Pattern Recognition 117 (2021) 107995

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Cross-scene foreground segmentation with supervised and unsupervised model communication

Dong Liang[a,*], Bin Kang[b], Xinyu Liu[a], Pan Gao[a], Xiaoyang Tan[a], Shun'ichi Kaneko[c]

[a] 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
[b] Department of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
[c] Metatec Cooperation, Yokohama 220-0004, Japan

ARTICLE INFO

Article history:
Received 26 May 2020
Revised 22 January 2021
Accepted 15 April 2021
Available online 24 April 2021

Keywords:
Foreground segmentation
Model communication
Cross-scene
Online updates

ABSTRACT

In this paper[1], we investigate cross-scene video foreground segmentation via supervised and unsupervised model communication. Traditional unsupervised background subtraction methods often face the challenging problem of updating the statistical background model online. In contrast, supervised foreground segmentation methods, such as those that are based on deep learning, rely on large amounts of training data, thereby limiting their cross-scene performance. Our method leverages segmented masks from a cross-scene trained deep model (spatio-temporal attention model (STAM), pyramid scene parsing network (PSPNet), or DeepLabV3+) to seed online updates for the statistical background model (CPB), thereby refining the foreground segmentation. More flexible than methods that require scene-specific training and more data-efficient than unsupervised models, our method outperforms state-of-the-art approaches on CDNet2014, WallFlower, and LIMU according to our experimental results. The proposed framework can be integrated into a video surveillance system in a plug-and-play form to realize cross-scene foreground segmentation.

1. Introduction

Video foreground segmentation plays a fundamental role in many vision systems [1]. For example, surveillance systems usually record video around the clock and generate massive amounts of data. However, keyframes that contain moving objects such as pedestrians and vehicles often occur very infrequently and are spread over time. Foreground segmentation can eliminate such redundancies and preserve keyframes for further analyses. However, foreground segmentation is challenging for nonstationary/dynamic scenes (e.g., scenes with outdoor illumination changes, indoor light turning on/off, swaying tree branches, water fountains, or any combination of these factors). Thus, simplistic background subtraction methods are ill-suited for foreground segmentation in dynamic scenes. Early studies focused on the use of statistical distributions to build unsupervised background modeling algorithms with background updating schemes to adapt to dynamic backgrounds.

There are two main strategies for updating a background model online to handle dynamic scenes [2]: (1) selective updating, in which a new sample is added to the model only if it is classified as a background sample, and (2) blind updating, in which every new sample is added to the model. Using selective updating, one must decide whether each pixel value belongs to the background or not. The simplest way to do this is to use the segmentation result as an updating decision. The problem is that any incorrect segmentation decision will result in persistent incorrect segmentation afterward. Blind updating does not suffer from this deadlock scenario since it does not involve any updating decisions; it allows intensity values that do not belong to the background to be added to the model. This leads to more false negatives as those foreground pixels erroneously become part of the model. Trade-offs must be made with the update rate, which regulates the spread at which the background model is updated. A high update rate leads to noisy segmentation due to sensitivity to small or temporary changes, whereas a low update rate, yields an outdated background model and results in false segmentation.

Modern deep learning-based foreground segmentation approaches can provide pixel-level annotation for each frame without any model updates. The main obstacle in introducing a supervised learning technique is that foreground segmentation is a scene-dependent task. Supervised models that are trained on

[*] Corresponding author.
E-mail address: liangdong@nuaa.edu.cn (D. Liang).
[1] Dong Liang, Bin Kang and Xinyu Liu contributed equally to this work.

https://doi.org/10.1016/j.patcog.2021.107995
0031-3203/© 2021 Elsevier Ltd. All rights reserved.

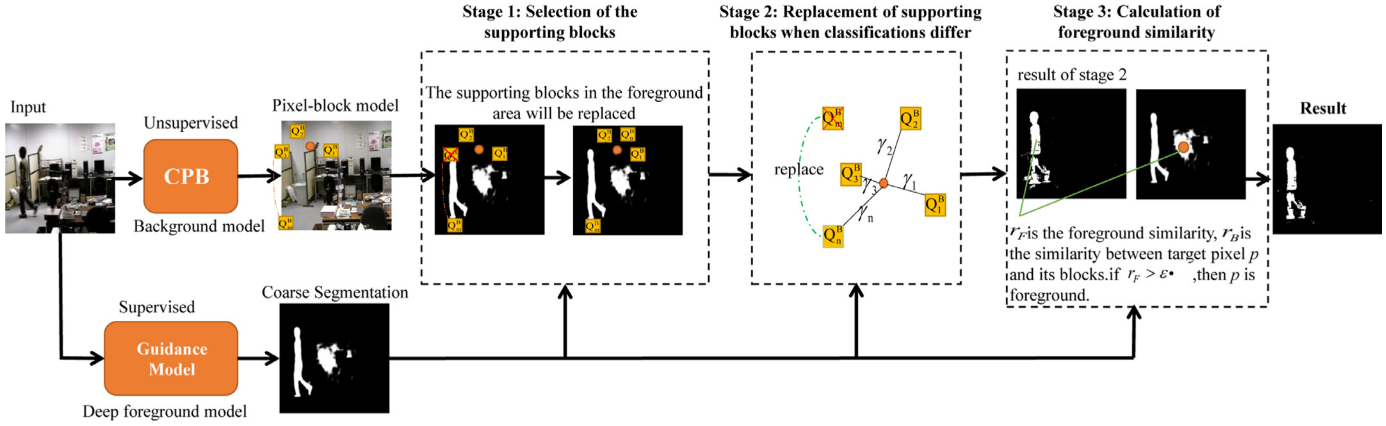
Stage 1: Selection of the supporting blocks
Stage 2: Replacement of supporting blocks when classifications differ
Stage 3: Calculation of foreground similarity
Input
Unsupervised
CPB
Background model
Pixel-block model
The supporting blocks in the foreground area will be replaced
replace
Q^B_m
Q^B_2
Q^B_3
Q^B_1
Q^B_n
γ_2
γ_3
γ_1
γ_n
result of stage 2
Result
$r_F$is the foreground similarity, $r_B$is the similarity between target pixel p and its blocks.if $r_F > \varepsilon$• ,then p is foreground.
Supervised
Guidance Model
Deep foreground model
Coarse Segmentation

Fig. 1. Proposed method.

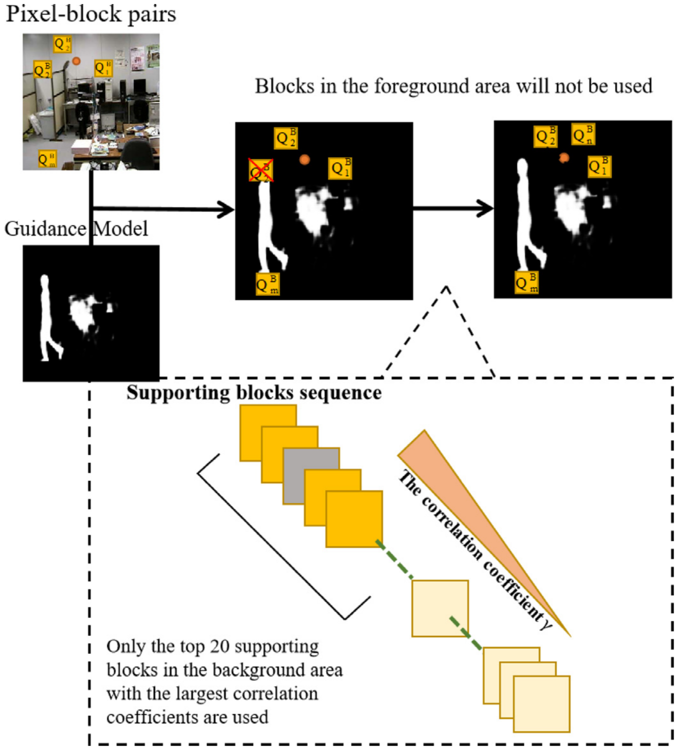
Pixel-block pairs
Blocks in the foreground area will not be used
Q^B_2
Q^B_1
Q^B_m
Guidance Model
Supporting blocks sequence
The correlation coefficient γ
Only the top 20 supporting blocks in the background area with the largest correlation coefficients are used

Fig. 2. Selection of the supporting blocks and model updates.

scene-specific data do not generalize well in real-world scenarios, and cross-scene models do not perform well on individual scenes.

Essentially, video foreground segmentation is an empirical segmentation problem that is closely related to the scene's appearance, motion, and semantics. In this paper, we propose a new foreground segmentation framework via supervised and unsupervised model communication: Using a deep model as seeds, our method can facilitate the accurate online update of the unsupervised statistical background model and realize refined foreground segmentation. We use the cooccurrence pixel-block model (CPB) [3,4] as the unsupervised background model and we use the spatiotemporal attention model (STAM) [5], pyramid scene parsing network (PSPNet) [6] and DeepLabV3+ [7] as the supervised segmentation guidance models. The CPB model compares each observed pixel with its supporting blocks to determine whether the pixel belongs to the foreground. The supporting blocks are selected via spatial correlation [8,9]. This method's training process relies on the calculation of the linear correlation between pixels, for which online update is too expensive. Therefore, the segmentation performance of this method will gradually decrease over time. STAM is a cross-scene foreground segmentation deep model. PSPNet and DeepLabV3+ are state-of-the-art semantic segmentation models. The proposed framework is illustrated in Fig. 1. To bridge the gap between unsupervised and supervised guidance models for foreground detection, the guidance model is used to obtain coarse segmentation results. Then, through three stages, namely, (1) selection of the supporting blocks, (2) replacement of the supporting blocks, and (3) calculation of the foreground similarity, CPB realizes model updates and completes fine-grained segmentation.

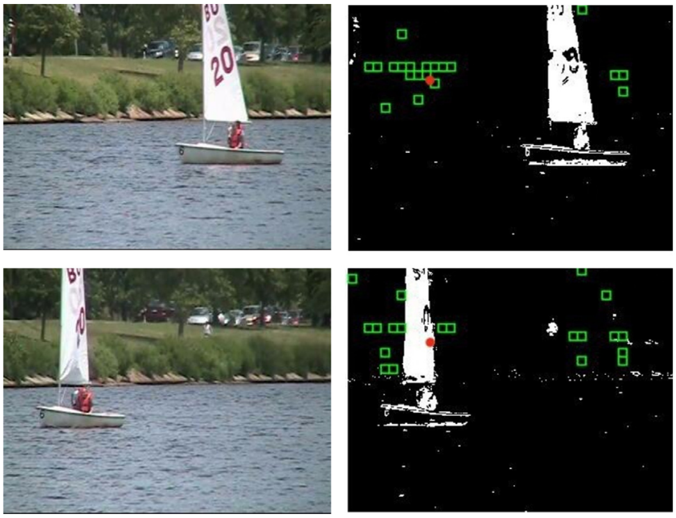

Fig. 3. A target pixel (the red point) that has different supporting blocks (the green boxes) at different times. When a boat passes by, the model always selects the supporting blocks in the background area. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The contributions of this study include the following:

(1) Coarse-to-fine segmentation. The proposed interaction of supervised and unsupervised models can realize fine-grained foreground segmentation.

(2) Unsupervised background model updates. The unsupervised statistical background model can update and avoid deadlock by using segmented masks as external selective-updating cues.

(3) This method is more flexible than deep learning-based methods that depend on scene-specific training. Compared with

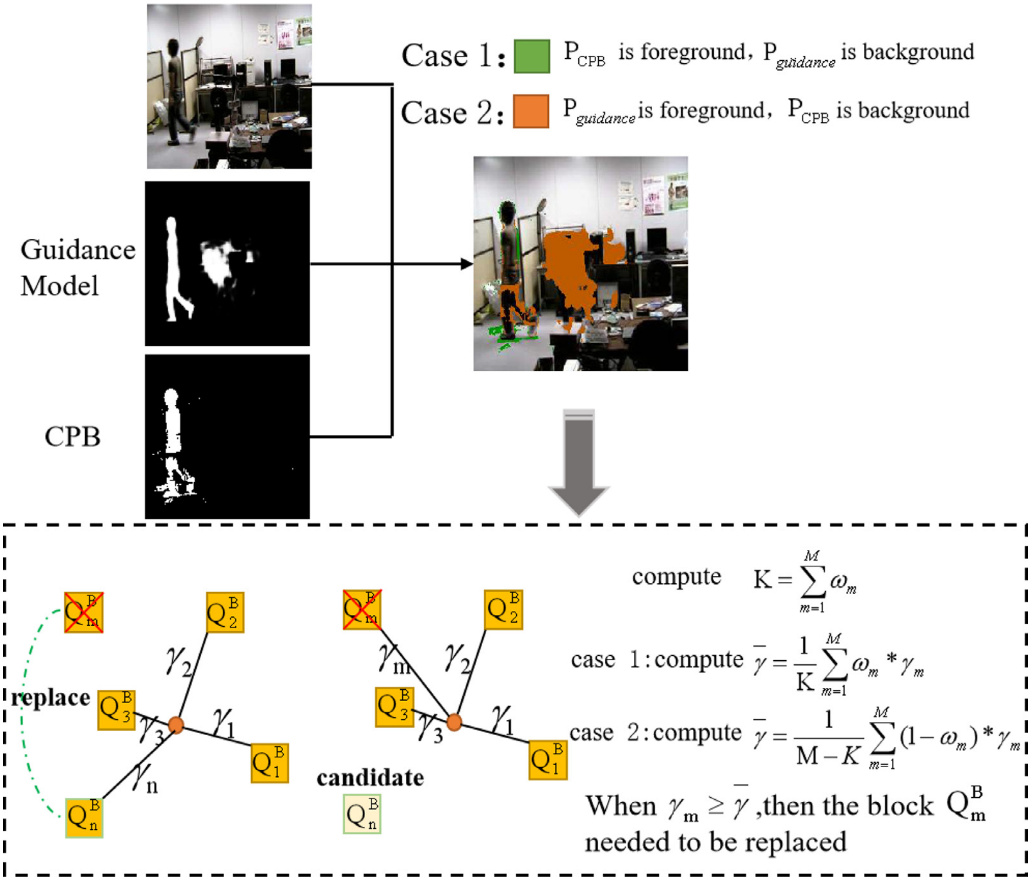
Case 1: 🟩 P_{CPB} is foreground, $P_{guidance}$ is background
Case 2: 🟧 $P_{guidance}$ is foreground, P_{CPB} is background
Guidance Model
CPB
Q_m^B
Q_2^B
Q_3^B
Q_1^B
Q_n^B
γ_2
γ_1
γ_3
γ_n
γ_m
replace
candidate
compute $K = \sum_{m=1}^{M} \omega_m$
case 1: compute $\bar{\gamma} = \frac{1}{K} \sum_{m=1}^{M} \omega_m * \gamma_m$
case 2: compute $\bar{\gamma} = \frac{1}{M-K} \sum_{m=1}^{M} (1-\omega_m) * \gamma_m$
When $\gamma_m \geq \bar{\gamma}$, then the block Q_m^B needed to be replaced

Fig. 4. Replacement of supporting blocks when classifications differ.

unsupervised models, it reduces the number of training samples and utilizes training datasets with no human intervention.

The remainder of this paper is organized as follows. In Section 2, we discuss related work. Section 3 describes the proposed method in detail. The experimental results are presented and discussed in Section 4, and the final conclusions of this study are presented in Section 5.

2. Related studies

2.1. Unsupervised background subtraction

2.1.1. Background model

Since observations of the background in image sequences can be considered stochastic events, many statistical approaches [10,11] have been employed to model the background. Background models can be classified into two categories: Independent pixel-wise models employ the statistical processing of time-domain observations for each pixel. Most earlier background modeling approaches fall into this category, which include the well-known single Gaussian [12] model, Gaussian mixture model (GMM) [13], kernel density estimation model (KDE) [2], and hidden Markov models (HMMs) [14].

The methods of the second category exploits the spatial dependencies of pixels in the background. Oliver [15] employed an eigenspace decomposition approach in which the background was modeled by the eigenvectors that corresponded to the largest eigenvalues. Sheikh [16] used the joint representation of image pixels in a local spatial distribution and color information to competitively build both background and foreground KDE models in a decision framework. Heikkilä and Pietikäinen [17] used a local binary pattern (LBP) to subtract the background and detect moving objects in real-time. Reference [18] learned a tensor subspace representation by adaptively updating the sample mean and an eigenbasis for each unfolding matrix. Cooccurrence pixel-pair background model [8,9] employs an alignment of supporting pixels for the target pixel, which maintains a stable intensity difference in training frames without any restriction of locations. The intensity difference of the pixel pairs enables the background model to tolerate noise and be illumination-invariant. Methods that have been proposed in recent years include [19,20].

2.1.2. Advanced background updating strategies

In addition to selective and blind updating, several advanced strategies are available for updating the background model online. The updating strategy of Vibe [21] incorporates three important components: memoryless updates to ensure a smoothly decaying lifespan for the samples that are stored in the background pixel models; random time subsampling to extend the time windows that are covered by the background pixel models; and spatial propagation of background pixel samples to ensure spatial consistency and to enable the adaptation of the background pixel models that are masked by the foreground. SuBSENSE [22] updates pixel models using a conservative, stochastic, two-step approach: Samples are replaced randomly instead of according to their last modification to ensure that a solid history of long and short-term background representations can be retained in the pixel models. Since new samples can only be inserted when a local pixel is recognized as background, this approach prevents static foreground objects from being assimilated too fast, which often occurs for methods that use blind updating. The second step is spatial diffusion, which enables homogeneous regions with the background to be absorbed

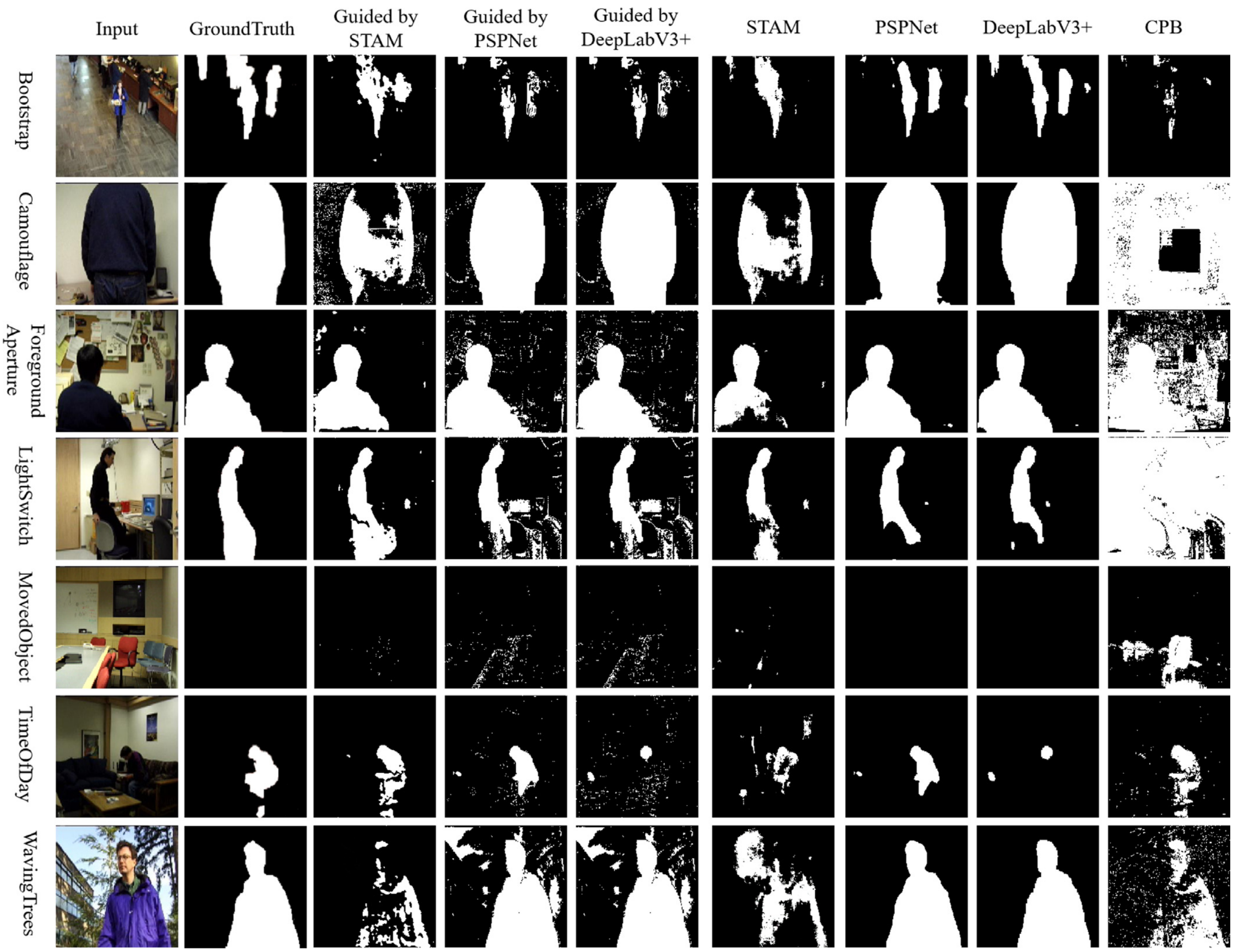

Fig. 5. Comparison of various methods on WallFlower.

much faster. This step increases the background model's spatial coherency to the extent that it can tolerate limited camera motion. BMOG [23], which is based on a mixture of Gaussians, explores a classification mechanism and combines color space discrimination capabilities with hysteresis and a dynamic update rate for background model updating. The update rate is set to a fixed minimum value when a pixel is transferred from background to foreground.

2.2. Methods that are based on convolutional neural networks

2.2.1. Foreground segmentation

Background subtraction based on a convolutional neural network is first proposed in [24]. DeepBS [25] utilizes a convolutional neural network and a spatial-median filter in cross scenes. As the foreground is detected based on independent frames, the neighboring frames' temporal relevance is ignored. Cascade CNN [26] is a semiautomatic method that reduces the required amount of training data. CNN branches for processing images of various sizes are cascaded together to help detect a multiscale foreground. SGSM-BS [27], which is an improved version of the RPCA method, uses the entropy rate superpixel segmentation model (ERS) and the structured Gaussian scale mixture model (SGSM) to simulate a group of pixels that belong to a moving object. DPDL [28] (deep pixel distribution learning) uses pixel-based features via RPoTP (random permutation of temporal pixels). It deliberately blurs the temporal correlation of the previous observation results of a single pixel. Reference [29] uses a convLSTM-based network to capture spatial and temporal features. The encoder features are imported into a spatiotemporal information propagation (STIT) module, which can better capture the spatiotemporal relationship between consecutive frames. FgSegNet [30,31] encodes the features of three scales of the same input image with three sets of CNN encoders.

All the approaches that are discussed above are supervised models, for which training sample annotation is time-consuming and laborious. They also tend to generalize poorly to scenes that are absent from the training data. Although various methods [5,25] that are trained on large-scale multi-scene datasets can segment various scenes, they typically perform even worse than unsupervised background subtraction methods on untrained scenes.

2.2.2. Semantic segmentation

Semantic segmentation methods have made remarkable progress due to the development of convolutional neural networks. BFP [32] learns the boundary as an additional semantic class to help the network become aware of the boundary layout. A boundary aware feature propagation (BFP) module is introduced for sharing local features within their regions. CCL [33], uses a context-contrast local convolutional network to generate multilevel and multiscale context-aware local features, and an aggregation scheme, namely, gated sum, is proposed for selecting features

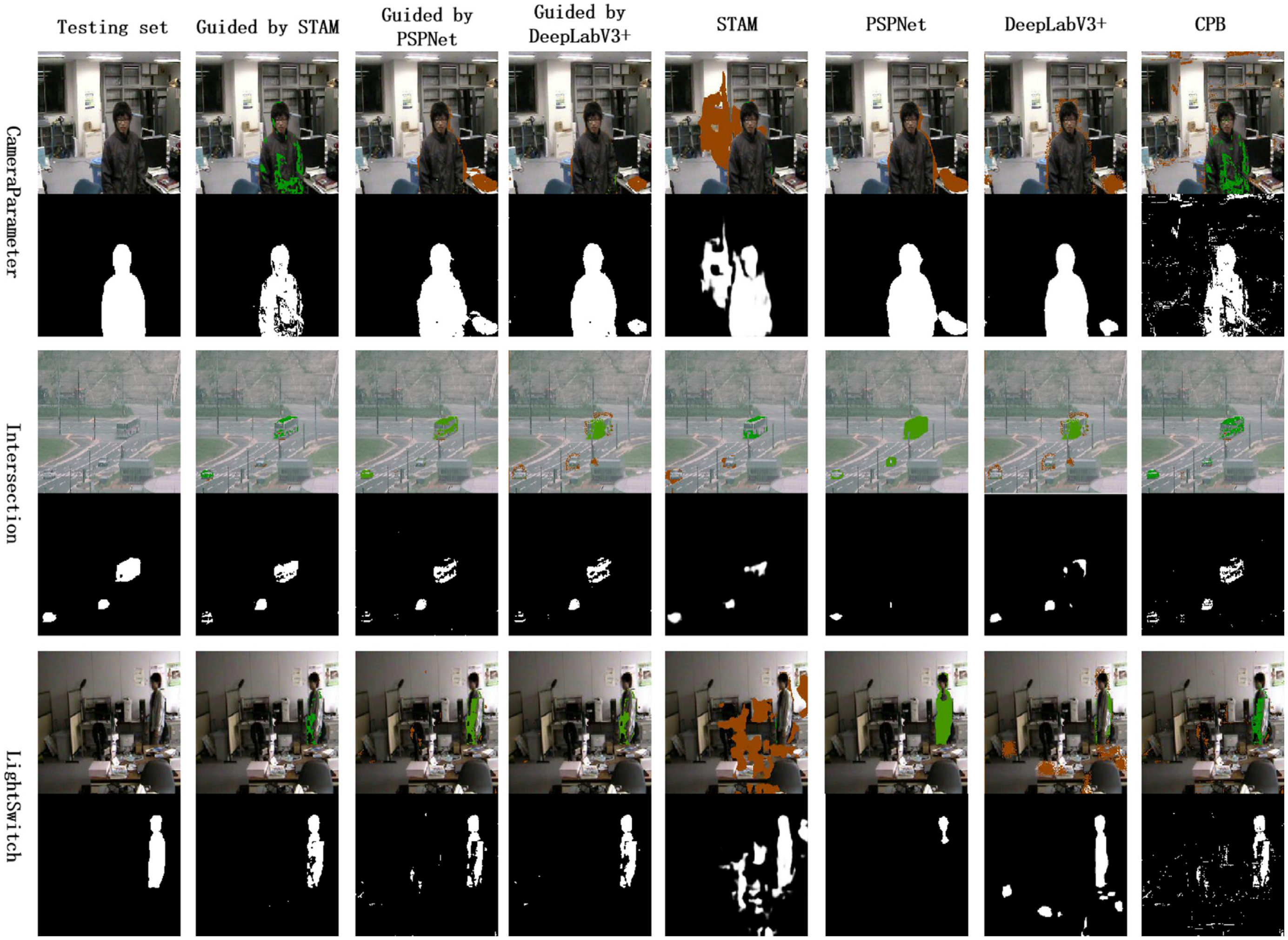

Fig. 6. Comparison on LIMU. False positives are marked in orange, and false negatives are marked in green. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of various scales and segmenting objects at multiple scales. It is improved in [34] via the introduction of a boundary delineation refinement (BDR) model with a small computational cost. A shape-variant context (SCVNet) [35] is proposed for customizing the shape/scale of the context for each pixel instead of using simple smooth context information. This study also uses a labeling denoising model for helping reduce errors that are caused by noisy low-level features. Although semantic segmentation approaches can provide high-level annotation for each pixel, they ignore the temporal relevance and motion cues, which are crucial for video foreground segmentation.

3. Proposed approach

3.1. Supervised model as segmentation guide

In this subsection, we introduce the three supervised models that are used in the proposed method as segmentation guides.

3.1.1. STAM

The spatiotemporal attention model (STAM) [5] is an attention-guided weight-able connection encoder-decoder that maintains the useful connections of the symmetric layer and suppresses the invalid connections. It aggregates features from both the decoder and encoder by introducing an attention module in the decoding stage. The high-level features provide global information to guide the attention module to select suitable low-level features. The static frame and its optical flow (motion cue) feed two encoders, and the attention modules reorganize them to reconstruct the foreground at the pixel level. In contrast to the model without motion cues and the attention module, this model jointly learns appearance and temporal information to optimize the efficiency of feature aggregation. We train STAM using 5% random training data with its ground truth on the CDNet2014 dataset [36].

3.1.2. PSPNet

The pyramid scene parsing network (PSPNet) [6] uses a pre-trained ResNet model with atrous convolution to extract feature maps. The main role of atrous convolution is to enlarge the receptive field. A pyramid pooling module with a depth of 4 is used to obtain context information. Features of various depths are obtained through pooling operations of different scales based on input features. These pyramid features are directly up-sampled to the same size as the input features and connected with the input features. The process of feature merging fuses the detailed features and global features of the target. PSPNet provides sufficient global context information for pixel-level scene analysis. The pyramid pooling module collects and integrates context information on various scales, which is more representative than global pooling.

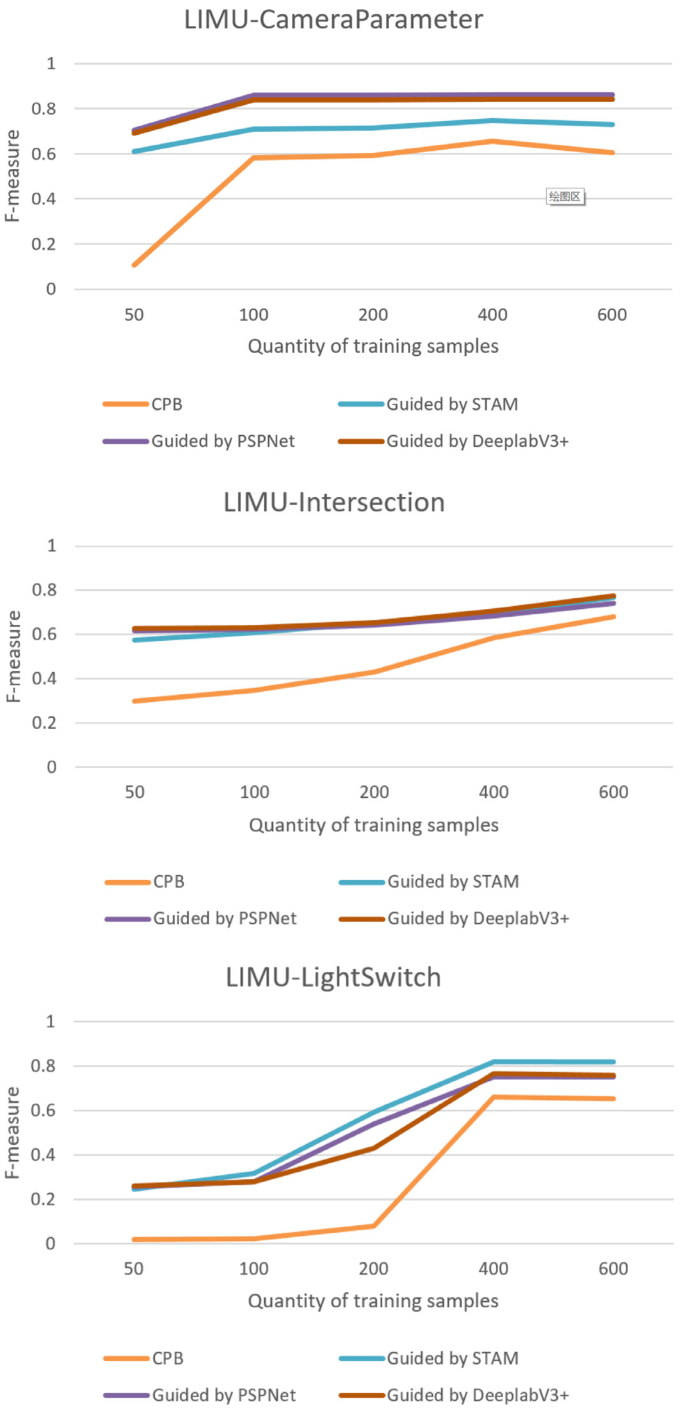

Fig. 7. Comparison of training sets of various sizes.

3.1.3. Deeplabv3+

The spatial pyramid pooling module can encode multiscale contextual information by probing the incoming features with filters or pooling operations at multiple rates and multiple effective fields-of-view, while the encode-decoder can capture sharper object boundaries by gradually recovering the spatial information. DeepLabV3+ [7] combines the advantages of both methods. It extends DeepLabv3 by adding a simple yet effective decoder module to refine the segmentation results especially along object boundaries. It further explores the Xception model and applies depthwise separable convolution to both the atrous spatial pyramid pooling and decoder modules, thereby resulting in a faster and stronger encoder-decoder network.

For both of these semantic segmentation methods, the ADE20K [37] training dataset is used to obtain the trained models.

3.2. Co-occurrence pixel-Block model

The CPB [3,4] model compares the target pixel p with its supporting block Q^B to determine whether p belongs to the foreground.

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

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

$$\gamma(p, Q_m^B) = \frac{C_{p,\bar{Q}_m^B}}{\sigma_p \cdot \sigma_{\bar{Q}_m^B}}. \tag{2}$$

where $C_{p,\bar{Q}_m^B}$ is the intensity covariance of the target pixel p and its supporting blocks Q_m^B. σ_p and $\sigma_{\bar{Q}_m^B}$ 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, namely, the difference in intensity follows a single Gaussian distribution:

$$(I_p - \bar{I}_{Q_m^B}) \sim N(b_m, \sigma_m^2). \tag{3}$$

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

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

$$\omega_m = \begin{cases} 1 & \text{if } |I_p - \bar{I}_{Q_m^B}| \le \eta \cdot \sigma_m \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where η is a threshold of the Gaussian model. Considering the difference in correlation between each target pixel and its supporting blocks, their correlation coefficients γ_m are introduced as weights. CPB will classify a target pixel p as a foreground pixel when the following conditions are satisfied:

$$\sum_{m=1}^{M} \gamma_m \cdot \omega_m > \lambda \cdot \sum_{m=1}^{M} \gamma_m \tag{5}$$

where λ is the relevance decision threshold.

The CPB model relies on the correlation coefficient between target pixel p and its supporting blocks $\{Q_m^B\}_{m=1,2,\ldots M}$, which is difficult to update after training initialization. The lack of updates causes the model's performance to degrade over time and limits the online applicability.

3.3. Model communication

3.3.1. Stage 1: Selection of the supporting blocks

When a foreground object covers the supporting block, it will destroy the Gaussian relationship between the supporting block and its target pixel. The state of the pixel-block pair, which is defined by Eq. 4 is temporarily invalid, thereby resulting in segmentation errors. Moreover, the segmentation result that is provided by the guidance model has a high probability of belonging to the foreground. The CPB model's performance can be improved by avoiding the selection of supporting blocks in the foreground area. As shown in Fig. 2, only the supporting blocks in the background area are used, while the blocks in the foreground, which are shaded gray, are temporarily discarded. The candidate supporting blocks, which are shaded light yellow, are obtained during the training process. Fig. 3 illustrates the selection process of supporting blocks when a boat passes by.

3.3.2. Stage 2: Replacement of supporting blocks when classifications differ.

The process of Stage 1 avoids the selection of the supporting blocks that are in the foreground area. However, supporting blocks can still be in the foreground area due to potential differences between the guidance model's result and the ground truth. Moreover, the pixel-block model that is obtained during the training process may degrade over time because the background is not static, e.g., due to cloud drift in the sky or entry/exit of vehicles in a parking lot. As a result, foreground or background "noise" might arise in the segmentation process.

Pixels for which the CPB and guidance model classifications differ are represented by green and orange areas in Fig. 4. They are divided into two cases: Case 1, in which CPB regards the pixel as foreground while the guidance model classifies it as background, and Case 2, in which CPB regards the pixel as background while the guidance model classifies it as foreground. When the result from the CPB model differs from that of the guidance model, the supporting blocks that correspond to high correlation coefficient values must be responsible for potential errors, which may already be in the state of structural failure. The strategy is as follows:

$$K = \sum_{m=1}^{M} \omega_m \tag{6}$$

$$\overline{\gamma} = \begin{cases} \frac{1}{K} \sum_{m=1}^{M} \gamma_m \cdot \omega_m & \text{Case 1} \\ \frac{1}{M-K} \sum_{m=1}^{M} \gamma_m \cdot (1-\omega_m) & \text{Case 2} \end{cases} \tag{7}$$

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

$$\gamma_m \geq \overline{\gamma} \tag{8}$$

then Q_m^B must be replaced.

As illustrated in Fig. 4, when considering target pixels that differ in terms of classification, their supporting blocks, which are selected via Eq. 8 will be temporarily discarded, and the candidate supporting blocks (represented by yellow squares) will replace them as the new supporting blocks of the target pixels.

3.3.3. Stage 3: Calculation of the foreground similarity

The pixel-block model's construction is based on the correlation of their long-term eigenvalue sequences. The supporting block with a high correlation coefficient should be in an area that is homogeneous with its target pixel. When pixel p is in the foreground and CPB misclassifies it as a background pixel (Case 2, as described in Section 3.3.2), we calculate the similarity of the Euclidean distance in the feature 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 satisfies the following equation, p will be classified as foreground:

$$r_F > \varepsilon \cdot r_B \tag{9}$$

where ε is the similarity decision threshold.

3.4. Discussion on the quantity and quality of the training samples

In this section, we discuss the impacts of the size and quality of the training dataset on the performance of the proposed method.

3.4.1. Quantity of training samples

Various methods such as KDE and CPB require a large amount of training data because they need precise statistical histograms or correlation coefficients to build a background model. The proposed method focuses more on selecting and updating the supporting block Q_m^B during the segmentation stage. The coarse segmentation that is produced by the supervised guidance models and the updated supporting blocks provide performance compensation for the CPB model. Therefore, our method is less susceptible to insufficient background model training due to the training data size, thereby enabling it to maintain stable performance under training sets of various sizes.

3.4.2. Quality of the training samples

Most traditional unsupervised background modeling methods, including the CPB model, tend to perform well with training sets that contain as few foreground objects as possible. In a scenario in which almost all the samples in the training set contain foreground objects such as pedestrians or vehicles, the target pixel p and its supporting block Q_m^B would maintain a stable relationship over time. Moreover, the history of a specified target pixel p can be categorized into background and foreground stages. Its supporting blocks can also be divided into two classes: those that maintain a stable relationship with p when they are in the background and those that maintain a stable relationship with p when they are in the foreground. This will cause CPB to consider both the background and the moving objects as background and lead to many false negatives. When the proposed method faces this scenario, the blocks that maintain a stable relationship with p but contain the foreground will be replaced according to the guidance model. As a result, the proposed method is more tolerant and flexible in training set selection. Relevant experimental verification will be conducted in the next section.

4. Experiments

4.1. Experimental settings and implementation details

For foreground segmentation model STAM [5], we use the training protocol that is recommended in DeepBS [25], which randomly selects 5% of the samples with the corresponding ground truths of each subset from CDNet2014 [36]. All the other deep supervised foreground segmentation models, which include DeepBS [25], Cascade CNN [26], and FgSegNet [30], are also trained on the same training set as STAM.

For semantic segmentation models, since there is no semantic annotation in the foreground segmentation dataset, DeepLabV3+ [7] and PSPNet [6] are trained on ADE20K [37]. We define various classes as foreground according to the protocol that is recommended in [38], which include person, car, cushion, box, book, boat, bus, truck, bottle, van, bag, and bicycle.

For the CPB model and five other unsupervised background subtraction models, namely, SuBSENSE [22], KDE [2], GMM [13], BMOG [23] and PBAS [39], we use the following training strategy: On the CDNet2014 and LIMU datasets, we choose the first 400 frames for training. On the WallFlower dataset, we adopt the strategy that is recommended by the dataset itself, namely, we use the provided 200 frames as the training set. The experimental parameter settings for CPB are presented in Table 1. The segmented foregrounds are obtained without any postprocessing.

4.2. Results and evaluation

4.2.1. Cross-scene testing on CDNet2014

Experiments are conducted on a total of 11 subsets in CDNet2014, bad weather (BDW), baseline (BSL), camera jitter (CJT),

Table 1
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

dynamic background (DBG), intermittent object motion (IOM), low frame rate (LFR), night videos (NVD), shadow (SHD), thermal (THM) and turbulence (TBL). Since STAM is trained on CDNet2014 with 5% random samples with the ground truths for each subset, we do not use it as a guidance model for fairness. DeepBS, Cascade CNN, and FgSegNet are also removed for the same reason. Instead, we use semantic segmentation models – DeepLabV3+ and PSPNet, trained on ADE20K as guidance models. Five unsupervised background subtraction models, namely, SuBSENSE, KDE, GMM, BMOG, and PBAS, are used for comparison.

The testing results are presented in Table 2, and CPB that is guided by PSPNet ranks first among all unsupervised methods in terms of overall F-measure, while CPB that is guided by DeepLabV3+ ranks second. considering the 11 individual subsets, CPB that is guided by DeepLabV3+ ranks at the top for 3, and CPB that is guided by PSPNet ranks at the top for 6. For two subsets, namely, baseline (BSL) and thermal (THM), CPB that is guided by PSPNet and CPB that is guided by DeepLabV3+ show low performance for the following reasons: (1) The unsupervised background subtraction model can handle simple scenes well. However, the semantic segmentation model is not guaranteed to distinguish between real foregrounds such as a moving and a parked car. (2) The coverage of the ADE20K dataset is insufficient. For thermal (THM), all the videos are captured by an infrared camera and are not covered by ADE20K. Finally, we observe that both types of guided CPB models outperform their original CPB model, and their performances are relatively stable.

4.2.2. Cross-scene testing on WallFlower

Next, we perform cross-scene experiments on WallFlower [40]. WallFlower has strict regulations on the training set. This dataset also contains many huge objects, and each scene has only one image in the test set.

The experimental results on WallFlower are presented in Table 3, Table 4 and Fig. 5. Using the F-measure as the evaluation metric in Table 3, CPB that is guided by STAM realizes the highest score on LightSwitch and outperforms CPB in all scenes. In other scenes, the best performances are realized by PSPNet and DeepLabV3+.

From the visualized results in Fig. 5, we observe that the performances of CPB models that are guided by PSPNet and DeepLabV3+ degrade mainly due to the poor performance of the original CPB and the presence of large foreground objects. When an object is too large (or the false-positive area is too large), there may not be sufficiently many qualified candidate supporting blocks for CPB to complete the model updating stages, thereby resulting in an incomplete foreground or more false positives.

There is no foreground object in the ground truth of subset MovedObject. Therefore, specificity = TN / (TN + FP) is selected as the evaluation metric instead of the F-measure. From Table 4 and the fifth row of Fig. 5 which corresponds to MovedObject, PSPNet and DeepLabV3+ obtain perfect results, but this is by accident since the foreground class that we defined does not include the class armchair.

Although the proposed method does not outperform PSPNet or DeepLabV3+ on WallFlower, it outperforms CPB and other background subtraction models.

4.2.3. Cross-scene testing on LIMU

LIMU [41], with subsets CameraParameter, Intersection, and LightSwitch, is a standard video surveillance dataset for both indoor and outdoor scenes. Experimental results on this dataset are presented in Table 5 and Fig. 6. In terms of the F-measure, CPB that is guided by STAM ranks first on LightSwitch. CPB that is guided by PSPNet ranks first on CameraParameter. CPB that is guided by DeepLabv3+ ranks first on Intersection and overall. Correspondingly, the overall F-measure is 0.3154 higher than that of STAM, 0.2563 higher than that of PSPNet, and 0.2883 higher than that of DeepLabV3+. All three methods outperform CPB by more than 10% in terms of the overall F-measure. From Fig. 6, the proposed method suppresses false positives well (marked in orange).

In subset LightSwitch, when the lights are turned off, all the guidance models perform poorly due to low illumination. CPB has many false negatives in the inferred foreground and scattered false positives in the inferred background. The results for the guidance models are the opposite: their foregrounds are complete, and their background parts have no scattered points. Nevertheless, the guidance models could have false-positive results that are caused by fluctuations in light or other factors. In these three scenarios, the proposed model can learn different but useful information from both and significantly outperform each of the two modules alone.

4.3. Discussion of the results on wallflower and LIMU

In most of the sequences of WallFlower, the results that are obtained by the proposed method are similar to those of the guidance models (STAM, PSPNet, and DeepLabV3+) alone. In contrast, in all LIMU sequences, the results are significantly better than those from each of the two modules (CPB and guidance models) alone. It can be explained by the algorithm and dataset characteristics.

In WallFlower, most of the foreground objects occupy a large area in the image, and these scenes are not difficult for the guidance models; thus, the guidance models could segment most of the foreground area. Due to the proposed algorithm principle in Stage 1 (described in Section 3.3.1), most of the original supporting blocks with a large linear correlation coefficient γ_m were eliminated due to the large foreground area. After Stage 1, γ_m of the updated supporting blocks was small because most of the supporting blocks with large γ_m were eliminated. Therefore, from the left part of Eq. 5, $\sum_{m=1}^{M} \gamma_m$, had a relatively small value, and, as a result, CPB was not sensitive to foreground segmentation. The dominant factor for foreground detection was Eq. 9 in Stage 3 (described in Section 3.3.3), which led to the foreground mainly following the guidance model's segmentation results. However, according to the results on WallFlower, the final segmentation showed a slight degradation when the guidance model had nearly perfect segmentation while the foreground was large due to the involvement of CPB.

In LIMU, where most of the foreground objects are of regular sizes compared with the whole scene, the proposed method could reconcile the results from both CPB and the guidance model according to Stage 2 (described in Section 3.3.2) without any bias. In constrast, when the guidance model is unreliable (for example, in LIMU-LightSwitch, STAM is not sensitive to the foreground when the illumination is very low), CPB could follow its original supporting blocks and be free from the interference of the guidance model according to Stage 2 (described in Section 3.3.2).

In summary, the proposed framework is robust even when there is severe performance deterioration in either the CPB or guidance models, even in an extreme case.

Table 2
F-measure values of various methods on CDNet2014. Since STAM, DeepBS, Cascade CNN and FgSegNet are trained in CDNet2014, in this experiment we do not use STAM as a guidance model and do not use DeepBS, Cascade CNN or FgSegNet for comparison.

Methods in Scenes	BDW	BSL	CJT	DBG	IOM	SHD	THM	TBL	LFR	NVD	PTZ	Overall
CPB [4]	0.860	0.814	0.790	0.771	0.601	0.813	0.760	0.729	0.615	0.520	0.504	0.633
CPB guided by PSPNet	**0.865**	0.850	**0.861**	**0.850**	0.761	0.886	0.544	**0.852**	**0.889**	**0.604**	0.521	**0.772**
PSPNet [6]	0.503	0.759	0.586	0.654	0.435	0.276	0.180	0.460	0.869	0.357	0.118	0.605
CPB guided by DeepLabV3+	0.860	0.848	0.840	0.815	**0.768**	**0.895**	0.797	0.835	0.727	0.549	**0.545**	0.753
DeepLabV3+ [7]	0.648	0.541	0.484	0.569	0.588	0.823	0.579	0.363	0.507	0.220	0.476	0.676
SuBSENSE [22]	0.862	**0.950**	0.815	0.818	0.657	0.865	**0.817**	0.779	0.645	0.560	0.348	0.741
KDE [2]	0.757	0.909	0.572	0.596	0.409	0.803	0.742	0.448	0.548	0.437	0.037	0.595
GMM [13]	0.738	0.825	0.597	0.633	0.521	0.732	0.662	0.466	0.537	0.410	0.152	0.571
BMOG [23]	0.784	0.830	0.749	0.793	0.529	0.840	0.635	0.693	0.610	0.498	0.235	0.725
PBAS [39]	0.698	0.802	0.646	0.601	0.471	0.702	0.465	0.665	0.536	0.512	0.478	0.576

Table 3
F-measure values of various methods on WallFlower.

Methods	Bootstrap	Camouflage	ForegroundAperture	LightSwitch	TimeOfDay	WavingTrees	Overall
CPB [4]	0.6518	0.6112	0.5900	0.7157	0.7564	0.7033	0.6714
CPB guided by STAM	0.7560	0.6884	0.9402	**0.9097**	0.7949	0.6665	0.7929
STAM [5]	0.7414	0.7369	0.8292	0.9090	0.3429	0.5325	0.6820
CPB guided by PSPNet	0.5341	0.9849	0.9204	0.7048	0.8147	0.8696	0.7132
PSPNet [6]	0.7203	0.9876	**0.9887**	0.7710	**0.8195**	**0.9891**	**0.9731**
CPB guided by DeepLabV3+	0.5927	0.9867	0.9237	0.6886	0.2604	0.8682	0.7122
DeepLabV3+ [7]	**0.8327**	**0.9941**	0.9884	0.7375	0.1909	0.9850	0.7881
SuBSENSE [22]	0.4192	0.9535	0.6635	0.3201	0.7107	0.9597	0.6711
KDE [2]	0.5887	0.8498	0.5726	0.2879	0.7425	0.9854	0.6832
GMM [13]	0.5306	0.8307	0.5778	0.2296	0.7203	0.9767	0.6443
BMOG [23]	0.5484	0.8310	0.5785	0.3546	0.7295	0.9840	0.6535
PBAS [39]	0.2857	0.8922	0.6459	0.2212	0.4875	0.8421	0.5624
DeepBS [25]	0.7479	0.9857	0.6583	0.6114	0.5494	0.9546	0.7512
Cascade CNN [26]	0.5238	0.6778	0.7935	0.5883	0.3771	0.2874	0.5413
FgSegNet [30]	0.3587	0.1210	0.4119	0.6815	0.4222	0.3456	0.3902

Table 4
Specificity on WallFlower - Moved Object.

Guided by STAM	Guided by PSPNet	Guided by DeepLabV3+	STAM [5]	PSPNet [6]	DeepLabV3+ [7]	Cascade CNN [26]	FgSegNet [30]	CPB [4]
0.9977	0.9836	0.9838	0.9949	**1.0**	**1.0**	0.7763	0.8470	0.8922

Table 5
F-measure values of various methods on LIMU.

Methods	CameraParameter	Intersection	LightSwitch	Overall
CPB [4]	0.6545	0.6778	0.6633	0.6652
CPB guided by STAM	0.7484	0.7672	**0.8211**	0.7789
STAM [5]	0.6742	0.6237	0.0953	0.4644
CPB guided by PSPNet	**0.8631**	0.7443	0.7527	0.7868
PSPNet [6]	0.8408	0.1317	0.6190	0.5305
CPB guided by DeepLabV3+	0.8419	**0.7707**	0.7688	**0.7931**
DeepLabV3+ [7]	0.6965	0.5624	0.2555	0.5048
SuBSENSE [22]	0.6744	0.6530	0.6934	0.6753
KDE [2]	0.6456	0.6483	0.6754	0.6561
GMM [13]	0.6372	0.6423	0.6743	0.6519
BMOG [23]	0.6584	0.6830	0.6749	0.6793
PBAS [39]	0.6582	0.6412	0.4591	0.5862
DeepBS [25]	0.6705	0.5545	0.6332	0.6073
Cascade CNN [26]	0.1025	0.0453	0.0277	0.0585
FgSegNet [30]	0.2668	0.1428	0.0414	0.1503

4.4. Ablation studies

In this subsection, we examine the importance of the three stages in the proposed method when it is applied to LIMU. As presented in Table 6, in Stages 1 - 3, our proposed method's performance gradually improves. In Stage 1, the performance is improved by avoiding the selection of supporting blocks in the foreground area. In Stage 2, the candidate supporting blocks are updated via Eqs. 7 and 8, thereby resulting in a significant improvement.

4.5. Comparison on various numbers of training samples

The performance of the proposed method is compared with that of CPB under various training data sizes in Fig. 7. As the number of frames in the training set increases, the original CPB model's performance rapidly improves. Although the proposed method shows a more stable trend, it substantially outperforms CPB, especially under a low training sample volume. This suggests that the proposed framework can save training time by requiring fewer training samples in real-world applications. In subset LightSwitch,

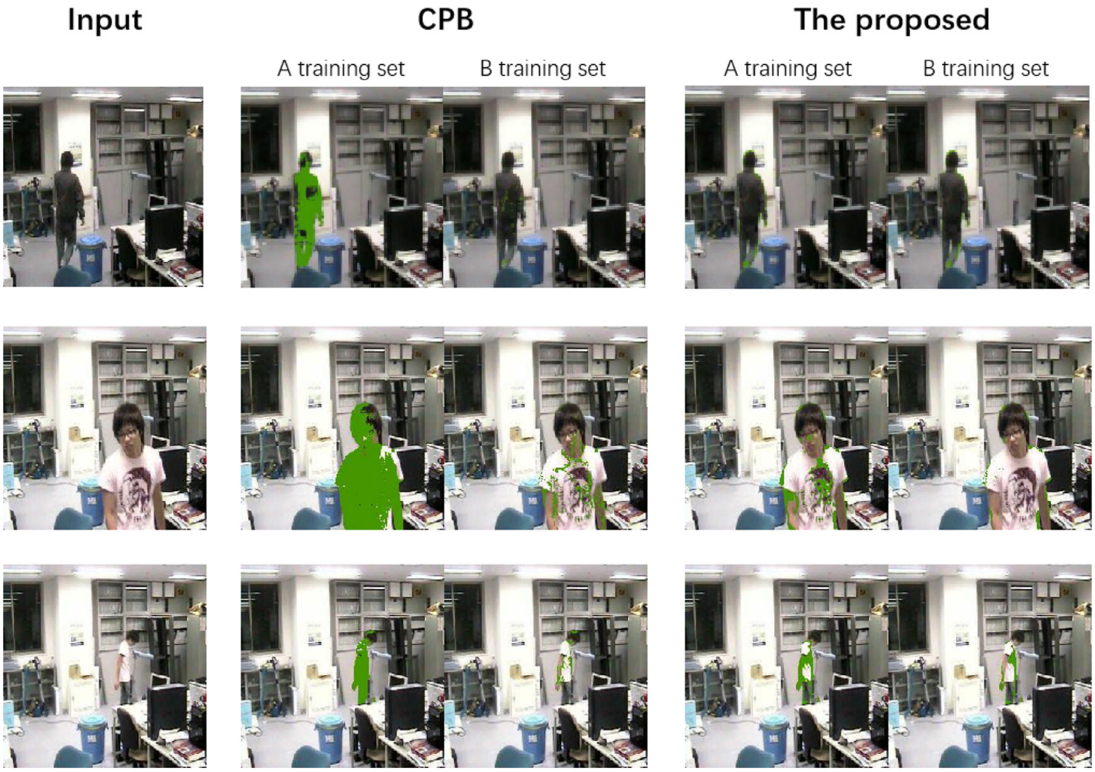

Fig. 8. False negatives with various training set qulities.

Table 6
Ablation studies on LIMU.

Methods	Stage 1	Stage 2	Stage 3	F-measure
CPB [4]				0.6652
CPB	✓			0.6831
guided by	✓	✓		0.7519
STAM	✓	✓	✓	0.7789
CPB	✓			0.6950
guided by	✓	✓		0.7703
PSPNet	✓	✓	✓	0.7868
CPB	✓			0.6991
guided by	✓	✓		0.7838
DeepLabV3+	✓	✓	✓	0.7931

when turning off lights, none of the guidance models performed well due to low illumination; hence the results depend more on the number of training samples for CPB to cover low-light conditions.

4.6. Comparison on training samples that vary in terms of quality

We examine the effect of training set quality on the segmentation results. We use CPB and the proposed method with guidence by STAM to conduct experiments on LIMU. The experiment setup is as follows: For each subdataset, we choose 150 frames with a relatively high percentage of foreground objects as training set A and 150 frames with almost no foreground objects as training set B. Then, we perform segmentation on the same test set.

The visualization results for subdataset CameraParameter are presented in Fig. 8. The second and third columns are the segmentation results that were obtained after training the CPB model on A and B, respectively. The fourth and fifth columns are the segmentation results of the proposed method after training on A and B, respectively. Compared to CPB, which generated many false-negative classifications (marked green in Fig. 8) under training set A, the proposed method performs more stably.

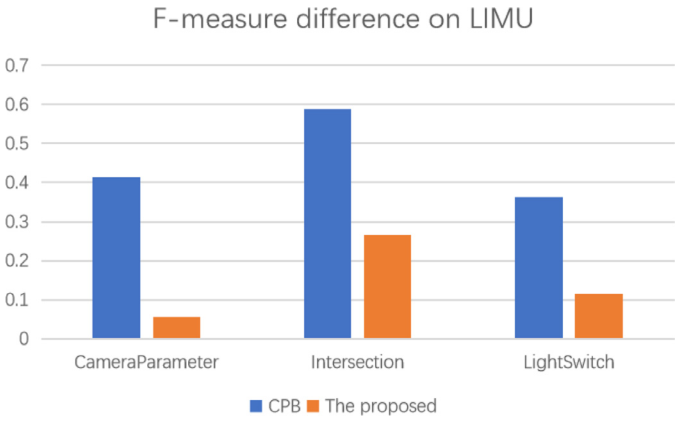

Fig. 9. Training set quality comparision. The F-measure difference between the segmentation results after training on A and B.

In Fig. 9, we plot the change in the F-measure between the segmentation results after training on A and B. The difference in the F-measure of the proposed method is much smaller than that of CPB. The proposed method is less sensitive to the training sample quality and more flexible in selecting training frames.

5. Conclusions

In this paper, we investigate foreground segmentation in dynamic scenes via supervised and unsupervised model communication. Based on the CPB model, three supervised segmentation methods are introduced for completing a coarse-to-fine foreground segmentation and updating the background model. The experimental results demonstrate that the proposed method outperforms CPB on all the experimental datasets. On the CDnet2014 dataset, the proposed method ranks in the top two among all the considered methods. On the WallFlower dataset, the proposed method outperforms all the background subtraction methods. On the LIMU dataset, the proposed method ranks in the top three.

In summary, based on the overall performance, the proposed method significantly outperforms its component methods (three supervised segmentation methods and CPB) in cross-scene tests. The proposed method is more flexible than CPB in terms of the training set size and quality.

The main limitation of the proposed method is its inability to identify qualified spatial supporting blocks when the foreground occupies a large proportion of the frame, which leads to degraded final segmentation results compared to the guidance models.

We plan to evaluate additional guidance models and explore a better selection of guidance models for specified scenarios in the future.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank Dr. Fei Peng from Facebook and Professor Huiyu Zhou from University of Leicester for the language checking. This work is partially supported by AI+ Project of NUAA (XZA20003), National Science Foundation of China (61772268, 61801242, 61976115), and research project (315025305).

References

[1] A. Vacavant, T. Chateau, A. Wilhelm, A benchmark dataset for outdoor foreground/background extraction, Asian Conference on Computer Vision (2012) 291–300.

[2] A. Elgammal, R. Duraiswami, D. Harwood, L.S. Davis, Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, Proc IEEE 90 (7) (2002) 1151–1163.

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

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

[5] D. Liang, J. Pan, H. Sun, H. Zhou, Spatio-temporal attention model for foreground detection in cross-scene surveillance videos, Sensors 19 (23) (2019) 5142.

[6] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, CVPR, 2017.

[7] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, CVPR (2018) 833–851.

[8] D. Liang, S. Kaneko, M. Hashimoto, K. Iwata, X. Zhao, Co-occurrence probability-based pixel pairs background model for robust object detection in dynamic scenes, Pattern Recognit 48 (4) (2015) 1374–1390.

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

[10] P. Spagnolo, T. D'Orazio, M. Leo, A. Distante, Advances in background updating and shadow removing for motion detection algorithms, in: ICASSP, 6, 2005, pp. 2377–2380.

[11] P. Spagnolo, T. Orazio, M. Leo, A. Distante, Moving object segmentation by background subtraction and temporal analysis, Image Vision Comput. 24 (5) (2006) 411–423.

[12] C.R. Wren, A. Azarbayejani, T. Darrell, A.P. Pentland, Pfinder: real-time tracking of the human body, IEEE TPAMI 19 (7) (1997) 780–785.

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

[14] J. Rittscher, J. Kato, S. Joga, A. Blake, A probabilistic background model for tracking, ECCV (2000) 336–350.

[15] N.M. Oliver, B. Rosario, A.P. Pentland, A bayesian computer vision system for modeling human interactions, IEEE TPAMI 22 (8) (2000) 831–843.

[16] Y. Sheikh, M. Shah, Bayesian modeling of dynamic scenes for object detection, IEEE TPAMI 27 (11) (2005) 1778–1792.

[17] M. Heikkila, M. Pietikainen, A texture-based method for modeling the background and detecting moving objects, IEEE TPAMI 28 (4) (2006) 657–662.

[18] W. Hu, X. Li, X. Zhang, X. Shi, S. Maybank, Z. Zhang, Incremental tensor subspace learning and its applications to foreground segmentation and tracking, IJCV 91 (3) (2011) 303–327.

[19] A.T. Chen, M. Biglariabhari, K.I. Wang, Superbe: computationally light background estimation with superpixels, Journal of Real-time Image Processing 16 (6) (2019) 2319–2335.

[20] S.M. Roy, A. Ghosh, Real-time record sensitive background classifier (rsbc), Expert Syst Appl 119 (2019) 104–117.

[21] O. Barnich, M. Van Droogenbroeck, Vibe: a universal background subtraction algorithm for video sequences, IEEE Trans. Image Process. 20 (6) (2011) 1709–1724.

[22] P. Stcharles, G. Bilodeau, R. Bergevin, Subsense: a universal change detection method with local adaptive sensitivity, IEEE Trans. Image Process. 24 (1) (2015) 359–373.

[23] I. Martins, P. Carvalho, L. Cortereal, J.L. Albacastro, Bmog: boosted gaussian mixture model with controlled complexity for background subtraction, Pattern Analysis and Applications 21 (3) (2018) 641–654.

[24] M. Braham, M. Van Droogenbroeck, Deep background subtraction with scene-specific convolutional neural networks, International Conference on Systems (2016) 1–4.

[25] M. Babaee, D.T. Dinh, G. Rigoll, A deep convolutional neural network for background subtraction, Pattern Recognit 76 (2018) 635–649.

[26] Y. Wang, Z. Luo, P. Jodoin, Interactive deep learning method for segmenting moving objects, Pattern Recognit Lett 96 (2017) 66–75.

[27] G. Shi, T. Huang, W. Dong, J. Wu, X. Xie, Robust foreground estimation via structured gaussian scale mixture modeling., IEEE Trans. Image Process. 27 (10) (2018) 4810–4824.

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

[29] M. Qiu, X. Li, A fully convolutional encoder-decoder spatial-temporal network for real-time background subtraction, IEEE Access 7 (2019) 85949–85958.

[30] L.A. Lim, H.Y. Keles, Foreground segmentation using convolutional neural networks for multiscale feature encoding, Pattern Recognit Lett 112 (2018) 256–262.

[31] L.A. Lim, H.Y. Keles, Learning multi-scale features for foreground segmentation, Pattern Analysis and Applications (2019) 1–12.

[32] H. Ding, X. Jiang, A.Q. Liu, N.M. Thalmann, G. Wang, Boundary-aware feature propagation for scene segmentation, ICCV (2019) 6819–6829.

[33] H. Ding, Context contrasted feature and gated multi-scale aggregation for scene segmentation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[34] H. Ding, X. Jiang, B. Shuai, A.Q. Liu, G. Wang, Semantic segmentation with context encoding and multi-path decoding, IEEE Trans. Image Process. 29 (2020) 3520–3533.

[35] H. Ding, X. Jiang, B. Shuai, A.Q. Liu, G. Wang, Semantic correlation promoted shape-variant context for segmentation, CVPR (2019) 8885–8894.

[36] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, P. Ishwar, Changedetection.net: a new change detection benchmark dataset, CVPR workshop (2012) 1–8.

[37] B. Zhou, H. Zhao, X. Puig, S. Fidler, Scene parsing through ADE20K dataset., IEEE conference on computer vision and pattern recognition(CVPR) (2017) 633–641.

[38] M. Braham, S. Pierard, M. Van Droogenbroeck, Semantic background subtraction, IEEE ICIP (2017) 4552–4556.

[39] M. Hofmann, P. Tiefenbacher, G. Rigoll, Background segmentation with feedback: the pixel-based adaptive segmenter, Computer Vision and Pattern Recognition Workshops (2012) 38–43.

[40] K. Toyama, J. Krumm, B. Brumitt, B.R. Meyers, Wallflower: principles and practice of background maintenance, ICCV (1999) 255–261.

[41] , 2021. (http://limu.ait.kyushu-u.ac.jp/dataset/en/)

Dong Liang: received the B.S. degree in Telecommunication Engineering and the M.S. degree in Circuits and Systems from Lanzhou University, China, in 2008 and 2011, respectively. In 2015, he received Ph.D. at Graduate School of IST, Hokkaido University, Japan. He is now an Assistant Professor in Pattern Recognition and Neural Computing Lab., College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA). His research interests include computer vision and pattern recognition. He received the Excellence Research Award from Hokkaido University in 2013.

Bin Kang: received the M.S. degree in Circuits and Systems, and the Ph.D. degree in Electrical Engineering from Lanzhou University and Nanjing University of Posts and Telecommunications, in 2011 and 2016, respectively. He is currently work with the College of Internet of Things, Nanjing University of Posts and Telecommunications. His research interests include unsupervised learning, visual tracking and segmentation.

Xinyu Liu: received the B.S. degree in Nanjing University of Aeronautics and Astronautics, China. He is now a master student in College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His research interests include visual segmentation.

Pan Gao: received the B.Eng. degree in computer science and technology from Sichuan Normal University, Chengdu, China, in 2009, and the Ph.D. degree in electronic engineering from the University of Southern Queensland (USQ), Toowoomba, Australia, in 2017. He was a Post-Doctoral Research Fellow at the School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland, working on the V-SENSE project funded by the Science Foundation Ireland. He has been an Associate Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include Multiview/3D video coding, volumetric video processing and compression, and graph signal processing. He received the Publication Excellence Award from USQ in 2015.

Xiaoyang Tan: received the Ph.D. degree from Nanjing University, Nanjing, China, in 2005. He was a Post-Doctoral Researcher with the LEAR Team, INRIAR Rhone-Alpes, Grenoble, France, from 2006 to 2007. He is currently a Professor with the Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He has authored or co-authored over 50 conference and journal papers. His research interests include deep learning, reinforcement learning, and Bayesian learning. He and his colleagues were awarded the IEEE Signal Processing Society Best Paper in 2015.

Shun'ichi Kaneko: received the B.S. degree in Precision Engineering and the M.S. degree in Information Engineering from Hokkaido University, Japan, in 1978 and 1980, respectively, and then the Ph.D. degree in System Engineering from the University of Tokyo, Japan, in 1990. He had been a research assistant of the Department of Computer Science since 1980 to 1991, an associate Professor of the Department of Electronic engineering since 1991 to 1995, and an associate Professor of the Department of Bio-application and Systems Engineering since 1996 to 1996, in Tokyo University of Agriculture and Technology, Japan. He is currently an Professor at the Graduate School of Information Science and Technology, Hokkaido University, Japan. His research interests include machine and robot vision, image sensing and understanding, and robust image registration.