Background Modeling through Spatiotemporal Edge Feature and Color

Byeongwoo Kim[1], Adín Ramírez Rivera[2*], Oksam Chae[1], and Jaemyun Kim[1**]

[1] Department of Computer Science and Engineering, Kyung Hee University 1732, Deogyeong-daero, Giheung-gu, Yongin-si, Gyeonggi-do, 17104, South Korea {byeongwoo, oschaeg, jaemyunkim}@khu.ac.kr
[2] Institute of Computing, University of Campinas (UNICAMP), Campinas, Brazil
adin@ic.unicamp.br

Abstract. In this paper, we propose a new spatiotemporal edge feature for background modeling that can extract spatial and temporal (motion) features by considering the background model and current information. Previous work on background modeling considers mainly the spatial domain, which misses key temporal information. In our proposal, we create spatiotemporal edge features by using current and past background information by identifying the amount of change from past to present. By finding these differences, we can accurately detect the movement of objects that is more robust to noise and illumination variations. Moreover, our proposed background-modeling technique adapts to background changes that occur over time through a dynamic model update strategy. Additionally, we are enhancing the spatiotemporal edge features with color to maintain the characteristics of each other. Finally, we evaluated our proposed method on the publicly available CDNET 2012 dataset and compared with state-of-the-art methods. Our extensive evaluation and analysis show that our method outperforms previous methods on this dataset.

Keywords: Background Modeling · Spatiotemporal · Fusion

1 Introduction

Detection of foreground objects from videos is an initial step for many computer vision applications, like surveillance, activity recognition, human pose estimation, traffic flow estimation, etc. One of the widely used methods for this task is Background Subtraction (BS). Typically, this method requires to create a background model and then new frames from the videos are compared with that model to detect the foreground. Thus, it is important to generate a background model that contains the background information of the scene by including its different variations. If the background model includes consistent background information, then the foreground can be detected by comparing the current frame with the model. However, since the real video images contain various obstacles, such as illumination variations, shadows, waving trees, or flowing water, it is impossible to accurately detect the foreground by using a difference between RGB pixel values.

Since there are several sources of movement within a video, we have to decide which of these situations will be detected as foreground. Therefore, the goal of background modeling is to detect the motion of the moving objects by accepting non-interesting motions (e.g., the waving trees, flowing water, and illumination variations) as a moving background. To detect the foreground, it is common to extract spatial features from the image and compare them. Therefore, there are extensive studies on spatial descriptors for background modeling.

One of the typical background modeling techniques is texture-based background modeling. Local Binary Pattern (LBP) [3] compares the intensity between the center pixel and neighboring pixels and encode them in binary code. It is robust to illumination variations but is very sensitive to noise. Local Binary Similarity Pattern (LBSP) [1] encodes similarity rather than direct intensity comparison with adjacent pixels to solve the LBP problem. However, it is sensitive to shadows and background variations and degrades performance in flat regions. Local Binary Similarity Segmenter (LOBSTER) [9] tried to solve the problem by using both LBSP

* Funded in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under grant No. 307425/2017-7.
** Corresponding Author

and color information together alongside a random background update strategy based on Sample Consensus (SACON) [12]. Furthermore, the Self-Balanced Sensitivity Segmenter (SubSENSE) [10] attempted to adapt quickly to changes by dynamically adjusting its internal parameters. However, it is still sensitive to noise and when the color of the background and foreground are similar, and then its performance degrades in flat regions.

Edge-based background modeling is robust to illumination variations but sensitive to noise. Also, edges in the same position in continuous frames are not always with the same magnitude and direction. To solve this problem, an edge-segment-based method was proposed [7, 8]. It uses edge magnitude and shape information by connecting adjacent edges. However, it is difficult to accurately detect the inside of the foreground due to a lack of edge information in flat regions. Local Hybrid Pattern (LHP) [5] tried to solve this problem by using edge and color information together, but using only one direction did not show characteristics in various directions. Local Top Directional Pattern (LTDP) [6] uses the compass masks instead of pixel information to accurately extract changes of an edge in different directions and encodes only the direction with the largest edge response. Therefore, it can extract straight lines, edge, and other shapes.

Pixel-based background modeling uses the intensity and color information of each pixel. It is sensitive to changes caused by the appearance of new objects, but also sensitive to illumination variations. This method cannot capture the background variations. Mixture of Gaussians (MoG) [11] used k Gaussian-models to solve this problem. However, the disadvantage of MoG is that it is difficult to determine the value of k between the fast and slow variations. Pixel Based Adaptive Segmenter (PBAS) [4] creates the background model based on the pixel intensity recorded recently. PBAS determines the background and foreground with different thresholds and uses a random update strategy. Moreover, the thresholds are dynamically adjusted to show high performance.

In this paper, we use the spatiotemporal edge feature and color information based on SuBSENSE to detect the foreground. This can accurately detect pixel changes due to the existence of moving objects. The spatiotemporal edge feature is the average of the differences between the pixels of the current frame and the pixels of the past frames in the background model. Besides, we use five spatiotemporal edge features in different directions by comparing the pixels in the 7×7 block in the current frame and the pixels of the background model. Therefore, it is robust to noise and can detect the object's change in any direction. LOBSTER and SuBSENSE use LBSP and color information to detect different foreground and use the `and` operator to detect the foreground again. However, we try to use both the spatiotemporal edge feature and color simultaneously using a fusion method that uses a sigmoid graph with the edge magnitude in the x-axis.

2 Spatiotemporal Edge Feature

Previous background modeling methods create features with information from a single frame and within consecutive frames. Then, the created features are compared with those in the background model to detect the foreground. However, features created with information extracted in this manner are not robust enough to accurately detect pixel variations due to the foreground and moving background. Moreover, it is difficult to create features with accurate information in flat or complex regions. We present a solution to this problem by proposing a spatiotemporal edge feature that is generated based on the change between the current frame and the stable background information held in the background model.

We propose the spatiotemporal edge feature that by jointly modeling the appearance and motion of the objects in the frame obtains higher accuracy when detecting moving objects. Given the neighborhoods of two pixels p and q, their difference is

$$D(\mathcal{N}_p, \mathcal{N}_q) = \frac{1}{9} \sum_{\substack{i \in \mathcal{N}_p, \\ j \in \mathcal{N}_q}} |i - j|, \tag{1}$$

where i and j are parallel indexed pixels over the neighborhoods $\mathcal{N}_p$ and $\mathcal{N}_q$ (of size 3×3) centered at pixels p and q, respectively, i.e., i and j are the values of their respective neighborhoods that are traversed at parallel locations. The differences are performed channel-wise. Hence, the final element will contain the same amount of channels as the original pixel (these were omitted for brevity). We are computing the difference between a neighborhood centered at pixel p, $\mathcal{N}_p$, and subtracting it to another neighborhood centered at pixel q, $\mathcal{N}_q$. We define the feature in terms of neighborhoods since we take advantage of our background model, instead

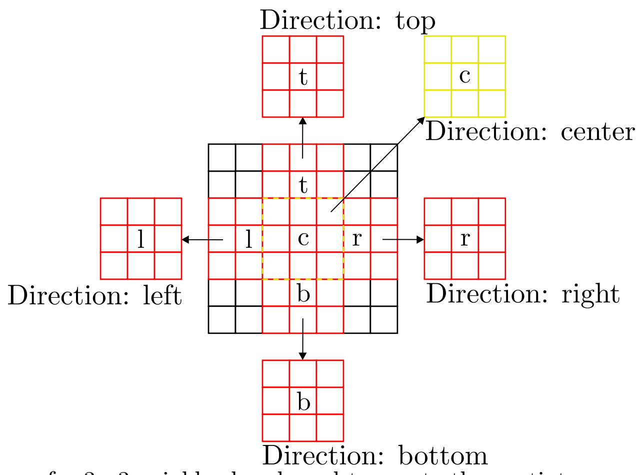

Fig. 1. Directions of a 3×3 neighborhood used to create the spatiotemporal edge features.

of densely computing the features for every frame. To obtain robust motion-features, we use five directions to get variations of the current neighborhood at the pixel of interest. Hence, for every pixel p and given a neighborhood $\mathcal{N}_q$ for comparison, we define a set of features

$$\mathcal{F}(p, \mathcal{N}_q) = \{D(\mathcal{N}_r, \mathcal{N}_q) : r \in \mathcal{D}(p)\} \tag{2}$$

based on five directions defined by

$$\mathcal{D}(p) = \{(p_x, p_y), (p_x \pm \Delta_x, p_y), (p_x, p_y \pm \Delta_y)\}\,, \tag{3}$$

where p is the pixel's position with horizontal and vertical components p_x and p_y, respectively, and where Δ_x and Δ_y are the displacements on each axis. In our definition of 3×3 blocks, we used a displacement $\Delta_x = \Delta_y = 2$ to minimize the overlap between the blocks, and still maintain consistency over the regions. We show an illustration of the five directions in Fig. 1.

3 Background Modeling

Model Definition. We model the scene pixel-wise through a set of codebooks that represent the local context of the pixel by including its spatiotemporal and color information. Let $\mathcal{B}_i(p)$ be the ith background model at pixel p's position as the tuple of the spatiotemporal edge features at p, $\mathcal{F}(p, \mathcal{N}^*)$, and the current pixel's neighborhood information, $\mathcal{N}_p$, such as

$$\mathcal{B}_i(p) = (\mathcal{F}(p, \mathcal{N}^*), \mathcal{N}_p)\,. \tag{4}$$

Note that we need a reference neighborhood $\mathcal{N}^*$ to compute the features for the given pixel. Given a match with the jth background model at pixel p (cf. Section 4), the corresponding neighborhood $\mathcal{N}^* \in \mathcal{B}_j(p)$ will be our reference neighborhood for the model. Moreover, the use of past background models as reference points for the features, instead of frame-to-frame features, creates a more robust representation of the changes between the current frame and the model. For initialization, we use a background frame that serves as reference neighborhoods for the first frame of the sequence. The whole background model for every pixel p is the set $\mathbb{B}(p) = \{\mathcal{B}_i(p)\}_{i=1}^{N}$ for a set number of codebooks N. In our proposal, we maintain a similar number of models per pixel as previous works [10], i.e., $N = 50$, that is supported by our experimental results on parameter selection. We illustrate the model in Fig. 2.

Model Matching. If a pixel is matched against $M_{\min}$ or more models within its set of codebooks for that particular pixel, then we consider it as background. Otherwise, it is considered as foreground. If the pixel is considered background, its information is used to randomly replace one of the N models. We performed an analysis of the parameters settings of our method in Section 5.1.

To match a pixel from the incoming frames with its set of models, we measure the distance of its features against each model. If the distance is close enough (according to a threshold, see below for details on their

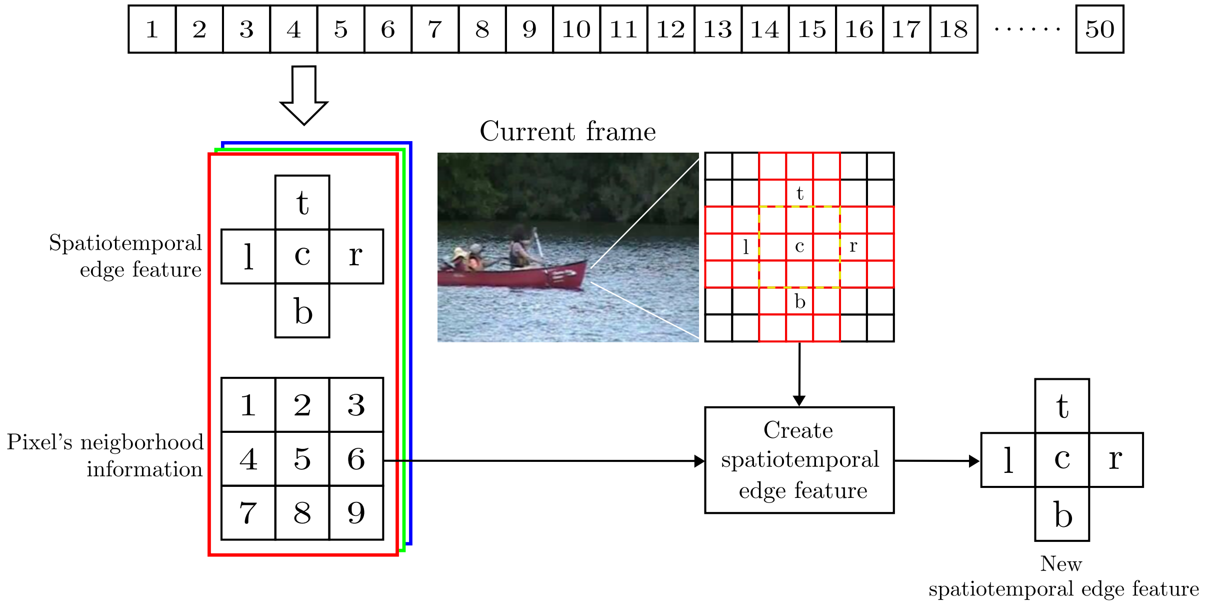
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 50
Current frame
Spatiotemporal edge feature
t
l c r
b
Pixel's neigborhood information
1 2 3
4 5 6
7 8 9
Create spatiotemporal edge feature
New spatiotemporal edge feature

Fig. 2. The background model consists of the five-direction spatiotemporal edge features and the pixels within the 3×3 block. Given a new frame, the new five-direction spatiotemporal edge features are created using nine pixels of the 3×3 block in the background model.

computation), we consider it a match. First, we compute the feature of the current frame at each pixel using the background models at its location as a reference. That is, at pixel p, for one of its background models, we use its neighborhood $\mathcal{N}^* \in \mathcal{B}(p)$ to compute the features of the current pixel $C = \mathcal{F}(p, \mathcal{N}^*)$. Then, we compare the distance against the background model's features $B = \mathcal{F}(\cdot, \cdot) \in \mathcal{B}(p)$ by

$$d_f(p, \mathcal{B}) = d_f(C, B) = \frac{1}{5} \sum_{\substack{c \in C \\ b \in B}} \sum_{n=1}^{3} \left| c[n] - b[n] \right|, \tag{5}$$

where the difference between the codes is performed channel-wise (n), and the codes are selected from the same direction in both sets C and B (the indexing was omitted for brevity). Similarly, we compute the difference between the color of the current pixel, $c_c = \mathcal{N}_p(\mathbb{c})$, where $\mathbb{c}$ represents the center position of the neighborhood $\mathcal{N}_p$, and the background color $c_b = \mathcal{N}_b(\mathbb{c})$, where $\mathcal{N}_b \in \mathcal{B}(p)$ is the neighborhood information stored on the background model, by

$$d_c(p, \mathcal{B}) = d_c(c_c, c_b) = \sum_{n=1}^{3} \left| c_c[n] - c_b[n] \right|, \tag{6}$$

where a color comprises three channels: R, G, and B. The range for both distances is within 0–765.

In contrast to existing methods [10] that use intersection between different detection maps, we propose to rely on a distance fusion method defined by

$$d^*(p, \mathcal{B}) = \alpha(p) d_f(p, \mathcal{B}) + (1 - \alpha(p))\, d_c(p, \mathcal{B}), \tag{7}$$

where α is determined by a saturated edge magnitude of the current pixel to combine d_f and d_c, $\mathcal{B}$ is one of the N models for the pixel p. Given the edge magnitude $e(p)$ at pixel p's position, we compute the scaling parameter by

$$\alpha(p) = \frac{0.8}{1 + \exp(-6e(p) + 2.8)} + 0.2, \tag{8}$$

where the parameters of the function define a minimum of 0.2 and a maximum of 0.8 for the sigmoid function. (These values give soft step function.) This is to prevent drastic changes in regions where feature or color is unilaterally strong. Hence, we allow a mixture of both distances within those limiting proportions.

Model Update. If the current pixel is determined to be background, the information of the current pixel is updated into the background model by randomly replacing one of its N models at the pixel's position.

Similarly, a random adjacent pixel is selected too, and one of its models is also randomly updated. This random update is similar to the SuBSENSE method [10].

4 Foreground Detection

We do the foreground detection at two steps based on our match confidence. At the first step, we compare the spatiotemporal edge feature (5) and color information (6) independently, and select pixels that we consider foreground with high confidence for both features. Otherwise, we use the fused information (7) to make an informed decision.

We define our foreground function (1 is foreground, 0 is background) as

$$F\left(p, \mathbb{B}\left(p\right)\right) = \begin{cases} 1 & \exists_{M_{\min}} \mathcal{B} \in \mathbb{B} : d_f(p, \mathcal{B}) > T_f^H(p) \text{ and } d_c(p, \mathcal{B}) > T_c^H(p), \\ 0 & \forall \mathcal{B} \in \mathbb{B} : d_f(p, \mathcal{B}) < T_f^L(p) \text{ and } d_c(p, \mathcal{B}) < T_c^L(p), \\ 1 & \exists_{M_{\min}} \mathcal{B} \in \mathbb{B} : d^*(p, \mathcal{B}) > T_*(p), \end{cases} \tag{9}$$

where $\mathbb{B}(p)$ corresponds to the set of models for the pixel p, and $\exists_{M_{\min}}$ represents the existence of at least $M_{\min}$ elements. For the spatiotemporal edge features, we use two thresholds (high and low) $T_f^H(p)$ and $T_f^L(p)$, respectively, per pixel p. Similarly, for the color features we use $T_c^H(p)$ and $T_c^L(p)$. For the fused distance, we have a singular threshold $T_*(p)$. The first two cases correspond to the certain cases in which both features agree on foreground or background by been higher or lower than its respective thresholds for one or all of the models, respectively. The third case is when there is a disagreement between the features (we omit the negation of the two previous conditions for brevity). Consequently, we use the fused distance to reach a consensus.

The thresholds reflect the tails on the distribution of distances that represent, either, foreground or background at the seen contexts per pixel. However, defining the threshold that best divides them is not trivial. Instead of setting the thresholds, we set the proportion of learned distributions of these distances. That is, given a prior distribution (σ^0) we maintain and update a running standard deviation each frame. Let $\sigma^t(p)$ be the standard deviation at time t and pixel p. We define its update as

$$\sigma^t(p) = \beta d(p) + (1 - \beta)\sigma^{t-1}(p), \tag{10}$$

were β represents the learning rate for the distributions, and $d(p)$ the distance that corresponds to the distribution of σ. We set $\beta = 0.05$ for a quick adaptation in the first 100 frames. After that, we set $\beta = 0.01$ to adapt slower to sequence the changes. We will maintain three parameters, one per distance (5, 6, and 7), namely, σ_f^t, σ_c^t, and σ_*^t at each time frame (we omit the frame t henceforth for brevity). Each distribution is updated (10) throughout the sequence to represent the changes in the scene.

We define the thresholds based on a factor of how far is the distance from the mean (based on standard deviations) by

$$T_{\{f,c\}}^{\{H,L\}}(p) = \sigma_{\{f,c\}}(p) F_{\{f,c\}}^{\{H,L\}}, \tag{11}$$

$$T_*(p) = \sigma_* F_*, \tag{12}$$

where $F_{\{f,c\}}^{\{H,L\}}$ and F_* are five the factors, one per threshold. We found the best factors experimentally, cf. Section 5.1.

5 Experimental Results

To evaluate the performance of the proposed approach, we use F_1 measure [5, 6] as a standard evaluation metric. We conduct background subtraction experiments on the Change Detection dataset 2012 (CDNET) [2]. This dataset is considered as challenging since the videos have diverse real-world scenes captured by CCTV surveillance cameras, and are categorized into six groups: baseline, camera jitter, dynamic background, intermittent object motion, shadow, and thermal. The dataset provides pixel-wise ground-truth information for all frames of videos.

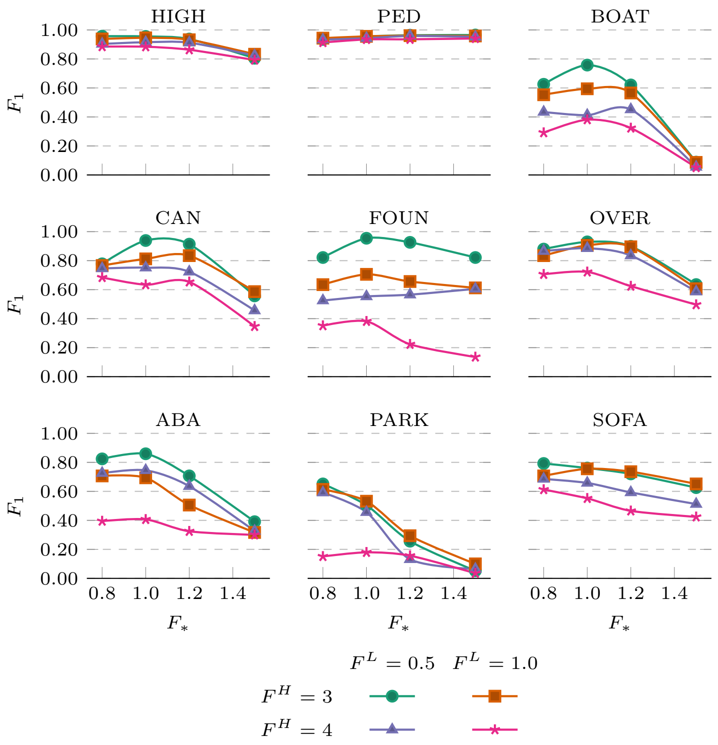

Fig. 3. The effect of varying the factors F^H, F^L, and F_* on the performance of the proposed method measured by F_1 on the CDNET database.

Table 1. Mean of F_1 measure by changing the different factor parameters.

F_*	(F^H, F^L)			
	$(3, 0.5)$	$(4, 0.5)$	$(3, 1.0)$	$(4, 1.0)$
0.8	0.807 ± 0.113	0.744 ± 0.139	0.712 ± 0.172	0.554 ± 0.269
1.0	$\mathbf{0.846 \pm 0.151}$	0.767 ± 0.151	0.702 ± 0.197	0.564 ± 0.252
1.2	0.772 ± 0.229	0.709 ± 0.224	0.645 ± 0.256	0.508 ± 0.278
1.5	0.549 ± 0.318	0.528 ± 0.303	0.487 ± 0.305	0.391 ± 0.315

In our evaluations, we make use of nine videos taken from three categories: Highway (HIGH), Pedestrians (PED), Boats (BOAT), Canoe (CAN), Fountain (FOUN), Overpass (OVER), Abandoned Box (ABA), Parking (PARK), and Sofa (SOFA). It should be noted that videos along with moving objects contain several other complications affecting the performance of the detection: illumination variations in the frames of HIGH and PED; dynamic backgrounds like waving in trees and water in BOAT, CAN, FOUN, and OVER; as well as sudden changes in background, and stopping of moving objects in ADA, PARK, and SOFA.

5.1 Parameter Setting

There are several important parameters of the proposed approach that should be set appropriately to expect reliable performance. We consider parameters related to the thresholds mentioned in Section 4. We perform experiments in various cases with different values to determine these best parameters.

Threshold Parameters. Through empirical studies, we attempt to find the optimal values of $F_{\{f,c\}}^{\{H,L\}}$ and F_* by considering the effect of the threshold on the performance. In the experiments, the factors were grouped by high and low, such as $F_f^H = F_c^H = F^H$ and $F_f^L = F_c^L = F^L$, to simplify the parameter search. Our search space was $F^H \in \{3, 4\}$, $F^L \in \{0.5, 1\}$, and $F_* \in \{0.8, 1.0, 1.2, 1.5\}$. Fig. 3 shows the F_1 measure of each dataset according to the different factors in 16 cases. We show the average of all F_1 measures in the Table 1. Thus, we found the best configuration at $F^H = 3$, $F^L = 0.5$, and $F_* = 1.0$.

From these experiments we can see that if F^H or F^L is too high the foreground regions are reduced (false negatives increase). Conversely, if F^H and F^L are too low, dynamic backgrounds and shadows appear as

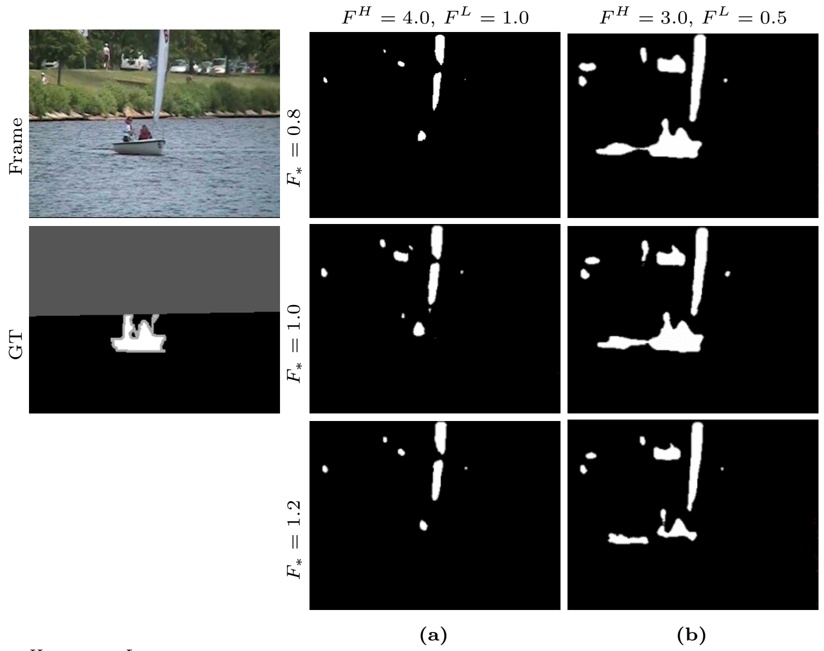

Fig. 4. Example of F^H and F^L effects in the BOAT dataset. (a) Detections when the foreground is not correctly detected (too many false negatives). And (b) detections when the foreground is over detected (too many false positives).

Table 2. Mean of F_1 measure by changing the different background model parameters.

N	$M_{\min}$		
	2	3	4
30	0.791 ± 0.161	0.819 ± 0.166	0.809 ± 0.163
40	0.800 ± 0.165	0.824 ± 0.161	0.819 ± 0.165
50	0.812 ± 0.163	$\mathbf{0.850 \pm 0.141}$	0.838 ± 0.140
60	0.804 ± 0.175	0.838 ± 0.160	0.830 ± 0.153

foreground (false positives increase). For example, when $F^H = 4$ and $F^L = 1$ the foreground is not detected properly, and when $F^H = 3$ and $F^L = 0.5$ the foreground is over detected, as shown in Fig. 4.

Background Model Parameters. We also conducted experiments to understand the effect of background model parameters. Table 2 shows the F_1 measures of the average of the CDNET sequences while varying N and $M_{\min}$ in 12 cases. We set our search space as $N \in \{30, 40, 50, 60\}$ and $M_{\min} \in \{2, 3, 4\}$. When N is too large there is a drop in performance due to the large model size. Conversely, if it is too small, the background cannot be modeled exactly. In the case when $M_{\min}$ is too large, it is easy to detect the exact background, but if it is too small, the foreground can be detected as background. We selected $N = 50$ and $M_{\min} = 3$ as the optimal configuration.

Table 3. Quantitative evaluation based on F_1 measure of different features on our background modeling setup.

Methods	LBP	LDP	LBSP	Proposed
HIGH	0.914	0.949	0.944	**0.955**
PED	0.907	0.953	0.954	**0.955**
BOAT	0.656	0.636	0.693	**0.758**
CAN	0.756	0.884	0.792	**0.939**
FOUN	0.905	0.906	0.944	**0.954**
OVER	0.806	0.793	0.857	**0.929**
ABA	0.656	**0.879**	0.849	0.859
PARK	0.405	**0.528**	0.444	0.504
SOFA	0.695	0.732	0.742	**0.761**
Avg.	0.744	0.807	0.802	**0.846**

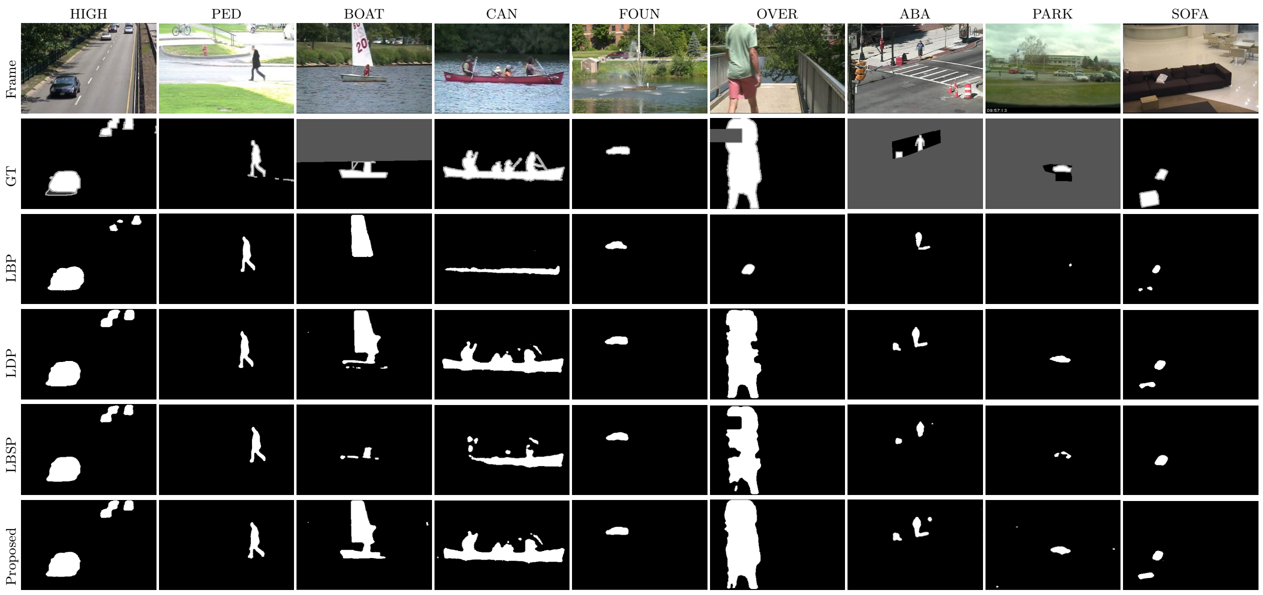

Fig. 5. Detection examples on the sequences of CDNET when using different features on the proposed setup. GT stands for ground truth.

Table 4. Quantitative evaluation by comparing F_1 measures of several methods on CDNET.

Methods	ViBe	PBAS	GMM	SuBSENSE	Proposed
HIGH	0.855	0.945	0.924	0.944	**0.955**
PED	0.808	0.936	0.954	0.954	**0.955**
BOAT	0.433	0.361	0.729	0.693	**0.758**
CAN	0.779	0.720	0.882	0.792	**0.939**
FOUN	0.714	0.936	0.803	0.944	**0.954**
OVER	0.746	0.793	0.872	0.857	**0.929**
ABA	0.614	0.690	0.539	0.849	**0.859**
PARK	0.388	0.174	**0.749**	0.444	0.504
SOFA	0.546	0.738	0.645	0.742	**0.761**
Avg.	0.654	0.699	0.789	0.802	**0.846**

5.2 Performance Analysis

To demonstrate the efficiency of proposed features, we compare our proposed spatiotemporal edge feature against well-know descriptors, such as LBP, LDP, and LBSP, by using such features instead of our proposal in our setup. Table 3 shows the F_1 measures on the different categories. The results demonstrate that our spatiotemporal edge features outperform others in most of the experiments. Although LDP shows a better result in ABA and PARK categories, the proposed approach provides comparable results with more stable performance, as shown by the higher average F_1 measure.

Along with quantitative results, we also provide qualitative results in Figs. 5 and 6. In Fig. 5, we can see over-detection due to the sensitivity to pixel illumination changes (i.e., shadows) in ABA and PARK videos.

Fig. 6 is a comparison of the proposed method with the other methods. Additionally, we show a quantitative comparison in Table 4. In PARK, F_1 measures of GMM is higher, but the proposed method is better overall.

6 Conclusion

In this paper, we propose a spatiotemporal edge features that take advantage of both spatial and temporal domains. This feature is robust to noise while being responsive to detect pixel variations occurring due to foreground appearance changes. We combine the spatiotemporal edge and color features to improve the prediction of foreground based on a set of adaptive thresholds. Thus, the foreground is detected by complementing the weak points of each other feature. Besides, internal parameters can be adjusted dynamically to quickly adapt to illumination or background variations. This shows that the overall proposed method provides better performance in different situations.

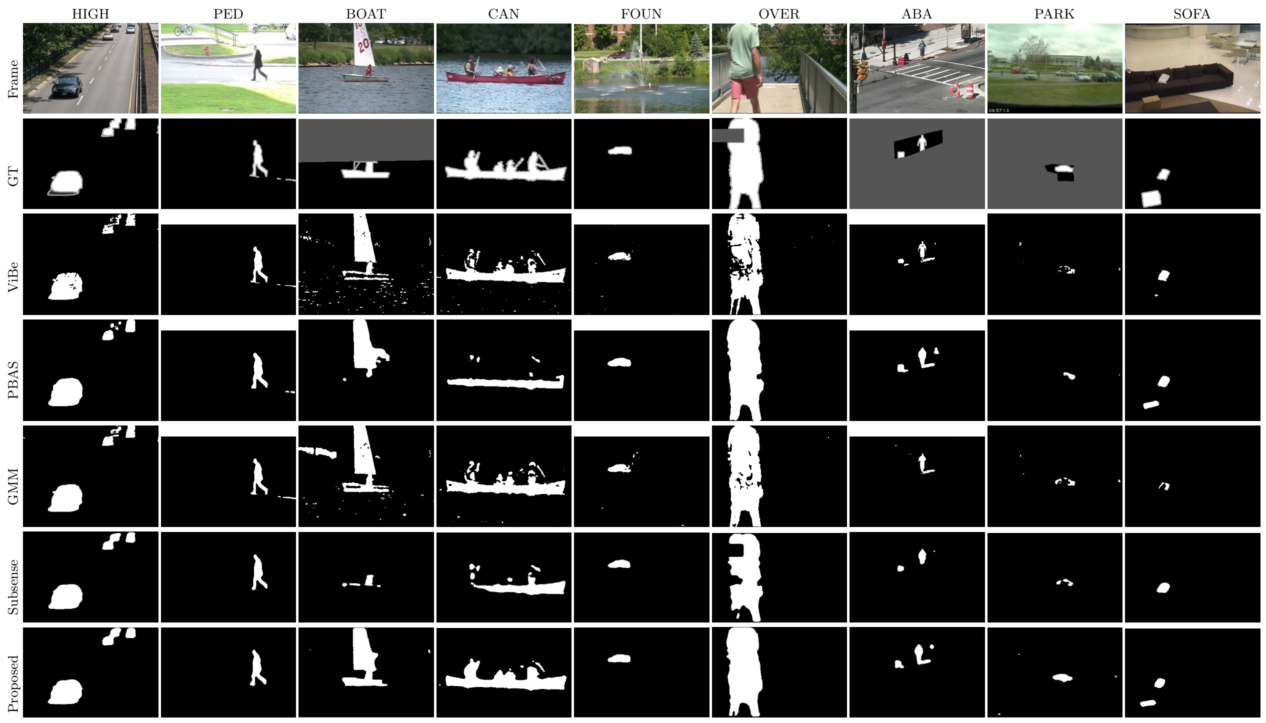

Fig. 6. Detection examples on the sequences of CDNET when comparing other methods. GT stands for ground truth.

References

1. Bilodeau, G.A., Jodoin, J.P., Saunier, N.: Change detection in feature space using local binary similarity patterns. In: International Conference on Computer and Robot Vision, pp. 106–112 (2013)
2. Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J., Ishwar, P.: Changedetection.net: A new change detection benchmark dataset. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8, IEEE (2012)
3. Heikkila, M., Pietikäinen, M.: A texture-based method for modeling the background and detecting moving objects. IEEE transactions on pattern analysis and machine intelligence **28**(4), 657–662 (2006)
4. Hofmann, M., Tiefenbacher, P., Rigoll, G.: Background segmentation with feedback: The pixel-based adaptive segmenter. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, pp. 38–43, IEEE (2012)
5. Kim, J., Ramírez Rivera, A., Ryu, B., Chae, O.: Simultaneous foreground detection and classification with hybrid features. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3307–3315 (2015)
6. Md Rifat, A., Farkhod, M., Oksam, C., Jaemyun, K.: Background subtraction based on fusion of color and local patterns. In: Asian Conference on Computer Vision, pp. 214–230 (2019)
7. Murshed, M., Ramírez Rivera, A., Chae, O.: Statistical background modeling: an edge segment based moving object detection approach. In: 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 300–306, IEEE (2010)
8. Ramírez Rivera, A., Murshed, M., Kim, J., Chae, O.: Background modeling through statistical edge-segment distributions. IEEE Transactions on Circuits and Systems for Video Technology **23**(8), 1375–1387 (2013)
9. St-Charles, P.L., Bilodeau, G.A.: Improving background subtraction using local binary similarity patterns. In: IEEE Winter Conference on Applications of Computer Vision, pp. 509–515, IEEE (2014)
10. St-Charles, P.L., Bilodeau, G.A., Bergevin, R.: SuBSENSE: A universal change detection method with local adaptive sensitivity. IEEE Transactions on Image Processing **24**(1), 359–373 (2015)
11. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 246–252, IEEE (1999)

12. Wang, H., Suter, D.: A consensus-based method for tracking: Modelling background scenario and foreground appearance. Pattern Recognition **40**(3), 1091–1105 (2006)