extract edges from video frames using canny edge detector [19] and represent them as a structure of edge segments [20]. We do not process intensity values individually but rather all the edge pixels of a segment are processed together. Our proposed binary edge frequency accumulation helps to initialize our statistical background edge distribution model without the necessity of motion free frames. Edge specific automatic thresholds are generated from the statistical distribution that can separate true



(a) (b)

Figure 1. Necessity for statistical background model: (a) a sample background reference image and (b) edges made from the accumulation of 50 superimposed background reference edge images

(a) (b) (c)

Figure 2. Characteristics of background edge segments in consecutive frames: (a)-(b) edge segments from two consecutive frames, and (c) difference between (a) and (b)

background edges. To solve the problem of dynamic background, background model updating scheme is used. Our proposed background model can tolerate camera jittering or calibration error in a limited scale. It takes less time to process since we do not need to search every edge pixel individually unlike the traditional methods. Thus, our method utilizes the robustness of edge segment structure and utilizes statistical background model to facilitate fast and flexible background edge segment matching for the detection of true moving object edges.

2. **The Proposed Edge Segment Based Statistical Background Model.** Accuracy of matching background edges in an image sequence suffers from position and shape variation in different frames due to illumination change, background movement, camera movement and noise. This variation differs from edge to edge even for the same scene. Figure 2 illustrates the characteristics of background edges for consecutive frames. Figures 2(a) and 2(b) show two consecutive background edge images. Figure 2(c) shows the edge difference image of those two images. From Figure 2(c), the region labeled '1' contains the moving edges from a tree with noise and the region labeled '2' has illumination variation. Thus, in the edge difference image, we have more compact edge segments in those regions compared with the other regions. It is obvious from Figure 2 that, we should model every background edge segment individually unlike traditional global common distance threshold based methods [1, 10]. Edge segment based statistical background model can estimate background edge behavior by observing a number of reference frames and thus can keep the statistics of motion variations, shape, and segment size variation for every background edge segment. This statistics helps to improve background edge matching accuracy by setting edge specific automatic threshold. In order to accumulate

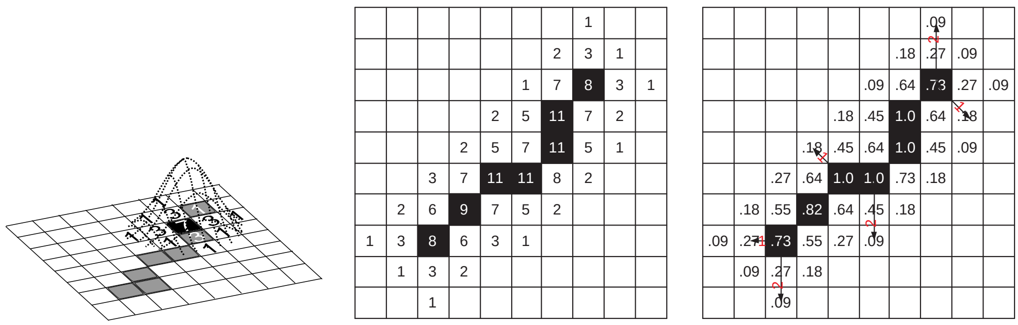
background behavior, edges e_j^i from sample image frame i are extracted using Canny edge operator [19]. Now, let us define the set of all extracted edges from frame i as a binary edge map E^i. A Gaussian kernel approximated mask $G(.)$ as shown in Figure 3(a) is places on every edge pixel positions over E^i and convolution is performed to create edge distribution map D_{E^i}:

$$D_{E^i} = \sum_{\forall e_j^i \in E^i} G(e_j^i) \tag{1}$$

The convolution outcome of the mask over a given edge segment is shown in Figure 3(b) (edge positions are shaded in black). Now, form a set of edge distribution maps, extracted from N frames, and we superimpose them and add them together to get binary background edge frequency accumulation map A_E:

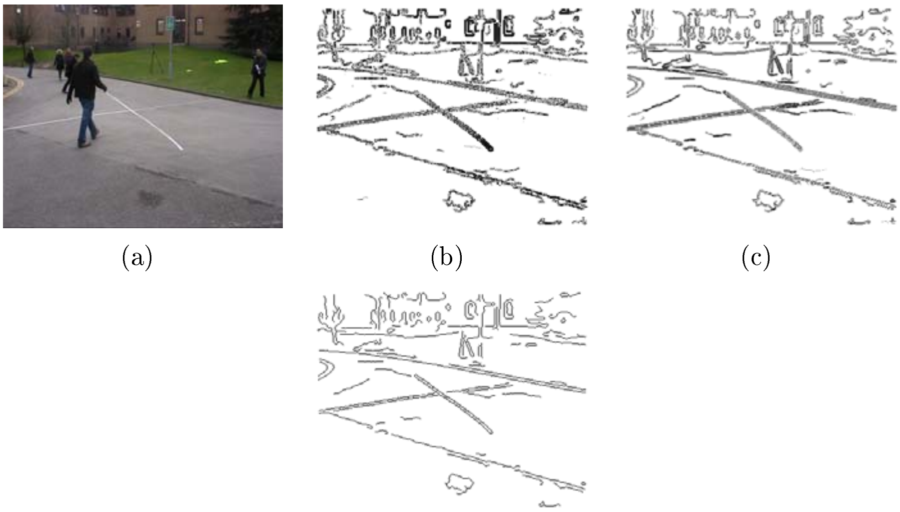
$$A_E = \sum_{i=1}^{N} D_{E^i} \tag{2}$$

Figure 4(b) shows an example for the binary background edge frequency accumulation. The accumulation map A_E generated this way contains useful information like edge motion statistics, edge shape variation statistics, edge flickering statistics, as well as the influence from moving objects that were present at the sequence during the accumulation period. To make A_E independent from the training sequences, we need to remove the edge contribution for all the moving objects that pass through those sequences. Thus we clearly see the advantage of the proposed binary edge frequency accumulation over existing gradient based accumulation [1]. The edge distribution of the selected ROI region in Figure 5(a) is shown in Figure 5(b). Figure 5(c) is a cut of distribution of Figure 5(b) at column 60. Here, moving objects contribution in the accumulation map is shown in red color, while background edge pixels create high accumulation and is shown in blue color. From Figures 5(b) and 5(c), it is obvious that the higher peaks in A_E correspond to the contribution of edges from the background where very small peak is from the influence of moving object edges. A clever selection of a threshold can suppress the effect of moving objects over this accumulation. The same ROI region for the gradient accumulation is



(a) (b) (c)

Figure 3. Gaussian convolution over a sample edge segment: (a) a Gaussian kernel approximated mask, (b) convolution outcome using the mask in (a) and every edge pixel position (dark pixels) of an edge segment, and (c) normalized relative probability values for getting an edge pixel for each pixel location in the distribution (labeled arrow indicates eight neighborhood distance values from edge mean position to some sample points over the distribution)



(a) (b) (c)

(d)

FIGURE 4. Creation of Ψ_{SD} and E^B_{SD} from simple sequence: (a) a sample frame from a sequence, (b) accumulation map A_E of the sequence in (a), (c) Ψ_{SD} obtained by threshold and normalizing (b), and (d) E^B_{SD} obtained by non-maxima suppression thinning algorithm over (c)

shown in Figure 5(e). From Figure 5(f), which shows the same cut at column 60, we see that it is very difficult to set threshold to separate the edge contribution of moving object from the sequence. Thus the use of binary edge frequency accumulation gives us wider boundary to set threshold so that we can separate moving object's contribution from the true background edge accumulation. Now, we assume that the slowest moving object will have at least the speed of one pixel-per-frame. Even if a moving object moves along a linear trajectory path, the highest accumulation value over the accumulation map would be at most the pixel wise highest dimension (κ) times maximum value of the kernel mask $\max(G(.))$ for that moving object. Hence, we can easily set a threshold anywhere between $\kappa \times \max(G(.))$ and $\alpha \times N \times \max(G(.))$. The parameter α indicates background edge occurrence probability per frame. For a true background edge usually $0.5 \leq \alpha \leq 1$, we empirically found that the threshold $\alpha = 0.25$ gives good separation for large training frames ($N \geq 50$). Now, we normalize the background edge accumulation map. Figure 3(c) illustrates normalized background edge frequency accumulation map that is made by dividing every pixel position value of the distribution in Figure 3(b) with the highest value found in the distribution. Normalization gives the relative probability value for getting an edge pixel for that position in the distribution. From Figure 3(c), the labeled arrow indicates the eight neighborhood distance D_8 values from the edge mean position to some sample points over the distribution, where the computation of edge means position is briefly discussed in Section 2.1. D_8 measures the edge pixel movement from its mean position. Thus, the distribution map generated this way after threshold and normalizing A_E is called Statistical Edge Distribution Map Ψ_{SD}. Figure 4 shows an example of Ψ_{SD}, created from a sample sequence.

2.1. **Creation of static background edge list.** We create Static Background Edge List (E^B_{SD}) directly from Ψ_{SD} by using a non-maxima suppression thinning algorithm over it. Thus we extract thin edge segments from Ψ_{SD} through the edge mean position (peak) for every edge's distribution map as shown in Figure 4(d). Unique labels are assigned for the newly extracted edge segments. Edge segment labeling maps for all (E^B_{SD}) segments

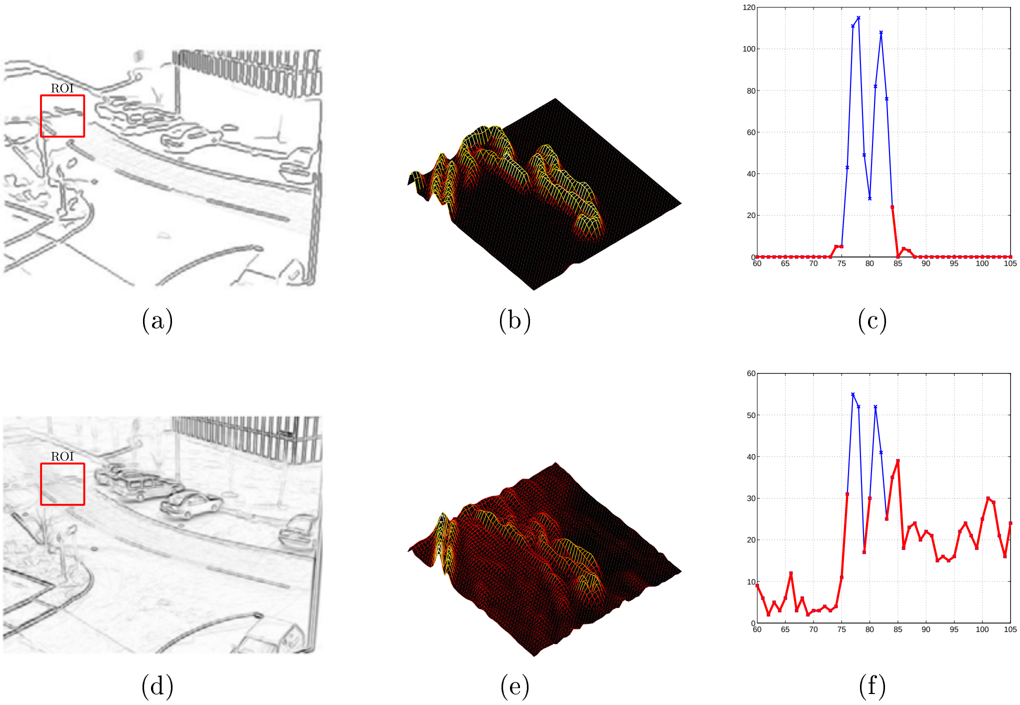


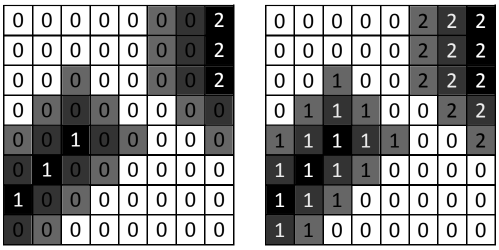
Figure 5. The advantages of binary background edge frequency accumulation map: (a) background edge accumulated map with selected ROI, (b) edge distribution map of the ROI in (a), (c) a cut of the distribution in (b) at column 60, (d) background edge gradient accumulated map for the same area as in (a), (e) accumulated gradient distribution map of the same ROI, and (f) a cut of the distribution in (e) at column 60 (moving objects contribution in the accumulation map is shown in red color, while background edge pixels create high accumulation and is shown in blue color)

are also computed using Ψ_{SD}. Figure 6 illustrates two edge distribution maps along with corresponding thinned edge segments in dark color with their labels shown in its distribution. Thus, by utilizing Ψ_{SD}, every background edge segment j can create edge labeling map Ψ^j_{SD} as shown in Figure 6(b). The labeled area represents a background edge's motion statistics throughout the training sequence. The area also indicates the search boundary for a candidate background edge segment. Now, for each edge segment e^B_j on the E^B_{SD}, edge appearance relative mean probability μ^B_j is calculated by taking the average of all the edge pixel location (x) values from e^B_j over Ψ_{SD}.

$$\mu^B_j = \frac{1}{n_j} \sum_{\forall x \in e^B_j} \Psi_{SD}(x) \tag{3}$$

here n_j is the number of edge points on the edge segment e^B_j. Assume a threshold and normalized statistical distribution map, Ψ_{SD} in Figure 3(c), where the edge mean positions are highlighted in black shade, μ^B_j on the edge mean position is 0.896. Thus the distribution has a relative probability of having a background segment at its mean position by 0.896. The variance in distance (σ^B_j) of the probability distribution is calculated by:

$$\sigma^B_j = \frac{1}{T_j} \sum_{\forall x \in \Psi^i_{SD}} D_8(x) \times abs\left(\mu^B_j - \Psi_{SD}(x)\right) \tag{4}$$



(a) (b)

FIGURE 6. Edge distribution map of two sample background edge segments with their search boundary: (a) edge distribution for two segments where edge positions are labeled with segment identifier and (b) edge labeling map for the two edge segments for their associated distribution in (a)

here, T_j is the total number of points in the labeled region Ψ^j_{SD}, $D_8(x)$ is the eight-connectivity distance from nearest edge mean location in E^B_{SD} to the point x, and $\Psi_{SD}(x)$ is the normalized accumulation score, i.e., the relative probability value at position x. Using Equation (4), $\sigma^B_j = 0.624$ is found for the distribution in Figure 3(c). Now, for all e^B_j segments in the E^B_{SD} list, corresponding σ^B_j are stored as distance threshold for its associated background edge segment.

2.2. **Flexible background edge segment matching.** The background edge segment matching in our proposed method utilizes background edge segment's statistics. Thus, the candidate background edge segment will be examined for a match using a wider edge distribution map, if corresponding background edges have high motion variation statistics. A narrow edge distribution map is applicable for those where background segment has less motion variation history. Thus the proposed method utilizes edge specific flexibility during matching background edge segment that drastically reduces false alarm rate. To match a candidate background edge segment j, the system first looks for the corresponding background edge segment using the edge labeling map Ψ^j_{SD} discussed in Section 2.1. Thus we get background edge segment e^B_j along with its variance in distance σ^B_j threshold. If no corresponding background segment is found from Ψ^j_{SD}, then the segment is a candidate moving edge segment; otherwise, for a given sample edge segment e^i_j at frame i, to determine if it is a background edge, we compute average eight neighborhood distance $D^B_{CB}(e^i_j)$ from all points of e^i_j to the closest point of background segment in E^B_{SD}:

$$D^B_{CB}(e^i_j) = \frac{1}{n_j} \sum_{\forall x \in e^i_j} D_8(x) \tag{5}$$

here, e^i_j is a background edge segment if it satisfies $D^B_{CB}(e^i_j) < 2 \times \sigma^B_j$ and e^i_j is a candidate moving edge segment otherwise. The edge list formed by adding all candidate moving edge segments at current frame i is thus the Candidate Moving Edge List E^i_{CM}.

$$E^i_{CM} = \underset{e^i_j \in E^i \wedge D^B_{CB}(e^i_j) \geq 2\times\sigma^B_j}{\forall j \{e^i_j\}} \tag{6}$$

3. **Moving Edge Detection.** The framework of proposed moving edge detection system is given in Figure 7. Here the proposed Statistical Edge Distribution Map (Ψ_{SD}) is used for flexible background edge segment matching. In order to adapt to dynamic change in the scene, i.e., for moving edges that are detected at the same position for a long time (moving edges detected from a stopped moving car), the proposed method uses

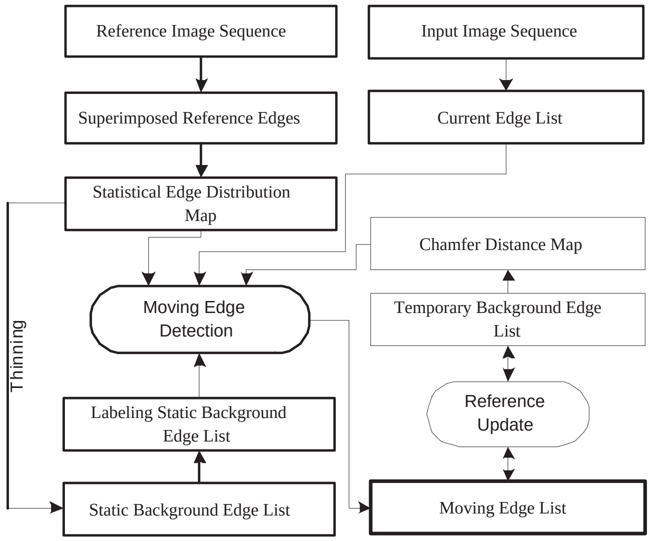
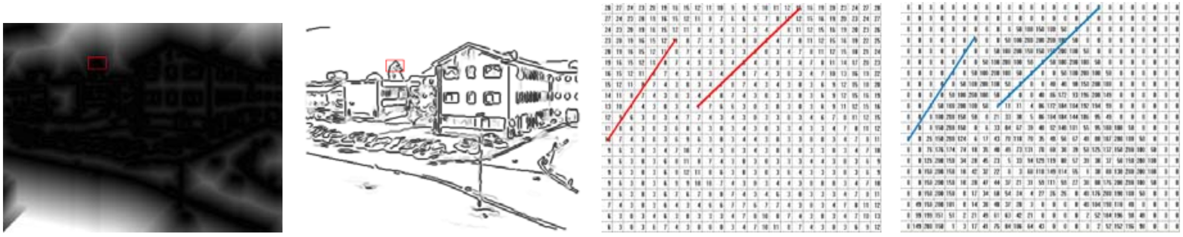


FIGURE 7. The framework of the proposed moving edge detection method

temporary background edge list (E^i_{TB}). Here, E^i_{TB} is composed of moving edge segments from previous frames. Hence, the proposed method uses Ψ_{SD} for flexible background edge segment matching followed by a verification method that uses E^i_{TB} in a Chamfer-3/4 distance map (Ψ_{CD}) [21]. A detailed description on moving edge verification method using E^i_{TB} segments by utilizing chamfer distance map and the updating procedure of E^i_{TB} can be found in the work of Hossain et al. [1]. Moving edge segments at current frame i is thus the list E^i_{ME}:

$$E^i_{ME} = \underset{e^i_j \in E^i_{CM} \wedge e^i_j \notin E^i_{TB}}{\forall j \{e^i_j\}} \tag{7}$$

4. **Results and Discussion.** We tested our method on several video sequences including corridor, parking lot, road scene and video sequences from PETS database which could be downloaded from *ftp://ftp.pets.rdg.ac.uk/pub/*. All images were scaled to size 640 × 520 with background motion, illumination change and noise. The proposed system was able to detect almost all of the moving objects in the sequences. We used visual C++ and MTES [22], an image processing environment tool. Our system can process 10 frames per second. The assumption under which our algorithm works is that the color of moving objects will not be exactly the same as the color of the background. We also assume that sufficient light is present at the scene so that canny edge detector can work although there is illumination variation. Additionally, during the background model learning period as least 100 frames are available to build the background model. The approximated computational time (millisecond) at different stages to process each frame in a 2.5GHz CPU running with 2GB of memory is given in Table 1. In each input frame, the edge extraction step extracts edge segments using Canny edge detector [19]. Edge extraction step takes approximately 50ms, which is the highest time taken among all steps. Thus, we have an option to speed-up the system even up to 20 frames per second by using an optimized canny edge detection algorithm. The next step represents collection of edge pixels as edge segment using an edge identifier. Only 10ms is needed for this step. The removal of scattered noisy edge segments that takes 5ms is performed afterwards. The following step generates Distance Image for the temporary reference edge list to adapt the change in the environment. Only 5ms is needed for this step. Finally, moving edge matching uses the Statistical Model along with the Distance Image takes 20ms.



(a) (b) (c) (d)

FIGURE 8. Distance map is used for background edge matching in Hossain et al. method and the proposed method: (a) Ψ_{CD} of a background with ROI in red color, (b) Ψ_{SD} of a background with ROI in red color, (c) the distance values of Ψ_{CD} in the selected ROI region in (a) and two superimposed line segments, and (d) edge accumulation values of Ψ_{SD} in the same ROI region and two line segments in the same position as in (c)

Figure 8 demonstrates the performance of normalized statistical distribution map Ψ_{SD} over chamfer-3/4 distance map Ψ_{CD} for background edge matching. In chamfer distance matching, every background edge segment is matched using a common threshold as is used in Hossain et al. method [1]. But every background segment has its own motion statistics. An edge segment that has high movement statistics should be matched with a wider search area and high threshold. Non-moving edge segments should benefit from low threshold and narrow search region. Figures 8(a) and 8(b) show Ψ_{CD} as well as ΨSD of a sequence of background frames where the ROI is highlighted in red color. The same ROI regions are also exposed numerically in Figures 8(c) and 8(d). Again, two synthetic edges are placed over Ψ_{CD} and Ψ_{SD} in Figures 8(c) and 8(d) respectively. In the background edge matching model with Ψ_{CD}, the numeric value in every small cell indicates distance score an edge pixel from the test image will obtain if it is overlapped on that point whereas in the background model with Ψ_{SD} numeric value indicates edge accumulation scores obtained from training images that resembles the relative probability of having a background edge pixel over that point. In this example, for Ψ_{CD}, every entry value '0' indicates the background edge segment positions. On the other hand for Ψ_{SD} every value '200' represents the background edge mean position with highest probability for the corresponding background segment. (Note that, here normalization is not used in the Ψ_{SD}, due to example simplicity.) The matching score of the left most line over Ψ_{CD} is 2.938, i.e., according to Ψ_{CD} the line is just three pixels apart from the reference background edge image. Thus Ψ_{CD} will detect the above line as background edge by the Hossain et al. [1] method. But in Ψ_{SD} the score of the same line is '0'. Thus, the proposed method accurately detects the line as a moving edge line. The red color in Figure 8(c) indicates false detection where the blue color in Figure 8(d) represents correct detection. Here we see that background edges will be detected more accurately while using Ψ_{SD} because of its statistical search boundary and automatic threshold selection.

Figure 9 shows a comparison among several moving edge detection methods using some selected frames of some sequences. In Figures 9(a) and 9(b), we have illumination variation due to the movements of the cloud in the sunny environment. Figures 9(c) and 9(d) have over exposed image sequences with illumination variation and reflection from the windows, cars and other moving objects. The ground truth for the selected frames is shown on the second row of Figure 9. Kim and Hwang detected moving objects from sequence images by using edge differencing method with a combination of three edge maps [5]. Using edge pixel differencing method they compute current moving edges and

Table 1. Computational time (millisecond) to process every frame in the proposed method

Processing Steps	Mean Time (ms)
Edge Extraction	50
Edge Segment Representation	10
Removal of Noisy Pixels	5
Generation of Distance Image	5
Edge Matching	20
Total	90

temporary moving edges. Finally, moving edges are determined by applying logical or operation between them. Their method does not update the background model. Thus, the method cannot handle dynamic background and results in higher false alarm. To solve the problem of handling dynamic background, Jain et al. [6] proposed a method that models the background based on sub pixel edge map by the representation of edge position and orientation using a mixture of Gaussians Model. Their method has a high computational cost due to the use of increased number of Gaussians that requires update at every frame. Due to illumination variation, Jain et al. [6] as well as Kim and Hwang [5] detect a lot of scattered edge pixels as shown in the third and fourth row of Figure 9 respectively. Dailey et al. [11] computes moving object without using any background. In their method, two edge maps are extracted from the edge difference image of three consecutive frames. Finally, the moving edges are extracted by applying logical AND operation between the two edge maps. Here, the method makes exact matching between edge pixels from the two edge maps. Due to random noise or small camera movement, edge pixel position may change in consecutive frames. Therefore, exact edge matching is not sufficient to detect moving edges. The method also fails to detect slowly moving objects and thus cannot be used for real-time applications. The detection result for the Dailey and Cathey Method [11] is shown in the fifth row of Figure 9. Dailey's method is very sensitive to camera movements. Although in the given sequence there are no camera movements, Dailey's method cannot give compact shape of the moving object. Instead, it gives scattered pixels in the moving object region. Edge segment based approach introduced by Hossain et al. [1], see sixth row of Figure 9, utilizes edge segment structure while detecting moving objects. Their method can control camera movement in a limited scale but it requires some initial motion free training frames for generating the background. For matching background edge segments as well as temporary background edge segments (edge segments from a car that stops moving in the scene), they have used the same chamfer distance [21] based matching method. Moreover, their method uses a common threshold value for all the background segments. Selection of a lower threshold results in matching of edges with small movement variation. On the other hand, higher threshold increases false matching of moving edge and background edge. Moving object detection in our method is given in the last row of Figure 9. Here we utilize movement statistics of every background edge segment effectively. Due to the segment based statistical nature of our proposed method, it overcomes all the difficulties and outperforms all the existing edge based moving objects detection methods. The proposed method can also be applied in the edge segment based object tracking systems [10, 23] for better accuracy. Edges are less sensitive to illumination change compared with color or texture. Thus, edges are useful tool for boundary based object tracking algorithms. Additionally, the proposed method can be used to track non-rigid human body part analysis which has a great demand for

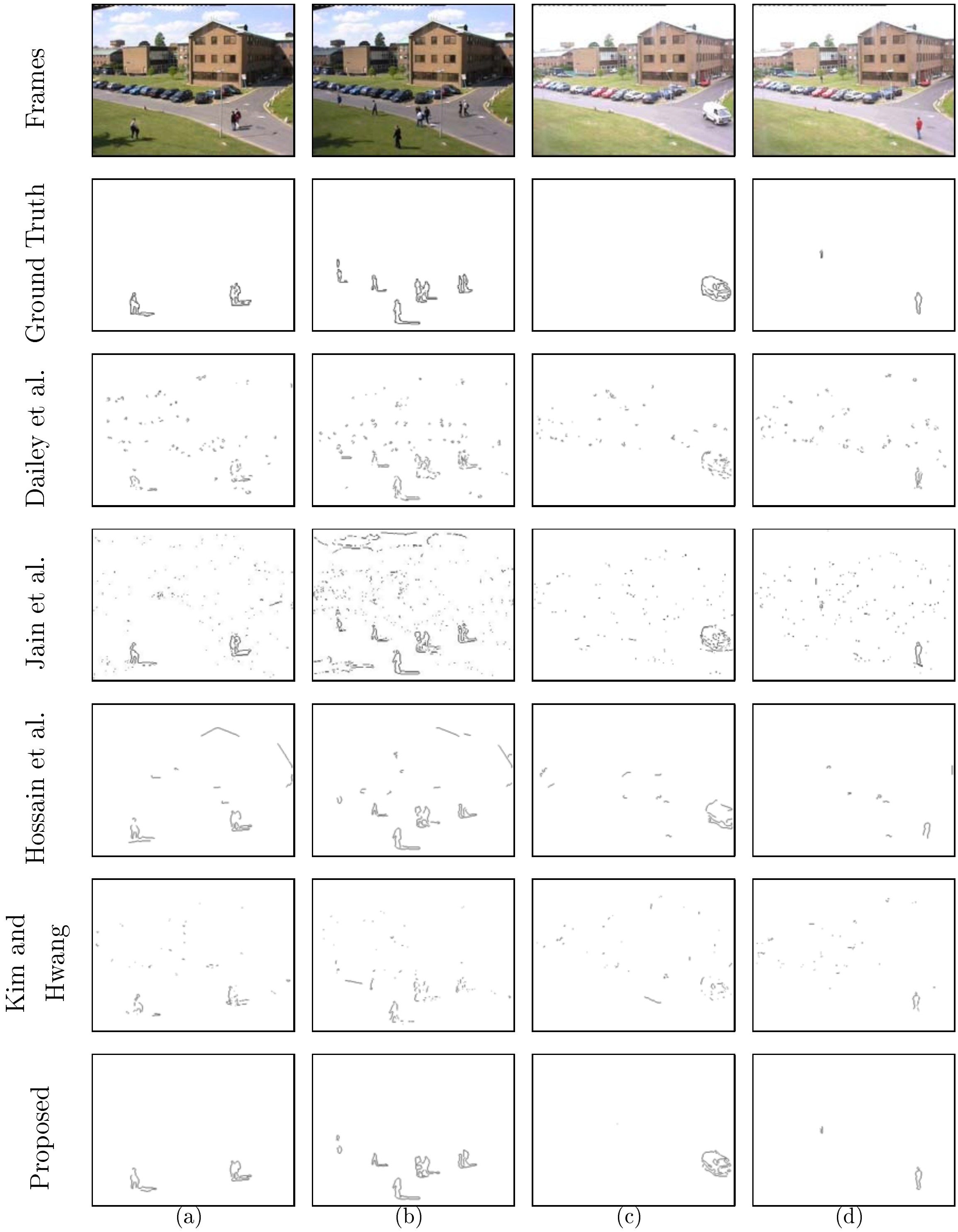
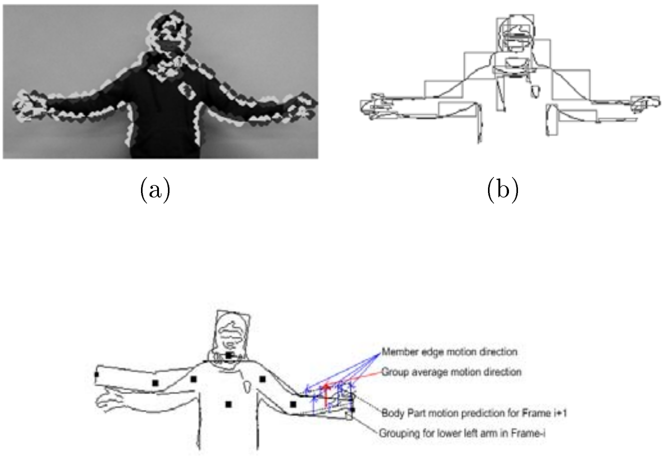


FIGURE 9. Subjective evaluation in different challenging environments: (a)-(b) columns correspond to frame no.1461 and frame no.2661 from sequence PETS "Data Set 3", and (c)-(d) columns correspond to frame no.891 and frame no.2601 from sequence PETS "Data Set 4"



(a) (b)

(c)

FIGURE 10. Illustrating a possible application using moving edge segments: (a) extracting side colors from moving segments, (b) computing edge motion by segment tracking, and (c) body part tracking utilizing a human body model

automated human body part detection and tracking [24]. Human body tracking is difficult due to the large variation of movements for different body parts. Edge segments from limbs (legs and arms) show high movement variation while walking or running, where as head and torso segments move slowly. In the proposed method, we can assign weight to moving segments so that different level of flexibility can be applied during segment matching. Figure 10 describes a possible application of moving edge segments. Here we can build a segment tracker [23] to track every moving edge segments as is shown in Figure 10(b). Segments with similar motion and similar side color can form a group. We can compute average group motion. While tracking a group, member edge segment that deviate from the average group motion or whose side intensity does not match with the group's intensity range can be eliminated. Utilizing a human model, we can initialize a body part tracker. This tracker can predict possible body part location as depicted in Figure 10(c) for the next frame.

To evaluate the performance of the proposed system quantitatively, we compare the detected moving edge segments with the ground truth that is obtained manually. The metric used for performance evaluation is based on three criteria: Precision, Recall and F-Measure that are defined in Equation (8), Equation (9) and Equation (10). Precision measures the accuracy of detecting moving edges while Recall computes the effectiveness of the extracted actual moving edge segments. Thus, F-measure, which measures the harmonic mean of precision and recall, gives a single measure of performance for the proposed system.

$$\text{Precision} = \frac{\text{Extracted moving edge pixels}}{\text{Total extracted edge pixels}} \tag{8}$$

$$\text{Recall} = \frac{\text{Extracted moving edge pixels}}{\text{Total actual moving edge pixels}} \tag{9}$$

$$\text{F-Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

Figure 11(a) shows the precision of detected moving edges in different methods. Here we have used five different datasets where each of the datasets has around hundred frames.

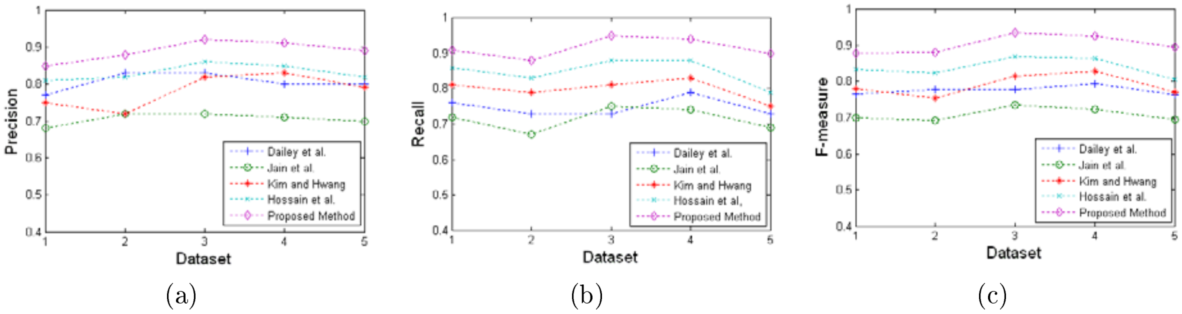


FIGURE 11. (a) Precision, (b) Recall, and (c) F-Measure of detected moving edges by different methods

Dataset 1 and Dataset 2 have illumination variation due to the movement of the cloud and for the moving tree branches. Dataset 3 is taken from indoor environment. Over exposed images are tested in Dataset 4 that has noise and reflectance from objects. Finally, a combined sequence that has all the above mentioned effects is tested in Dataset 5. Figure 11(b) demonstrates the recall of detected moving edges. As shown in Figures 11(a) and 11(b), recall is higher in the proposed method while compared with other methods. This is due to the fact that we have modeled the background statistically, i.e., flexibility in matching is given to those segments that have high movement information. Moreover, existing pixel based methods lags by scattered moving edge pixels that are often mismatched which eventually leads to lower recall value. Figure 11(c) illustrates the performance of the proposed method in terms of F-Measure. It describes the overall edge detection accuracy as well as effectiveness. As is shown in the figure, the proposed method outperforms all the existing methods since the method shows high value for both precision and recall parameters. Thus the proposed method is stable and more reliable than other methods discussed in this paper.

5. **Conclusion.** This paper describes the suitability of using edge segment based statistical background model for background edge segment matching while detecting moving objects. The proposed method can be used in video surveillance based applications. The strength of our approach lies in the ability to separate true background edge segments from moving edge segments. To achieve this, we utilize statistical background model for flexible background edge segment matching and chamfer distance based matching for the verification of moving edge segments. Moreover, the background initialization step in the proposed method does not require motion free training image sequences. The example figures described in this paper clearly justify the use of statistical background model, which is highly efficient under illumination variation condition and shape change situation. For segmenting the moving objects, we can use an efficient watershed based segmentation algorithm [25], where the region of interest (ROI) can be obtained by utilizing method [7]. The proposed background modeling technique can be used in conjunction with our previous work [23] for tracking objects with higher accuracy since the work [23] realies on the detected moving edge segments.

However, the proposed method has some limitations. In our method, the minimum number of training frames to build the background model is assumed to be 100. So the proposed method will not detect reliable moving segments until it is trained with at least 100 frames. Once sufficient number of background frames is learned, the system can allow taking arbitrary background frames for learning. Another limitation of our method is that, currently our method works with gray scale images. We are not using color information found in RGB images. To utilize color images, we can use color gradient instead of gray scale gradient for the computation of edges; additionally we can store every edge's side color information for better accuracy in edge matching.

In our future work, we will eliminate highly probable background regions from the background model using the background edge distribution. This will save edge segment matching time and thus speed up the system significantly. Moreover, we will incorporate multi-frame based edge segment matching algorithm along with edge's side color distribution matching for the detection and tracking of moving edges for more sophisticated vision based applications like airport security and human activity recognition.

Acknowledgment. This work was jointly supported by the Korea Research Foundation Grant funded by the Korean Government (MEST) (No. 2011-0003183 and No. 2011-0017151).

REFERENCES

[1] 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.E90-B, no.12, pp.3654-3664, 2007.
[2] Z. Musa and J. Watada, Video tracking system: A survey, *ICIC Express Letters*, vol.2, no.1, pp.65-72, 2008.
[3] W. Zhang, Q. J. Wu and H. B. Yin, Moving vehicles detection based on adaptive motion histogram, *Digit. Signal Process.*, vol.20, no.3, pp.793-805, 2010.
[4] B. Sugandi, H. Kim, J. K. Tan and S. Ishikawa, Real time tracking and identification of moving persons by using a camera in outdoor environment, *International Journal of Innovative Computing, Information and Control*, vol.5, no.5, pp.1179-1188, 2009.
[5] C. Kim and J.-N. Hwang, Fast and automatic video object segmentation and tracking for content-based applications, *IEEE Trans. Circuits Syst. Video Techn.*, vol.12, no.2, pp.122-129, 2002.
[6] V. Jain, B. Kimia and J. Mundy, Background modeling based on subpixel edges, *Proc. of IEEE International Conference on Image*, vol.6, pp.321-324, 2007.
[7] M. Yokoyama and T. Poggio, A contour-based moving object detection and tracking, *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp.271-276, 2005.
[8] B. Sugandi, H. Kim, J. K. Tan and S. Ishikawa, Tracking of moving persons using multi-camera employing peripheral increment sign correlation image, *ICIC Express Letters*, vol.1, no.2, pp.177-183, 2007.
[9] M. A. A. Dewan, M. J. Hossain and O. Chae, An adaptive motion segmentation for automated video surveillance, *EURASIP J. Adv. Signal Process*, pp.1-13, 2008.
[10] M. A. A. Dewan, M. J. Hossain and O. Chae, Background independent moving object segmentation for video surveillance, *IEICE Transactions*, vol.92-B, no.2, pp.585-598, 2009.
[11] D. J. Dailey, F. W. Cathey and S. Pumrin, An algorithm to estimate mean traffic speed using uncalibrated cameras, *IEEE Transactions on Intelligent Transportation Systems*, vol.1, no.2, pp.98-107, 2000.
[12] W. Hu, T. Tan, L. Wang and S. Maybank, A survey on visual surveillance of object motion and behaviors, *IEEE Transactions on Systems, Man and Cybernetics*, vol.34, pp.334-352, 2004.
[13] A. Yilmaz, O. Javed and M. Shah, Object tracking: A survey, *ACM Comput. Surv.*, vol.38, no.4, 2006.
[14] R. J. Radke, S. Andra, O. Al-Kofahi and B. Roysam, Image change detection algorithms: A systematic survey, *IEEE Transactions on Image Processing*, vol.14, pp.294-307, 2005.
[15] Z. B. Musa and J. Watada, Multi-camera tracking system for human motions in different areas and situations, *International Journal of Innovative Computing, Information and Control*, vol.4, no.5, pp.1213-1222, 2008.
[16] A. Lipton, H. Fujiyoshi and R. Patil, Moving target classification and tracking from real-time video, *IEEE Workshop on Applications of Computer Vision*, pp.8-14, 1998.
[17] D. Meyer, J. Denzler and H. Niemann, Model based extraction of articulated objects in image sequences for gait analysis, *IEEE Int. Conf. on Image Processing*, pp.78-81, 1997.
[18] B. K. P. Horn and B. G. Schunck, Determining optical flow, *Artificial Intelligence*, vol.17, pp.185-203, 1981.

[19] J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Int.*, vol.8, no.6, pp.679-698, 1986.

[20] Y. Ahn, K. Ahn and O. Chae, Detection of moving objects edges to implement home security system in a wireless environment, *Computational Science and Its Applications – ICCSA*, pp.1044-1051, 2004.

[21] G. Borgefors, Hierarchical chamfer matching: A parametric edge matching algorithm, *IEEE Trans. Pattern Anal. Mach. Int.*, vol.10, no.6, pp.849-865, 1988.

[22] V. S. Sunderam, G. D. van Albada, P. M. A. Sloot and J. Dongarra, Theoretical computer science and general issues, *Computational Science – ICCS 2005, Part I*, vol.3514, 2005.

[23] M. Murshed, M. H. Kabir and O. Chae, Moving object tracking – An edge segment based approach, *International Journal of Innovative Computing, Information and Control*, vol.7, no.7(A), pp.3963-3979, 2011.

[24] M. Bergtholdt, J. Kappes, S. Schmidt and C. Schnorr, A study of parts-based object class detection using complete graphs, *International Journal of Computer Vision*, vol.87, pp.93-117, 2010.

[25] L. Vincent and P. Soille, Watersheds om digital spaces: An efficient algorithm based on immersion simulations, *IEEE Trans. Pattern Anal. Mach. Int.*, vol.13, no.6, pp.583-598, 1991.