Background Modeling using Adaptive Properties of Hybrid Features

Jaemyun Kim[1], Adín Ramírez Rivera[2], Byeongwoo Kim[1], Kaushik Roy[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
{jaemyunkim, byung-woo, kaushik.roy, oschae}@khu.ac.kr
[2]Institute of Computing, University of Campinas (UNICAMP), Campinas, Brazil
adin@ic.unicamp.br

Abstract

In this paper, we propose Local Adaptive Hybrid Pattern (LAHP) to use intrinsic properties, both edge and color, of each pixel adaptively (inspired from LHP based methods) while using a single feature representation (inspired from LOBSTER). The proposed LAHP encodes edge and color information, and an adaptive factor based on gradient magnitude together as a single feature. We introduce a way to calculate the code distance of LNESP to reduce sensitive to edge distortions. Furthermore, we modify ADM (Adaptive Dictionary Model) to manage LAHP and extend a feature matching scheme of ADM to adjacent background models to reduce sensitivity to background motions. Results show promising performance than other methods in standard datasets.

1. Introduction

In these days the number of video-based surveillance applications is increasing, such as vision-based traffic system [1], video segmentation [2], [3], human behavior analysis [4], [5], etc. Background subtraction is a method to detect moving objects and to segment their regions. It uses the subtraction operation between the background model and the incoming frame. Thus, most cases of vision-based surveillance require a background modeling technique. The main challenge of this technique comprises the background changes due to diversity in illumination, background motions (*e.g.*, waving trees or rippling water), noise, shadows, etc. Several methods [1], [2], [6]–[24] have been proposed to solve this problems using different approaches.

Pixel-based methods model the background using intensity or color information on each pixel. Single-modal based approaches [6], [7] model the background using single model but they suffer from illumination changes. Mixture of Gaussians (MoG) [8] uses k Gaussians models to represent background variations. However, the selection of k modes to identify between fast and slow variations is hard. Kim *et al*. [9] proposed a codebook based background modeling by using codewords (color and six tuples) with variable size to represent background variations. However, this may generate a large number of codewords in the model when the background changes. Hoffman *et al*.'s [11] work, or the previously proposed Visual Background extractor (ViBe) [10], model the background using a history of samples that do not assume a shape on the distributions based on SAmple CONsensus (SACON) model [25], [26] and random update strategy. Despite these efforts, pixel-based methods are not free from introducing illumination-based errors to the modeled scene.

In contrast, texture- and edge-based methods were proposed as less-sensitive to illumination variations. Heikkilä and Pietikäinen [12] proposed a mixture of histograms of Local Binary Patterns (LBP). However, LBP is sensitive to noise. To reduce this drawback, Scale Invariant Local Ternary Pattern (SILTP) [13] encodes a texture pattern using an adaptive threshold based on pixel intensity, but this requires twice the bit length than LBP. Also, Silva *et al*. [27] proposed eXtended Center-Symmetric Local Binary Pattern (XCS-LBP) to be robust to illumination changes and noise. However, texture-based methods have problems representing consistency when the image contains flat regions.

Edge-based methods model the background relying on edges or gradient responses from the image. These approaches are more stable to illumination changes than pixel-based methods. However, edges do not have the exact same position, shape, or length on consecutive frames (we refer to these problems as edge distortions). Traditional edge-pixel-based methods use binary edge information [1], [2] or models [14] on each edge pixel. They have, however, many false alarms by the edge distortions. To overcome these problems, Hossain *et al*. [15] used edge-segment (*i.e.*, concatenation of connected edges) instead of edge pixels. This may misclassify moving-edge-segment as background and requires higher computation to match each edge-segment.

*Corresponding author

IEEE AVSS 2017, August 2017, Lecce, ITALY

Statistical edge-segment-based methods [16], [17] model the edge variations in the scene to overcome previously introduced drawbacks. Each accumulated region in the model refines their statistical information to represent its edge-segment distribution. However, these edge-based methods are sensitive to background motions and support only foreground edges in the foreground detection as well.

Some studies model background using hybrid approaches [18]–[24] that mix both pixel and edge (or texture) at the feature level. Jabri *et al*. [18] proposed a color and edge-information based background modeling that models mean and variance for each color channel and Sobel responses separately. Javed *et al*. [19] proposed a hierarchical modeling approach based on MoG at pixel-level. In region-level, they count foreground edge pixels that lie on the pixel-based foreground region. Noh and Jeon [20] proposed Scene Adaptive Local Binary Pattern (SALBP) based a multiple cue system. They modeled background using color and texture (*i.e*., HSV and SALBP) separately. However, these methods require different types of background models. St-Charles and Bilodeau proposed LOBSTER (LOcal Binary Similarity segmenTER) that used both local binary similarity pattern (LBSP) [28] and color together as a single feature and its modeling based on ViBe. And St-Charles *et al*. [22] proposed SubSENSE (Self-Balanced SENsitivity SEgmenter) to improve LOBSTER based on pixel based adaptive segmenter (PBAS) [11] that is modified version of ViBe [10]. But their LBSP takes 16 bits to represent the texture pattern for each pixel, and their feature matching is based on logical AND operator and requires satisfaction of two conditions (*i.e*., texture and color). Kim *et al*. [23] proposed a Local Hybrid Pattern (LHP) based background modeling that maintains heterogeneous features (*i.e*., edge and pixel) adaptively. A code image based on LHP separates an image as two different regions (edge and inner), and encodes their respective properties. And Hong *et al*. [24] proposed an edge shape pattern, called Local Neighbor Edge Shape Pattern (LNESP), for improving edge representation of the LHP structure. LHP-based methods show the robustness of illumination changes. However, they are sensitive to dynamic background motion because they do not have distance calculation for the edge feature matching.

In this paper, we propose Local Adaptive Hybrid Pattern (LAHP) to use intrinsic properties, both edge and color, of each pixel adaptively (inspired from LHP based methods) while using a single feature representation (inspired from LOBSTER). Previous LHP representation could not use intensity or color information when the pixel is classified an edge pixel, vice versa. The proposed LAHP is consisted to three different factors, *i.e*., edge code, inner code, and gradient magnitude. Then, LAHP at a pixel can use edge and inner information adaptively by the gradient magnitude on the pixel. Also, we can classify that the pixel is in edge or inner region based on the gradient magnitude. Furthermore, we propose a way that is similarity calculation of LNESP [24] for taking flexibility on the edge code matching. This may help to reduce edge distortion problem in consecutive frames.

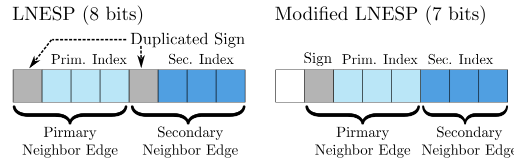

Figure 1. 7 bits LNESP representation by reducing duplicated signed bit from original LNESP. Each index of the neighbor edge pixel in the original LNESP has the same sign of the direction.

To adjust the dynamic background motion in the scene, we modify Adaptive Dictionary Model (ADM) [23] to manage LHAP instead of LHP and We calculate background probabilities based on Gaussian weighting for the nine code probabilities obtained from nine background models in the 3×3 region. In the modeling, conservative update strategy which the background updates when the pixel is classified to background is applied to model the background by clear background information, like ViBe [10]. But this strategy requires background propagation to learn real backgrounds, *i.e*., real background regions that appear by a parked car when it moves. Our feature matching scheme can preform similar effect like that because we consider adjacent background models.

Experimental results show better performance that proposed method is robust to dynamic background motions as well as other complex situations, such as illumination changes, many foregrounds in the scene, and global camera movements.

2. Local Adaptive Hybrid Pattern

We propose Local Adaptive Hybrid Pattern (LAHP) to use edge and pixel properties of the local neighborhoods adaptively. Note that the edge and intensity information are inverse properties. The more an edge is present in a local neighborhood the less meaningful the intensity information is in that neighborhood, and vice versa. Due to this fact, we propose an adaptive measuring algorithm for our codes that adaptively weights the information present in the local neighborhoods. Since the gradient magnitude is a reliable source of information on the amount of edgeness of a neighborhood, we exploit it further. Consequently, our code comprises edge, intensity, and gradient magnitude information that we combine to produce our background model and detect the foreground.

2.1. Code Definition

We design LAHP to represent edge shape and pixel information adaptively (through the gradient magnitude). Let $C(x)$ be a LAHP feature vector at pixel location x,

$$C(x) = \{C_{\mathrm{E}}(x), C_{\mathrm{I}}(x), G_{\mathrm{m}}(x)\}, \tag{1}$$

where C_{E} is an edge code, C_{I} is an inner code, and G_{m} is a gradient magnitude at pixel location x.

To create the edge and intensity codes we have several options. For instance, Kim *et al*. [23] designed LHP to use several edge or texture patterns into the edge code, and Hong *et al*. [24] tested several descriptors with their proposed LNESP code. In our case, for the edge code, C_{E}, we use 7 bits LNESP representation by reducing duplicated directional signed bits, as shown in Fig. 1. Because each index of the neighbor edge pixel of x in the original LNESP [24] has the same sign of the direction. Our proposed edge code is generated as

$$C_{\mathrm{E}}(x) = \begin{cases} 2^6 sgn\big(d_1(x)\big) + & \text{if } \big(S_{d_1}(x) \geq T_s \wedge \\ 2^3 d_1(x) + d_2(x) & \quad S_{d_2}(x) \geq T_s\big), \\ \emptyset & \text{otherwise}, \end{cases} \tag{2}$$

where d_1 and d_2 are primary and secondary neighbor edge pixel indixes (*i.e*., their primary and secondary edge directions), $S_{d_{\{1,2\}}}$ are the scoring functions that weight each direction—refer to Hong *et al*. [24] (1)—, and T_s corresponds to a score threshold—as proposed in the original LNESP [24]. If either $S_{d_1}(x)$ or $S_{d_2}(x)$ are less than T_s, then we set a default edge code, $\emptyset$, because they are not meaningful enough directions. Then, our proposed edge code encodes a sign of the direction and indixes of two neighbor edge pixels which has the first and second highest score representing edge shape pattern (*i.e*., principal directions).

On the other hand, our intensity code, C_{I}, uses RGB color value, *i.e*., a 24 bits single channel by concatenating color values from each channel (R, G, and B) instead of the grayscale intensity value to get more accurate pixel representation. The intensity code on LAHP is defined as

$$C_{\mathrm{I}}(x) = 2^{16}R(x) + 2^8 G(x) + B(x), \tag{3}$$

where $R(x)$, $G(x)$, and $B(x)$ are the red, green, and blue color values at pixel location x, respectively.

As we mentioned before, the gradient magnitude G_{m} grasps the disposition of the local neighborhood, centered at the pixel x, of containing a majority of edge or intensity information. To compute the gradient magnitude we use the response from Sobel operator

$$G_{\mathrm{m}} = \sqrt{G_x(x) + G_y(x)}, \tag{4}$$

where G_x and G_y are the Sobel operators in the horizontal and vertical directions, respectively.

2.2. Code matching

Previous LHP based methods are sensitive to dynamic background motions, such as waving trees or flowing water, because they do not have a similarity measure for the edge code. That is, previous methods try to match the codes exactly. However, that is not always possible. Instead, for feature vector distance calculation, we compute edge and intensity distances individually and combine them into a code distance using adaptive factors, extracted from the gradient magnitude, as follows. We define the distance between two codes C^1 and C^2 as

$$\begin{aligned} dist(C^1, C^2) =& \omega(G^1_{\mathrm{m}}, G^2_{\mathrm{m}}) \cdot dist_{\mathrm{E}}(C^1_{\mathrm{E}}, C^2_{\mathrm{E}}) + \\ & \big(1 - \omega(G^1_{\mathrm{m}}, G^2_{\mathrm{m}})\big) \cdot dist_{\mathrm{I}}(C^1_{\mathrm{I}}, C^2_{\mathrm{I}}), \end{aligned} \tag{5}$$

where ω is a ratio based on two adaptive factors G^1_{m} and G^2_{m} from each codes C^1_E and C^2_E by

$$\omega(G^1_{\mathrm{m}}, G^2_{\mathrm{m}}) = \begin{cases} S\big(\max(G^1_{\mathrm{m}}, G^2_{\mathrm{m}})\big) & \text{if } C^1_{\mathrm{E}} \neq \emptyset \wedge C^2_{\mathrm{E}} \neq \emptyset, \\ S(G^1_{\mathrm{m}}) & \text{if } C^1_{\mathrm{E}} \neq \emptyset \wedge C^2_{\mathrm{E}} = \emptyset, \\ S(G^2_{\mathrm{m}}) & \text{if } C^1_{\mathrm{E}} = \emptyset \wedge C^2_{\mathrm{E}} \neq \emptyset, \\ S\big(\min(G^1_{\mathrm{m}}, G^2_{\mathrm{m}})\big) & \text{otherwise}, \end{cases} \tag{6}$$

where $\emptyset$ denotes the default code, S is a Sigmoid function for the weight activation, that is defined as

$$S(g) = \frac{1}{1 + e^{-Ag - B}}, \tag{7}$$

$$g = \frac{G_{\mathrm{m}}}{G_{\mathrm{m,max}}}, \tag{8}$$

where A and B are slope and shift of Sigmoid function (we set $A = 10$ and $B = 5$, as shown in Fig. 2), g is a normalized gradient magnitude, and $G_{\mathrm{m,max}}$ is a maximum possible gradient magnitude by Sobel operator. And edge distance $dist_{\mathrm{E}}$ is computed by direction similarity as defined by

$$dist_{\mathrm{E}} = \begin{cases} dist^{dir}_{\mathrm{E}} & \text{if } C^1_{\mathrm{E}} \neq \emptyset \wedge C^2_{\mathrm{E}} \neq \emptyset, \\ dist_{\emptyset} & \text{otherwise}, \end{cases} \tag{9}$$

$$dist^{dir}_{\mathrm{E}} = w_{\mathrm{E}} \cdot dist^{\mathrm{dir}}_1 + (1 - w_{\mathrm{E}}) \cdot dist^{\mathrm{dir}}_2, \tag{10}$$

$$dist^{\mathrm{dir}}_i(d^1_i, d^2_i) = \begin{cases} dd_i & \text{if } dd_i \leq d_{\mathrm{max}}, \\ 2d_{\mathrm{max}} - dd_i & \text{otherwise}, \end{cases} \tag{11}$$

where w_{E} is a weight value for primary edge pixel on LNESP (we set $w_{\mathrm{E}} = 0.67$ to give twice weight on primary edge than the secondary edge) and d_{max} is the maximum possible directional distance and $dd_i = |d^1_i - d^2_i|$ is a distance between two directions from C^1_E and C^2_E. And inner distance $dist_{\mathrm{I}}$ are computed as

$$dist_{\mathrm{I}} = \begin{cases} |R^1 - R^2| + & \text{if } |R^1 - R^2| < T_{\mathrm{I}} \\ |G^1 - G^2| + & \quad \wedge |G^1 - G^2| < T_{\mathrm{I}} \\ |B^1 - B^2| & \quad \wedge |B^1 - B^2| < T_{\mathrm{I}}, \\ 255 * 3 & \text{otherwise}, \end{cases} \tag{12}$$

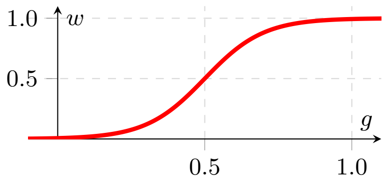

Figure 2. Sigmoid function for the weight adaptively based on gradient response.

where T_{I} is a color channel threshold, R^1 and R^2 are red values from code C^1 and code C^2, respectively, and G and B are the same as R case. Finally, we normalize the inner distance $dist_{\text{I}}$ to have the same distance range with $dist_{\text{E}}$ (in this paper $[0, 4]$).

3. Background Modeling

To model the background we intend to learn the features of the local neighborhoods by modifying an Adaptive Dictionary Model (ADM). The ADM models the background by using LAHP codes (instead of raw pixel information), and extends a feature matching to adjacent background models to reduce sensitivity to background motions. Let $B = \{D_x\}$ be the dictionary model that is a set of dictionaries, and D_x be a dictionary with a limit N on the number of codes it supports defined as

$$D_x = \{(c_0, w_0), (c_1, w_1), \ldots, (c_{N-1}, w_{N-1})\}, \tag{13}$$

where x is a pixel location, C_i is the ith LAHP code and w_i is associated weight.

To adjust the original background appearance that was hidden regions by a static object, we update the background model based on a conservative strategy, like ViBe, that updates a pixel only when it is classified as background. This means that once we learn our background model, we will only propagate that information into the rest of the scene. In other words, foreground will never be incorporated into the background and objects will be always detected as foreground even if they are stationary for a long time. (Note that this is just a design decision that may suit some scenarios.)

In our proposed method, we try to match the incoming background code c at pixel x to the background model within the local neighborhood, $\mathcal{N}_x$, of size 3×3 centered at pixel X (*i.e.*, nine different background probabilities) to handle changes in the background. Formally, 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 $t+1$. For a given pixel x, we need to update its associated dictionary D_x^t to create D_x^{t+1}. We first check if we can find the incoming code c within the dictionaries, D, in its neighborhood, $\mathcal{N}_x$. That is,

$$i^* = \arg\min_i \{dist(c, d_i) : (d_i, w_i) \in D, D \in \mathcal{N}_x\}, \tag{14}$$

where $dist$ is defined as (5). Consequently, (d_{i^*}, w_{i^*}) is the closest match in the existing dictionaries of the neighborhood of x. (We abuse the notation of the indexes to simplify the definition of the tuples in the set of dictionaries by assuming that every tuple will have a unique identifier.) Since we assume that the incoming code is background, it is guaranteed to exist within the local neighborhood—cf. Section 4. Then, we compute D_x^{t+1} by updating the associated weight of each tuple in the dictionary $(d_i, w_i) \in D^t$ through

$$w_i^{t+1} = u(w_i, d_i, i^*), \tag{15}$$

where

$$u(w_i, d_i, i^*) = \begin{cases} (1-\alpha)w_i + \alpha & \text{if } i = i^*, \\ (1-\alpha)w_i & \text{otherwise}, \end{cases} \tag{16}$$

where i^* is the index of the best matched code (14), and α is the learning rate. If $(d_{i^*}, w_{i^*}) \notin D_x$, then we need to update the dictionary and add the best match. Thus, we are propagating the existing background through the model. We can update D_x^{t+1}, through

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

where $\backslash$ is the difference set operator, (d_j, w_j) is the tuple with the lowest weight w_j in D_x^t, and $|D_x^t|$ is the number of elements in D_x.

4. Foreground Detection

To take advantage of both edge and intensity information in foreground detection, we use a method proposed by Kim *et al.* [23]. We extract foreground candidates based on the background probability, p, defined by

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

where T_p is a threshold of the background probability, and $p(x)$ is the probability map of been background at pixel location x. As mentioned above, we match the code c with nine different background models and find the best match. We define the probability of finding c at the ith neighbor in the neighborhood $\mathcal{N}_x$ of x as

$$p_i = \begin{cases} \text{norm}(w_y) & \begin{array}{l} \text{if } \exists\ (d_y, w_y) \in D_i : \\ \quad \arg\min_y \{dist(c, d_y)\}, \end{array} \\ 0 & \text{otherwise}, \end{cases} \tag{19}$$

where c is the incoming code code, D_i is the dictionary at the ith neighbor of x, and $\text{norm}(w_y)$ is a normalization function $\text{norm}(w_y) = w_y / \sum_j w_j$. Then, the final probability of been background for the pixel x is given by the

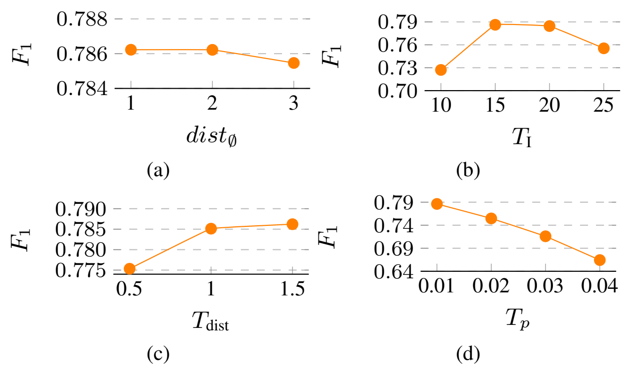

Figure 3. The average F-measure of the proposed method over all the datasets when changing (a) default distance $dist_\emptyset$ between default code edge shape code in LNESP, (b) color channel threshold T_I, (c) feature distance threshold T_dist, and (c) background probability threshold T_p.

weighted sum of the probabilities in its neighborhood, such as

$$p(x) = \sum_i p_i \cdot \mathcal{G}_i, \tag{20}$$

where i is an index in $\mathcal{N}_x$, p_i is defined as in (19), and $\mathcal{G}_i$ is the value of a Gaussian mask with the same dimensions as $\mathcal{N}_x$.

5. Experimental Results

Proposed method is designed to have less sensitivity of background motions, such as waving tree or flowing water, as well as illumination changes or other complex situations. To evaluate these, we examined the proposed method using several datasets from the ChangeDetection database [29]. In detail, we selected six datasets (Boats, Canoe, Fall, Fountain01, Fountain02, and Overpass) from Dynamic Background category, two datasets (Highway and Pedestrians) from Baseline category, and two datasets (Badminton and Traffic) from Camera Jitter category. Datasets in the Dynamic category, Both Boats and Canoe sequences include flowing water as background, Fountain01 and Fountain02 include Water movement fountain, Falling movement of branches, and overpass includes both water flowing and waving branches. In Baseline category, Highway includes complex situations of waving trees, many foregrounds, and small illumination variation and Pedestrians, and Pedestrians shows static situation with small illumination changes.

We compared our proposed method against other methods, MoG [8], Visual Background Extractor (ViBe) [10], LOcal Binary Similarity segmenTER (LOBSTER) [21], and LNESP [24].

For evaluate the propose method, F-measure (F) shows how well the proposed method matches to the ground truth defined by $F = 2(Pr)(Re)/(Pr + Re)$, where $Pr = TP/(TP + FP)$ is precision and $Re = TP/(TP + FN)$ is recall, where TP are true positives, FP are false positives and FN are false negatives. In this measurement, TP are the region of interest (ROI) for all the detected foregrounds.

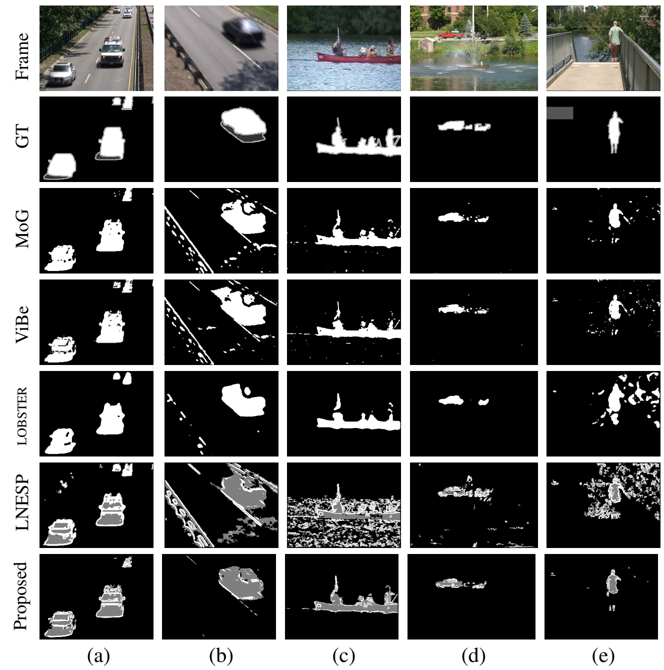

Figure 4. Examples of detection each of the different dataset sequences. (a) Highway, (b) Traffic, (c) Canoe, (d) Fountain02, and (e) Overpass from ChangeDetection database. GT is ground truth.

Table 1. Quantitative results comparison as F-measure (%)

Methods	MoG	ViBe	LOBSTER	LNESP	Proposed
Highway	**0.9240**	0.8831	0.8989	0.8825	0.9155
Pedestrians	**0.9536**	0.9493	0.9305	0.9053	0.9516
Badminton	0.6912	0.6694	0.7986	0.4517	**0.9078**
Traffic	0.6636	0.6406	0.7375	0.4052	**0.8347**
Boats	**0.7287**	0.4002	0.5822	0.0546	0.6836
Canoe	0.8817	0.8564	0.9299	0.2297	**0.9632**
Fall	0.4358	0.3990	0.2493	0.0902	**0.6034**
Fountain01	0.0763	0.1000	0.1651	0.0234	**0.5905**
Fountain02	0.8035	0.8308	0.8367	0.2241	**0.9421**
Overpass	0.8719	0.8050	0.7019	0.2590	**0.9345**
Avg.	0.7030	0.6534	0.6831	0.3526	**0.8327**

Proposed method several parameters that default distance $dist_\emptyset$ in LNESP (distance between default code ($C_\mathrm{E} = \emptyset$) and edge shape pattern code ($C_\mathrm{E} \neq \emptyset$)), color channel threshold T_I, feature distance threshold T_dist, and background probability threshold T_p. In Fig. 3, we set $dist_\emptyset = 2$ as , $T_\mathrm{I} = 15$, $T_\mathrm{dist} = 1.5$, and $T_p = 0.01$ by practical examination. For the other extra-parameters we follow setup as Hong *et al.* [24] for LNESP, *i.e.*, $T_s = 40$ and $w = 0.7$, and ADM learning rate $\alpha = 0.006$.

In Fig. 4, we show qualitative results of other methods and our proposed method. First, our proposed method improved the performance of LNESP as comparing in last two rows at Fig. 4. Especially, proposed method shows robustness to background motions, but the previous LNESP shows too sensitive because it does not have a distance measurement between edge codes. For the other methods, all of other methods are sensitive to camera jitter (as shown in Fig. 4(b)) that have global background motion by camera movements. For Canoe and Fountain02 datasets, other methods have noisy foregrounds or many holes in foreground regions. Proposed method shows better-defined foreground region. In Overpass dataset, other methods also detect branches as foregrounds because it has a lot of movement at branches on a tree but proposed method shows better detection results.

Also,we show quantitative evaluations of all the other methods with our proposed method on Table 1. In all the datasets, except in Highway, Pedestrians, and Boats datasets, our proposed method is better than previous methods.

6. Conclusions

In this paper, we proposed Local Adaptive Hybrid Pattern (LAHP) to use edge and inner information adaptively for a feature matching based on its gradient magnitude. For edge code in LAHP, we applied original LESP with distance calculation to get a flexibility on the edge matching for reducing edge distortion problems. To have less sensitivity to dynamic background motions, we check adjacent background models within 3×3 region and calculate its background probability based on them. We performed several experiments in which the proposed method obtained promising results for the dynamic background motions as well as other complex situations, such as illumination changes, many foregrounds in the scene, and global camera movements.

References

[1] D. Dailey, F. Cathey, and S. Pumrin, "An algorithm to estimate mean traffic speed using uncalibrated cameras," *IEEE Trans. Intell. Transp. Syst.*, 2000.

[2] C. Kim and J. Hwang, "Fast and automatic video object segmentation and tracking for content-based applications," *IEEE Trans. Circuits Syst. Video Technol.*, 2002.

[3] J. Kim, A. Ramirez Rivera, B. Ryu, K. Ahn, and O. Chae, "Unattended object detection based on edge-segment distributions," in *Proc. IEEE AVSS*, 2014, pp. 283–288.

[4] P. Banerjee and S. Sengupta, "Human motion detection and tracking for video surveillance," in *In National Conference on Communication*, IITBombay, India, Feb. 2008.

[5] N. M. Oliver, B. Rosario, and A. P. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug. 2000.

[6] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Mach. Vis. Appl.*, pp. 187–193, 1995.

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

[8] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE CVPR*, vol. 2, 1999, pp. 246–252.

[9] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, Jun. 2005.

[10] O. Barnich and M. Van Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[11] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *IEEE CVPRW*, Jun. 2012, pp. 38–43.

[12] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.

[13] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, and S. Z. Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes.," in *Proc. IEEE CVPR*, 2010.

[14] V. Jain, B. Kimia, and J. Mundy, "Background modeling based on subpixel edges," in *ICIP 2007*, Sep. 2007.

[15] M. Hossain, M. Dewan, and O. Chae, "Moving object detection for real time video surveillance: An edge based approach," *IEICE Transactions*, 2007.

[16] A. Ramirez Rivera, M. Murshed, J. Kim, and O. Chae, "Background modeling through statistical edge-segment distributions," *IEEE Trans. Circuits Syst. Video Technol.*, 2013, ISSN: 1051–8215.

[17] J. Kim, M. Murshed, A. Ramirez Rivera, and O. Chae, "Background modelling using edge-segment distributions," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.

[18] S. Jabri, Z. Duric, and H. Wechsler, "Detection and location of people in video images using adaptive fusion of color and edge information," in *Proc. ICPR*, 2000, pp. 627–630.

[19] O. Javed, K. Shafique, and M. Shah, "A hierarchical approach to robust background subtraction using color and gradient information," in *Proc. MOTION '02*, 2002.

[20] S. Noh and M. Jeon, "A new framework for background subtraction using multiple cues," in *Computer Vision – ACCV 2012*, vol. 7726, Springer Berlin Heidelberg, 2013, pp. 493–506.

[21] P.-L. St-Charles and G.-A. Bilodeau, "Improving background subtraction using local binary similarity patterns," in *Proc. IEEE WACV*, 2014, pp. 509–515.

[22] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Subsense: A universal change detection method with local adaptive sensitivity," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 359–373, 2015.

[23] J. Kim, A. Ramirez Rivera, B. Ryu, and O. Chae, "Simultaneous foreground detection and classification with hybrid features," in *Proc. IEEE ICCV*, Jun. 2015, pp. 3307–3315.

[24] S. Hong, J. Kim, A. R. Rivera, G. Song, and O. Chae, "Edge shape pattern for background modeling based on hybrid local codes," in *Proc.IEEE AVSS*, 2016, pp. 1–7.

[25] H. Wang and D. Suter, "Background subtraction based on a robust consensus method," in *Proc. ICPR 2006*, vol. 1, 2006, pp. 223–226.

[26] H. Wang and D. Suter, "A consensus-based method for tracking: Modelling background scenario and foreground appearance," *Pattern Recogn.*, vol. 40, no. 3, pp. 1091–1105, Mar. 2007.

[27] C. Silva, T. Bouwmans, and C. Frélicot, "An extended center-symmetric local binary pattern for background modeling and subtraction in videos," in *Porc. VISAPP*, Berlin, Germany, 2015.

[28] G.-A. Bilodeau, J.-P. Jodoin, and N. Saunier, “Change detection in feature space using local binary similarity patterns,” in *Computer and Robot Vision (CRV), 2013 International Conference on*, IEEE, 2013, pp. 106–112.

[29] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, “Changedetection.net: A new change detection benchmark dataset,” in *CVPRW*, 2012.