An Edge-based Approach for Robust Foreground Detection

Sebastian Gruenwedel, Peter Van Hese and Wilfried Philips

Ghent University TELIN-IPI-IBBT, Sint Pietersnieuwstraat 41, 9000 Gent, Belgium
Tel: +32 9 264 34 12, Fax: +32 9 264 42 95
sebastian.gruenwedel@telin.ugent.be

Abstract. Foreground segmentation is an essential task in many image processing applications and a commonly used approach to obtain foreground objects from the background. Many techniques exist, but due to shadows and changes in illumination the segmentation of foreground objects from the background remains challenging. In this paper, we present a powerful framework for detections of moving objects in real-time video processing applications under various lighting changes. The novel approach is based on a combination of edge detection and recursive smoothing techniques. We use edge dependencies as statistical features of foreground and background regions and define the foreground as regions containing moving edges. The background is described by short- and long-term estimates. Experiments prove the robustness of our method in the presence of lighting changes in sequences compared to other widely used background subtraction techniques.

Keywords: foreground detection, foreground edge detection, background subtraction, video surveillance, video processing

1 Introduction

Foreground/background segmentation is a crucial pre-processing step in many applications, aimed at the separation of moving objects (the foreground) from an expected scene (the background). Many techniques use this operation as part of their work flow. For instance, tracking algorithms may focus on foreground regions to detect moving objects and therefore speed up object-matching [13].
There are many techniques to detect moving objects in indoor and outdoor sequences [3]. Nevertheless, most of the techniques perform poorly when lighting changes suddenly. Especially in the case of indoor scenarios, there are problems to distinguish between foreground and background regions when sudden and/or partial lighting changes occur. Therefore, a robust detection of foreground objects under such circumstances is needed.
The Gaussian Mixture Model (GMM) method of [14] uses a variable number of Gaussians to model the color value distribution of each pixel as a multi-modal signal. This parametric approach adapts the model parameters to statistical

changes. As such it can adapt to lighting changes. However, this adaptation requires several frames during which the performance is generally very poor. Another very similar approach is presented in [5] wherein color and gradient information are explicitly modeled as time adaptive Gaussian mixtures.

The recently published ViBe [1] is a sample-based approach for modeling the color value distribution of pixels. The sample set is updated according to a random process that substitutes old pixel values for new ones. It exploits spatial information by using the neighborhood of the current pixel as well as by adaptation to lighting changes. This method is more robust to noise changes than GMM, as described in [1]. However, the GMM based method adapts poorly to fast local and/or global lighting changes due to the slow adaptation of the background model. The performance of ViBe already improved for such changes, but is still not robust enough. The adaptation to a lighting change involves several frames, but after a few changes in a short time period both methods loose their ability to distinguish between foreground and background. Therefore moving objects will no longer be detected and the performance is insufficient.

The method in [7] divides the scene in overlapping squared patches, followed by building intensity and gradient kernel histograms for each patch. The paper shows that contour based features are more robust than color features regarding changes in illumination. In [4], a region-based method describing local texture characteristics is presented as a modification of the Local Binary Patterns [8]. Each pixel is modeled as a group of adaptive local binary pattern histograms that are calculated over a circular region around the pixel. Similar to this approach is the method described in [12] which uses the texture analysis in combination with invariant color measurements in RGB space to detect foreground objects. The two aspects are linearly combined resulting in a multi-layer background subtraction method, which is modeled and evaluated similarly to GMM. These models are particularly robust to shadows.

In this paper we propose a new method to subtract foreground (FG) from background (BG) by detecting moving edges in real-time video processing applications, in particular for tracking. We use edge dependencies as statistical features of foreground and background regions and define foreground as regions containing moving edges, and background as regions containing static edges of a scene. In particular, we are interested in finding edges on moving objects. The proposed method estimates static edges which is in contrast to changes in intensity of GMM and ViBe.

The novelty is the background modeling which uses gradient estimates in x- and y-direction. The x and y components of the gradient are estimated independently using adaptive recursive smoothing techniques for each pixel. Based on the gradient estimates, detection of moving edges becomes feasible. An edge is defined as a sharp change in the image intensity function. We use thresholding on the current gradient estimates and our background modeling to obtain foreground edges. Edge detection in general includes the relationship to neighboring pixels and is in theory independent of lighting changes [2]. Even if in practice this is not the case, lighting changes only affect the edge strength; adaptation is

not needed and therefore the method copes better with local and global lighting changes.
We compare the results of the proposed method with the results of two state-of-the-art FG/BG segmentation techniques. To do so, we artificially fill the interior of moving objects by clustering edges and filling those clusters with a convex hull technique. The results are obtained from several indoor sequences in the presence of local and global lighting changes. In particular, we choose the Gaussian Mixture Model (GMM) [10] based method by [14] and the sample-based approach ViBe [1] as a comparison to the proposed method. We show that our method performs best in sequences under changes in illumination. As an evaluation measure we compare the position of moving people obtained by [6] to ground truth data.
This paper is structured as follows. In Section 2 the proposed method including the background model is explained. In Section 3 experimental results will be discussed in detail and we will show that our method performs best for the tested sequences in presence of lighting changes.

2 Background Subtraction using Moving Edges

As it is often the case, edges are detected by computing the edge strength, usually a first order derivative expression such as the gradient magnitude, and searching for local maxima. In our method, we define foreground as regions of moving edges and use first order derivatives in x- and y-direction as input treating each direction independently. We estimate the x and y component of the gradient per pixel over time using a recursive smoothing technique. The smoothing is applied with a low learning factor and estimates the background of a scene, further referred to as long-term background edge model. Due to the low learning factor, changes in the gradient estimates will be incorporated slowly by the background models. By comparing the background edge models to the recent gradient estimates, we might detect more edges than actually present because of the low learning factor. However, this situation is prevented using a second smoothing approach, referred to as short-term background edge model, based on recursive smoothing with a higher learning factor. The two models per direction are used jointly to obtain a foreground gradient estimate per direction, containing only regions where motion occurs in the image.
In Figure 1 the block scheme of our method is shown. First, we calculate the gradient estimates in x- and y-directions, represented by two matrices $G_{x,t}$ and $G_{y,t}$, for the input image of frame t using a discrete differentiation operator (e.g. Sobel operator). In the next step, we compare our long-term background edge models with the current gradient estimates for each direction and obtain two binary foreground masks, $F^l_{x,t}$ and $F^l_{y,t}$, using hysteresis thresholding with two thresholds T_{low} and T_{high}. The same procedure is done for the short-term models resulting in two binary masks, $F^s_{x,t}$ and $F^s_{y,t}$, using only the threshold T_{low}. The comparison per model is done using the differences between the background edge models in x- and y-direction and the x and y component of the gradient

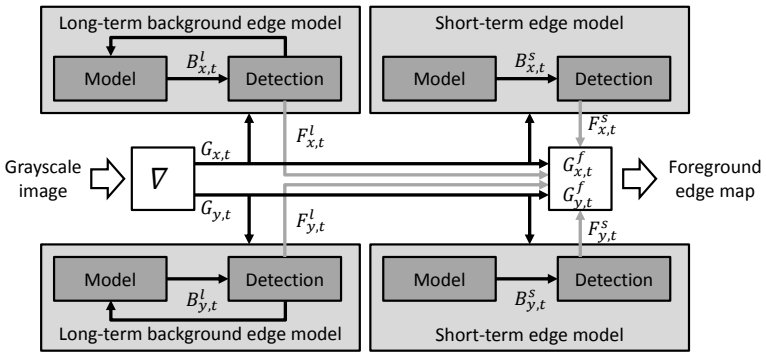


Fig. 1. The input image is passed to differentiation operator resulting in the x and y component of the gradient. The gradient is compared to our background models resulting in foreground gradient estimates.

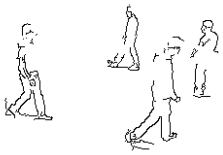
instead of the absolute value of differences, which results in a better detection of moving edges. We define $G^f_{x,t}$ and $G^f_{y,t}$ as the foreground gradient estimates in x- and y-direction, respectively. The foreground gradient estimate in x, $G^f_{x,t}$, contains the x component of the gradient, $G_{x,t}$, in foreground regions if, and only if, the binary masks $F^s_{x,t}$ and $F^l_{x,t}$ are one, otherwise zero values; and vice versa in y-direction. In the final step, edges are extracted from the foreground gradient estimates using a non-maximum suppression technique together with two thresholds T_{low} and T_{high}. The resulting moving edges of our method are exemplified in Figure 2.

The model updates are performed using the recursive smoothing (running average) technique [9] with two different learning factors α_s and α_l, for the short-term and long-term model, respectively. In order to meet the real-time criteria we choose the simplest form of exponential smoothing, i.e. the mean value is a cumulative frame-by-frame estimate. In summary, we use four different parameters for our method: α_s and α_l for both short-term and long-term models in x- and y-direction as well as T_{low} and T_{high}. A detailed discussion of the short-term and long-term models can be found in Section 2.1 and 2.2. We will only focus on the explanation of the x-direction since the calculations for the y-direction are analogous.

2.1 Short-term Model

The short-term models of x and y are responsible to smooth the x and y component of the gradient over a recent number of frames according to the learning factor α_s. The model is needed, in combination with the long-term model, to suppress noise and to robustly detect moving edges. The model update is performed using a recursive smoothing technique with the learning rate α_s, which is higher than the learning rate α_l of the long-term background edge models. We define the difference between the averaged gradient estimate, $B^s_{x,t}$, and the

(a)



(b)

Fig. 2. Segmentation result of an input frame (a) using the proposed method (b).

current gradient estimate, $G_{x,t}$, as $d^s_{x,t}(x,y) = G_{x,t}(x,y) - B^s_{x,t}(x,y)$ at location (x,y). Formally, the model $B^s_{x,t}$ is updated according to:

$$B^s_{x,t}(x,y) = B^s_{x,t-1}(x,y) + \alpha_s d^s_{x,t}(x,y) \tag{1}$$

where $\alpha_s \in [0,1]$ is the learning rate. The learning rate α_s is constant and usually around 0.1.
Frame difference is a special case of the short-term model with $\alpha_s = 1$ and the simplest case of motion detection; $\alpha_s < 1$ models the smoothing over a recent number of frames rather than the last one. To obtain the binary mask $F^s_{x,t}$, we threshold the difference between the $B^s_{x,t}$ and the gradient estimate $G_{x,t}$. Formally, we get

$$F^s_{x,t}(x,y) = \begin{cases} 1, & \left|d^s_{x,t}(x,y)\right| > T_{low} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $F^s_{x,t}$ represents a binary mask, specifying the presence of motion with 1 and otherwise 0 for each pixel. The threshold T_{low} is the same as in the long-term modeling.

2.2 Long-term Background Edge Model

The long-term model $B^l_{x,t}$ basically contains averaged gradient estimates over a long time period and therefore describes static edges in the background. The model uses the same running average technique as the short-term model, but with a very low learning factor $\alpha_l \in [0,1]$ (around 0.01). The difference between the long-term gradient estimate, $B^l_{x,t}$, and the current gradient estimate, $G_{x,t}$, is defined as $d^l_{x,t}(x,y) = G_{x,t}(x,y) - B^l_{x,t}(x,y)$ at location (x,y). The update is calculated as follows:

$$B^l_{x,t}(x,y) = \begin{cases} B^l_{x,t-1}(x,y) + \alpha_l d^l_{x,t}(x,y), & \text{if } F^s_{x,t}(x,y) = F^l_{x,t}(x,y) = 0 \\ B^l_{x,t-1}(x,y), & \text{otherwise} \end{cases} \tag{3}$$

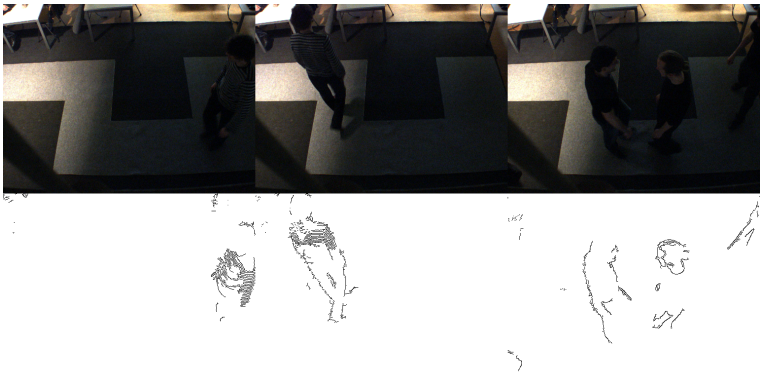


Fig. 3. Detected moving edges of the proposed method in a partly illuminated scene. The first row shows the input frames and the second row the segmentation results. Our method produces reliable results even in dark regions.

where $B^l_{x,t}(x,y)$ corresponds to the long-term model in x-direction in the t-th input frame at location (x,y). The model $B^l_{x,t}$ is selectively updated according to the binary masks $F^s_{x,t}$ and $F^l_{x,t}$, i.e., the model is only updated in regions where moving edges are not detected. This makes sure that moving edges are not included in the long-term model. The learning factor $\alpha_l \in [0,1]$ is constant and describes the adaptation speed of the long-term model.
Selective updating could cause a propagation of false detections in time because the model is not updated in region of moving edges. This situation is prevented using a second model, the short-term model.
The binary mask $F^l_{x,t}$ is determined by the comparison of the background model $B^l_{x,t}$ and the input gradient estimates $G_{x,t}$. Formally, we calculate the mask $F^l_{x,t}$ as follows:

$$F^l_{x,t} = \text{hyst}\left(\left|d^l_{x,t}\right|, T_{low}, T_{high}\right) \tag{4}$$

The resulting mask contains 1 for foreground and 0 for background regions for each pixel. The function $\text{hyst}(\cdot)$ corresponds to a hysteresis thresholding of the absolute value of the difference between long-term gradient estimate and current gradient estimate. All pixel values larger than T_{high} are immediately accepted as foreground and vice versa; values smaller than T_{low} are immediately rejected. Pixel values inbetween the two thresholds are accepted if they are in the neighborhood (8-connected) of a pixel that has a larger value than T_{high}.

2.3 Detection of Moving Edges

In the final step, moving edges are generated from the foreground gradient estimates $G^f_{x,t}$ and $G^f_{y,t}$. $G^f_{x,t}$ is found by setting foreground regions according to binary masks $F^s_{x,t}$ and $F^l_{x,t}$ to the gradient estimate in x-direction, $G_{x,t}$, and zero otherwise. In this stage we make sure that we take only foreground regions

into account. The calculation is defined as follows:

$$G^f_{x,t}(x,y) = \begin{cases} G_{x,t}(x,y), & F^s_{x,t}(x,y) = F^l_{x,t}(x,y) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $G^f_{x,t}$ contains the gradient estimate at location (x, y) and zero values otherwise.
To obtain thin edges, we calculate the edge map for time t using the non-maximum suppression technique like in many edge algorithms [2] by combining the foreground gradient estimate in x- and y-direction, $G^f_{x,t}$ and $G^f_{y,t}$. Non-maximum suppression searches for the local maximum in the gradient direction. As in Figure 2 already illustrated, our method produces edges describing moving objects.

3 Results & Discussion

We first tested the performance of the proposed method under different lighting conditions and visually compared the results to the Gaussian Mixture Model [14] and the ViBe technique [1].
In the second step, we performed an evaluation of these three background segmentation methods by using the foreground silhouettes from each method as input for constructing occupancy maps by Dempster-Shafer reasoning in a multi-camera network according to [6]. The soundness of the maps per time instance is then used as an evaluation measure for the different FG/BG methods. In particular, these maps are useful for monitoring the activities of people and tracking applications, with the intention to find a correct trajectory of a person.
The data set, we used for comparison, consists of indoor sequences which were captured by a network of four cameras (780x580 pixels at 20 FPS) with overlapping views in an 8.8m by 9.2m room. Recordings were taken for about one minute during which ground truth positions of each person were annotated at one second intervals. We also tested the proposed method on outdoor sequences with similar results.
For the sequences we use a fixed learning factor of $\alpha_s = 0.1$ for the short-term and $\alpha_l = 0.01$ for the long-term models. The proposed framework is not very sensitive to the learning factor α_l, provided that the factor is reasonable small (0.01 to 0.05). However, due to the fact that the short-term model is responsible for the smoothing of recent activities in the foreground, the learning rate α_s is important and specifies the adaptation speed to changes in the foreground. Usually the factor is about ten times bigger than α_l.

3.1 Visual Evaluation of the Proposed Method under Different Lighting Conditions

In the first step we compared the performance visually on three exemplary different sequences with evaluation measures of how well moving people are segmented

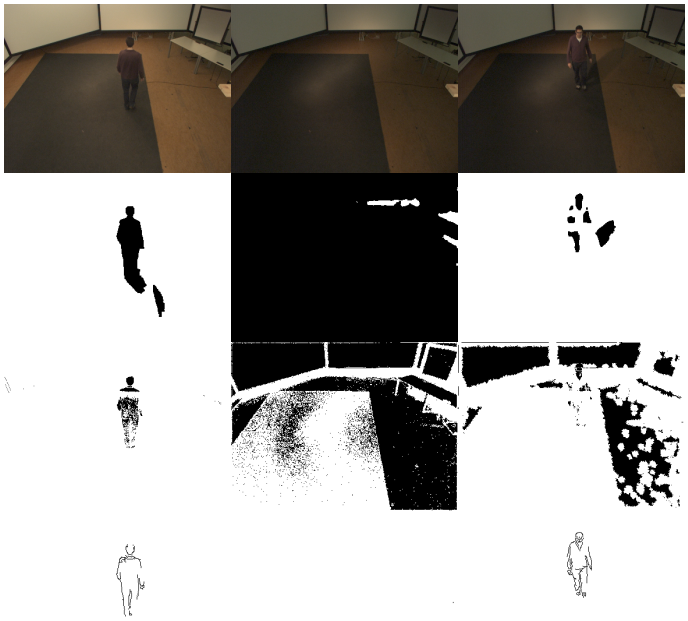


Fig. 4. Exemplary frames of a global lighting change. Black pixels correspond to foreground regions. First column: input frames; Second column: results of GMM [14]; Third column: results of ViBe [1]; Forth column: results of our proposed method. The second row shows the scene directly after a lighting change. Our method is not affected by this change.

for GMM and ViBe, and how well moving edges are detected on people. In Figure 3, the segmentation result of a partial illuminated sequence is shown. Our method produces reliable results even in dark regions, i.e., edges are still found on moving people. The results of our method are similar to those of change detection techniques, but differ favorably in the presence of lighting changes as shown in Figures 4 and 5.

In Figure 4 exemplary segmentation results of the whole sequence for a global lighting change are shown. The second column shows the scene directly after a lighting change. It is clearly visible that our method is not affected by this change. The detection of edges on the walking person is still reliable, even in poorly illuminated parts of the scene. GMM and ViBe suffer from the adaptation to the lighting change and especially ViBe fails to segment the person in the scene.

Figure 5 shows an example of global and local lighting changes. In this sequence, four people are moving around with the light changing at first globally and then locally in the scene. This makes it difficult to find a proper segmentation of moving people. The first column illustrates the results of all methods at the beginning

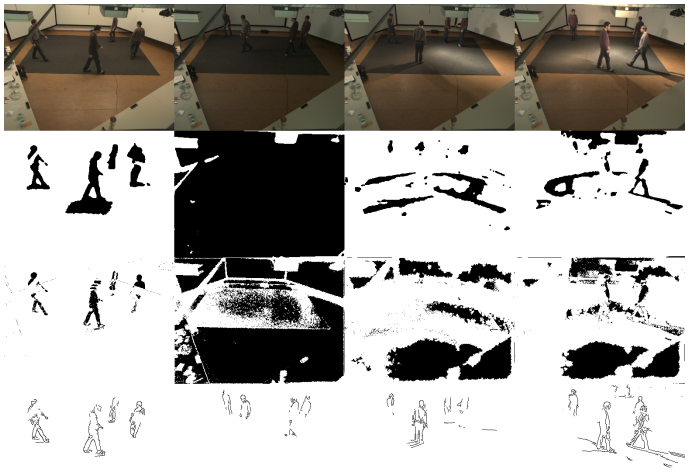


Fig. 5. Exemplary frames of a global and local lighting changes. Black pixels correspond to foreground regions. First row: input frames; Second row: results for GMM by [14]; Third row: results for ViBe by [1]; Forth row: results for the proposed method. The proposed method is less influenced by lighting changes.

of the sequence. In the second column a global illumination change occurred and GMM and ViBe suffer from this lighting change while our proposed method provides some edges on moving people. The third and fourth column contain local lighting changes. ViBe fails completely in this case because the adaptation to lighting changes is quite slow. Even GMM has problems to adapt to these changes and to find a good segmentation. Our method performs best in these cases and provides a good segmentation of moving edges. Due to the local lighting changes shadows are partially segmented by our method as depicted in the last column.

As shown in the example sequences, our method is less influenced by lighting changes and hence more robust. The results of our method under different light conditions are only affected in less detections of edges onto the objects or partial detection of shadows. Less edges are detected due to the poor lighting on the objects which results in too small intensity differences.

3.2 Numerical Evaluation of the Proposed Method for the Construction of Occupancy Maps

To quantitatively compare all described methods, we used an exemplary sequence which includes local and global lighting changes (example frames shown in Figure 5). We performed an evaluation based on the foreground silhouette from each method as an input for constructing occupancy maps by Dempster-Shafer reasoning in a multi-camera network [6]. This comparison is especially

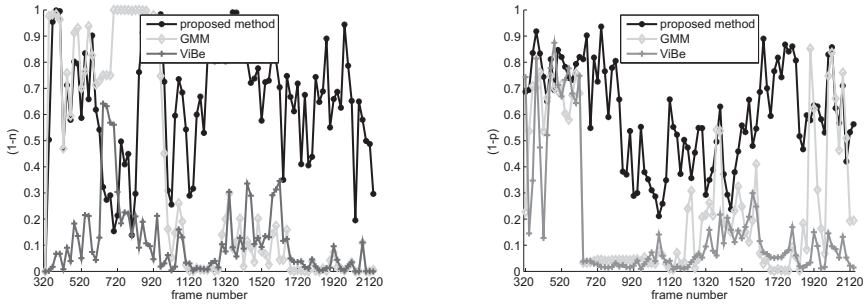


Fig. 6. Comparison of GMM, ViBe and the proposed method for each frame in the sequence. In (a) $(1-n)$ and in (b) $(1-p)$ for each method is shown. Higher values indicate better performance. The proposed method outperforms the other two methods.

of interest for tracking applications as it is a measure of how often the tracking might be lost. Due to the fact that edges cannot be compared directly with foreground masks of FG/BG segmentation techniques we clustered edges using a nearest-neighbor technique and combined them by a convex hull to represent silhouettes of moving people. The convex hull is constructed around a cluster of edges and and usually results in a sub-optimal solution to construct the silhouette of a person. A convex hull for a set of edge points is generally the minimal convex set containing these points. However, this is only used for comparison with FG/BG methods to construct an occupancy map.

We used occupancy maps (i.e. a top view of the scene) together with Dempster-Shafer reasoning, as explained in [6], to obtain the person's position in the scene. An occupancy map is calculated using different camera views and fusing foreground silhouettes onto the ground plane. In this sequence we have four different views of the scene and per second manually annotated ground truth positions of each person. For each occupancy map the positions of people were compared to ground truth data.

To evaluate the soundness of all maps per time instance we use two measures, n and p, as described in [11]: n represents a measure of evidence at a person's position (within a radius of $10cm$, $n = 0$ is the ideal case) and p as a measure of no evidence outside the positions ($p = 0$ is the ideal case). For p, we choose a radius of $70cm$ around the person's position. Those measures provide a reasonable evaluation of FG/BG methods, as stated in [11], e.g. for tracking applications. The ideal case for a method should be that $n = 0$ and $p = 0$, which means that all objects are detected and the evidence of a person is concentrated around the ground truth position.

In Figure 6 the evaluation over all 1800 frames ($90s$) is shown. The Figure 6 (a) denotes the measure $(1-n)$ and Figure 6 (b) $(1-p)$. The ideal case would be that for each frame the measure $(1-n)$ and $(1-p)$ is close to one. The results show that after lighting changes occur (frame 600), our method performs best

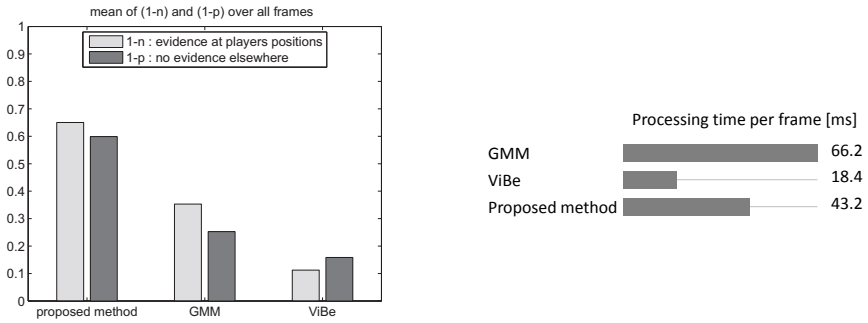


Fig. 7. Comparison of GMM, ViBe and the proposed method: (a) the mean of $(1-n)$ and $(1-p)$ over all frames, (b) the processing time of all methods calculated in full resolution (780x580 pixels). The proposed method clearly outperforms the other methods and is able to run in real-time.

for this sequence.
The mean of $(1-n)$ and $(1-p)$ is shown in Figure 7(a). Our method has a performance of 60%, which is the double of GMM. Althrough, for some frames the results are still not satisfying which is due to the fact that only half of the people are segmented, resulting in lower evidences of occupancy. However, GMM and ViBe fail almost completely in the presence of lighting changes for this sequence. The processing time of our method is higher than ViBe, but still better than GMM based methods (Figure 7(b)).
To sum up, our method performs best in the presence of lighting changes compared to GMM and ViBe which is due to the fact that our model is based on the detection of edges, which are much less influenced by lighting changes and therefore more robust.

4 Conclusion

In this paper, we presented a novel approach for background subtraction using moving edges. We showed that our method produces results similar to state-of-the-art foreground/background methods [14, 1], but performs much better in the presence of lighting changes. The parameters of our method do not need fine tuning (short-term and long-term learning factors) since the results are satisfying in a wide range of environments with fixed set parameters.
The problem of changing light conditions is still a critical issue for foreground segmentation techniques and needs further investigation; however, our proposed method based on edge information could solve this drawback and is a step towards the robustness against illumination changes. This edge-based approach can be used to model the lighting changes and thus help to find a better segmentation of foreground objects. A minor drawback of the proposed method are

the not yet fully light-insensitive thresholds; further exploration to automatically adapt the thresholds to the light changes is required. Furthermore, tracking approaches could make use of moving edges because edges are a common feature of choice in these applications.

References

1. Barnich, O., Van Droogenbroeck, M.: Vibe: a powerful random technique to estimate the background in video sequences. In: IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 945–948 (2009)
2. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8(6), 679–698 (1986)
3. Cristani, M., Farenzena, M., Bloisi, D., Murino, V.: Background subtraction for automated multisensor surveillance: a comprehensive review. Journal on Advances in Signal Processing 2010, 24 (2010)
4. Heikkila, M., Pietikainen, M.: A texture-based method for modeling the background and detecting moving objects. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28(4), 657–662 (2006)
5. Klare, B., Sarkar, S.: Background subtraction in varying illuminations using an ensemble based on an enlarged feature set. Computer Vision and Pattern Recognition Workshop 0, 66–73 (2009)
6. Morbee, M., Tessens, L., Aghajan, H., Philips, W.: Dempster-shafer based multi--view occupancy maps. Electronics Letters 46(5), 341 –343 (march 2010)
7. Noriega, P., Bernier, O.: Real time illumination invariant background subtraction using local kernel histograms. British Machine Vision Association (BMVC) (2006)
8. Ojala, T., Pietikainen, M., Harwood, D.: Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In: Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on. vol. 1, pp. 582–585. IEEE (1994)
9. Piccardi, M.: Background subtraction techniques: a review. In: IEEE International Conference on Systems, Man and Cybernetics. vol. 4, pp. 3099–3104. IEEE (October 2004)
10. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. 22, 747–757 (August 2000)
11. Van Hese, P., Grünwedel, S., Niño Castañeda, J., Jelaca, V., Philips, W.: Evaluation of background/foreground segmentation methods for multi-view occupancy maps. In: Proceedings of the 2nd international conference on positioning and context-awareness (PoCA-2011). p. 37 (2011)
12. Yao, J., Odobez, J.: Multi-layer background subtraction based on color and texture. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. pp. 1–8. IEEE (2007)
13. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. 38(4), 13 (2006)
14. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. vol. 2, pp. 28 – 31 Vol.2 (2004)