2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Coarse-to-fine Foreground Segmentation based on Co-occurrence Pixel-Block and Spatio-Temporal Attention Model

Dong Liang and Xinyu Liu
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics,
MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, 211106, China
Corresponding author: Dong Liang (liangdong@nuaa.edu.cn)

***Abstract*—Foreground segmentation in dynamic scene is an important task in video surveillance. The unsupervised background subtraction method based on background statistics modeling has difficulties in updating. On the other hand, the supervised foreground segmentation method based on deep learning relies on the large-scale of accurately annotated training data, which limits its cross-scene performance. In this paper, we propose a foreground segmentation method from coarse to fine. First, a across-scenes trained Spatio-Temporal Attention Model (STAM) is used to achieve coarse segmentation, which does not require training on specific scene. Then the coarse segmentation is used as a reference to help Co-occurrence Pixel-Block Model (CPB) complete the fine segmentation, and at the same time help CPB to update its background model. This method is more flexible than those deep-learning-based methods which depends on the specific-scene training, and realizes the accurate online dynamic update of the background model. Experimental results on WallFlower and LIMU validate our method outperforms STAM, CPB and other methods of participating in comparison.**

I. Introduction

[1] Foreground segmentation plays an important role in intelligent video monitoring [1]. Traditional foreground segmentation methods usually rely on the background statistical modeling which is an unsupervised training process. Usually, the eigenvalues of each pixel are sampled and counted in the time domain, to build the statistical model. For example, Gaussian Mixture Model (GMM) [2] or Kernel Density Estimation (KDE) [3]. Spatial-dependence model [4], [5], [6], which exploit spatial-dependence among pixels to build local or global models, is widely used to explore context information but performs poor when the background is texture-less. Background subtraction methods proposed in recent years include [7], [8].

The main problem of these methods lie in updating strategy which is using a learning rate function. However, a manually set learning rate always has a well-known trade-off problem. In order to adapt to the sudden change of illumination, the learning rate is usually high. As a result, slowly moving objects or temporarily stopped objects will be detected as background. BMOG [9] which based on Mixture of Gaussians explores a novel classifcation mechanism that combines color space discrimination capabilities with hysteresis and a dynamic learning rate for background model update. The learning rate will turn into a fixed minimum value when a pixel turning from background to foreground. In the opposite case, the learning rate will increase. However, the strategy is unstable which means any wrong detection may cause subsequent errors.

On the other hand, foreground segmentation methods based on convolutional neural network have also emerged in recent years [10], [11], [12], [13]. DeepBS [14] utilizes a trained convolutional neural network and a spatial-median filterer to implement foreground detection in various video scenes. As the foreground is detected based on independent frame, the temporal relevance of the neighbouring frames is ignored. Cascade CNN [15] proposed a semi-automatic method which release pressure on amount of training data. CNN branches processing images in different size are cascaded together that helps the cascade CNN to detect foreground objects in multi-scale. FgSegNet [16], [17] encodes the features of three different scales of the same input image with three sets of CNN encoders. TCNN (transposed convolutional neural network) is used to decode the multi-scale features to obtain the pixel-level foreground segmentation mask.

As is known to all, the training samples of segmentation tasks need to be manually annotated, which is expensive. It is the difficulty in implementing the rapid labelling and training that limits its popularization and application in video monitoring tasks. In order to improve the accuracy of pixel level segmentation in the scene that has been trained, these models are often over-fitting, resulting in its poor generalization ability to the untrained scene. Some methods [14], [18], gained capability of foreground segmentation under cross-scene, after training with the large-scale multi-scene data. At present, the performance of foreground segmentation based on convolutional neural network in untrained scenes is generally worse than that

[1]Dong Liang and Xinyu Liu contributes equally to this work. This work is supported by the National Key R&D Program of China under Grant 2017YFB0802300.

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412814

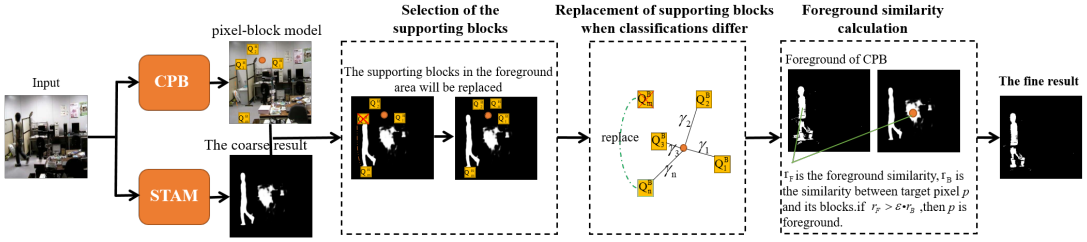

Fig. 1. The proposed method

of traditional background modeling methods.

In recent years, we proposed the Co-occurrence Pixel-Block Model (CPB) [19], [20]. Its early work included [21], [22]. This method uses the spatial correlation between pixels to segment the foreground, so that the background model can automatically adapt to the dynamic background. However, the training process of this method relies on the calculation of linear correlation between pixels under the condition of large amount of data, which makes it difficult to update online. The segmentation performance of this method will gradually decrease over time. And it makes this method more suitable for offline applications. Recently, we proposed a Spatio-Temporal Attention Model (STAM) [18] for foreground segmentation cross-scene. It combines spatio-temporal information and uses the attention module to fuse the features of the encoder and decoder. It also takes the single frame as well as its optical flow information as input which guarantees its sensitivity to moving targets. However, the cross-scene segmentation STAM obtains is usually coarse, indicating that the result needs further refinement.

In this paper, we propose a foreground segmentation method from coarse to fine. First, a trained STAM is used to achieve a coarse segmentation across scenes. Then the coarse segmentation result is used as a reference to help CPB complete the fine segmentation. At the same time the proposed method also helps CPB to have its background model updated online. On WallFlower [23] and LIMU [24] dataset that are not training by STAM, the performance of the proposed method is better than that of the single STAM model and CPB model, as well as a lot of other methods participating in comparison.

II. Description of the Proposed Method

The proposed method framework is shown in Figure 1. First, STAM is used to get the coarse segmentation result. Then, Through three steps which are selection of the supporting blocks, replacement of supporting blocks and calculation of foreground similarity, the coarse result is used to help CPB complete the fine segmentation.

A. Spatio-Temporal Attention Model (STAM)

STAM [18] can be seen as an attention-guided weightable connection encoder-decoder, to preserve the effective connections and suppress the invalid connection. It

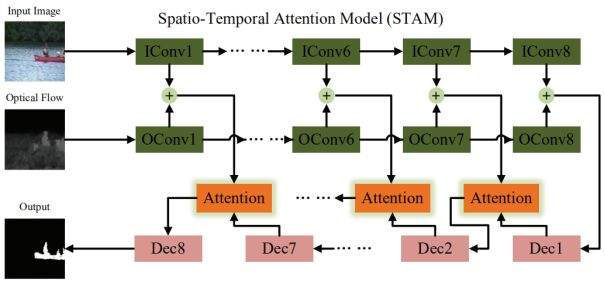

Fig. 2. Spatio-Temporal Attention Model (STAM)

integrates the features of the decoder and encoder by introducing the attention module in the decoding stage. The high-level features provide global information to guide the attention module to select appropriate low-level features. These low-level features are helpful for binary prediction of the input frame. As shown in Figure 2, the model combines spatial and temporal information, and the attention module is employed to mix encoder features together with decoder ones. The blocks in green represent the encoder layers and "IConv" and "OConv" are two encoders fed with static image and optical flow, respectively. They have the same structure and eight convolution layers. Additionally, the decoder has eight layers and up-sampling processed in each layer and seven attention modules are applied to make features mixtures. The blocks in pink and orange represent the decoder layers and attention modules. The plus sign in green means the addition in pixel-level. The static frame and its optical flow (motion cue) feed two encoders, and reorganized by attention modules to reconstruct the foreground in pixel-level. Compared to the model without motion cue, this model introduce useful temporal informations.

The feature fusion method we adopted in the attention mechanism in [18] is shown in the left part of Figure 3, which is marked as the A mode. The attention module merges high-level and low-level features under the guidance of the former ones. $Y1$ and $Y2$ are the encoder features of the input frame and optical flow, respectively, and X is the decoder feature. H, W and C are the height, width and channel number of the feature map, respectively. It applies a single convolution operation $conv()$ onto X followed by a sigmoid activation function σ that makes the weights

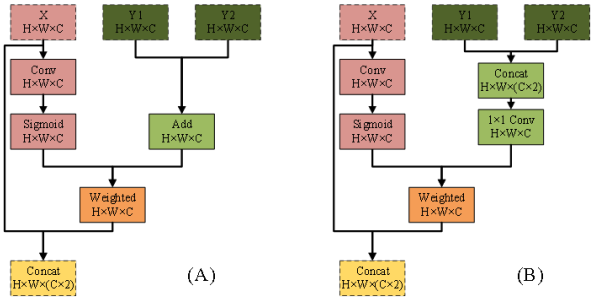

Fig. 3. Comparison of two different feature fusion methods

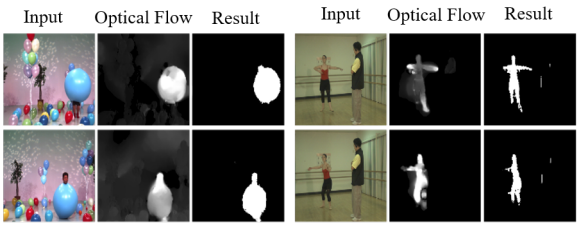

Fig. 4. Segmentation results of STAM on the MSR 3D Video dataset

belong to 0 to 1. Where b is the bias value of a convolution operator. Then it uses $f_{weights}$ to weight the sum of the encoder features. Finally, the decoder feature X and the re-weighted features are concatenated f_{output} as the input of next convolutional layer.

$$f_{weights} = \sigma(conv(X) + b) \tag{1}$$

$$f_{output} = concat(f_{weights} \otimes (Y1 \oplus Y2), X) \tag{2}$$

where $\otimes$ and $\oplus$ denote the pixel-wise multiplication and sum operation, and $concat(,)$ is a concatenate process on two features.

The fusion method shown in the right part of Figure 3 is marked as the B mode. The difference of the two modes is that different methods are used to process the appearance and motion features in the same layer. A mode merges two input features by adding corresponding pixels while in B mode, we first connect these two features by channel, the corresponding channel number is doubled, and then use the 1×1 convolution to process the result to reduce the dimensions of the channel. Feichtenhofer [25] pointed out that because each channel in the network expresses different semantic information and the semantic information of each channel is arbitrary. The fusion method of directly adding corresponding channels and pixels cannot guarantee that the semantic information expressed by the two features involved in the operation is consistent. While B first connects the two channels without considering the correlation between the different channels. And then uses the subsequent layer to learn the correlation information between the different channels. Experiments prove that B mode is a better choice, which can improve the model's F-measure and increase Recall.

STAM uses randomly 5% training data with groundtruth on the CDNet2014 dataset [26], and we can get foreground segmentation result on other datasets without retraining it on specific scene. Figure 4 shows the segmentation results of STAM on the MSR 3D Video dataset [27], which is used in the field of view synthesis [28], [29]. In the scene on the left, the background has flickering lights and the horizontal movement of the entire screen. The scene on the right has another human body and shadows which are unfavourable for foreground segmentation.

With a trained STAM model, we can get foreground segmentation results on many other datasets directly. There is no need to retrain it on specific scene because pixel-level labeling takes time and effort. However, the cross-scene segmentation results STAM gives are usually coarse, indicating that the results need further refinement.

B. Co-occurrence Pixel-Block Model (CPB)

The CPB model includes two stages: training process and detection process.This method compares the target pixel p with its supporting block Q^B to determine whether p belongs to foreground.

The co-occurrence supporting blocks of target pixel p is defined as $\{Q_M^B\}_{m=1,2,...,M} = \{Q_1^B, Q_2^B, ..., Q_M^B\}$. Those supporting blocks are selected by using Pearson product-moment correlation coefficient.

$$\{Q_M^B\}_{m=1,2,...,M} = \{Q^B | M \ largest \ \gamma(p, Q^B)\}. \tag{3}$$

$$\gamma(p, Q_m^B) = \frac{C_{p,\overline{Q}_m^B}}{\sigma_p \cdot \sigma_{\overline{Q}_m^B}}. \tag{4}$$

$C_{p,\overline{Q}_m^B}$ is the intensity covariance of the target pixel p and its supporting block Q_m^B. σ_p and $\sigma_{\overline{Q}_m^B}$ are the standard deviations of the intensity values of p and Q_m^B, respectively. Each target pixel p corresponds to several supporting blocks Q^B. They maintain a stable relationship over time, that is, the difference in intensity follows a single Gaussian distribution:

$$(I_p - \overline{I}_{Q_m^B}) \sim N(b_m, \sigma_m^2). \tag{5}$$

I_p is the intensity value of target pixel p and $\overline{I}_{Q_m^B}$ is the average intensity value of supporting block Q_m^B.

After the training process, the CPB model obtains all the supporting blocks $\{Q_M^B\}_{m=1,2,...,M}$ of each target pixel p. The state of each pixel-block pair (p, Q_m^B) is defined as follows:

$$\omega_m = \begin{cases} 1 & \text{if } |I_p - \overline{I}_{Q_m^B}| \leq \eta \cdot \sigma_m \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

η is a threshold of Gaussian model. Considering the difference in correlation between each target pixel and its supporting blocks, their correlation coefficients γ_m are

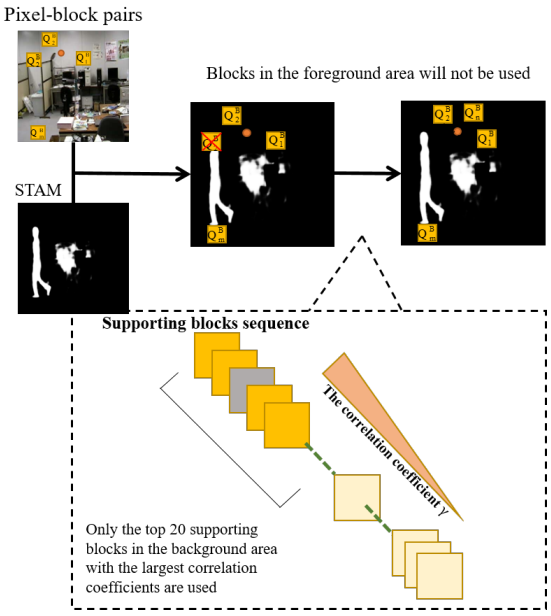

Fig. 5. Selection of supporting blocks

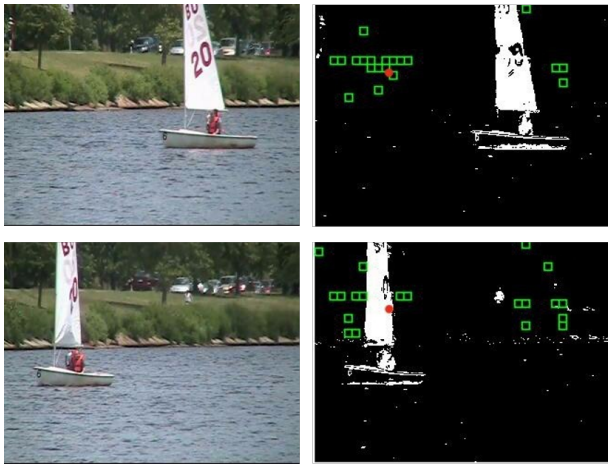

Fig. 6. The same target pixel (the red one in the figure) has different supporting blocks (the green ones in the figure) at different time. When a ship passes by, the model will always select the supporting blocks in the background area.

introduced as weights. CPB will classify the target pixel p as foreground when the following conditions are met:

$$\sum_{m=1}^{M} \gamma_m \cdot \omega_m > \lambda \cdot \sum_{m=1}^{M} \gamma_m \tag{7}$$

λ is relevance decision threshold.

The CPB model relies on the strong correlation between target pixel p and its supporting blocks $\{Q_m^B\}_{m=1,2,...M}$, which cannot be updated after training process. It's the lack of update capability which would cause model performance to degrade over time and limits the widespread use of the model.

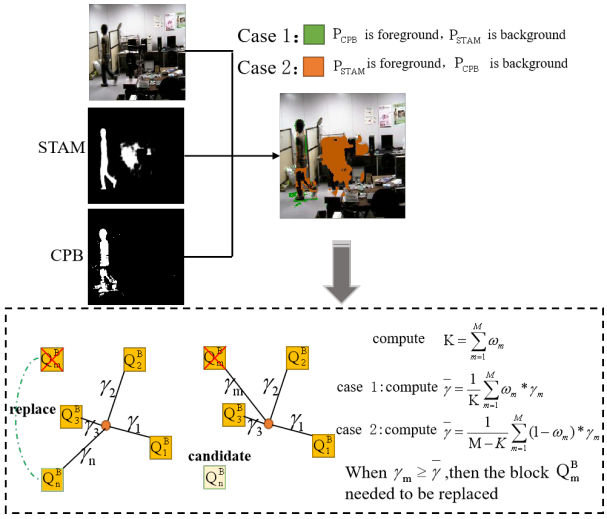

Fig. 7. Replacement of supporting blocks when classifications differ

C. Selection of the supporting blocks

When the foreground object covers the supporting block, it will obviously cause the Gaussian relationship between the supporting block and its target pixel to be destroyed. The state of pixel-block pair which is defined by Equation 6 is temporarily invalid, resulting in segmentation errors. The segmentation result given by the STAM model has a high probability as foreground. The performance of the CPB model will be effectively improved by avoiding the selection of supporting blocks in the foreground area. It can be seen in Figure 5 that only the supporting blocks in the background area will be used, while the blocks in foreground marked by gray will be temporarily discarded. The candidate supporting blocks marked by light yellow are obtained during training process. It turns out the proposed method works very well especially when there is a large foreground area which means a large number of pixel-block pairs are in a structure of failure. Figure 6 shows the selection process of supporting blocks when a boat passing by.

D. Replacement of supporting blocks when classifications differ

From above, we try not to choose supporting blocks that are in the foreground area. However, first, there is a certain gap between the STAM's segmentation result and the groundtruth, so it is possible that some of the supporting blocks are still in foreground area. Secondly, the pixel-block model which is obtained during the training process may generate a degredation over time, because the background is not static, such as cloud drift in the sky or entry/exit of vehicles in the parking lot. As a result, foreground or background "noise" might arise in the detection process.

Those pixels that CPB and STAM model's classifications differ, as shown in Figure 7, the green and orange areas. They are specifically divided into: Case 1) CPB considers it

as foreground while STAM classifies it as background; Case 2) CPB considers it as backfround while STAM classifies it as foreground.

Consider the difference of the linear correlation coefficient γ_m between each supporting block Q_m^B and the target pixel p. CPB uses the correlation coefficient γ_m as the weight value. When the result is different from that of the STAM's, the supporting blocks corresponding to the high correlation coefficient value need to be responsible for potential errors, which may already be in the state of structural failure. The strategy is as follows:

$$K = \sum_{m=1}^{M} \omega_m \tag{8}$$

$$\overline{\gamma} = \begin{cases} \frac{1}{K} \sum_{m=1}^{M} \gamma_m \cdot \omega_m & \text{case 1} \\ \frac{1}{M-K} \sum_{m=1}^{M} \gamma_m \cdot (1-\omega_m) & \text{case 2} \end{cases} \tag{9}$$

If the correlation coefficient between the supporting block $Q_m^B$and target pixel p satisfies:

$$\gamma_m \geq \overline{\gamma} \tag{10}$$

Then Q_m^B needs to be replaced.

As shown in Figure 7, when considering those target pixels that have different classifications, their supporting blocks which are selected by Equation 10 will be temporarily discarded, and the candidate supporting blocks represented by the light yellow will replace them as the new supporting blocks of the target pixel.

When STAM and CPB have different segmentation results at the target pixel p, we take STAM's results as a guide. Considering the possibility of errors in the segmentation results of CPB, we decide to replace the supporting blocks that may already be in a structural failure state with candidate blocks. At the same time, the robustness of CPB also means that this solution will not cause the degradation of segmentation performance. The coming relevant comparative experimental results also confirm the rationality of the proposed solution.

E. Calculation of foreground similarity

The construction of the pixel-block model is based on the correlation of their eigenvalues in the long-term domain. The supporting block with a high correlation coefficient should be an area that is homogeneous with its target pixel. When the pixel p is the foreground and it is misclassified by CPB as the background (which is, case 2), calculate the similarity (the Euclidean distance in image space) between the pixel and the surrounding foreground pixels r_F, and calculate the average similarity between the pixel and all its supporting blocks r_B. If it meets the following equation, p will be classified as foreground.

$$r_F > \varepsilon \cdot r_B \tag{11}$$

ε is similarity decision threshold.

III. Experiments

A. Settings

The CDNet2014 dataset contains a large number of different scenes. The STAM model is trained on the CDNet2014 dataset by taking random 5% data and its groundtruth. In order to verify the generalization ability of the proposed method under cross-scene, we use WallFlower and LIMU datasets. In other words, we train STAM on CDNet2014 and do foreground segmentation on WallFlower and LIMU. In the comparison experiment, Cascade CNN and FgSegNet are supervised learning methods and they take the same strategy as STAM.

For the CPB model we use the following strategy. On WallFlower dataset, we adopt the strategy provided by the dataset itself, that is, using the provided 200 frames as training set. On LIMU dataset, we choose the first 400 frames for training. The experimental parameter settings are shown in Table I.

B. Results and evaluation

Comparative experiments were performed on a total of 10 scenes of WallFlower and LIMU. The experimental setting allows us to verify the cross-scene performance of the proposed method against other methods.

The experimental results on the WallFlower dataset are shown in Table II, III and Figure 8. Taking F-measure as the evaluation, the method proposed in this paper achieves the highest score on Bootstrap, ForegroundAperture, LightSwitch, TimeOfDay, and performs better than CPB on all scenes. It also performs best on average across all scenes. An average increase of 0.1215 over CPB and 0.1109 over STAM. MovedObject is a scene which is used to test the update ability of the background model. There is no foreground target in the groundtruth. So Specificity = TN / (TN + FP) is selected as the evaluation. From Table III, it can be seen that the method proposed in this paper ranks first among all methods. As shown in the fifth line of Figure 8, the armchair in the scene has been moved which causes CPB detect it as foreground. In contrast, the proposed method performs well in this situation. WavingTrees is a scene that branches sway back and forth. GMM is very effective in modeling the multimodal distribution background, especially this kind of scene with tiny repetitive motion.

For the LIMU dataset, comparative experiments

TABLE I
PARAMETER SETTINGS

Parameter	Value
number of supporting blocks K	20
number of candidate supporting blocks	10
Gaussian model threshold η	2.5
Relevance decision threshold λ	0.5
Similarity decision threshold ε	0.8

TABLE II
F-MEASURE OF DIFFERENT METHODS ON WALLFLOWER

Method	Bootstrap	Camouflage	ForegroundAperture	LightSwitch	TimeOfDay	WavingTrees	Overall
The proposed	**0.7560**	0.6884	**0.9402**	**0.9097**	**0.7949**	0.6665	**0.7929**
STAM[18]	0.7414	0.7369	0.8292	0.9090	0.3429	0.5325	0.6820
DeepBS[14]	0.7479	**0.9857**	0.6583	0.6114	0.5494	0.9546	0.7512
Cascade CNN[15]	0.5238	0.6778	0.7935	0.5883	0.3771	0.2874	0.5413
FgSegNet[16]	0.3587	0.1210	0.4119	0.6815	0.4222	0.3456	0.3902
CPB[19]	0.6518	0.6112	0.5900	0.7157	0.7564	0.7033	0.6714
SuBSENSE[6]	0.4192	0.9535	0.6635	0.3201	0.7107	0.9597	0.6711
GMM[2]	0.5306	0.8307	0.5778	0.2296	0.7203	**0.9767**	0.6443
PBAS[30]	0.2857	0.8922	0.6459	0.2212	0.4875	0.8421	0.5624

TABLE III
SPECIFITY OF DIFFERENT METHODS ON MOVEDOBJECT

The proposed	STAM[18]	Cascade CNN[15]	FgSegNet[16]	CPB[19]
0.9977	0.9949	0.7763	0.8470	0.8922

Fig. 8. The comparison of different methods on WallFlower

were performed on CameraParameter, Intersection, and LightSwitch. The experimental results are shown in Table IV and Figure 9. Taking F-measure as the evaluation, The proposed method ranks first on all three scenes. The average F-measure is 0.3154 higher than STAM and 0.1137 higher than CPB. From Figure 9 we can also see that the proposed method well suppresses false positive which is marked with orange.

IV. Conclusion

Based on CPB model, we use the STAM segmentation result as a guide to complete a coarse-to-fine foreground segmentation and help CPB to have its background model updated. Experiments show that the proposed method has a higher performance than CPB on all the experimental datasets. In a total of 10 scenes of WallFlower and LIMU datasets that have not been trained by STAM, the proposed method ranks first on 8 datasets except Camouflage and WavingTrees. In summary, the proposed method is significantly better than STAM and CPB in cross-scene, and is superior to other methods that participate in comparison. Future work will be to further explore the replaceability of the STAM module in the proposed method and the corresponding comparison after applying different methods as segmentation references.

References

[1] A. Vacavant, T. Chateau, A. Wilhelm, and L. LequiÍlvre, "A benchmark dataset for outdoor foreground/background extraction," *Assian Conference on Computer Vision*, pp. 291–300, 2012.

[2] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, vol. 2, pp. 246–252, 1999.

[3] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.

TABLE IV
F-MEASURE OF DIFFERENT METHODS ON LIMU

Method	CameraParameter	Intersection	LightSwitch	Overall
The proposed	**0.7484**	**0.7672**	**0.8211**	**0.7789**
STAM[18]	0.6742	0.6237	0.0953	0.4644
Cascade CNN[15]	0.1025	0.0453	0.0277	0.0585
FgSegNet[16]	0.2668	0.1428	0.0414	0.1503
CPB[19]	0.6545	0.6778	0.6633	0.6652

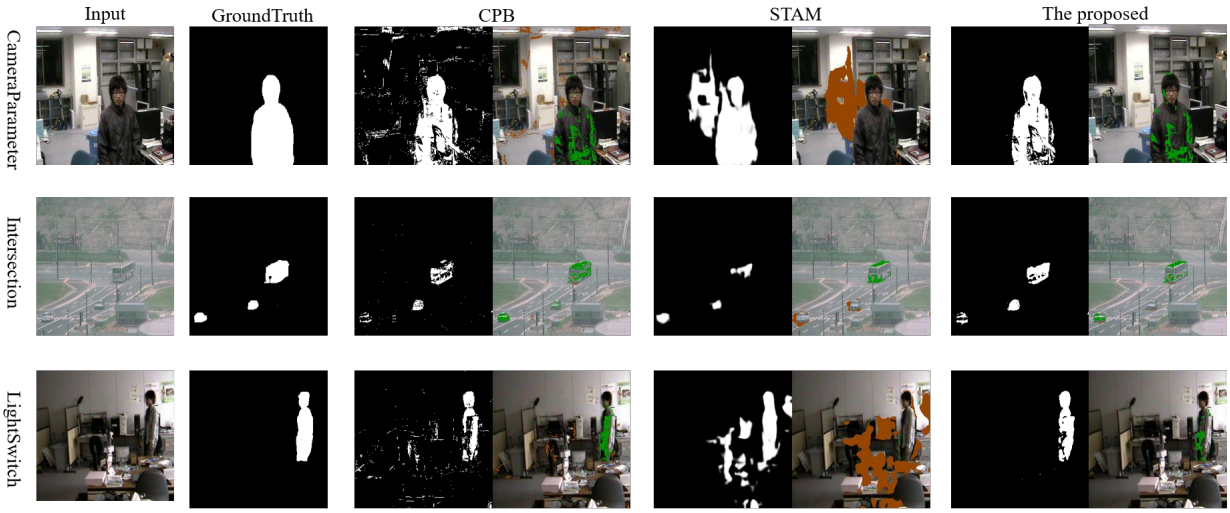

Fig. 9. The comparison of different methods on LIMU

[4] P. Jodoin, M. Mignotte, and J. Konrad, "Statistical background subtraction using spatial cues," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 12, pp. 1758–1763, 2007.

[5] 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.

[6] P. Stcharles, G. Bilodeau, and R. Bergevin, "Subsense: A universal change detection method with local adaptive sensitivity," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 359–373, 2015.

[7] A. T. Chen, M. Biglariabhari, and K. I. Wang, "Superbe: computationally light background estimation with superpixels," *Journal of Real-time Image Processing*, vol. 16, no. 6, pp. 2319–2335, 2019.

[8] S. M. Roy and A. Ghosh, "Real-time record sensitive background classifier (rsbc)," *Expert Systems With Applications*, vol. 119, pp. 104–117, 2019.

[9] I. Martins, P. Carvalho, L. Cortereal, and J. L. Albacastro, "Bmog: boosted gaussian mixture model with controlled complexity for background subtraction," *Pattern Analysis and Applications*, vol. 21, no. 3, pp. 641–654, 2018.

[10] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," *International Conference on Systems*, pp. 1–4, 2016.

[11] G. Shi, T. Huang, W. Dong, J. Wu, and X. Xie, "Robust foreground estimation via structured gaussian scale mixture modeling." *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4810–4824, 2018.

[12] C. Zhao, T. Cham, X. Ren, J. Cai, and H. Zhu, "Background subtraction based on deep pixel distribution learning," *International Conference on Multimedia and Expo*, pp. 1–6, 2018.

[13] M. Qiu and X. Li, "A fully convolutional encoder-decoder spatial-temporal network for real-time background subtraction," *IEEE Access*, vol. 7, pp. 85 949–85 958, 2019.

[14] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for background subtraction." *arXiv: Computer Vision and Pattern Recognition*, 2017.

[15] Y. Wang, Z. Luo, and P. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognition Letters*, vol. 96, pp. 66–75, 2017.

[16] L. A. Lim and H. Y. Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognition Letters*, vol. 112, pp. 256–262, 2018.

[17] L. A. Lim and H. Y. Keles, "Learning multi-scale features for foreground segmentation," *Pattern Analysis and Applications*, pp. 1–12, 2019.

[18] D. Liang, J. Pan, H. Sun, and H. Zhou, "Spatio-temporal attention model for foreground detection in cross-scene surveillance videos," *Sensors*, vol. 19, no. 23, p. 5142, 2019.

[19] W. Zhou, S. Kaneko, M. Hashimoto, Y. Satoh, and D. Liang, "A co-occurrence background model with hypothesis on degradation modification for object detection in strong background changes," *International Conference on Pattern Recognition*, pp. 1743–1748, 2018.

[20] W. Zhou, S. Kaneko, M. Hashimoto, Y. Satoh, and D. Liang, "Foreground detection based on co-occurrence background model with hypothesis on degradation modification in dynamic scenes," *Signal Processing*, vol. 160, pp. 66–79, 2019.

[21] D. Liang, S. Kaneko, M. Hashimoto, K. Iwata, and X. Zhao, "Co-occurrence probability-based pixel pairs background model for robust object detection in dynamic scenes," *Pattern Recognition*, vol. 48, no. 4, pp. 1374–1390, 2015.

[22] D. Liang, S. Kaneko, H. Sun, and B. Kang, "Adaptive local spatial modeling for online change detection under abrupt dynamic background," *International Conference on Image Processing*, pp. 2020–2024, 2017.

[23] K. Toyama, J. Krumm, B. Brumitt, and B. R. Meyers, "Wallflower: principles and practice of background maintenance," vol. 1, pp. 255–261, 1999.

[24] http://limu.ait.kyushu-u.ac.jp/dataset/en/.

[25] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[26] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," pp. 1–8, 2012.

[27] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 600–608, 2004.

[28] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand, "Depth image-based rendering with advanced texture synthesis for 3-d video," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 453–465, 2011.

[29] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3d video," in *Applications of Digital Image Processing XXXII*, vol. 7443. International Society for Optics and Photonics, 2009, p. 74430T.

[30] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," *Computer Vision and Pattern Recognition Workshops*, pp. 38–43, 2012.