Edge Shape Pattern for Background Modeling based on Hybrid Local Codes

Seokjin Hong[1], Jaemyun Kim[1], Adín Ramírez Rivera[2], Gihun Song[1] and Oksam Chae*[1]

[1]Department of Computer Science and Engineering, Kyung Hee University

1732, Deogyeong-daero, Giheung-gu, Yongin-si, Gyeonggi-do, 17104, South Korea

{paperwalls, sense21c, gihunsong, oschae}@khu.ac.kr

[2]Vergara 432, Escuela de Informática y Telecomunicaciones, Universidad Diego Portales, Santiago, Chile

adin.ramirez@mail.udp.cl

Abstract

In this paper, we propose a novel edge descriptor method for background modeling. In comparison to previous edge-based local-pattern methods, it is more robust to noise and illumination variations due to the use of principal gradient information in a local neighborhood. For the background modeling problem, we combined the proposed method with the Local Hybrid Pattern and experimented with an adaptive-dictionary-model based background modeling method. We show in the quantitative evaluations that the proposed methods is better than other local edge descriptors when applied to the same framework. Furthermore, we show that our proposed method is more powerful than other state of the art methods on standard datasets for the background modeling problem.

1. Introduction

In video streams captured from a stationary camera, foreground detection is often one of the essential tasks in many video surveillance systems. Background modeling methods are commonly used in research as the first step to extract foreground. The main idea of background modeling is to compare the current frame with a reference background model. Despite extensive research related to background modeling, challenges still remain in this area because of non-stationary backgrounds. Moving backgrounds, like waving trees, illumination variation, the addition or removal of foregrounds, cause non-stationary backgrounds problems. Camera jitter and noise caused by sensor are the reason that a stable background model is hard to generate.

Pixel-based methods are often used in research to represent video streams by using the information of pixels, such as color or intensity. Maintaining the single background model, like average intensity of pixels, is the simplest way to detect foregrounds [12]. However, this simple approach cannot handle moving background because of its unimodal models (*i.e.*, can model only one condition). Some researchers proposed multimodal models to overcome this limitation. Mixture of Gaussians [14] is the most famous multimodal model. This method can represent moving background, like waving trees by using k Gaussians to model each pixels. But it depends on the definition of the k values. Heikkilä and Pietikäinen proposed a texture-based background modeling method [4]. This method uses Local Binary Pattern (LBP) [11] instead of pixel information. The advantage of texture-based method is its robustness to illumination variations. However, LBP tends to be sensitive to noise, because a slight variation of the neighboring pixels can cause entirely different texture patterns. Also, this problem often occurs in flat regions. Xue *et al*. [16] proposed hybrid center symmetric local pattern based background modeling methods to overcome this problem. It works better than LBP, but this method detects foreground coarsely because it models background as blocks instead of pixels. Researchers proposed a novel method which combines texture-based and pixel-based methods to get advantages of both worlds [2, 17, 18]. However, these methods need more computer resources, like memory, to maintain a more complex background model.

Edge-based methods represent the background model by using location of edge pixels. Edges are more robust to illumination variations than pixel intensity. However, it still has problems. In video streams, edge positions may not exactly be the same at each consecutive frames. Also shape and length of the edge may changed because of noise. To overcome the problem of edge-based methods, edge-segment-based methods were proposed [6]. They take the advantage of the edge existence and shape information by concatenating adjacent edges. However, fundamental problems of the edge-based methods can be inherited. Statistical edge-segment-based methods solve the edge-variation problem by accumulating the edge position in practice [9]. A post-

*Corresponding author

978-1-5090-3811-4/16/$31.00 ©2016 IEEE IEEE AVSS 2016, August 2016, Colorado Springs, CO, USA

processing step is required to extract the regions because these methods detect foregrounds as the edges only. Kim *et al*. [8] proposed Local Hybrid Pattern (LHP) as the principal feature to generate the scene background model which adaptively divides edge and inner information. The characteristic of the LHP code is that it represent both edge and inner information, such as color or grayscale values. It uses a flag bit to express either edge or pixel information. In detail, they generate 8 bit LHP code including a flag bit. Each pixel represents edge or inner information with less than 7 bits (because of the flag bit). They use the direction of the maximum gradient response of neighboring pixels and ternary pattern to generate edge code. This method is robust against illumination variations, however, it may not represent various directions of the edge response because of its usage of the primary direction.

Given that any local pattern can be used on the LHP framework, there is a plethora of methods that arise from the combinations of existing local pattern methods. For example, LBP [11] has been used to represent local information by using neighboring pixels. LBP is generated by comparing intensity of the center pixel with each neighboring pixel, and flag a bit string based on whether its intensity is bigger than the center pixel. It is robust to illumination variations; however, it may generate different patterns whenever noise is included on the image. Local Directional Pattern (LDP) [7], on the other hand, was proposed to overcome the disadvantages of LBP by using edge responses obtained by the result of Kirsch compass masks instead of pixel information. It generates 8 bit code by choosing k neighboring maximum edge responses. Each bit represents the state of 8-neighboring pixels. The bits associated with each kth neighboring maximum edge responses will be set 1, otherwise they will be set 0. LDP is more robust against noise than LBP, however, if the last kth maximum response is not high enough, it may cause a problem due to noise during code generation. Kim *et al*. [8] used a coding scheme that uses the primary directional number and its orthogonal gradient magnitude in their method. Because of the usage of gradient magnitude, it is more robust against illumination variations than LBP. However, this scheme considers only one direction. Thus, it may not represent complex edges with various high edge responses.

In this paper, we propose a novel approach to generate local information patterns. Our proposed method includes directional information of local neighboring pixels through a set of restrictions imposed by the structure of the neighborhood. Such restrictions make it more robust to noise and other variations, as it will prevent it from encoding wrong information into the model. Thus, it is robust against noise and illumination variations (in comparison to intensity-based approaches) and can represent multiple edge directions and corners. In detail, we use the gra-

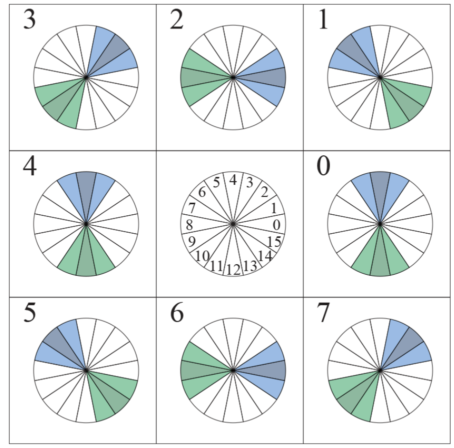

Figure 1: Structural restrictions of each neighboring pixel through a weight definition W_i. Each number in the upper corner represents the index i of neighboring pixels. Gradient orientation is quantized to 16 directions through a q function. The circle located on the center shows the indexes of quantized angle $\theta(x)$. The sectors filled with dark color denote a strong weight (Ω), surrounding sectors filled with light color denote a weaker weight (ω), and white denotes zero weight. To consider sense of direction, we use blue and green.

dient response of a Sobel filtered image, and use the magnitude information to obtain the principal directions. We check that these directions comply the structural restrictions, and if they do, we encode them as a structural edge pattern code into the LHP framework. That is, we extend the previously proposed LHP coding algorithm by replacing the edge representation. Furthermore, we generate the background model using an Adaptive Dictionary Model. Results obtained by using our method show that they are more accurate than other methods in standard datasets.

2. Proposed Edge Pattern Coding Scheme

The main idea of the proposed algorithm is to create a signature of the local neighborhood edge information. We encode the principal directions that are within the structure of a local neighborhood. The directions that deviate from this restrictions are discarded as noise. The principal directions are encoded using directional numbers [13].

In detail, to compute the code for a given pixel, x, we evaluate the gradient magnitude responses, G_m, of each y_i neighboring pixel for the current pixel x. Since we only care about principal directions in the local neighborhood to infer its structure, we consider a subset of possible structural edges on the local neighborhood: horizontal line, vertical line, and eight corners (four in the cross and four in the diagonals)—as shown in Fig. 1. To assert these directions we impose a weight on the representative magnitudes of the local pixels that depends on their edge angle and their position in the neighborhood we are evaluating. Formally, we compute the score, S_i, of each neighboring pixel y_i through

$$S_i = \begin{cases} G_m(y_i) W_i\left(\theta(y_i)\right) & \text{if } G_m(y_i) \geq T_m, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where G_m is the gradient magnitude response computed from the blurred Sobel operator, W_i is a weight function that imposes a restriction on the possible directions according to the ith position in the local neighborhood, θ is a function that returns the angle of the pixel in the argument, and T_m is a threshold that filters weak edge responses. The weight function W_i returns three possible values, a strong weight, Ω, a weak weight, ω, and zero weight according to the structural restriction of the ith position in the neighborhood. Formally, we define W_i according to its location $i \in \{0, 1, \ldots, 7\}$ as

$$W_{\{0,4\}}(\alpha) = \begin{cases} \Omega & \text{if } q(\alpha) \in \{4, 12\}, \\ \omega & \text{if } q(\alpha) \in \{3, 5, 11, 13\}, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

$$W_{\{2,6\}}(\alpha) = \begin{cases} \Omega & \text{if } q(\alpha) \in \{0, 8\}, \\ \omega & \text{if } q(\alpha) \in \{1, 7, 9, 15\}, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

$$W_{\{1,5\}}(\alpha) = \begin{cases} \Omega & \text{if } q(\alpha) \in \{6, 14\}, \\ \omega & \text{if } q(\alpha) \in \{5, 7, 13, 15\}, \\ 0 & \text{otherwise}, \end{cases} \tag{4}$$

$$W_{\{3,7\}}(\alpha) = \begin{cases} \Omega & \text{if } q(\alpha) \in \{2, 10\}, \\ \omega & \text{if } q(\alpha) \in \{1, 3, 9, 11\}, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

where $W_{\{i,j\}}$ denotes that the same function applies to the locations in the set, *i.e.*, i as well to j, α is the angle of the pixel we are evaluating, and q is a function that quantizes the angle α into 16 bins. Fig. 1 depicts the structural restrictions imposed by W_i. For each position i in the local neighborhood, we established a principal direction (which have a strong weight shown in darker color) and small deviations from it (which have a weaker weight shown in lighter color), as well as a sign of the direction (shown by the different color in each location). To assign the weight to a given angle, we quantized the angle domain into 16 bins, and each bin has an associated weight that is returned by W_i. For example, the 0th position will have an horizontal edge going through it, thus we expect an edge response perpendicular to it. Any other angle at position 0 will be treated as noise.

Once we computed the scores of every neighboring pixel, we choose the two prominent pixels through

$$d_k(x) = \arg\max_{j}{}_{\mathrm{k}} \{S_j(x) \mid 0 \leq j \leq 7\}, \tag{6}$$

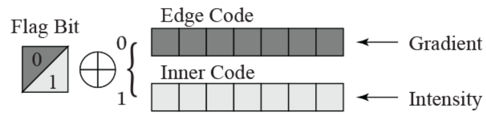

Figure 2: Simple explanation of the LHP code. It can represent two different information, whether it is edge or inner code, by using flag bit.

where d_k is the index of the kth maximum value in the set of scores from every neighboring pixel of x, and $\arg\max_{\mathrm{k}}$ is a maximum operator that returns the kth maximum element. In this paper, we focus on two meaningful directions to generate the proposed edge pattern code. Thus, we only generate d_1 and d_2 as the first and secondary directions, respectively. Finally, the proposed edge pattern code C_{EP} is generated as

$$C_{\text{EP}}(x) = \begin{cases} 16 C_s\left(d_1(x)\right) + & \text{if } (S_{d_1}(x) \geq T_s \\ C_s\left(d_2(x)\right) & \wedge\, S_{d_2}(x) \geq T_s), \\ 0 & \text{otherwise}, \end{cases} \tag{7}$$

where C_s is a function that transform the index and the sign information into a three bit code, and T_s corresponds to score threshold. If either $S_{d_1}(x)$ or $S_{d_2}(x)$ are less than T_s, then we do not generate a code because they are not meaningful enough directions. Note that we use d_1 instead of $d_1(x)$ in the subscripts (same for d_2) for the sake of simplicity—*cf*. (6). The code generation is defined as

$$C_s(i) = 8 sgn(i) + i, \tag{8}$$

where sgn is a function that returns the sign of the pixel at position i. Note that we are abusing the notation of the position of the pixel in the ith neighboring position for the sake of simplicity in the definition of sgn. In rigor, sgn should receive the y_i position to compute it, but that will make complexify the definition of (7).

3. Background Modeling

The Local Hybrid Pattern (LHP) is a local feature descriptor that mixes edge and internal information. And it is used with an Adaptive Dictionary Model that models background by Kim *et al.* [8]. In this section, we explain our modifications to LHP by mixing it with our proposed method to model the background.

3.1. Local Hybrid Pattern

To train a video stream captured from stationary camera to background model, we use LHP as the feature that represents two different sets of information, edge and texture (inner), by using a flag bit. In Fig. 2, LHP code is represented as the edge or inner code whether the flag bit is 0 or not.

We should generate a binary edge image f as the flag map to consider every pixels as either edge or inner code before modeling background, because edge codes have to be generated on the edge pixels only. We use the results of the proposed coding scheme to make the decision on what is and is not an edge through setting the flag f by

$$f(x) = \begin{cases} 1 & \text{if } C_{\text{EP}}(x) = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

where x is the pixel position, and we consider pixel x as the edge if $C_{\text{EP}}(x) = 0$. We generate the code

$$\text{LHP}(x) = \begin{cases} 2^{n-1} f(x) + C_{\text{EP}}(x) & \text{if } f(x) = 0, \\ 2^{n-1} f(x) + C_{\text{inner}}(x) & \text{otherwise,} \end{cases} \tag{10}$$

where $C_{\text{inner}}(x)$ is code method which computes inner code by maintaining the high bits of the intensity value through

$$C_{\text{inner}}(x) = I(x) \gg b, \tag{11}$$

where $\gg$ represents the shift-left b bits operator to the left argument, and I is the intensity image.

3.2. Adaptive Dictionary Model

We use Adaptive Dictionary Model (ADM) based background model method. Let $B = \{D_x\}$ be the model as a set of dictionaries and D_x is a dictionary defined as

$$D_x = \{(c_0, w_0), (c_1, w_1), \ldots, (c_{N-1}, w_{N-1})\}, \tag{12}$$

where x is a pixel position, c_i is the ith code and w_i is associated weight. Every dictionary has a limit, N, on the number of codes.

To consider background changes, such as waving trees, we need to update background model. Let $B^t = \{D_x^t : \forall\, x\}$ be the current background model at time t and B^{t+1} be the next model at time $t+1$, we compute B^{t+1} by updating each associated weight of all the codes in the dictionaries belonging to B^t through $w_i^{t+1} = u(c, d_i, w_i)$, where the latter is defined as

$$u(c, d_i, w_i) = \begin{cases} (1-\alpha) w_i + \alpha & \text{if } m(c, d_i) = 1, \\ (1-\alpha) w_i & \text{otherwise,} \end{cases} \tag{13}$$

where c is the code at pixel position x at time t computed by (10), (d_i, w_i) is the ith tuple which belongs to D_x^t, α is the learning rate and $m(c, d_i)$ is a function which checks whether c is matched to d_i through

$$m(c, d) = \begin{cases} m_{\text{edge}}(c, d) & \text{if } c \text{ is edge,} \\ m_{\text{inner}}(c, d) & \text{otherwise,} \end{cases} \tag{14}$$

where m_{edge} and m_{inner} are matching functions for each case, such as edge and inner. These functions are defined as

$$m_{\text{edge}}(c, d) = \begin{cases} 1 & \text{if } c = d, \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

$$m_{\text{inner}}(c, d) = \begin{cases} 1 & \text{if } |c - d| \le T_i, \\ 0 & \text{otherwise,} \end{cases} \tag{16}$$

where T_i is a threshold that checks the similarity between c and d. Now we can manage D_x, that is, a new code c at time t that is not the matched code d belonging to D^t needs to be added to D^t to generate background model B^{t+1}. We achieve it through

$$D_x^{t+1} = \begin{cases} \Big\{ D_x^t \backslash \big\{ (d_j, w_j) \big\} \Big\} \cup \big\{ (c, \alpha) \big\} & \text{if } |D_x| \ge N, \\ D_x^t \cup \big\{ (c, \alpha) \big\} & \text{otherwise,} \end{cases} \tag{17}$$

where $\backslash$ is the difference set operator, (d_j, w_j) is the tuple comprised of code d_j and its weight w_j, and $|D_x|$ is the number of elements in D_x.

4. Foreground Detection

Similarly to previous section, we use foreground detection method proposed by Kim *et al*. [8]. To mix the advantages of both edge and inner, first we generate candidate foregrounds of edge and inner as following:

$$F_{\text{cand}}(x) = \begin{cases} 1 & \text{if } p(x) < T_c, \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where T_c is a threshold of the background probability, and $p(x)$ is the probability map defined by

$$p(x) = \mathcal{P}\left(\text{LHP}^t(x), D_x^t\right), \tag{19}$$

where $\text{LHP}^t(x)$ is the code and D_x^t is the dictionary of the pixel x at time t. The function $\mathcal{P}$ returns the probability of the dictionary as follows

$$\mathcal{P}(c, D) = \begin{cases} \text{norm}(w_i) & \text{if } \exists\, d \in D : m(c, d) = 1, \\ 0 & \text{otherwise,} \end{cases} \tag{20}$$

where c is a code, D is a dictionary and $\text{norm}(w_i)$ is a normalization function $\text{norm}(w_i) = w_i / \sum_j w_j$.

After doing that, we need to fill the holes in candidate foregrounds by performing a closing operation. Finally, we can classify foreground by finding intersected regions between edge and inner candidate foregrounds.

5. Experimental Results

In this section, we examine the performance of our proposed method against several datasets from the ChangeDetection database [3] which consists of six different categories. In detail, we chose five datasets from three different categories: Highway (HIGH), Pedestrians (PED) and PETS 2006 (P2006) from Baseline, Parking (PARK) from Intermittent Object Motion and Bungalows (BUN) from Shadow. Both HIGH and PED sequences include illumination variations, P2006 include object which is left on the floor and PARK have new background that is made by the parked car.

We compared our proposed method against three different edge code descriptors using ADM based background modeling method using LHP [8]. As well, we compared our proposed method against four state of the art methods, Pixel-Based Adaptive Segmenter (PBAS) [5], Spatially Coherent Self-Organizing Background Subtraction (SC-SOBS) [10], Visual Background Extractor (ViBe) [1] and Vibe+ [15].

We use F-measure (F) to measure how well each methods matches the ground truth defined as $F = 2(Pr)(Re)/(Pr + Re)$, where Pr is precision, and Re is recall, and are defined $Pr = TP/(TP + FP)$, $Re = TP/(TP + FN)$, where TP are true positives, FP are false positives and FN are false negatives. Before measurement, we assume that all the detected foregrounds in the region of interest (ROI) are TP.

Proposed method has four parameters: T_m, T_s and the weight function weights Ω and ω. For the following experiments we set $\Omega = 1$, and for the thresholds and weight ω we did a parameter search using the F_1 measure by changing T_s, T_m and ω simultaneously. In here, we report the average score of the best case $T_m = 45$, $T_s = 40$ and $\omega = 0.7$ using the other in Figs. 3(a), 3(b) and 3(c), respectively. All the methods compared to proposed method in this paper can be included in 7 bits. But our proposed method requires 8 bit. We can include the proposed edge pattern code in LHP by extending the code size of LHP. But to evaluate it fairly, we did not extend it, but used a lookup table instead which uniformly converts 8 bit proposed code to 7 bit.

First we evaluate the contribution of the proposed method as an edge feature descriptor into the background model by using different codes in the proposed framework instead of proposed method. Namely, we compared against original LHP, LBP, and LDP. The parameters for each method are as follows. We use similar setup as Kim *et al.* [8] for LHP, *i.e.*, $T_{\text{edge}} = 100$. We set $\text{LBP}_{2,6}$ which generates code using 6 neighboring pixels on a circle of radius 2. LDP_3 is generated by considering the top 3 neighboring edge responses. Inner code quantization parameter $b = 1$, ADM learning rate $\alpha = 0.006$ and $T_{\text{cand}} = 0.18$.

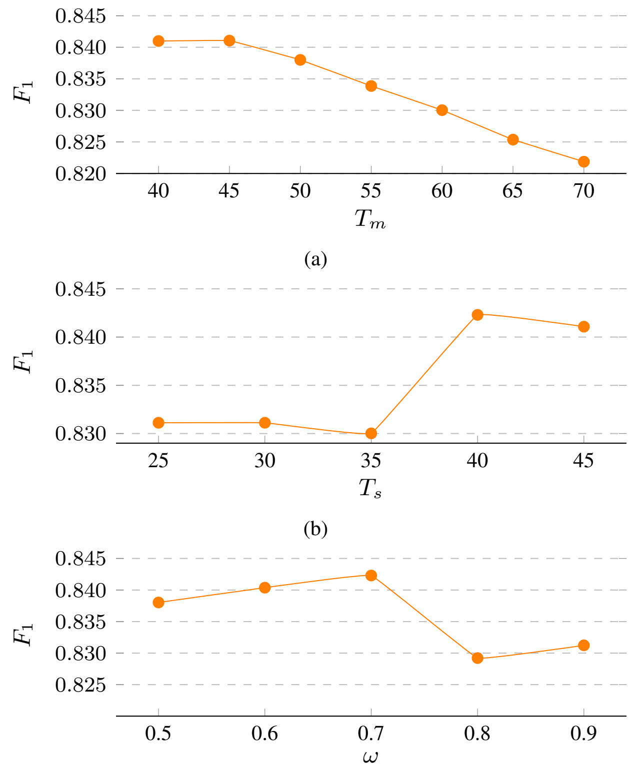

Figure 3: The average F-measure of the proposed method over all the datasets when changing (a) the magnitude threshold (T_m), (b) the score threshold (T_s) and (c) the weight (ω).

We show quantitative evaluations of all the descriptors and our proposed method on Table 1. In all the datasets, except in HIGH, our proposed method is better than previous methods. In HIGH case, there are many FP in the moving background region, such as waving trees, due to the discriminative rate of the codes in proposed method. Thus, the proposed algorithm represents the directions of the dynamic edges differently. However, proposed method shows better results than other methods (see Fig. 4).

Our proposed method represents various edge information by using neighboring directions and it is more robust against noise and illumination variations than other coding methods. Fig. 5 shows illumination variations in HIGH and overlapping edges in PARK. In HIGH dataset, the vertical line in red box is a background edge. LBP and LDP detect many false edge candidate because they generate different code (due to sensitivity to illumination variations). LHP and our proposed method are more robust to illumination variations because both use principal neighboring edge information. In PARK dataset, the car, as ground truth, is overlapping with the horizontal line in the background. In this case, LHP cannot detect enough edge candidate because it considers one principal neighboring edge informa-

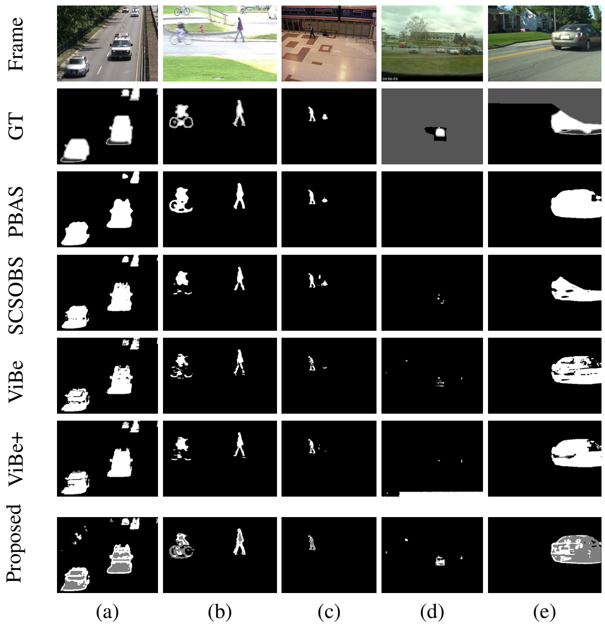

Figure 4: Examples of detection each of the different dataset sequences. (a) HIGH, (b) PED, (c) P2006, (d) PARK, (e) BUN from ChangeDetection database. GT is ground truth

Table 1: Quantitative results of edge descriptors as F-measure (%).

Descriptor	LHP	$LBP_{2,6}$	LDP_3	Proposed
HIGH	91.99	91.52	91.45	90.41
PED	90.99	90.66	90.38	91.08
P2006	71.07	71.38	71.59	72.43
PARK	62.21	65.61	68.12	71.59
BUN	95.48	95.65	95.59	95.65
Avg.	82.35	82.96	83.43	84.23

tion. It means that LHP cannot reliably distinguish overlapping edges. Furthermore, candidate foreground edge of proposed method is more stable (continuous) than others, because proposed method considers two principal neighboring pixels within its encoded information. Thus, proposed method shows better results consistently in comparison to others. The detection of good edge candidates is important when we use LHP method, thus the introduction of proposed method provides an improvement of the overall algorithm.

Quantitative and qualitative evaluations of state of the art methods against our proposed method are shown in Table 2 and Fig. 4, respectively. In this case, we generate results with the ADM-based background modeling method using LHP which include proposed method as edge code with post processing median filter. We can get better result than other methods in PARK sequences because when parked car leaves, our proposed method can update more faster than others.

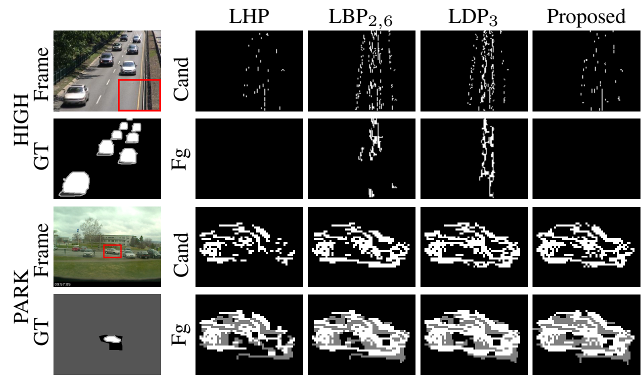

Figure 5: Examples of candidate foreground edge and detection results. First column is input frame and ground truth of each dataset (HIGH and PARK), respectively. Second group of columns show the result of different methods in each dataset.

Table 2: Quantitative results of background modeling methods as F-measure (%)

Methods	PBAS	SC-SOBS	ViBe	ViBe+	Proposed
HIGH	92.59	89.64	85.46	89.09	91.77
PED	91.30	80.89	80.77	81.95	91.74
P2006	85.99	79.70	70.65	72.98	72.73
PARK	16.48	40.05	38.77	26.32	74.02
BUN	94.82	93.82	91.40	94.23	96.02
Avg.	76.24	76.82	73.41	72.91	85.26

6. Conclusions

In this paper, a novel edge descriptor is proposed that encodes directional information reliably by imposing structural constraints on the possible edge directions that appear in the local neighborhood that is analyzed. Proposed method takes the advantage of using a quantize angle space to provide with weights to score the different pixels, and, finally, produces a stable code. We use the proposed method in the LHP background modeling framework and obtained better results than existing methods in several datasets. We compared our proposed method to other local descriptors under the same LHP framework, and we obtained better results. Moreover, we show that the LHP plus our proposed method combination yields a better performance in standard datasets in comparison to other state-of-the-art background modeling methods.

Acknowledgment

This research was supported by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2015R1A2A2A01006412) and a grant funded by CONICYT, under FONDECYT Iniciación No. 11130098.

References

[1] O. Barnich and M. Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *Image Processing, IEEE Transactions on*, 20(6):1709–1724, 2011.

[2] T. W. Chua, Y. Wang, and K. Leman. Adaptive texture-color based background subtraction for video surveillance. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 49–52. IEEE, 2012.

[3] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection. net: A new change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–8. IEEE, 2012.

[4] M. Heikkila and M. Pietikäinen. 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.

[5] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 38–43. IEEE, 2012.

[6] M. J. Hossain, M. A. A. Dewan, and C. Oksam. Moving object detection for real time video surveillance: An edge based approach. *IEICE transactions on communications*, 90(12):3654–3664, 2007.

[7] T. Jabid, M. H. Kabir, and O. Chae. Local directional pattern (ldp) for face recognition. In *2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE)*, 2010.

[8] J. Kim, A. Ramirez Rivera, B. Ryu, and O. Chae. Simultaneous foreground detection and classification with hybrid features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3307–3315, 2015.

[9] J. Kim, A. Ramírez Rivera, G. Song, B. Ryu, and O. Chae. Edge-segment-based background modeling: Non-parametric online background update. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pages 214–219. IEEE, 2013.

[10] L. Maddalena and A. Petrosino. The sobs algorithm: what are the limits? In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 21–26. IEEE, 2012.

[11] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[12] C.-J. Pai, H.-R. Tyan, Y.-M. Liang, H.-Y. M. Liao, and S.-W. Chen. Pedestrian detection and tracking at crossroads. *Pattern Recognition*, 37(5):1025–1034, 2004.

[13] A. Ramírez Rivera, J. Rojas Castillo, and O. Chae. Local directional number pattern for face analysis: Face and expression recognition. *IEEE Transactions on Image Processing*, 2012.

[14] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[15] M. Van Droogenbroeck and O. Paquot. Background subtraction: Experiments and improvements for vibe. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 32–37. IEEE, 2012.

[16] G. Xue, L. Song, J. Sun, and M. Wu. Hybrid center-symmetric local pattern for dynamic background subtraction. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.

[17] J. Yao and J.-M. Odobez. Multi-layer background subtraction based on color and texture. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[18] Z. Zhang, C. Wang, B. Xiao, S. Liu, and W. Zhou. Multi-scale fusion of texture and color for background modeling. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 154–159. IEEE, 2012.