Background Subtraction Based on Integration of Alternative Cues in Freely Moving Camera

Chenqiu Zhao, Aneeshan Sain, Ying Qu, Yongxin Ge, and Haibo Hu

***Abstract*— Previous approaches to background subtraction in freely moving camera typically focus on improving the accuracy of motion estimation. In this paper, we propose that the accurate background subtraction is possible with the integration of alternative cues about foreground and background. We also put forward a novel background subtraction framework called the integration of foreground and background cues. Here, the foreground cues are extracted by the Gaussian mixture model compensated with image alignment, while the background cues are obtained from the spatiotemporal features filtered by the homography transformation. Subsequently, the integration is devised as a hierarchical competition procedure based on super-pixels under multiple levels with the underlying motivation to utilize the exclusiveness between these cues for the compensation of their corresponding defects. The result of competition between foreground and background cues in a particular super-pixel is used as the proximity, and the foreground is segmented by combining super-pixels with proximity under multiple levels. Comprehensive evaluations using standard benchmarks demonstrate the superiority of our work compared with the state-of-the-art.**

***Index Terms*— Background subtraction, freely moving camera, foreground and background cues, exclusiveness, super-pixels.**

I. Introduction

BACKGROUND subtraction is a fundamental task of computer vision, with a diversified range of applications such as in video monitoring [1], optical motion capture [2], and multimedia applications [3]. The task aims at classifying each pixel into two classes, namely, foreground class which represents the moving objects, and background class which represents the background scene. It has been assumed traditionally that video sequences are captured from a stationary camera [1].

Manuscript received May 1, 2018; revised June 23, 2018; accepted June 29, 2018. Date of publication July 9, 2018; date of current version July 1, 2019. This work was supported in part by the National Natural Science Foundation of China under Grants 61602069, 61402062, and 61772093, and in part by the Chongqing Research Program of Basic Research and Frontier Technology under Grants cstc2016jcyjA0468, cstc2015jcyjA40037, and cstc2017jcyjB0305. This paper was recommended by Associate Editor A. Loui. *(Corresponding authors: Yongxin Ge; Haibo Hu.)*

C. Zhao is with the Department of Computing Science, University of Alberta, Edmonton, AB T6G 2E1, Canada (e-mail: zhaochenqiu@gmail.com).

A. Sain was with the Institute of Engineering and Management, Maulana Abul Kalam Azad University of Technology, Kolkata 700064, India (e-mail: saneeshan95@gmail.com).

Y. Qu is with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: ying.qu@cqu.edu.cn).

Y. Ge and H. Hu are with the Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing 400044, China, and also with the School of Big Data and Software Engineering, Chongqing University, Chongqing 401331, China (e-mail: yongxinge@cqu.edu.cn; haibo.hu@cqu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2018.2854273

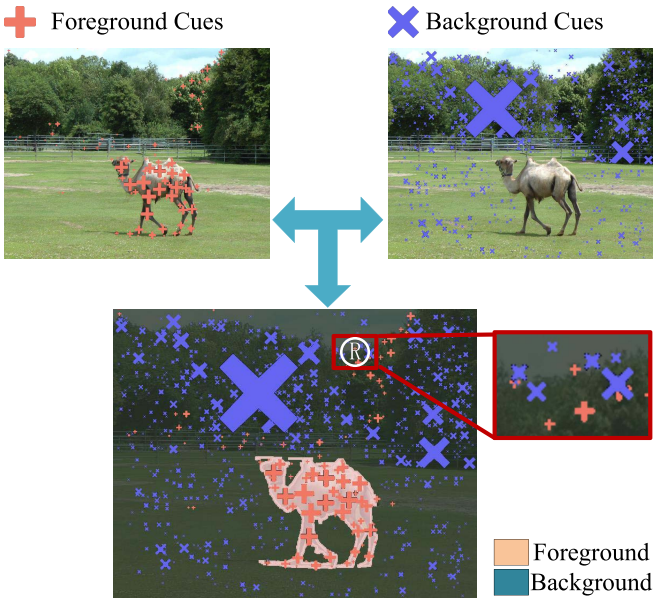


Fig. 1. The illustration of background subtraction via integration of alternative foreground and background cues. Although there are several foreground cues captured in region R, which belongs to background, the integration of foreground and background cues caused the correct labeling of the R as the background.

Under this assumption, the moving object becomes the unique source of motion and contributes to the variation in the intensity of the pixels. In previous approaches the background was subtracted by analyzing variations using a statistical model, such as the Gaussian Mixture Model (GMM) [4]. However, in the frames captured by a freely moving camera, the pixels located in background no longer maintain their respective positions. In fact, the background behaves as another source of motion, resulting in irregular observations in a given pixel. These irregular variations invalidate the algorithms proposed for stationary cameras, thus posing a new challenge to the background subtraction issue.

In a freely moving camera, the motion of foreground overlaps with that of the background thus making accurate estimations difficult. Moreover the complexity of natural scenes also adds up to the challenges. Under the proposed approach there is no need to estimate the motion accurately because the integration of alternative cues is used as a solution. As shown in Fig. 1, due to the complexity of natural scenes, foreground cues are displayed not only on moving objects but also on certain portions of the background like the region R in Fig. 1. Usually, these waving trees are falsely detected as

1051-8215 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

moving objects. Fortunately, benefited from the contribution of background cues, the integration of these cues counteracts these errors, thus leading to a promising binary mask of moving objects. Inspired by these insights, we have focussed on integrating alternative cues rather than improving the accuracy of motion estimation, and a novel framework based on the **I**ntegration of **F**oreground and **B**ackground (IFB) cues is presented for background subtraction in a freely moving camera.

In the IFB framework, foreground cues are captured by the Gaussian Mixture Model (GMM) [4] extended by image alignment, which is used as the compensation of camera motion. Meanwhile, background cues are represented by the spatio-temporal features as filtered by homography transformation [5], where it has been considered as relatively static between features. Subsequently, considering the exclusiveness between foreground and background cues, a competition procedure, based on super-pixels under multiple levels, is proposed to integrate these alternative cues, where the strategy of multiple levels is employed to improve the accuracy of proposed approach. For each super-pixel, the statistic of foreground cues is competed with one of the background cues, and the result is used as the proximity. Thereafter, several proximity images are captured by the combination of super-pixels with proximity under multiple levels where the densities of super-pixels are increased. Finally, the mean of these proximity images is compared with the threshold for the final foreground segmentation. The contributions of this paper are summarized as follows:

1) We propose a novel framework called IFB to achieve background subtraction in a freely moving camera. Unlike earlier cases, here, the IFB model focuses on integrating alternative foreground and background cues rather than improving the accuracy of the motion estimation. Both these cues can be used simultaneously to compensate for their respective defects, owing to their mutual exclusiveness.
   The integration of these cues in super-pixels under multiple levels has been found to improve the efficiency and accuracy of our IFB model.
2) We extend the GMM by an image alignment based on geometric constraints between the background image and the current frame to capture foreground cues. Technically a background image is a image without moving objects, which is captured from, and also the representation of, the GMM model.
3) We propose a method based on spatio-temporal features to capture background cues. The motion of the spatio-temporal features is estimated by homography transformation in image sequences, while the features with the same camera motion are used as background cues. In particular, the SIFT [6], SURF [7] and KAZE [8] features are used as the examples of spatio-temporal features.

II. Related Work

A number of impressive algorithms have recently been proposed to detect moving objects when using a freely movin camera. We have discussed these algorithms briefly in the next three sections. Firstly, algorithms based on spatio-temporal features are reviewed in Section II-A. Algorithms based on geometric constraints are then discussed in Section II-B. Finally, as super-pixels are used for integration, we discuss algorithms based on super-pixels in Section II-C.

A. Algorithms Based on Spatio-Temporal Features

Traditional background subtraction algorithms have assumed the camera to be stationary [9]–[11], and the statistic models such as the Gaussian Mixture Model [4], [12] have been used widely to handle this problem. However, without the assumption of stationary camera, it becomes a considerable challenge [13], in which the difficulty fundamentally arises from the simultaneous motion of the camera and the moving object. A considerable number of earlier works have classified features using motion information in an attempt to address this issue. Approaches based on the trajectory (e.g. [13]–[20]) and the optical flow (e.g. [21]–[28]) in particular, are popular for background subtraction in a freely moving camera.

Sundaram *et al.* [19] tracked the motion by point trajectories. To handle the texture-less areas, Ochs *et al.* [17] and Brox and Malik [19] utilized the color information to analyze the long term trajectory. Similarly, the shape of trajectory is also used to subtract the background [14]. In addition, Elqursh and Elgammal [16] proposed the Bayesian filtering framework for the classification of trajectories, and Cui *et al.* [17] classified the trajectories of foreground and background through a statistic model. Hence, Berger and Seversky [21] utilized the linear subspace of trajectories to describe the background. Specifically, Wu *et al.* [13] assumed that the motion of moving objects is stronger than the camera motion, and segmented the moving objects by the difference between two kinds of motion.

The trajectory can be considered as the integration of optical flow in time sequence [29], which is captured between two consecutive frames and has wide applications in motion estimation [30]–[33], object segmentation [34], [35] and motion segmentation [36]–[38]. Kwak *et al.* [21] initialized the motion field using optical flow, and maintained the background model by Bayesian filters, which is related to the Bayesian inference of optical flow estimation [39] and the background subtraction method proposed by Lim and Han [27]. In addition to that, Zamalieva *et al.* [25] utilized the Markov Random Fields [40] to optimize the propagated model. Similarly, Zhu and Elgammal [29] considered background subtraction as a multi-label problem rather than a binary-label problem, and employed the Markov Random Fields [40] to perform pixel-wise labeling. Moreover, Narayana *et al.* [23] used the orientations of optical flow to improve the robustness of the algorithms, enabling greater consistency with scenes from real-world motion. Motivated by the same issue, Sun *et al.* [24] formulated a fully connected layered model to capture long-range correlations in natural scenes. Bideau and Learned-Miller [28] combined the angle and magnitude of the optical flow to maximize the differences between the motion of objects. Recently, Tokmakov *et al.* [37] utilized the end-to-end deep learning network to learn a coarse representation

of optical flow field features for decreasing the errors in optical flow.

However, optical flow is sensitive to illumination variation due to the assumption of brightness consistency. To handle it, Mohamed *et al.* [41] proposed the illumination-robust optical flow, which is based on the texture rather than the brightness, and the Kumar *et al.* [42] decomposed the brightness into reflectance and illumination components by a stochastic brightness. Unfortunately, due to the diversity of natural scenes, these approaches perform unsatisfactorily in such complex natural scenes. Moreover, the computational cost is another defect of optical flow [43]. Even with the help of recent advances [44], [45] related to deep learning, capturing the accurate optical flow in various complex natural scenes is still quite challenging a task. Hence, it is possible that moving objects produce a motion pattern similar to the camera, due to the unpredictability of such moving objects. In this condition, neither the trajectory nor the optical flow approaches can label the foreground for a moving camera correctly, thus failing to work in complex scenes or video, obtained from a fast-moving camera. In this study, features with motion information are only utilized to capture background cues, where they are integrated with foreground cues for background subtraction in a freely moving camera. It is the exclusiveness between these two kinds of cues that preserves the efficiency of the proposed method.

B. Algorithms Based on Geometric Constraints

A number of algorithms have been developed to counteract motion using geometric constraints (e.g. [17], [46]–[55]), because of the invalidation of traditional algorithms as a result of camera motion.

Lim *et al.* [53] initialized background and foreground labels using a fundamental matrix, in which graph-cut algorithms were used for foreground segmentation. However, the degeneracy of the fundamental matrix means that the approach in [53] is unsuitable for detecting small moving objects. On the contrary, Liu and Gleicher [56] assumed that the background is dominant in a scene. A region is labeled as foreground when its fundamental matrix of motion is different from the one described as the global background motion. Unfortunately, the effect of parallax of objects in different layers means that the efficiency of the approach in [55] is low in complex scenes, which is the case in the approach proposed by Mittal and Huttenlocher [57]. In an attempt to address this, Chen *et al.* [54] tackled several parts of the background at once. A global constraint was then utilized to optimize the accuracy of the transformation. Considering the effect of different layers, regions with the same transformation only show part of the background. To solve this problem, Patwardhan *et al.* [57] detected the foreground in multiple pixel layers. Bugeau and Perez [59] utilized the probability value to validate the estimates of optical flows, allowing the Markov Random Fields framework [40] to be utilized to label pixels as foreground or background. In addition to that, Yuan *et al.* [50] proposed the "Plane+Parallx", which was first mentioned in [52]. The framework in [52] is used to apply 2D layer homographies and identify dominant layers to prevent the estimation of background motion. Moreover, Zamalieva and Yilmaz [46] proposed a 2.5D background model to describe the scene in terms of both its appearance as well as its geometry. Zhang *et al.* [48] performed a full 3D recovery using a combination of structure with motion and bundle adjustment. They evolved epipolar planes between the moving camera center and the 3D scene, in a development of their previous work in [51]. Hence, Bideau *et al.* [38] proposed a statistical model based on perspective projection constraints and combined the results of semantic segmentation, which is captured by deep learning network for motion segmentation.

Moving objects are the unique source of pixel variation following the counteraction provided by geometric constraints. In several previous methods the foreground was segmented by the analysis of pixel variation via a statistical model. However, these foregrounds contain an abundance of noise since the counteraction is not perfect. Fortunately, it is enough to generate foreground cues for integration, which is the essentially the difference between the proposed approach and those used in previous studies.

C. Algorithms Based on Super-Pixels

Neighboring information is often utilized to improve the robustness of background subtraction (e.g., [13], [59]–[62]). Pixels are not independent [59] and should be related to their neighborhoods, especially those that are similar to each other. Region-based algorithms described in [50] and [63] do not account for the similarity between pixel intensity, which is usually weaker at the border of an object. In the approach outlined in [13] a mean-shift is applied to refine the foreground captured by the trajectory, and this improves the robustness of foreground detection in freely-moving cameras. However, the accuracy of the trajectory also limits the efficiency of the method of Wu *et al.* [13]. Hence, Zhu *et al.* [60] refine the results of saliency detection by using super-pixels [61], which is similar to the mixing process of background and foreground cues. The main difference between the proposed approach and Zhu *et al.*'s work [60] is the procedure of using super-pixels. Zhu *et al.* [60] model the salient object as the optimization of saliency values of super-pixels. In our IFB model, the super-pixels are used for the integration of foreground and background cues. The statistic of these cues in a particular super-pixel is used as its confidence as foreground or background, and the moving objects are segmented by the combination of these labeled super-pixels. The foreground and background cues being mutually exclusive, the defects of these cues are compensated, which enhances the robustness and efficiency of our IFB background model.

III. The IFB Framework

In this section, the details of the IFB framework have been explained. The pipeline of IFB framework is shown in the Fig. 2, which can be divided into three parts corresponding to Section III-A, Section III-B and Section III-C. In Section III-A and Section III-B respectively, we introduce the process of capturing foreground and background cues. Thereafter,

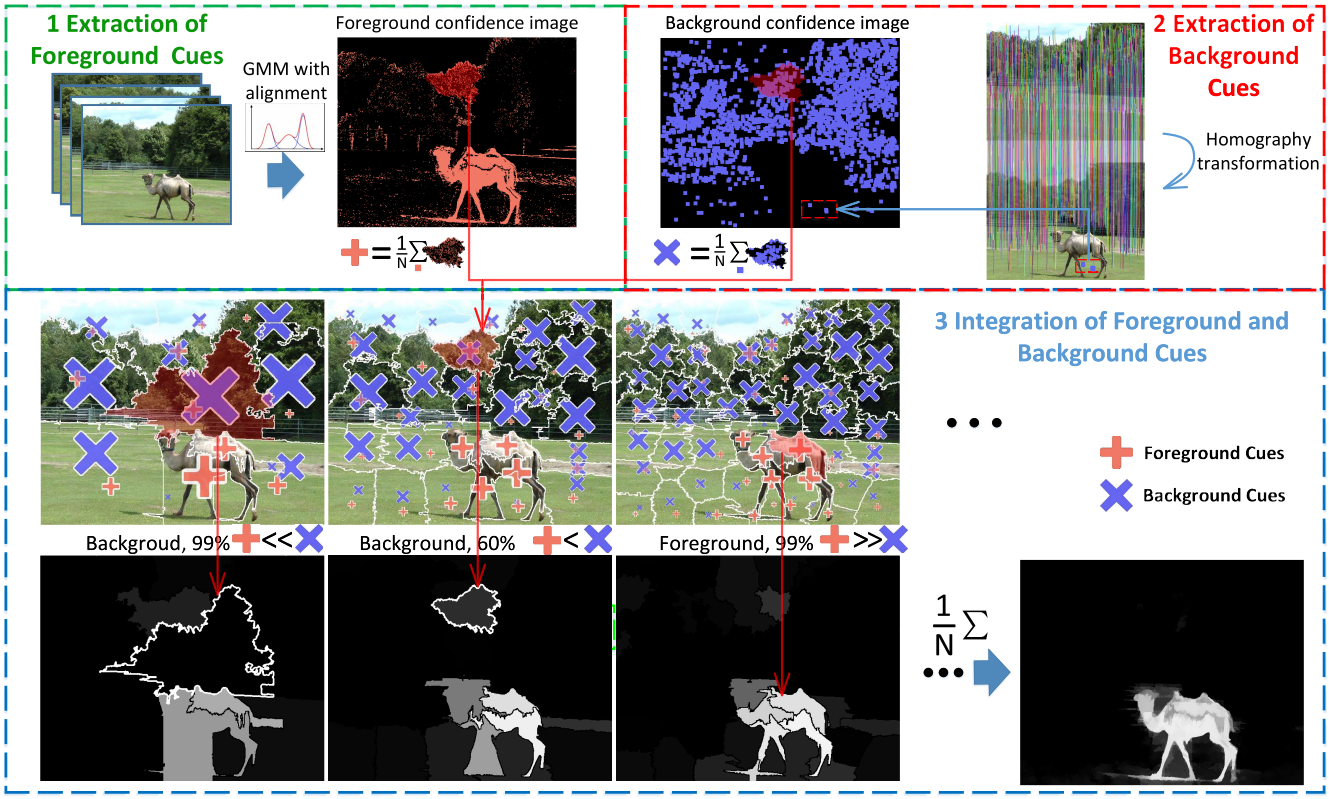


Fig. 2. The illustration of the extraction foreground and background cues as well as the integration of these cues. The foreground and background cues are extracted from the summation of confidence values preserved in foreground and background confidence image respectively. The result of competition between the two cues defines the proximity of super-pixels, while the size of cues represents the intensity of proximity. Larger the cues, higher the possibility. The super-pixels with definite proximity values, constitute the proximity images under multiple levels, and the mean of these proximity images is used as the final result for foreground segmentation.

the integration of these cues by super-pixels under multiple levels is explained in Section III-C.

A. Foreground Cues Extraction

The foreground cues are defined as the proximity of super-pixels that are considered as foreground. It is extracted by the mean of confidence values contained in a particular super-pixel, which is shown as follows:

$$\mathcal{P}_{fg}(Sp) = \frac{1}{||Sp||_1} \sum_{x,y \in Sp} \mathcal{C}_{fg}(x, y), \tag{1}$$

where $\mathcal{P}_{fg}(Sp)$ is foreground cues as well as the proximity of the super-pixel Sp considered as foreground, $\mathcal{C}_{fg}(x, y)$ is the confidence value of pixel at location (x, y) and $||Sp||_1$ represents the number of pixels contained in super-pixel Sp. The confidence values of pixels in current frame are preserved in a foreground confidence image $\mathcal{C}_{fg}$, which is shown as the part 1 of Fig. 2. The foreground confidence image is captured by the Gaussian Mixture Model (GMM) [4] compensated with a frame alignment step. Although, an alignment step is insufficient to counteract the camera motion perfectly, it is adequate for producing rough confidence image for extraction of cues.

The GMM is a classical pixel-based statistical method for background subtraction in a stationary camera. It approximates the variation of pixels' intensity by multiple Gaussian functions, which are then combined with the weights. Let us denote the GMM established for a pixel at location (x', y') as $G(x', y') = \{\omega_k(x', y'), \theta_k(x', y') | k \in [1\ K]\}$, where the ω and θ presents the weights and parameters of each Gaussian function respectively, the k is the index of Gaussian function and the K is the number of Gaussian functions. In the video captured by stationary camera, the positions of pixels remain invariant. Therefore, the GMM located at the position (x', y') of previous frames is directly used in the pixel located at the same position of the current frame. However, in the video obtained from freely moving camera, the pixels no longer maintain their positions, even in two consecutive frames. In order to extend the GMM into freely moving camera, and to capture the confidence of pixel's intensity $\mathcal{I}(x, y)$, each model $G(x', y')$ in last frame is transformed into the corresponding position (x, y) of current frame by image alignment which is shown as follows:

$$\mathcal{C}_{fg}(x, y) = \sum_{k=1}^{K} \omega_k(x', y') \phi(\mathcal{I}(x, y) | \theta_k(x', y')), \tag{2}$$

where $\mathcal{C}_{fg}(x, y)$ is the confidence value preserved in foreground confidence image $\mathcal{C}_{fg}$, ϕ is the Gaussian function and $\theta = \{\mu, \sigma\}$ represent the mean and variance of Gaussian function.

The transformation between the positions of pixels (x, y) and the positions of GMM (x', y') is described by a homography matrix [5] $\mathcal{H}_{bk} = [\vec{h}_1\ \vec{h}_2\ \vec{h}_3]^T$, which is shown as follows:

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \mathcal{H}_{bk} \cdot \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \vec{h}_1 \\ \vec{h}_2 \\ \vec{h}_3 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}. \tag{3}$$

The procedure of transformation between the (x, y) and the (x', y') is captured by the decomposition of Eq. 3, which is shown below:

$$x = tr_x(\mathcal{H}_{bk}, x', y') = \frac{1}{\lambda}\vec{h}_1[x'\ y'\ 1]^T, \tag{4}$$

$$y = tr_y(\mathcal{H}_{bk}, x', y') = \frac{1}{\lambda}\vec{h}_2[x'\ y'\ 1]^T, \tag{5}$$

$$\lambda = \vec{h}_3[x'\ y'\ 1]^T, \tag{6}$$

where the $tr_x()$ and $tr_y()$ are denoted as the transformation functions, which are used to transform the position (x', y') of GMM model to the corresponding position (x, y) of the pixel in the current frame. The homography matrix $\mathcal{H}_{bk}$ is captured by Eq. 10. The details about the matrix $\mathcal{H}_{bk}$ are explained later, since it is captured from the procedure of capturing background cues.

By using Eq. 4 and Eq. 5, the GMM models located at positions of previous frame are transformed into the corresponding positions of the current frame. In particular, the models transformed into impossible positions are discarded, and new GMM models are created for these positions without any GMM model transformed in. For these models successfully transformed into new positions of the current frame, the updating procedure is done to adapt to the environmental change, which is shown below:

$$\begin{aligned}
\mu_k(x, y) &= \alpha_\mu \cdot \mu_k(x, y) + (1 - \alpha_\mu) \cdot \mathcal{I}(x, y), \\
\sigma_k(x, y) &= \alpha_\sigma \cdot \sigma_k(x, y) + (1 - \alpha_\sigma) \cdot (\mu_k(x, y) - \mathcal{I}(x, y))^2, \\
\omega_k(x, y) &= \frac{\alpha_\omega \omega_k(x, y)}{\sum_{k=1}^{K} \alpha_\omega \cdot \omega_k(x, y)}, \\
\alpha_\mu, \alpha_\sigma, \alpha_\omega &= \begin{cases} \alpha_\mu, \alpha_\sigma, \alpha_\omega, & if(\mathcal{I}(x, y) - \mu_k(x, y))^2 \\ & < \sigma_k(x, y)) \\ 1, & otherwise, \end{cases}
\end{aligned}$$

where the $\alpha_\mu, \alpha_\sigma, \alpha_\omega$ are the parameters for updating, and K is the number of Gaussian functions. In the experiment, the GMM is self-implemented and the default parameters are used for all experiments to emphasize the efficiency of our IFB framework instead of the GMM model, which is only used for the extraction of foreground cues. The updating rates of $\{\alpha_\mu, \alpha_\sigma, \alpha_\omega\} = \{0.99, 0.99, 1.05\}$ were used, and the Gaussian function with number $K = 3$ was used. Moreover, the first frame of video had been used for initializing the GMM to capture the initial cues.

B. Background Cues Extraction

Similar to the definition of foreground cues, the background cues are defined as the proximity of super-pixel that are considered as background. However, the extraction of background cues is independent from that of foreground cues, since the integration is devised as the competition between foreground and background cues, in which the cues must be independent with each other for competition. The background cues are extracted from the mean of confidence values preserved in the background confidence images $\mathcal{C}_{bk}$ as the part 2 of Fig. 2, which is shown as follows:

$$\mathcal{P}_{bk}(Sp) = \frac{1}{||Sp||_1} \sum_{x,y \in Sp} \mathcal{C}_{bk}(x, y), \tag{7}$$

where $\mathcal{P}_{bk}(Sp)$ is the background cues of the super-pixel Sp and $\mathcal{C}_{bk}(x, y)$ is the background confidence value of pixel at location (x, y).

Unlike the foreground confidence image, which is used to preserve the proximity captured by the extended Gaussian Mixture Model, the background confidence image preserves the key points of spatio-temporal features which is considered to be located in background. However, the traditional assumption of background, namely the static part of scenes, is invalid in the frames obtained from a moving camera, since the locations of objects vary thus making the static parts nonexistent. To resolve the issue, we assume the majority of relatively static part to be the background. In particular, the spatio-temporal features are used to describe the different parts of the scene, and features with the same motion estimation are recognized as relatively static. Under this assumption, the background is the part that includes the largest number of relatively static spatio-temporal features, and the background confidence image is used to preserve the positions of these spatio-temporal features.

The background confidence image are identified and captured based on the difference between the current frame and the background image. The background image $\mathcal{I}_{bk}$ is the image without moving objects, which is the representation of GMM model. The intensities of pixels in the background image are directly captured by the mean μ of the Gaussian function with the highest weight, as described below:

$$\mathcal{I}_{bk}(x, y) = \underset{\mu_k(x,y)}{\operatorname{argmax}} (\mu_k(x, y)^0 \cdot \omega_k(x, y)), \tag{8}$$

where $\mu_k(x, y)$ and $\omega_k(x, y)$ are the means and the weights of Gaussian functions located at (x, y). Next, let us denote the spatio-temporal features captured in the current frame and the background image as $\{f_1, f_2, \ldots f_M\}$ and $\{f'_1, f'_2 \ldots f'_{M'}\}$ respectively, where M and M' are the numbers of features. Commonly, each spatio-temporal feature contains at least two components, a descriptor $\vec{h}$ and a position (x, y) of key point. The descriptors of these features are cross-matched to find the common parts between the current frame and the background image. In particular, the squared Euclidean distance is used for matching. Then, the $\mathcal{E}$ is denoted as the set of couples of matched spatio-temporal features and is defined as follows:

$$\mathcal{E} = \langle f_i, f'_i \rangle, \quad f = \{(x, y), \vec{h}\}\ i \in [1\ N] \tag{9}$$

where i is the index of the features in the common part and N is the number of parts. In the experiment, the SIFT (Scale-Invariant Feature Transform) [6], the SURF (Speeded-Up Robust Features) [7] and the KAZE [8] features are used as

the examples of spatio-temporal features to show that the IFB framework is effective with the inputs of different features.

As mentioned above, the background is defined as part includes the largest number of spatio-temporal features following the same transformation matrix which is used to describe the background motion. Let's denote the transformation matrix of the background motion as $\mathcal{H}_{bk}$, it is mathematically shown as follows:

$$\mathcal{H}_{bk} = \underset{H}{\operatorname{argmax}} \sum_{i=1}^{N} g(\mathcal{L}(\mathcal{H}, f_i, f_i'), T_m), \tag{10}$$

where $g(\mathcal{L}(\mathcal{H}, f_i, f_i'), T_m)$ is used to judge whether the transformation between f_i and f_i' follows the matrix $\mathcal{H}$ in an error range described by threshold value T_m. The $g(x, y)$ is a piecewise function:

$$g(x, y) = \begin{cases} 1, & x < y \\ 0, & otherwise, \end{cases} \tag{11}$$

and the $\mathcal{L}()$ is the loss function for the evaluation of errors when we transform the positions of features f_i' in the background image to the corresponding positions of features f_i in the current frame by matrix $\mathcal{H} = [\vec{h}_1 \ \vec{h}_2 \ \vec{h}_3]^T$. The definition of $\mathcal{L}()$ is shown as follows:

$$\begin{aligned} &\mathcal{L}(\mathcal{H}, f_i, f_i') \\ &= \mathcal{L}(\mathcal{H}, (x_i, y_i), (x_i', y_i')) \\ &= ||x_i - tr_x(\mathcal{H}, x_i', y_i')||_1 + ||y_i - tr_y(\mathcal{H}, x_i', y_i')||_1 \\ &= |x_i - \frac{1}{\lambda}\vec{h}_1[x_i' \quad y_i' \quad 1]^T| + |y_i - \frac{1}{\lambda}\vec{h}_2[x_i' \quad y_i' \quad 1]^T|, \end{aligned} \tag{12}$$

where the tr_x and tr_y are the transformation functions which are shown in the Eq. 4 and Eq. 5 respectively. The (x_i, y_i) and the (x_i', y_i') are the positions of the key points of the features f_i and f_i' respectively, and the λ is defined in the Eq. 6.

To capture the $\mathcal{H}_{bk}$, homography transformation and RANSAC is used. For the completeness of this paper, we have briefly introduced the methods to capture the $\mathcal{H}_{bk}$ by homography transformation and RANSAC. Please refer to [5] for more details of principle. Due to the property of homography transformation, four couples of features in $\mathcal{E}$, which is shown in Eq. 9, is primarily needed to capture a homography matrix $\mathcal{H}$. Let us denote the positions of key points in these couples as $\{(x_i, y_i), (x_i', y_i') | i \in [1 \ N], N \geq 4\}$. Firstly, the Skey-symmetric matrix for each position is captured as follows:

$$h(x, y) = \begin{bmatrix} 0 & -1 & y \\ 1 & 0 & -x \\ y & x & 0 \end{bmatrix}, \tag{13}$$

where $h(x, y)$ return the Skew-symmetric matrix of position (x, y). Then, these Skey-symmetric matrices are combined as follows:

$$\mathcal{A} = [A_1 \quad A_2 \quad \cdots A_N]^T \quad N \geq 4, \tag{14}$$

$$A_i = h(x_i', y_i') \otimes h(x_i, y_i), \tag{15}$$

where $\otimes$ is denoted as the Kronecker product. The (x_i, y_i) and (x_i', y_i') are positions of the entries in the $\mathcal{E}$, which are also the positions of key points in the current frame and the background image respectively. Finally, the Singular Value Decomposition (SVD) results of matrix $\mathcal{A}$ is the solution of homography matrix between these four positions, which is shown as follows:

$$SVD(\mathcal{A}) = \mathcal{U} \cdot \mathcal{S} \cdot \mathcal{V}^T. \tag{16}$$

The "right singular vector" (a column from $\mathcal{V}$), which corresponds to the smallest singular value, is the solution of the homography matrix $\mathcal{H}$. Then, the RANSAC algorithm randomly selects four couples of $\mathcal{E}$ several times to get several homography matrices $\mathcal{H}_i$. The one which achieves highest value by Eq. 10 is used as the background motion estimation $\mathcal{H}_{bk}$. In the RANSAC algorithm, the number of selections is fixed as 400, which is the default parameter of RANSAC. The features in $\mathcal{E}$ follows background motion is considered as part of background and it is position is preserved in the background confidence image, which is shown as follows:

$$\mathcal{C}_{bk}(x_i^t \pm R, y_i^t \pm R) = \begin{cases} K_{bk} & d(\mathcal{H}_{bk}, f_i, f_i') < T_m, \\ 0 & otherwise \end{cases} \tag{17}$$

where the $\mathcal{C}_{bk}$ is the background confidence image. The R is a radius of key points and the K_{bk} is a constant value to represent the contribution of points. Since the key points are invisible, we simply change the point to a rectangle for visual clarity of the points. The R is the length of side. In these experiments, the R is fixed as 5.

C. Integration of Foreground and Background Cues

The main factor in our IFB model's efficiency is the exclusiveness between the foreground and the background cues, since any defect that exists in one kind of cue gets counteracted by the other. Based on this motivation, the integration is devised as a procedure of competition between the foreground cues and background cues, which is shown as the part 3 of Fig. 2. The competition result is used as the proximity of the super-pixels belonging to foreground or background. Thereafter, a proximity image, consisting of these super-pixels with proximity, is produced in a particular level. Subsequently, several proximity images are hierarchically captured at multiple levels. The mean of these proximity images is used as the final proximity image, which is then compared to a threshold for segmenting moving objects.

However, in the videos obtained by freely moving camera, the numbers of key points, preserved in the $\mathcal{C}_{bk}$ for the extraction of background cues, do not stay invariant, since the numbers of spatio-temporal features are varying with changing background scenes. On the contrary, the foreground confidence images have weak influence from the variation of scenes, since they are captured from the GMM model, which is established for each pixel. In order to handle the imbalance between the $\mathcal{C}_{fg}$ and the $\mathcal{C}_{bk}$, the confidence values preserved in $\mathcal{C}_{bk}$ is adjusted according to the foreground confidence image $\mathcal{C}_{fg}$ captured, which is described as follows:

$$\mathcal{C}_{bk}(x, y) = \frac{\alpha_\gamma \cdot \sum_{(x,y)} \mathcal{C}_{fg}(x, y)}{\sum_{(x,y)} g(0, |\mathcal{C}_{bk}(x, y)|)}, \tag{18}$$

where α_γ is the parameter to control the contribution of background cues, and the $\mathcal{C}_{fg}$ and the $\mathcal{C}_{bk}$ are the foreground and background confidence images respectively. The $g()$ is a piecewise function, which has been defined in Eq. 11.

It is well known that a pixel is not independent but related to its neighborhood, especially the one that is similar to it. Similar pixels have a high probability of sharing the same labels for foreground or background. Therefore, the super-pixels, which consist of similar pixels, are used to integrate the foreground and background cues. In particular, the SLIC [61] algorithm has been chosen to capture super-pixels, because it is based on k-means, which makes the sizes of super-pixels convenient and is controlled by the number of super-pixels. A geometric progression is utilized to control the numbers of super-pixels under multiple levels. Let us denote the numbers of super-pixels under multiple levels as: $\{\alpha_\beta \cdot 2^j | j \in [0\ J]\}$, and the super-pixels controlled by these numbers are denoted as follows:

$$Sp_i^j = \{\mathcal{I}(x_i^j, y_i^j) | i = [1\ \alpha_\beta \cdot 2^j],\ j \in [0\ J]\} \tag{19}$$

where j is the index of levels, i is the index of super-pixels under the level j and J is the user parameter to control the number of levels. The point (x_i^j, y_i^j) represents the position of pixels constituting the super-pixels Sp_i^j.

The integration of cues is devised as a competition process, with the consideration of the exclusiveness between the foreground and background cues. For each super-pixel, the foreground cues is competed with the background cues to capture the proximity about whether the super-pixel is considered as foreground or background, where the foreground cues and background cues are extracted from the means of confidence values as shown in Eq. 2 and Eq. 17 respectively. Mathematically, the competition procedure is shown as follows:

$$\begin{aligned}\mathcal{I}_p^j(x_i^j, y_i^j) &= \mathcal{P}_{fg}(Sp_i^j) - \mathcal{P}_{bk}(Sp_i^j), \text{ãŁŁ} \\ &= \frac{\sum\limits_{x_i^j, y_i^j \in Sp_i^j} \mathcal{C}_{fg}(x_i^j, y_i^j) - \sum\limits_{x_i^j, y_i^j \in Sp_i^j} \mathcal{C}_{bk}(x_i^j, y_i^j)}{||Sp_i^j||_1},\end{aligned} \tag{20}$$

where $\mathcal{P}_{fg}(Sp_i^j)$ and $\mathcal{P}_{bk}(Sp_i^j)$ are the foreground and the background cues located in the super-pixel Sp_i^j, (x_i^j, y_i^j) and $||Sp_i^j||$ are the positions and the number of pixels among to the super-pixel, and $\mathcal{I}_p^j(x_i^j, y_i^j)$ is the proximity image in a particular level. As mentioned earlier, the super-pixels captured in multiple levels constitute different proximity images, and the mean of the proximity images is compared to the threshold T_f for foreground segmentation, which is shown as follows:

$$\mathcal{M}(x, y) = g(T_f, \frac{1}{J}\sum_{j=1}^{J} \mathcal{I}_p^j(x, y)), \tag{21}$$

where the $\mathcal{M}(x, y)$ is the binary mask of moving objects. The J is the number of levels and $g()$ is the piecewise function defined in Eq. 11.

Unfortunately, the errors of alignment is accumulated during the transformation of GMM model by homography matrix. To resolve this issue, a strong updating strategy has been proposed to handle the accumulation of errors, which directly replaces the mean of GMM model by the pixels' intensity of the current frame. The strong updating is done after the foreground segmentation. In the Eq. 8, we captured the intensity of the background image from the mean u_k of Gaussian function with the highest weight. Therefore, in the strong updating procedure, the u_k with the highest weight is replaced by the intensity of corresponding pixel. In order to avoid the moving object, the strong updating is only done in the pixels classified as background, which is shown as follows:

$$\begin{aligned}\mu_k(x, y) &= (1 - \mathcal{M}(x, y))\mathcal{I}(x, y) + \mathcal{M}(x, y)\mu_k(x, y), \\ k &= \operatorname*{argmax}_k w_k.\end{aligned}$$

IV. Experiments

In this section, we have presented the comprehensive experiments for evaluating our IFB model. In the Section IV-A, the dataset and the metrics for evaluation are introduced. The benefit of exclusiveness between foreground and background cues has been demonstrated in the Section IV-B. We have also devised another IFB model in which the background cues are eliminated, and have compared this with the completed IFB model. Finally, in the Section IV-C, we have provided a quantitative evaluation of IFB. The IFB model is then compared with GBSSP [26], calMoSeg [27] and MLayer [28] in the FBMS [16] benchmark.

A. Evaluation Metric and Dataset

As far as our knowledge provides, there has been no acknowledged standard benchmark for evaluating background subtraction methods with a freely moving camera. A number of datasets have been proposed for evaluation (e.g., [14], [17], [24], [27], [28]), amongst which the FBMS [16] dataset is the largest. We have therefore chosen FMBS [16] as the benchmark for evaluating the IFB model. The FBMS dataset [16] is a very challenging dataset, containing 59 video sequences. The camera motion includes translation, rotation and scaling transformations, along with diversity of scenes that adds up to its complexity. Therefore, any comparison using the FMBS dataset is fair and comprehensive.

During the comparison, the Re(Recall), Pr(Precision) and Fm(F-measure) metrics have been used for evaluation. Re and Pr are measures of completeness and accurateness, respectively. Fm is a combination of Re and Pr. These metrics are defined as follows:

$$Re = \frac{TP}{TP + FN}, \quad Pr = \frac{TP}{TP + FP}, \quad Fm = \frac{2 \times Pr \times Re}{Pr + Re},$$

where TP and FP are True Positive and False Positive. Here, positive refers to foreground, while negative represents background. True denotes that the result of this detection is correct, while False means otherwise. Thus, TP means that the result of the detection is foreground as well as being the ground-truth.

B. Exclusiveness Between Foreground and Background Cues

In this section, we discuss the exclusiveness between foreground and background cues. The efficiency of the IFB has been derived from the exclusiveness between foreground and

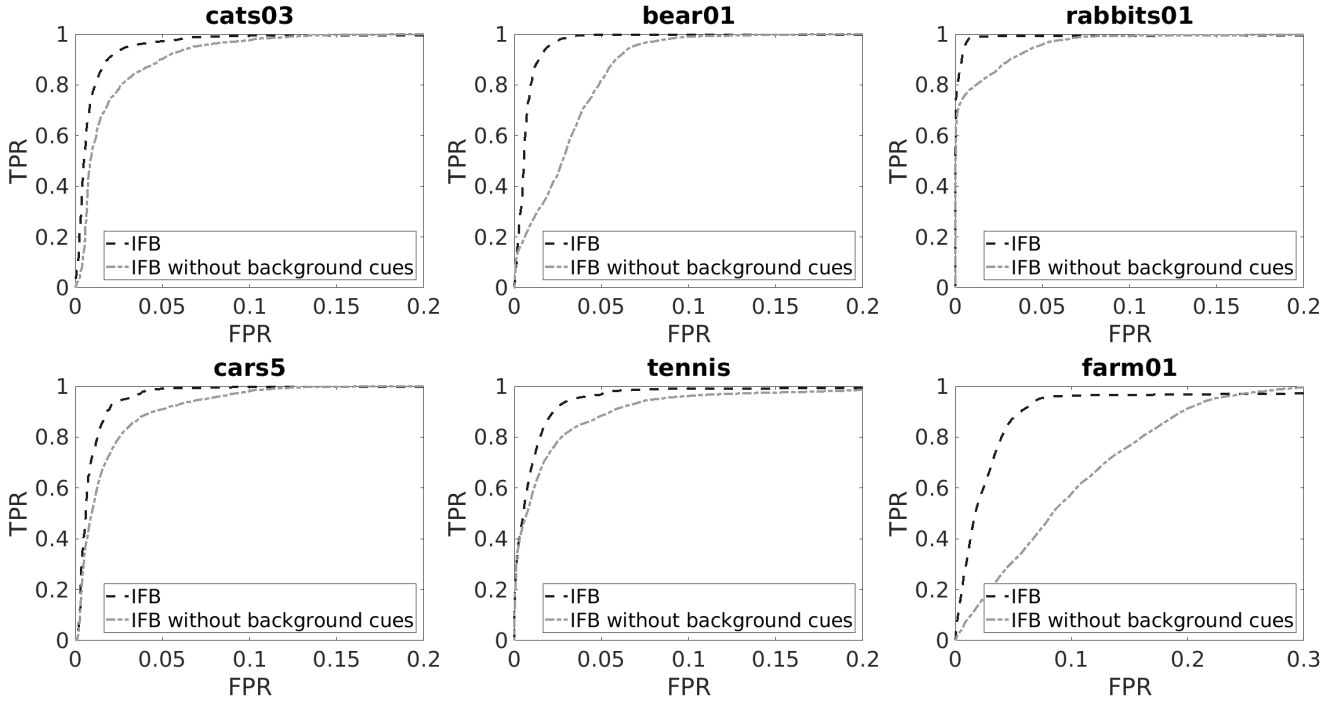


Fig. 3. The ROC curves of IFB and IFB without background cues in cats03, bear01, rabbits01, cars5, tennis and farm01 video sequences.

TABLE I
QUANTITATIVE COMPARISON BETWEEN IFB AND IFB_{nobk} USING RE,PR, FM METRICS

Videos	IFB_{nobk}			IFB		
	Re	Pr	Fm	Re	Pr	Fm
cats03	(**0.9472** , 0.5428 , 0.6901)			(0.9063 , **0.7955** , **0.8473**)		
bear01	(**0.9854** , 0.4806 , 0.6461)			(0.8695 , **0.8673** , **0.8684**)		
rabbits01	(**0.9530** , 0.4433 , 0.6051)			(0.9510 , **0.8730** , **0.9103**)		
cars5	(**0.9887** , 0.3693 , 0.5378)			(0.9437 , **0.7108** , **0.8109**)		
tennis	(**0.9364** , 0.4558 , 0.6132)			(0.8808 , **0.7345** , **0.8010**)		
farm01	(**0.9887** , 0.3897 , 0.5591)			(0.8954 , **0.7462** , **0.8140**)		
Average	(**0.9666** , 0.4469 , 0.6086)			(0.9078 , **0.7879** , **0.8420**)		

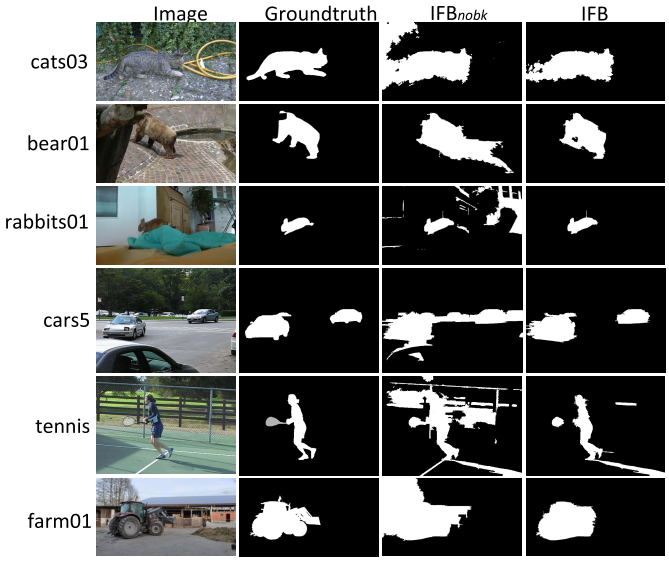


Fig. 4. Qualitative comparison between IFB and IFB_{nobk}.

background cues, without which the IFB cannot segment moving objects accurately. In the IFB model, the foreground cues are used to produce a rough foreground. Meanwhile, the background cues that show the background parts of scenes can be used as compensation for the errors of foreground cues. To demonstrate this, we devise another IFB model whose background cues are eliminated (IFB_{nobk}), and the IFB_{nobk} is compared with the IFB model to demonstrate the necessity of background cues. In particular, six videos from the FBMS dataset have been selected for proper comparisons. The quantitative and qualitative comparisons are shown in Fig. 4 and Table I, respectively. Moreover, the ROC curves of each method are shown in Fig. 3 to reduce the effect of subjectivity in the comparisons.

As shown in Table I, the IFB model has higher Fm scores than their counterparts without background cues. Since the Fm metric is the most important, it is justified to proclaim that the foreground produced by IFB is better than that segmented by IFB_{nobk}. In addition to that, all the foreground cues, whether right or wrong, are used in the foreground segmentation, owing to the lack of background cues. Thus, the foreground segmented by IFB_{nobk} is more complete, and has a higher Re score. However, since the foreground cues are rough cues and falsely detects several background parts as foreground, the Pr scores of IFB_{nobk} are lower than those of IFB. In the IFB model, since the background cues are used to compensate for defects in foreground cues, the IFB provides an obvious improvement in the Pr metrics but with tiny sacrifice in terms of the Re scores obtained. Ultimately, the Fm scores of the IFB

TABLE II

QUANTITATIVE COMPARISON OF THE IFB_{SU}, THE IFB_{KA}, THE IFB_{SI}, WHICH IS THE IFB MODEL WITH THE INPUT OF THE SURF [7], THE KAZE [8] AND THE SIFT [6] FEATURES RESPECTIVELY, AND THREE STATE-OF-THE-ART ALGORITHMS ON THE FBMS DATASET USING FM METRIC

Videos	GBSSP [27]	calMoSeg [28]	MLayer [29]	IFB_{SU}	IFB_{KA}	IFB_{SI}	Videos	GBSSP [27]	calMoSeg [28]	MLayer [29]	IFB_{SU}	IFB_{KA}	IFB_{SI}
dogs01	0.36	**0.82**	**0.82**	0.79	0.77	0.79	lion02	0.41	0.66	0.13	0.79	**0.81**	0.80
dogs02	0.59	0.62	0.82	0.91	**0.93**	0.92	ducks01	0.21	0.31	0.43	0.64	0.59	**0.70**
bear01	0.51	0.48	0.00	**0.86**	**0.86**	**0.86**	meerkats01	0.20	**0.75**	0.24	0.44	0.45	0.64
bear02	0.38	0.82	**0.88**	0.80	0.80	0.85	cars1	**0.93**	0.72	0.91	0.65	0.66	0.70
cats01	0.67	0.46	0.08	0.81	**0.84**	0.81	cars2	0.54	0.59	**0.85**	0.83	0.83	0.79
cats02	0.63	0.29	0.08	**0.86**	0.85	0.82	cars3	0.85	0.37	**0.94**	0.68	0.66	0.73
cats03	0.23	0.73	0.74	0.75	0.79	**0.84**	cars4	0.81	0.55	**0.90**	0.81	0.79	0.88
cats04	0.22	0.17	0.03	0.23	0.32	**0.53**	cars5	0.47	0.85	**0.86**	0.68	0.71	0.81
cats05	0.07	**0.75**	0.05	0.64	0.64	0.64	cars6	0.85	0.76	**0.92**	0.64	0.61	0.63
cats06	0.11	**0.64**	0.02	0.55	0.62	0.60	cars7	0.77	0.67	**0.90**	0.74	0.73	0.76
cats07	**0.85**	0.53	0.31	0.82	0.81	0.82	cars8	**0.90**	0.84	0.86	0.61	0.77	0.81
rabbits01	0.37	0.14	0.00	0.67	0.89	**0.91**	cars9	0.63	0.66	0.59	**0.68**	0.62	0.64
rabbits02	0.47	0.44	0.22	**0.87**	0.86	0.86	cars10	**0.77**	0.66	0.52	0.63	0.63	0.64
rabbits03	0.26	0.34	0.15	**0.60**	0.58	0.52	people1	0.84	0.73	**0.85**	0.53	0.63	0.53
rabbits04	0.25	0.23	0.02	0.49	0.41	**0.51**	people2	0.67	0.83	**0.96**	0.68	0.72	0.80
rabbits05	0.39	0.29	0.10	0.69	**0.70**	0.46	tennis	0.29	0.42	0.61	0.57	0.72	**0.80**
horses01	0.33	0.53	0.87	0.68	0.79	**0.88**	marple1	0.77	**0.80**	0.62	0.47	0.49	0.52
horses02	0.44	0.25	0.00	0.84	**0.90**	**0.90**	marple2	0.25	0.44	0.11	0.45	0.45	**0.47**
horses03	0.67	0.50	0.28	0.69	0.69	**0.73**	marple3	0.62	**0.83**	0.58	0.47	0.60	0.69
horses04	0.34	**0.50**	0.35	0.26	0.45	0.35	marple4	0.72	0.56	0.35	0.89	0.86	**0.91**
horses05	0.57	0.47	0.13	0.55	**0.65**	0.63	marple5	**0.72**	0.56	0.62	0.58	0.66	0.61
horses06	0.30	0.35	0.50	0.55	**0.57**	0.50	marple6	**0.58**	0.51	0.12	0.25	0.12	0.09
people03	**0.64**	0.65	0.00	0.51	0.50	0.59	marple7	0.59	**0.74**	0.45	0.64	0.73	0.68
people04	0.51	0.36	**0.66**	0.52	0.53	0.62	marple8	0.36	0.26	0.37	0.83	**0.87**	0.56
people05	0.22	0.23	**0.76**	0.46	0.49	0.56	marple9	**0.89**	0.59	0.20	0.21	0.28	0.14
camel01	0.28	0.13	0	**0.91**	0.90	**0.91**	marple10	0.46	**0.57**	0.40	0.36	0.53	0.48
farm01	0.64	0.59	0.02	0.68	0.75	**0.81**	marple11	0.65	0.54	0.21	0.56	0.66	**0.68**
giraffes01	**0.76**	0.43	0.20	0.70	0.64	0.69	marple12	0.67	**0.69**	0.07	0.64	0.65	0.34
goats01	0.23	**0.69**	0.42	0.46	0.56	0.64	marple13	**0.75**	0.68	0.68	0.71	0.74	0.72
lion01	0.53	0.56	0.18	0.82	**0.83**	**0.83**	Average	0.53	0.55	0.43	0.64	0.67	**0.68**

model are higher than those of IFB_{nobk}, and the importance of the background cues is prominent.

As shown in Fig. 3, the ROC curves of IFB are closer to the top-left points than to those of IFB_{nobk}. This phenomenon denotes that, benefits of the IFB model outweigh the costs in terms of foreground segmentation. The foregrounds shown in Fig. 4 also reflect this phenomenon, since the foregrounds of IFB have a higher accuracy with a negligible sacrifice in terms of completeness.

C. Quantitative Evaluation of IFB

The IFB model is compared with GBSSP [26], calMoSeg [27] and MLayer [28] in the FBMS [16] benchmark. Both of these compared algorithms are based on optical flow. In particular, the MLayer is based on the large displacement optical flow (LDOF) [30], which is a well-known optical flow extraction algorithm. The quantitative and qualitative comparisons are shown in Table II and Fig. 5 respectively. The results of the compared algorithms have been captured by the implementation provided by authors. For our IFB model, all results have been executed under the same arguments as those shown in Table III. Moreover, the results of IFB_{SU}, IFB_{KA}, IFB_{SI} are captured by the IFB model with the input of the SURF [7], the KAZE [8] and the SIFT [6] features respectively. Keeping in mind the restrictions on the length of this paper, only the results of IFB_{SI} have been shown in the Fig. 5, and the video sequences in the same category have been discussed together.

In the Table II, the Fm scores of different algorithms in the videos of FBMS dataset [16] are shown. Benefiting from the complementarity between foreground and background cues, the IFB_{SU}, IFB_{KA} and IFB_{SI} achieve an undeniable improvement in terms of the average Fm metric. Moreover, the results of IFB with different kinds of input are close, which goes on to prove that the efficiency of our IFB is independent of the input. In the dog01-02 videos, dogs are seen to move across and toward the cameras. The challenges of these video sequences are sourced from the complexity of the camera motion, which includes rotation and scaling transformations. For example, in the dog01 video sequence, a dog walks across the scene. The motion of camera corresponds to a translation transformation. In this condition, the optical flow has sufficient ability to classify the foreground and background. However, in the dog02 video sequence, the camera motion becomes more complex, due to which the optical flow cannot handle this situation. Consequently, both the GBSSP [26] and the calMoSeg [27] show relatively poor performances, as they are based on optical flow. In contrast to the above mentioned cases, owing to excellent performances of the large displacement optical flow(LDOF) [30], the MLayer [28], which is based on the LDOF, delivers satisfactory results. Similarly, the homography transformation utilized in the proposed approach is robust to both these kinds of transformation. As a result, our IFB model works well achieving high performances for both the video sequences.

In the bear01-bear02 video sequences, the bears are moving slowly, and camera motion is much stronger than the motion

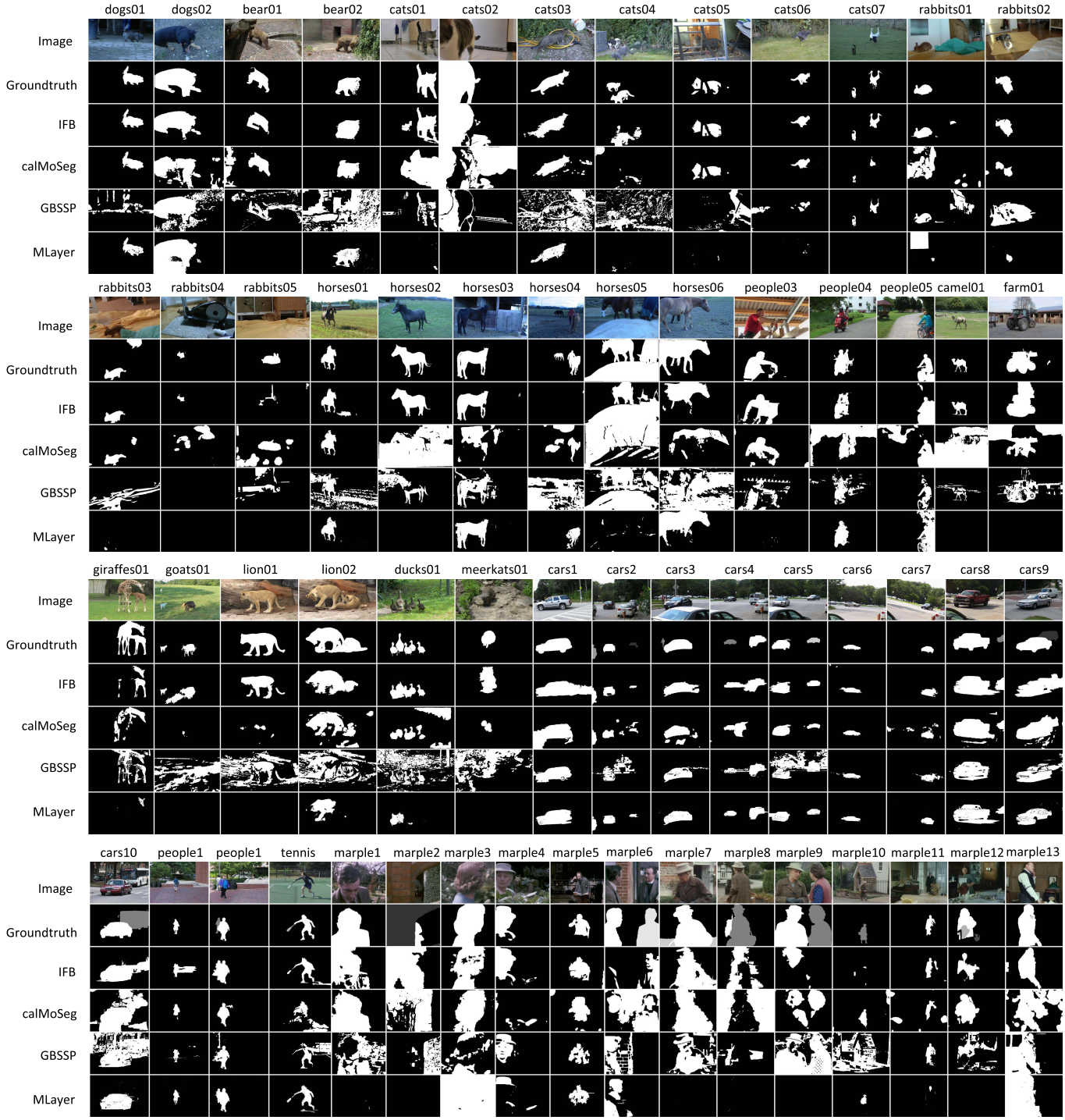


Fig. 5. Qualitative comparison of the IFB_{SI}, which is the IFB model with the input of the SIFT features [6], the GBSSP [26], the calMoSeg [27] and the MLayer [28] in FBMS dataset.

of moving objects. In such conditions, the optical flow is not sensitive enough to classify both the motion of camera as well as that of the moving objects. Therefore, in the bear01 video sequence, the MLayer [28] renders invalid. In our IFB model, the GMM model is used to capture the foreground cues. It is a statistic model and is equipped to handle slow moving objects. Hence, the background cues of IFB model are balanced by the foreground cues, as the IFB model has the ability of segmenting slow moving objects from strong camera motion.

The IFB model renders a high performance in the cats01-07 and rabbits01-05 video sequences as well. In the cats01-07 video sequences, the cats produce a sudden motion, which is the main challenge of these sequences, where the camera moves quickly in a short time. In this condition, it is difficult to use optical flow to describe the sudden motion with any sort of accuracy, which is why none of the GBSSP [26], calMoSeg [27] or the MLayer [28] work well in these kinds of videos. Hence, the foregrounds shown in

TABLE III

THE PARAMETER VALUE OF THE IFB FRAMEWORK AND THE GMM MODEL

IFB						GMM			
T_m	T_f	K_{bk}	α_γ	α_β	J	α_μ	α_σ	α_ω	K
16	75	255	2	16	7	0.99	0.99	1.05	3

Fig. 5 also demonstrate the problem of accuracy among the compared algorithms. On the contrary, the background image of our IFB model is updated at each frame. Therefore, there is no strong camera motion between the current frame and the background image. Hence, the homography transformation of IFB possesses a good ability to handle fast camera motion. As a result, our IFB model performs well for these video sequences. Unfortunately, however, our IFB model cannot address the camouflage of the cat's coat, as the foreground cues are captured from the GMM, which is based on color information. This is the reason why our IFB model achieves only limited Fm scores in the cats04 video sequence.

In the horses01-06 video sequences, the IFB model achieves significantly high improvement. In these video sequences, the horses are running fast in the natural scenes. Besides the strong camera motion, the complex natural scene is another challenge of these video sequences. It is highly challenging to produce an accurate motion estimation in the complex natural scenes, which is why, compared algorithms show poor performances in these videos. Fortunately, the rough cues are enough for our IFB model to produce the accurate foregrounds. As the integration of foreground and background cues improve the efficiency of IFB model, a high performance is demonstrated by our IFB model in these video sequences. Coming to the people03-05 and camel01 video sequences, we observe that the environment of the scenes is more complex. Particularly in the camel01 video sequence, the GBSSP [26] and the calMoSeg [27] only achieve Fm scores of 28% and 13% respectively, while the MLayer [28] is completely invalidated. However, in our IFB model, the extended GMM captures foreground cues for integration, and the alternative between the foreground and background cues improves the accuracy of the foreground segmentation. Finally, the IFB model achieves Fm scores of 91%, which is much higher than the other algorithms.

In the farm01, giraffes01, goats01, lion01-02, ducks01, meerkats01 and tennis video sequences, all the scenes are natural. The motion of these animals is unpredictable and the environment of the scenes is complex. In these kinds of scenes, it is very difficult to estimate the camera motion accurately. However, the focus of our IFB model is on integrating the alternative cues and accurate estimation of motion is unnecessary. The accuracy and efficiency are improved by the super-pixels used for integration. Therefore, our IFB model shows promising performances in these video sequences as well.

Unfortunately, in these videos of FBMS dataset [16], which contains the cars1-10, people1-2 and marple1-13, the numbers of frames in several videos are limited. For example, the cars1 video contains merely 19 frames. In this condition, the statistic model like GMM does not have enough frames to produce enough foreground cues for integration. Most parts of moving objects are recognized as background. Hence, without enough learning frames, the background image captured by GMM model includes moving objects, which affects the extraction of background cues. Both failure of foreground and background cues, contribute to the limited performances of IFB model in these video sequences. As determined experimentally, our IFB model works well when the video includes at least 40 frames. Moreover, in the marple6 and the marple9 video sequences, the moving objects are contained in the initial frames and are seen loitering during other time sequences. Similar to the situation of limited video frames, the IFB is unable to capture enough cues for integration, which is the main reason for unsatisfactory performance of the IFB in these video sequences.

TABLE IV

THE RUNTIME OF DIFFERENT PARTS OF IFB FRAMEWORK RAN IN THE FRAMES WITH RESOLUTION OF 340 × 270(SECONDS/FRAME)

Input	Features Extraction	RANSAC	Integration	Rest Part	Overall
SURF [8]	0.1426	0.1832	0.1069	0.0492	0.4819
KAZE [9]	0.2255	0.2406	0.1282	0.0433	0.6376
SIFT [7]	0.2297	0.1954	0.1158	0.046	0.5869

TABLE V

THE RUNTIME OF DIFFERENT PARTS OF IFB FRAMEWORK RAN IN THE FRAMES WITH RESOLUTION OF 680 × 540(SECONDS/FRAME)

Input	Features Extraction	RANSAC	Integration	Rest Part	Overall
SURF [8]	0.5498	0.4322	0.3636	0.1690	1.5145
KAZE [9]	0.7781	0.4519	0.3987	0.1725	1.8891
SIFT [7]	0.7213	0.4512	0.3663	0.1798	1.8172

D. Implementation Details and Time Complexity

The complete source code of our IFB framework is available at https://github.com/zhaochenqiu/IntegrationFgBk. The IFB framework has been implemented by the mixture of Matlab and C++. C++ has been used to implement several key parts (e.g. the transformation of GMM model, the integration of cues) of IFB framework, while Matlab has been used to combine these parts into a framework. The SURF [7], KAZE [8] and the SIFT [6] features are captured with the help of MexOpencv [64] with default parameters. In the integration parts, multiple threads have been used for acceleration.

The runtime of the proposed approach is shown in Table IV and Table V. In particular, the runtime of IFB framework is evaluated in frames with a resolution of 680 × 560 and 340 × 270. Besides the overall runtime of IFB, the runtime of key parts of IFB are also shown, which include the extraction of SURF [7], KAZE [8] and SIFT [6] features, the RANSAC, the integration of foreground and background cues, and the rest part of IFB framework. The experiments have been run on a PC having i7-6900k with octa-core processor in the Matlab 2017b and G++ environment under Ubuntu 16.04. In the evaluation experiments, all images have not been normalized.

V. CONCLUSIONS

In this paper, we have proposed the IFB framework for background subtraction in the case of a freely moving camera.

Unlike previous works where attempts were made to improve the accuracy of estimation of motion, our IFB focuses on integrating alternative foreground and background cues for foreground segmentation. In particular, foreground cues are detected by a GMM model with the estimation of background motion, while background cues are captured from spatio-temporal features filtered by homography transformation, where the SURF [7], KAZE [8] and SIFT [6] features are used as examples. Thereafter, super-pixels under multiple levels are utilized to integrate these cues. The efficiency of IFB as seen above, results from the exclusiveness between foreground and background cues, along with the utilization of super-pixels under multiple levels. Comprehensive experiments were performed to test the strength of our proposed method under various scenarios. Comparing the corresponding results with other state-of-the-art methods, proved the superior efficiency of our framework, thus demonstrating its potential for use in practical applications.

REFERENCES

[1] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Comput. Sci. Rev.*, vol. 11–12, pp. 31–66, May 2014.

[2] Z. Chen and T. Ellis, "A self-adaptive Gaussian mixture model," *Comput. Vis. Image Understand.*, vol. 122, pp. 35–46, May 2014.

[3] B. Han and L. S. Davis, "Density-based multifeature background subtraction with support vector machine," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1017–1023, May 2012.

[4] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, vol. 2, 2004, pp. 28–31.

[5] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[7] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2006, pp. 404–417.

[8] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 214–227.

[9] W. Kim and C. Jung, "Illumination-invariant background subtraction: Comparative review, models, and prospects," *IEEE Access*, vol. 5, pp. 8369–8384, 2017.

[10] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Understand.*, vol. 122, pp. 4–21, May 2014.

[11] K. Sehairi, F. Chouireb, and J. Meunier, "Comparative study of motion detection methods for video surveillance systems," *J. Electron. Imag.*, vol. 26, no. 2, pp. 26–29, 2017.

[12] M. Chen, X. Wei, Q. Yang, Q. Li, G. Wang, and M.-H. Yang, "Spatiotemporal gmm for background subtraction with superpixel hierarchy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1518–1525, Jun. 2018.

[13] Y. Wu, X. He, and T. Q. Nguyen, "Moving object detection with a freely moving camera via background motion subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 2, pp. 236–248, Feb. 2017.

[14] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sep. 2009, pp. 1219–1225.

[15] A. Elqursh and A. Elgammal, "Online moving camera background subtraction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer, 2012, pp. 228–241.

[16] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1187–1200, Jun. 2014.

[17] X. Cui, J. Huang, S. Zhang, and D. N. Metaxas, "Background subtraction using low rank and group sparsity constraints," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 612–625.

[18] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer, 2010, pp. 282–295.

[19] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by GPU-accelerated large displacement optical flow," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer, 2010, pp. 438–451.

[20] M. Berger and L. Seversky, "Subspace tracking under dynamic dimensionality for online background subtraction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1274–1281.

[21] S. Kwak, T. Lim, W. Nam, B. Han, and J. H. Han, "Generalized background subtraction based on hybrid inference by belief propagation and Bayesian filtering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2174–2181.

[22] D. Yuan and Y. Yu, "Moving object detection under moving camera by using normal flows," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2014, pp. 2517–2522.

[23] M. Narayana, A. Hanson, and E. Learned-Miller, "Coherent motion segmentation in moving camera videos using optical flow orientations," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 1577–1584.

[24] D. Sun, J. Wulff, E. B. Sudderth, H. Pfister, and M. J. Black, "A fully-connected layered model of foreground and background flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2451–2458.

[25] D. Zamalieva, A. Yilmaz, and J. W. Davis, "A multi-transformational model for background subtraction with moving cameras," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2014, pp. 803–817.

[26] J. Lim and B. Han, "Generalized background subtraction using superpixels with label integrated motion estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2014, pp. 173–187.

[27] P. Bideau and E. Learned-Miller, "It's moving! A probabilistic model for causal motion segmentation in moving camera videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 433–449.

[28] Y. Zhu and A. Elgammal, "A multilayer-based framework for online background subtraction with freely moving cameras," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5142–5151.

[29] T. Crivelli, M. Fradet, P.-H. Conze, P. Robert, and P. Pérez, "Robust optical flow integration," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 484–498, Jan. 2015.

[30] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.

[31] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1744–1757, Sep. 2012.

[32] G. Botella, A. Garcia, M. Rodriguez-Alvarez, E. Ros, U. Meyer-Baese, and M. C. Molina, "Robust bioinspired architecture for optical-flow computation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 4, pp. 616–629, Apr. 2010.

[33] G. Botella, U. Meyer-Baese, A. García, and M. Rodríguez, "Quantization analysis and enhancement of a VLSI gradient-based motion estimation architecture," *Digit. Signal Process.*, vol. 22, no. 6, pp. 1174–1187, 2012.

[34] J. Cheng, Y. H. Tsai, S. Wang, and M. H. Yang, "SegFlow: Joint learning for video object segmentation and optical flow," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 686–695.

[35] H. Xiao, J. Feng, G. Lin, Y. Liu, and M. Zhang, "MoNet: Deep motion exploitation for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1140–1148.

[36] K. Chen and D. A. Lorenz, "Image sequence interpolation based on optical flow, segmentation, and optimal control," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1020–1030, Mar. 2012.

[37] P. Tokmakov, K. Alahari, and C. Schmid, "Learning motion patterns in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 531–539.

[38] P. Bideau, A. RoyChowdhury, R. R. Menon, and E. Learned-Miller, "The best of both worlds: Combining CNNs and geometric constraints for hierarchical motion segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 508–517.

[39] P. Heas, C. Herzet, and E. Memin, "Bayesian inference of models and hyperparameters for robust optical-flow estimation," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1437–1451, Apr. 2012.

[40] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*, vol. 1. Providence, RI, USA: AMS, 1980.

[41] M. A. Mohamed, H. A. Rashwan, B. Mertsching, M. A. Garcia, and D. Puig, "Illumination-robust optical flow using a local directional pattern," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 9, pp. 1499–1508, Sep. 2014.

[42] A. Kumar, F. Tung, A. Wong, and D. A. Clausi, "A decoupled approach to illumination-robust optical flow estimation," *IEEE Trans. Image. Process.*, vol. 22, no. 10, pp. 4136–4147, Oct. 2013.

[43] E. Zhu, Y. Li, and Y. Shi, "Fast optical flow estimation without parallel architectures," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 11, pp. 2322–2332, Nov. 2017.

[44] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1647–1655.

[45] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2720–2729.

[46] D. Zamalieva and A. Yilmaz, "Background subtraction for the moving camera: A geometric approach," *Comput. Vis. Image Understand.*, vol. 127, pp. 73–85, Oct. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314214001349

[47] S. Dey, V. Reilly, I. Saleemi, and M. Shah, "Detection of independently moving objects in non-planar scenes via multi-frame monocular epipolar constraint," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 860–873.

[48] G. Zhang, J. Jia, W. Hua, and H. Bao, "Robust bilayer segmentation and motion/depth estimation with a handheld camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 603–617, Mar. 2011.

[49] A. Kundu, C. Jawahar, and K. Krishna, "Realtime moving object detection from a freely moving monocular camera," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2010, pp. 1635–1640.

[50] C. Yuan, G. Medioni, J. Kang, and I. Cohen, "Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1627–1641, Sep. 2007.

[51] G. Zhang, J. Jia, W. Xiong, T.-T. Wong, P.-A. Heng, and H. Bao, "Moving object extraction with a hand-held camera," in *Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.

[52] J. Kang, I. Cohen, G. Medioni, and C. Yuan, "Detection and tracking of moving objects from a moving platform in presence of strong parallax," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Oct. 2005, pp. 10–17.

[53] T. Lim, B. Han, and J. H. Han, "Modeling and segmentation of floating foreground and background in videos," *Pattern Recognit.*, vol. 45, no. 4, pp. 1696–1706, 2012.

[54] C. Chen, S. Li, H. Qin, and A. Hao, "Robust salient motion detection in non-stationary videos via novel integrated strategies of spatio-temporal coherency clues and low-rank analysis," *Pattern Recognit.*, vol. 52, pp. 410–432, Apr. 2016.

[55] F. Liu and M. Gleicher, "Learning color and locality cues for moving object detection and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 320–327.

[56] A. Mittal and D. Huttenlocher, "Scene modeling for wide area surveillance and image synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2000, pp. 160–167.

[57] K. A. Patwardhan, G. Sapiro, and V. Morellas, "Robust foreground detection in video using pixel layers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 746–751, Apr. 2008.

[58] A. Bugeau and P. Pérez, "Detection and segmentation of moving objects in highly dynamic scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.

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

[60] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proc. 27th IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 2814–2821.

[61] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

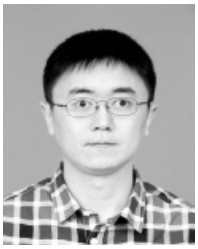
[62] S. Varadarajan, P. Miller, and H. Zhou, "Region-based mixture of Gaussians modelling for foreground detection in dynamic scenes," *Pattern Recognit.*, vol. 48, no. 11, pp. 3488–3503, 2015.

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

[64] K. Yamaguchi. *Mexopencv: Collection and a Development Kit of MATLAB MEX Functions for OpenCV Library*. [Online]. Available: http://www.cs.stonybrook.edu/~kyamagu/mexopencv

Chenqiu Zhao received the B.S. and M.S. degrees in software engineering from Chongqing University, Chongqing, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Multimedia Research Center, Department of Computing Science, University of Alberta. From 2017 to 2018, he was a Research Associate with the Institute for Media Innovation, Nanyang Technological University, Singapore. His current research interests include computer vision, video segmentation, pattern recognition, and deep learning.

Aneeshan Sain received the B.Tech. degree in electrical engineering from the Institute of Engineering and Management, Maulana Abul Kalam Azad University of Technology, Kolkata, India, in 2017, where he is currently pursuing the Ph.D. degree in the field of computer science. His current research interests include computer vision, deep learning, pattern recognition, and other related fields.



Ying Qu received the B.S. degree in applied mathematics from Chongqing Normal University, China, in 2008, and the M.S. degree in applied mathematics from Chongqing University, China, in 2012, where he is currently pursuing the Ph.D. degree in computer science with the College of Computer Science. His current research interests include computer vision, pattern recognition, and machine learning.

Yongxin Ge received the B.Sc. degree in information and computing science, the M.Sc. degree in operations research and control theory, and the Ph.D. degree in computer science and technology from Chongqing University, Chongqing, China, in 2003, 2006, and 2011, respectively. From 2008 to 2009, he was an Exchange Ph.D. Student with the Department of Computer Science, University of Alberta, Canada, under the support of the China Scholarship Council. He is currently an Associate Professor with the School of Big Data and Software Engineering, Chongqing University. He has authored over 40 technical papers in international journals and conferences. His research interests include image processing, pattern recognition, and computer vision.

Haibo Hu received the M.Sc. degree in software engineering and the Ph.D. degree in computer science from Chongqing University, China, in 2004 and 2012, respectively. He is currently an Associate Professor with the School of Big Data and Software Engineering, Chongqing University, where he is also a Research Associate with the Key Laboratory of Dependable Service Computing in the Cyber Physical Society, Ministry of Education. His current research interests include computer vision, pattern recognition, and software engineering.