2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)
October 8-12, 2022, Macau, China

Real-time Adaptive Background Subtraction for Traffic Scenarios at Signalized Intersections Based on Roadside Fish-eye Cameras

Jiahe Cao, Zhouqiao Zhao, *Student Member, IEEE*, Guoyuan Wu, *Senior Member, IEEE*, Matthew J. Barth, *Fellow, IEEE*, Yongkang Liu, Emrah Akin Sisbot, Kentaro Oguchi

2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) | 978-1-6654-6880-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITSC55140.2022.9922318

***Abstract*— Background subtraction (BS) has been a norm for moving object detection along a classical computer vision pipeline, especially when the labelled data is largely unavailable. It has been widely used for infrastructure-based sensing such as traffic surveillance with roadside cameras. Existing BS algorithms focus on detecting moving objects, while the temporal motionless objects are neglected. This leads to performance degradation in particular at signalized intersections where vehicles may stop to wait in red. In this paper, we propose a hierarchical adaptive BS method which can eliminate the cumulative errors for those temporally static objects based on images from roadside fish-eye cameras in real-time. The proposed method is validated in both the CARLA simulation and the real-world environment. The results show that our method outperforms ViBe and LOBSTER by about 45% and 39%, respectively, on recall, without compromising too much in precision.**

I. INTRODUCTION

Evolving with the advance in sensor technology and edge computing, research on next-generation infrastructure-based traffic surveillance systems has been active in recent years. By leveraging computer vision (CV) techniques, such systems can provide high-fidelity (e.g., object-level) perception information to enable cooperative automated driving in a mixed traffic environment [1].

Although deep learning-based CV approaches have gained momentum in the past few years, the conventional inference pipeline is still attractive, especially for roadside sensing, mainly due to its non-reliance of labelled data and generalizability. As one of the key steps along the pipeline, background subtraction (BS) which aims to separate moving objects (i.e., "foreground") from the static scene (i.e., "background") has been widely used for infrastructure-based traffic surveillance on freeways [2]. However, most of existing BS algorithms [3][4] are not capable of separating objects (e.g., vehicles) that keep stationary for a series of frames from the background, which is a commonly observed traffic scenario at a signalized intersection. In addition, to meet the requirement of 24/7 traffic surveillance, a dynamic BS strategy is needed to adapt to the changes in weather and lighting conditions.

To solve the aforementioned problems, we propose an innovative cascaded adaptive background subtraction algorithm based on Adaptive Frame Difference and modified Gaussian Mixture Model (GMM). This algorithm can dynamically update the regions without objects and reserve the regions with objects so that much fewer trials are left in background. For the previous BS algorithms, they aim at separating moving objects, while our method tries to remove objects no matter whether they are moving or stay still. Also for the fish-eye camera which gains much attention recently, some feature-based methods such as LBSP and LOBSTER fail to distinguish the foreground because of the major edge distortion in the fish-eye images. To validate the proposed algorithm, we apply it to both the CARLA simulation environment and the real-world testbed. The major contributions of this study are summarized below.

- Propose a hierarchical BS algorithm that can efficiently filter out temporally static objects frequently observed in traffic scenarios at signalized intersections.
- Compare with state-of-the-art BS algorithms using both simulation and real-world datasets.
- Create a fish-eye camera dataset with ground truth label in CARLA simulator.

The remainder of this paper is organized as follows. Section II reviews a few typical BS algorithms in detail. Then, the innovative cascaded BS algorithm is introduced in Section III. In Section IV, we compare the existing BS algorithms with the proposed algorithm in both the simulation environment and the real-world. Finally, we conclude the work and discuss future directions in Section V.

II. RELATED WORK

Since the 1990s, a number of methods have been proposed to improve the background subtraction (BS) aiming at making the detection process more robust to noise, background motion, and camera jitter [4] [2]. Generally, these algorithms have three common steps: *initialization*, *foreground detection*, and *background maintenance*. *Initialization* intends to provide the BS algorithm with a clean background at the beginning, which may lay a solid foundation for the subsequent steps. The quality of initialization also influences the quality of the generated background. This step can be realized using different statistical methods, including fuzzy model-based, kernel density-based, and feature-based [3]. The second step, *foreground detection*, separates the foreground objects from the image, and the background is left. In this step, the current frame is compared with the real-time generated background by subtraction or statistical method. *Background maintenance* updates the previously generated background with the information from the incoming frames. The rest of

Jiahe Cao, Zhouqiao Zhao, Guoyuan Wu, and Matthew J. Barth are with the Department of Electrical and Computer Engineering, the University of California at Riverside, Riverside, CA 92507 USA.

Yongkang Liu, Emrah Akin Sisbot, and Kentaro Oguchi are with the Toyota North America R&D Labs, Mountain View, CA 94043, USA.

this section will highlight and discuss some fundamentals and popular background subtraction algorithms.

A. Frame Difference

The most straightforward way to model background is based on mathematical theories, such as the temporal average [5] [6]. This type of methods can remove most moving objects from background but may include part of foreground into background when updating, because static foreground objects may exist in all frames for average. Thus, the adaptive-selection method is proposed [7]. In the adaptive-selection method, only the regions without moving objects are updated, but this still does not solve the problem thoroughly. For example, if objects move slowly or stop, they may be updated into background.

B. Gaussian Models

The Gaussian method includes single Gaussian distribution [8] [9] and *Gaussian Mixture Model* (GMM) [10] [11]. For single Gaussian method, every pixel inside each frame is modeled with a Gaussian distribution in the RGB space. Single Gaussian approach uses likelihood or *Mahalanobis distance* of current frame and background to differentiate foreground from background. GMM method uses weighted multiple Gaussian models for better modeling results, and it significantly outperforms the single Gaussian distribution method [11]. GMM method shows good performance on both the outdoor scene and low illumination scene. However, it cannot solve the aforementioned problem presented by the Frame Difference method.

C. Visual Background extractor (ViBe)

Compared to the Gaussian model methods, ViBe [12] tries to model background with a set of pixels. Assuming $v(x)$ is the color value of the pixel at position x, and v_i represents the sampled value at time i. The set of pixel values at position x is $M(x) = v_1, v_2, \ldots, v_N$. The N elements are sampled from previous video frames. By comparing the current pixel value $v(x)$ to the most closed points in the set $M(x)$, if the number of points within a sphere of radius R centered at $v(x)$ is larger than a predefined threshold, then $v(x)$ is traded as background. For the background updating, the new background pixel is added to $M(x)$, while a random old background pixel v_j is discarded. The random abandonment ensures a smooth and exponentially decaying lifespan for sample values that contribute to the pixel models.

D. The Pixel-Based Adaptive Segmenter (PBAS)

PBAS [13] combines sample consensus and ViBE together, and determines whether a pixel belongs to background or not by evaluating the sample consistence. It introduces control theory so that adaptive threshold and learning rate may vary with different background complexity. The more complex the background is, the higher probability of misjudgment it is. Thus, complex background should align with high threshold and low learning rate. PBAS shows outstanding robustness on slow illumination change.

E. Local Binary Similarity Patterns (LBSP)

LBSP [14] is a feature-based method for background subtraction. Binary feature descriptors are developed to evaluate the disparities in intensity between pairs of pixels under different configurations. LBSP reformulates the region R into a $n \times n$ square, and it can be obtained by comparing the center pixel with the neighboring pixels which may be all the pixels or a subset of P pixels in R. LBSP is defined as

$$LBSP_R(x_c, y_c) = \sum_{p=0}^{P-1} d(i_p - i_c) 2^p \quad (1)$$

where $d(x)$ is a threshold truncation function and i_c corresponds to the intensity of the center pixel (x_c, y_c). The updated LBSP may show its ability to express the local feature. However, when applied to the time-varying dataset, LBSP will add some corner point noises if the object has the same color gradient or similarity to the background despite the difference in color. Also, LBSP is likely to be affected by noises and blurred regions.

F. LOcal Binary Similarity segmenTER (LOBSTER)

LOBSTER [15] combines ViBe and LBSP, which obtains excellent detection performance in general scenes. LBSP feature and pixel value areas are all compared to the historical background model (similar to ViBe). If one of them is beyond the corresponding threshold or the union of them is within a specific interval, then the target region is regarded as background. Although LOBSTER shows the ability to separating the foreground, complexity of the model structure requires significant computation load, resulting in challenges for real-time performance.

III. METHODOLOGY

In this paper, the key idea to eliminate the temporally static objects is to dynamically update the regions without objects and reserve the regions with objects. Thus, the mask of foreground and background should be used to choose which region needs to be updated. The masking strategy needs to be carefully designed to balance the system performance when updating between moving and static objects, as well as illumination or environment change. For example, if the masking strategy is too aggressive, some static objects will be included into background. Nevertheless, if the strategy is too conservative, some regions may keep the same even when the illumination changes.

A. cascaded structure

To deal with this problem, we propose a hierarchical strategy which is inspired by both *Adaptive Frame Difference* (AFD) [16] [4] and *Gaussian Mixture Model* (GMM). Either AFD or GMM can not realize satisfactory background subtraction results. AFD may result in some image faults which cannot be updated in some regions because of the cumulative errors, thus regarded as foreground. GMM may mix the objects and background together and noises are left over on the generated background. Therefore, we combine these two methods into a cascaded structure for overcoming

the shortcomings of each method, as shown in Fig. 1, where AFD accounts for choosing updated region while GMM is responsible for updating region.

More specifically, the first (AFD) module is used to obtain the background mask with fewer foreground trails and the primitive low quality background $y_{t,1}$. Then, a GMM model g_1 is applied in the second module, which takes the mask and defective background as the input and generates a relatively clean background $y_{t,2}$. The third module is another AFD module, accepting an output $y_{t,3}$ from the second GMM model g_2 as the "background" to obtain the mask. g_2 keeps being directly updated so that its result has the same global color distribution as the ground truth, and some mild blurs of shadow and vehicle are acceptable for this module. The trail region from $y_{t,1}$ is identified and updated with the background $y_{t,2}$ from g_1 by AFD. $y_{t,3}$ can bring more robust information to the background in the color space while keeping the detail in $y_{t,2}$. The output $y_{t,4}$ is fed into the third GMM model g_3 and the final background y_t can be obtained. The detailed function of each module, i.e., AFD or GMM, is introduced in the rest of this section.

B. Adaptive Frame Difference (AFD) Module

The existing AFD methods only update the region without moving objects, which fails to remove static objects. Therefore, we change it to update the region without objects. We compare the current frame I to the current background B and apply a threshold truncation to obtain a mask of the foreground M. The reverse of mask M_{bg} is expected to be able to exclude the objects.

$$\begin{aligned} d &= (B^R - I^R)^2 + (B^G - I^G)^2 + (B^B - I^B)^2 \\ M &= Threshold(d) \\ M_{bg} &= \bar{M} \end{aligned} \tag{2}$$

Then the updated region and maintained region can be obtained and updated by

$$\begin{aligned} B_{new,bg} &= (1-lr)B \cdot M_{bg} + lr \cdot I \cdot M_{bg} \\ B_{new,fg} &= B \cdot M \\ B &= B_{new,bg} + B_{new,fg} \end{aligned} \tag{3}$$

The AFD module is designed to remove objects. However, the experiments show that the AFD module has a relatively narrow tolerance to illumination change, noise, and moving objects. If the threshold is set high, edges of some objects will be included; if it is set low, camera noises and illumination will not be updated. Therefore, a GMM module, following the AFD module, is applied for compensation.

C. Gaussian Mixture Model (GMM) Module

GMM module tries to model the distribution of pixels inside the image with N Gaussians, and each of them accounts for different image features and textures. The likelihood of a pixel being a background pixel is:

$$P(I_t) = \sum_{n=1}^{N} \frac{\alpha_n}{(2\pi)^{3/2}|\Sigma_n|^{1/2}} \exp^{\frac{1}{2}(I_t-\mu_n)^T \Sigma_n^{-1}(I_t-\mu_n)} \tag{4}$$

where α_n is the associated weight for the n-th Gaussian component and $\Sigma\alpha_n = 1$. For computational simplicity, the covariance matrix, Σ_n, is set to be diagonal because the R,G,B channels are assumed to be independent.

The standard deviation and mean of the measurements are used to determine if the pixel is background. If all channels' (RGB) values are within the threshold $||\mathbf{Z}_t - \mu_n|| < k\sigma_n$, the pixel is regarded as the background. Next, model of the background is updated by

$$\begin{aligned} \alpha_n^{'} &\leftarrow (1-\delta)\alpha_n^{'} + \delta \\ \mu_n^{'} &\leftarrow (1-\rho_n^{'})\mu_n^{'} + \rho_n^{'}\mathbf{Z}_t \\ \sigma_n^{2'} &\leftarrow (1-\delta)\alpha_n^{'} + \rho_n^{'}(\mathbf{Z}_t - \mu_n^{'})^T(\mathbf{Z}_t - \mu_n^{'}) \\ \rho_n^{'} &\leftarrow \sigma\mathcal{N}(\mathbf{Z}_t|\mu_n^{'}, \sigma_n^{'}) \end{aligned} \tag{5}$$

where ρ is an updating parameter and alpha is the user-defined learning rate. In this step, the weight of background model is strength and other models' weight is shorten by $\alpha_n \leftarrow (1-\alpha)\alpha_n$. If the pixel is judged as foreground, the models' are initialized as current measurement $\alpha_n = \sigma, \mu_n = \mathbf{Z}_t, \sigma_n^2 = \overline{\sigma}^2$.

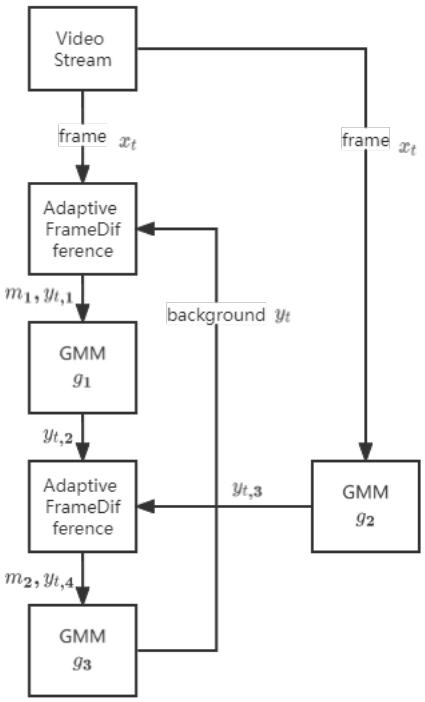


Fig. 1. The cascaded structure

IV. EXPERIMENT

To validate the performance of the proposed adaptive cascaded BS algorithm, we conducted the experiment in both the simulation environment and the real world. It is also noted that in this study we focused on the background subtraction on images from fish-eye cameras which are already installed in the real-world testbed. An effective way to measure the system performance is to compare the background mask with the ground truth background. However, it is difficult to obtain

Algorithm 1 Algorithm for Modified GMM

Input: Initialized Background X_0, Current Video Frames x_t, Learning Rate lr

Output: Background y_t

if (start video) **then**
  set $y_{t-1} = X_0$
end if
Calculate pixel-wise distance $d_1 = (x_t^R - y_{t-1}^R)^2 + (x_t^G - y_{t-1}^G)^2 + (x_t^B - y_{t-1}^B)^2$
Apply threshold function to get the binary mask. $m_1^{'} \leftarrow Threshold(d_1)$
Fill the inner holes in the mask, and apply image morphology transformation on the mask. $m_1 \leftarrow MorphoTrans(m_1^{'})$
Update the background region by $y_t^1 = (1 - lr) \cdot y_{t-1} \cdot \bar{m_1} + lr \cdot x_t \cdot \bar{m_1}$, while the foreground region keeps as last background.
$y_{t,2} \leftarrow g_1(y_{t,1})$
For k iterations, $y_{t,3} \leftarrow g_2(x_t)$
Calculate pixel-wise distance $d_2 = (y_{t,3}^R - y_{t,2}^R)^2 + (y_{t,3}^G - y_{t,2}^G)^2 + (y_{t,3}^B - y_{t,2}^B)^2$
Apply threshold function to get the binary mask. $m_2^{'} \leftarrow Threshold(d_1)$
Fill the inner holes in the mask, and apply image morphology transformation on the mask. $m_2 \leftarrow MorphoTrans(m_2^{'})$
Update the background region by $y_t^4 = (1 - lr) \cdot y_{t,2} \cdot \bar{m_2} + lr \cdot y_{t,3} \cdot \bar{m_2}$, while the foreground region keeps as last background.
$y_t \leftarrow g_3(y_t^4)$
return the background y_t

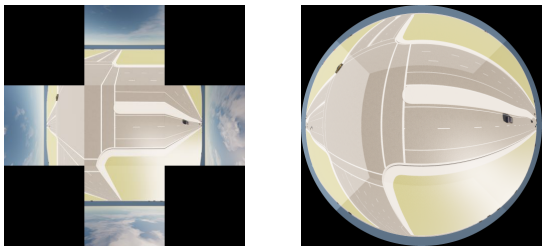
the real-time ground truth background in a real-world setting at the pixel level. Thus, we use the simulation-based dataset from CARLA [17] for the quantitative analysis, where the ground truth of the pixel-wise segmentation is available.

In the real-world experiment, a fisheye camera is deployed at a real-world intersection of University Avenue and Iowa Avenue in Riverside, California, to obtain the video stream of the traffic all day long. The fisheye camera faces vertically to the ground for the broadest cover of intersections and roads.

Also, the fisheye camera is different to ordinary cameras for its unique FOV and distortion that objects at the fisheye image edge are stretched. These features also leads to the failures of feature-based methods. It is necessary and critical to calibrate the fisheye images. Also, wide FOV lens distortion calibration can not be applied in fisheye camera for it is limit to a relative narrow FOV (generally 120°). In fact, fisheye camera calibration [18] is mature and widely used in fisheye camera system. In our experiment, fisheye camera calibration is used in both the simulation dataset and in the real-world dataset.

A. Simulation Experiment

To examine the feasibility of the proposed algorithm, we first implemented the algorithm in a CARLA-based simulation environment. We used the map Town10HD provided by CARLA for simulation, which presents a typical urban arterial network including a few signalized intersections. We chose one intersection in the map, as shown in Fig. 3, for collecting video frames at the traffic light where vehicles may stop to wait for the signals (to turn green). In the simulation environment, 32 vehicles were spawned in the map, randomly traversing the intersection. In addition, because there is no ready-to-use fish-eye camera model provided in CARLA, we adopted a transformation method called *cubic2fisheye* [19] to synthesize the fish-eye video by combining multiple regular cameras from different perspectives. The frame rate is 10 fps and the resolution is 960×960.



(a) The sewed up cubic camera image, and there is no edge error because of 90 degree FOV

(b) Fisheye image

Fig. 2. Cubic to Fisheye

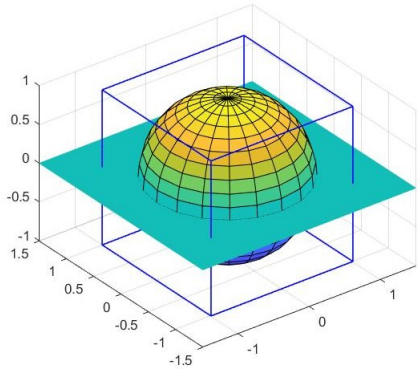


Fig. 3. The cubic camera model and the hemisphere

1) Cubic to Fisheye: We set five cameras at the same position with different angles, so they can take the image of each surface of a cubic, as shown in Fig. 2(a). These five cameras are located at the cubic center and faced to the five sides except the bottom. To ensure that every camera has no overlapped field of view (FOV) and no deficient FOV, we set the FOV of every camera 90 degrees, enabling the alignment of images from all cameras. With this configuration, the cubic camera may obtain RGB information from every direction except the bottom one. Then, we projected every pixel from the top half of the cubic surface to a

hemispherical surface. After getting this transformation, we applied a polynomial fitting model to project it on a 2-D plane. Fig. 3 shows the relationship between the azimuth angle and the radius on the 2-D plane, which is considered as a polynomial fitting model:

$$r(\theta) = k_1\theta + k_2\theta^2 + k_3\theta^5 + k_4\theta^7 + k_5\theta^9 \quad (6)$$

As is shown in Fig. 2(b), the fish-eye image can cover 180 degrees of FOV. The outer blue circular ring is the sky which means this image includes all-direction skyline.

2) Results and Analyses: Fig. 4 demonstrates a sample frame showing the synthesized fish-eye image, ground truth label, and the BS results of the proposed algorithm, GMM algorithm, ViBe, PBAS, as well as LOBSTER. We adopted the *confusion matrix* as the metric for the CARLA dataset. After applying the BS algorithms, the generated foreground masks were compared to the ground truth labels at the pixel level. The pixels of the overlapped region and non-overlapped region are obtained, so that the confusion matrix of each frame is available. Then the True Positive (*TP*), False Positive (*FN*), True Negative (*TN*), and False Positive (*FP*) are normalized and averaged across all the pixels in the dataset along the entire simulation time span. Other induced metrics, including precision, recalls, FPR, FNR, and F-Measure, are introduced for comparison.

- Precision (*Pr*): $TP/(TP+FP)$, larger is better
- Recall (*Re*): $TP/(TP+FN)$, larger is better
- False Positive Rate (*FPR*): $FP/(FP+TN)$, smaller is better
- False Negative Rate (*FNR*): $FN/(FP+TN)$, smaller is better
- Percentage of Wrong Classification (*PWC*/*PBC*) : $100\cdot(FP+FN)/(TP+FN+FP+TN)$, smaller is better
- F-Measure (*F1*): $(2\cdot Pr\cdot Re)/(Pr+Re)$, larger is better

The ground truth labels (foreground masks) are generated by the segmentation fish-eye camera, so that any vehicles' shadows are not included in the labels. However, most BS algorithms cannot distinguish the shadows from objects so that the shadows are regarded as part of objects. This is the main reason for those large FP values.

As can be observed from TABLE I, our method can significantly retain and subtract static foreground, while other methods may recognize some objects as background. Therefore, our method's *FN* is much lower than all other methods and our method reaches a higher recall. Compared to the SOTA (LOBSTER), our method shows better *TP* and *FN*. A hypothesis is that as a feature-based method, LOBSTER may be influenced by the distortion from the fish-eye camera and discriminate some foreground at edge region as background. It should be noted that the measurement was conducted at the pixel level in this study. Although the average precision is not very high, it is good enough for accurate detection and localization applications. In addition, our method outperforms LOBSTER in terms of F-Measure (a comprehensive metric) by 0.2096, showing the superiority.

B. Real-world Experiment

To evaluate the system performance under dynamic illumination and weather (e.g., windy) conditions, we collected a video clips/dataset from the fish-eye camera installed at the intersection of University Avenue and Iowa Avenue in Riverside, California. This video clip lasts for seven hours and is collected from morning to afternoon on 02/21/2022. The first half dataset is collected with cloudy condition and the rest part is collected with sunlight condition so the dataset includes shadow and sunlight variation. The frame rate is 10 fps and the resolution is 960×960.

Fig. 5 demonstrates a sample frame at 35 seconds showing the undistorted image from the fish-eye camera and the BS results of the proposed algorithm, GMM algorithm, as well as ViBE algorithm. Although it is challenging to quantify the performance with the real-world datasets, the visualization results indicate that our method is still reliable to identify the vehicles waiting at signals, trees swaying in the wind, and the rapid illumination changes due to the cloud movement. However, none of GMM, ViBe, PBAS and LOBSTER are capable of detecting temporally static objects due to iterative updating. They may consider those target vehicles as background. The longer the vehicles wait, the more likely they are classified as background. LOBSTER also includes some edge blobs into foreground mask due to the initialization being much longer than other methods.

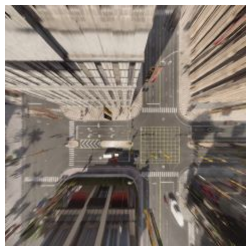
Fig. 6 shows a sample frame of shadow variation after 5.05 hours and the related BS results of the proposed algorithm. Our method successfully recovers shadow and illumination variation into background also separate vehicles from the frame no matter moving or idling. Other methods also recover shadow but some static vehicles or slowly variation are included into background.

Further more, our method also can process the real-world dataset in real-time. GMM, ViBe and PBAS can also meet the real-time requirement, while LOBSTER takes 3 times as long as our method.

V. Conclusion

In this paper, a cascaded adaptive background subtraction method is proposed for the need to separate the temporally static objects (e.g., vehicles waiting at signals) in the traffic scenario at signalized intersections. From the experiments in CARLA datasets, our method outperforms ViBe and LOBSTER by 44.89% and 38.48%, respectively, on recall, while keeping a relative high precision. In the real world datasets, the results also show that the proposed method has more reliable segmentation performance than GMM, ViBe, PBAS and LOBSTER.

In terms of future work, the hierarchy structure can be reinforced by the feature-based methods such as LBSP for ordinary camera videos. Moreover, a higher-efficiency realization will be explored by the implementation with CUDA coding so that the system can be run on GPU for faster speed.



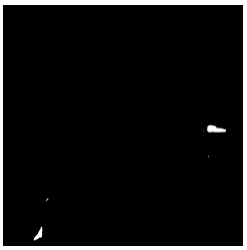
(a) One frame in CARLA dataset

(b) Ground truth label

(c) Our method

(d) GMM

(e) ViBe



(f) PBAS

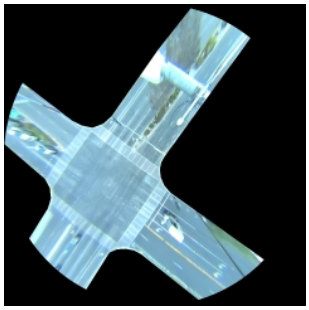
(g) LOBSTER

Fig. 4. CARLA dataset and experiments

TABLE I

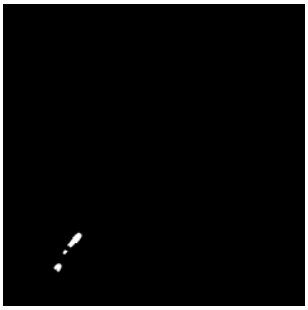
CONFUSION MATRIX OF METHODS

Method	TP	FN	TN	FP	Precison	Recall	FPR	FNR	*PWC*/*PBC*	F-Measure	FPS
Our Method	**1.242%**	**0.415%**	98.243%	0.618%	0.668	**0.750**	0.006	**0.004**	**0.010**	**0.707**	8.569
GMM	0.265%	1.426%	**98.665%**	**0.196%**	0.575	0.157	**0.002**	0.014	0.016	0.250	23.898
ViBe	0.509%	1.182%	**98.665%**	0.206%	0.712	0.301	0.002	0.012	0.014	0.430	29.689
PBAS	0.662%	1.029%	98.37%	0.489%	0.561	0.392	0.005	0.010	0.015	0.470	8.44
LOBSTER	0.616%	1.075%	98.665%	0.206%	**0.749**	0.364	0.002	0.011	0.013	0.497	3.297

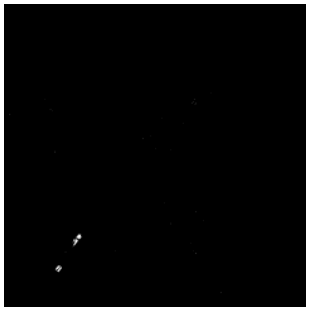


(a) One frame of real-world dataset

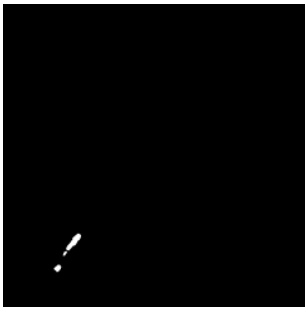
(b) Our method



(c) GMM



(d) ViBe



(e) PBAS

(f) LOBSTER

Fig. 5. Real world dataset and experiments at 35 seconds

(a) One frame of real-world dataset (b) Our method (c) GMM

(d) ViBe (e) PBAS (f) LOBSTER

Fig. 6. Real world dataset and experiments at 18180 seconds

ACKNOWLEDGMENTS

This research was funded by the Toyota Motor North America InfoTech Labs. The contents of this paper reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views of Toyota Motor North America.

REFERENCES

[1] Z. Bai, S. P. Nayak, X. Zhao, G. Wu, M. J. Barth, X. Qi, Y. Liu, and K. Oguchi, "Cyber mobility mirror: Deep learning-based real-time 3d object perception and reconstruction using roadside lidar," 2022.

[2] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, "Deep neural network concepts for background subtraction: A systematic review and comparative evaluation," *Neural Networks*, vol. 117, pp. 8–66, 2019.

[3] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Computer Vision and Image Understanding*, vol. 122, pp. 4–21, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314213002361

[4] Y. Benezeth, P. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms," in *2008 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[5] B. Lee and M. Hedley, "Background estimation for video surveillance," *IVCNZ 2002*, pp. 315–320, 01 2002.

[6] A. Lai and N. Yung, "A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence," in *1998 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 4, 1998, pp. 241–244 vol.4.

[7] M.-H. Sigari, N. Mozayani, and H. Pourreza, "Fuzzy running average and fuzzy background subtraction: Concepts and application," *International Journal of Computer Science and Network Security*, vol. 8, 01 2008.

[8] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol. 2, 1999, pp. 246–252 Vol. 2.

[9] Hayman and Eklundh, "Statistical background subtraction for a mobile observer," in *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003, pp. 67–74 vol.1.

[10] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, 2004, pp. 28–31 Vol.2.

[11] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern recognition letters*, vol. 27, no. 7, pp. 773–780, 2006.

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

[13] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *2012 IEEE computer society conference on computer vision and pattern recognition workshops*. IEEE, 2012, pp. 38–43.

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

[15] P.-L. St-Charles and G.-A. Bilodeau, "Improving background subtraction using local binary similarity patterns," 03 2014, pp. 509–515.

[16] M. M. Azab, H. A. Shedeed, and A. S. Hussein, "A new technique for background modeling and subtraction for motion detection in real-time videos," in *2010 IEEE International Conference on Image Processing*, 2010, pp. 3453–3456.

[17] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[18] E. Schwalbe, "Geometric modelling and calibration of fisheye lens camera systems," in *Proc. 2nd Panoramic Photogrammetry Workshop, Int. Archives of Photogrammetry and Remote Sensing*, vol. 36, no. Part 5, 2005, p. W8.

[19] P. ZIMMONS, "Spherical, cubic, and parabolic environment mappings," *http://www. cs. unc. edu/zimmons/cs238/maps/environment. html*.