A Content-Adaptive Resizing Framework for Boosting Computation Speed of Background Modeling Methods

Chun-Rong Huang, *Senior Member, IEEE*, Wei-Yun Huang, Yi-Sheng Liao,
Chien-Cheng Lee, *Member, IEEE*, and Yu-Wei Yeh

***Abstract*—Recently, most background modeling (BM) methods claim to achieve real-time efficiency for low-resolution and standard-definition surveillance videos. With the increasing resolutions of surveillance cameras, full high-definition (full HD) surveillance videos have become the main trend and thus processing high-resolution videos becomes a novel issue in intelligent video surveillance. In this article, we propose a novel content-adaptive resizing framework (CARF) to boost the computation speed of BM methods in high-resolution surveillance videos. For each frame, we apply superpixels to separate the content of the frame to homogeneous and boundary sets. Two novel downsampling and upsampling layers based on the homogeneous and boundary sets are proposed. The front one downsamples high-resolution frames to low-resolution frames for obtaining efficient foreground segmentation results based on BM methods. The later one upsamples the low-resolution foreground segmentation results to the original resolution frames based on the superpixels. By simultaneously coupling both layers, experimental results show that the proposed method can achieve better quantitative and qualitative results compared with state-of-the-art methods. Moreover, the computation speed of the proposed method without GPU accelerations is also significantly faster than that of the state-of-the-art methods. The source code is available at https://github.com/nchucvml/CARF.**

***Index Terms*—Background modeling (BM), frame downsampling, frame resizing, frame upsampling, superpixels.**

I. Introduction

Background modeling (BM) methods are shown to be effective and efficient for foreground segmentation in intelligent video surveillance (IVS) [1]. It serves as one of the most important preprocessing steps for many surveillance applications, such as video event analysis [2], [3], video synopsis [4], [5], and action recognition [6], [7]. As a preprocessing step, achieving real-time efficiency is necessary to avoid the computational bottleneck. Thus, many state-of-the-art BM methods claim that they can achieve real-time computation for processing low-resolution (320 × 240) or standard-definition (640 × 480) videos.

With the increasing resolutions of surveillance cameras, full high-definition (full HD) surveillance videos (1920 × 1080) have become the standard specifications in IVS. Full HD videos can record more details of environments with better resolutions and quality, but will require more computation time to apply BM methods. Obtaining foreground segmentation results of full HD videos by using current BM methods in real time becomes a novel and important issue in IVS. The motivation of this article is to propose a general and novel framework for boosting the computation speed of any BM methods without using hardware accelerations. Moreover, the quality of the obtained foreground segmentation results of BM methods with the proposed framework needs to be as similar as that of the original foreground segmentation results.

To reduce the computation time of BM methods, a naive idea is to downsample a full HD video to a low-resolution video via downsampling methods [8], [9]. Then, the low-resolution video is processed by using BM methods to obtain foreground segmentation results. In this way, the computation time of BM methods can be reduced. Nevertheless, during downsampling, the color information of several pixels of the full HD video is merged to retrieve the color information of a pixel of the downsampled low-resolution video. Thus, boundary information between foreground objects is generally unavailable and causes boundary blurs in the low-resolution video. As a result, the quality of foreground segmentation results obtained by using BM methods will degrade. Besides the degradation problem, the foreground segmentation results are in low-resolution forms. To obtain foreground segmentation results in the original resolution, the low-resolution results need to be upsampled based on upsampling methods [10]–[14]. Because the upsampling problem is an ill-posed problem, the upsampled results obtained from pixels of the low-resolution video will usually contain artifacts. The boundaries of segmented foreground objects are hard to be correctly recovered during upsampling. As a result, the quality of the upsampled

Manuscript received December 15, 2019; revised June 9, 2020; accepted August 15, 2020. Date of publication September 15, 2020; date of current version January 17, 2022. This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST109-2221-E-005-063, Grant MOST109-2221-E-155-054, Grant MOST109-2327-B-006-002, and Grant MOST107-2221-E-005-065-MY2. This article was recommended by Associate Editor J. Lu. *(Corresponding author: Chien-Cheng Lee.)*

Chun-Rong Huang, Wei-Yun Huang, and Yi-Sheng Liao are with the Department of Computer Science and Engineering, National Chung Hsing University, Taichung 402, Taiwan (e-mail: crhuang@nchu.edu.tw; rm3028@alumni.nchu.edu.tw; g105056037@mail.nchu.edu.tw).

Chien-Cheng Lee is with the Department of Electrical Engineering, Yuan Ze University, Taoyuan 320, Taiwan (e-mail: cclee@saturn.yzu.edu.tw).

Yu-Wei Yeh is with the Department of Communications Engineering, Yuan Ze University, Taoyuan 320, Taiwan (e-mail: s1044828@mail.yzu.edu.tw).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2020.3018872.

Digital Object Identifier 10.1109/TSMC.2020.3018872

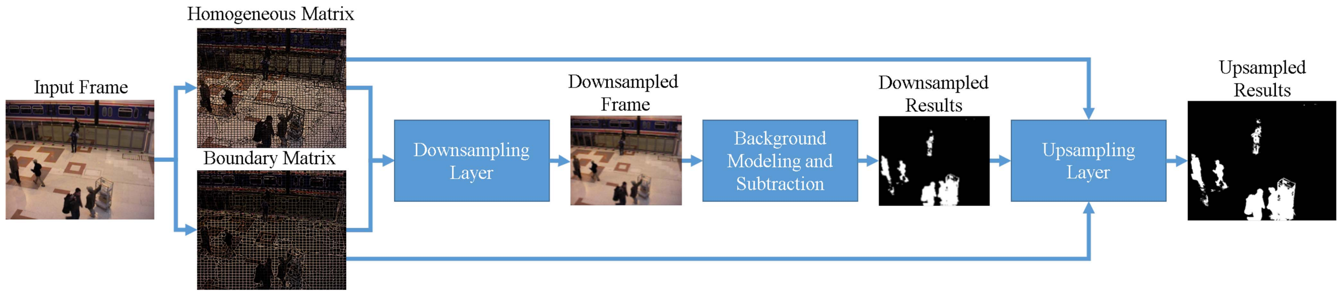


Fig. 1. Overview of the proposed CARF.

results is worse than that of the results processed from the original full HD videos.

To boost the computation speed of BM methods in full HD videos and obtain high-quality foreground segmentation results, two problems need to be solved. The first one is how to downsample the original full HD video to a high-quality low-resolution video, which contains clear boundaries of foreground objects to avoid degradation of performance of BM methods. The second one is how to upsample the low-resolution foreground segmentation results to the original full HD resolution and preserve the detected foreground boundaries of the upsampled results. Most existing downsampling and upsampling methods focus on either downsampling or upsampling images instead of coupling both image downsampling and upsampling steps simultaneously. Thus, existing methods cannot well solve these problems.

In this article, we propose a novel content-adaptive resizing framework (CARF), which couples both downsampling and upsampling layers simultaneously, to boost the computation speed of BM methods. As shown in Fig. 1, we first apply superpixels [15], which are computed based on the content of each frame, to separate the frame as a homogeneous matrix and a boundary matrix. Based on both matrices, the downsampling layer generates the low-resolution frame by using the proposed downsampling interpolation function. Then, the low-resolution frame is processed by using BM methods to obtain low-resolution foreground segmentation results. To obtain high-resolution results, the upsampling layer is applied to upsample the low-resolution results based on the superpixel information to preserve the foreground boundaries. Without pretraining, the downsampling and upsampling layers provide their own unified mappings based on the content of each frame, because each frame contains its own superpixel structure. As a result, our method can achieve the content-adaptive resizing and provide high-quality segmentation results. As shown in the experimental results, both of the quantitative and qualitative results of the proposed method are significantly better than those of the state-of-the-art methods including deep learning methods.

The contributions of this article are threefold. First, this article is the first work to boost the computation speed of BM methods by coupling both downsampling and upsampling layers. Second, the downsampling and upsampling can be achieved by adaptively fit the content of each frame for high-quality resizing results. Third, our method without GPU accelerations outperforms state-of-the-art methods in both computational efficiency and quantitative performance for boosting the computation speed of BM methods. The remaining parts of this article are organized as follows. In Section II, we will review the state-of-the-art methods. Section III describes the proposed method. Section IV shows the experimental results and comparisons with the state-of-the-art methods. Section V gives the conclusions.

II. Related Work

Because of the practical real-time issue of BM methods, hardware-based acceleration methods implemented using CPU and GPU are proposed. For example, Popa *et al.* [16] applied multicores and vector processing of CPUs to implement GMM [17] in the compressed domain. Recently, GPU-based implementations of BM methods have become a new trend due to the parallel processing ability of GPU cores. Pham *et al.* [18] proposed an improved version of GMM by using GPUs on HD videos. CUDA optimization techniques are also considered. Ye and Wan [19] proposed using GPU with the constant memory and asynchronous GPU implementation to accelerate the computation of GMM-based BM methods by using the computational capacity of CUDA cores on GPUs. Boghdady *et al.* [20] also implemented GMM-based BM methods from several simultaneous sources. They also provide a series of novel optimizations, including pinned memory, asynchronous memory transfer, and memory coalescing to improve the overall bandwidth usage. Kumar *et al.* [21] implemented GMM, and related post-processing steps, including morphological operations and blob labeling by exploiting the computational capacity of CUDA cores on GPUs. They also show that GPU implementation achieves significant speedup when performing morphological operations.

Besides the GPU implementation on GMM, implementations of more recent BM methods which require more computation time by using GPUs are also proposed. Poremba *et al.* [22] evaluated the performance of NVIDIA's compute unified device architecture and IBM's cell broadband engine architecture for accelerating different BM methods. They show that GPU implementations can improve the performance of using multithreaded dual-core CPU. Wilson and Tavakkoli [23] implemented a nonparametric statistical BM method by using the CUDA architecture. The statistical models for background pixels and the adaptive mechanism for classifying pixels are also implemented on the CUDA architecture. Qin *et al.* [24] proposed a Vibe-based [25] BM method by using Gabor wavelets filters to obtain foreground segmentation results. They improve the randomized expansion and updating the speed of their method by applying GPU accelerations. Song *et al.* [26] proposed a parallel-connected component labeling method to segment foregrounds by using pixelwise color histograms in GPUs. Foreground segmentation results will be clustered to obtain separate different foreground objects. They also implemented their algorithm by using CUDA. For the review of GPU-based BM implementation, please refer to [27]. Although using the multicore CPU and GPU can accelerate the computation speed of BM methods, it is necessary to reimplement each BM method based on the hardware architectures. It is thus hard to apply the multicore CPUs and GPUs to boost the computation of complicated BM methods such as [28]–[30]. In contrast, the proposed method can be generally applied to all of the BM methods and can also boost the computation speed of these complicated BM methods. To the best of our knowledge, this article is the first work to discuss the boosting of computation speed for different BM methods by using a unified framework of downsampling and upsampling layers without GPU accelerations. As a result, our method can be practically applied to all of the BM methods without modifications, which are required by the hardware-based acceleration methods.

III. Proposed Method

A. Problem Formulation

Given the t-th frame f_t of the surveillance video of the resolution $U \times V$, let $\mathbf{I}_t$ be the 2-D frame matrix of f_t, which contains the colors of the pixels of f_t. The matrix $\mathbf{I}_t$ is composed by a homogeneous matrix $\mathbf{H}_t$ of homogeneous regions and a boundary matrix $\mathbf{B}_t$ of the boundaries of objects of f_t as follows:

$$\mathbf{I}_t = \mathbf{H}_t + \mathbf{B}_t \tag{1}$$

where the dimensions of $\mathbf{I}_t$, $\mathbf{H}_t$, and $\mathbf{B}_t$ are also $U \times V$. Let $\mathbf{I}_t^D$ and $\mathbf{I}_t^U$ be the downsampled and the upsampled frames whose dimensions are $U' \times V'$ and $U \times V$, respectively. To obtain $\mathbf{I}_t^D$, a downsampling layer, which contains the pixel-based downsampling interpolation function $F_t^D(\cdot)$ for each f_t, is applied to $\mathbf{I}_t$ as follows:

$$\mathbf{I}_t^D = F_t^D(\mathbf{I}_t). \tag{2}$$

To recover the high-resolution frame $\mathbf{I}_t^U$, the upsampling interpolation function $F_t^U(\cdot)$ of the upsampling layer is applied to $\mathbf{I}_t^D$, and $\mathbf{I}_t^U$ is represented as follows:

$$\mathbf{I}_t^U = F_t^U\big(\mathbf{I}_t^D\big). \tag{3}$$

To obtain high-quality upsampled results which are as similar as the results of the original frames, we aim to solve the minimization problem as follows:

$$\mathbf{I}_t^{U*} = \underset{\mathbf{I}_t^U}{\arg\min}\big\|\mathbf{I}_t^U - \mathbf{I}_t\big\|^2 \tag{4}$$

where $\|\mathbf{I}_t^U - \mathbf{I}_t\|^2$ is the two norm between $\mathbf{I}_t^U$ and $\mathbf{I}_t$, $\mathbf{I}_t^{U*}$ is the solution of (4), and $\mathbf{I}_t$ is the constraint of the minimization problem. Then, we substitute (1)–(3) to (4) as follows:

$$\mathbf{I}_t^{U*} = \underset{\mathbf{I}_t^U}{\arg\min}\big\|F_t^U\big(F_t^D(\mathbf{H}_t + \mathbf{B}_t)\big) - (\mathbf{H}_t + \mathbf{B}_t)\big\|^2. \tag{5}$$

Nevertheless, solving the minimization problem is very time consuming. Thus, we propose novel downsampling and upsampling interpolation functions, $F_t^D(\cdot)$ and $F_t^U(\cdot)$, which can be adaptively defined based on the content of f_t to solve the optimization problem with high visual quality and computational efficiency. Then, $F_t^D(\cdot)$ and $F_t^U(\cdot)$ are used to downsample the full-HD frames and upsample the low-resolution foreground segmentation results. To represent the elements in $\mathbf{I}_t$, $\mathbf{H}_t$, and $\mathbf{B}_t$, we define the following symbols. Let $\mathbf{p}_k = [x_k\, y_k]^\top$ be the 2-D image position of the kth pixel p_k in $\mathbf{I}_t$. $\mathbf{I}_t(p_k)$ represents the color vector $[r_k\, g_k\, b_k]^\top$ of p_k, where r_k, g_k, and b_k are the red, green, and blue elements of p_k, respectively. Similarly, $\mathbf{H}_t(p_k)$ and $\mathbf{B}_t(p_k)$ are the RGB color vectors of p_k in $\mathbf{H}_t$ and $\mathbf{B}_t$, respectively. Based on (1), $\mathbf{I}_t(p_k) = \mathbf{H}_t(p_k) + \mathbf{B}_t(p_k)$. In the following, we will introduce how to efficiently separate $\mathbf{I}_t$ to $\mathbf{H}_t$ and $\mathbf{B}_t$ based on the content of f_t. Then, CARF is presented to boost the computation speed of BM methods.

B. Content-Based Frame Separation

To separate the frame matrix $\mathbf{I}_t$ to $\mathbf{H}_t$ and $\mathbf{B}_t$ based on the content of f_t, superpixels [15] are applied. As described in [15], to make superpixels adaptively adhere to boundaries of objects in f_t, a color quantized image is generated by dividing pixels into groups based on their colors. Then, a pixel-superpixel assignment is applied to adaptively determine superpixels based on spatially connected and visually coherent groups of pixels of objects. The obtained superpixels can then correctly adhere to the boundaries of objects in f_t and thus are content adaptive for objects in each frame without any pretraining process. In the following, we will introduce how to generate content adaptive $\mathbf{H}_t$ and $\mathbf{B}_t$ from $\mathbf{I}_t$.

Assume that f_t is oversegmented by $U' \times V'$ superpixels. Each superpixel s_i represents a union of pixels as $s_i = \{p_k | p_k \in s_i\}$. For a superpixel s_i, we separate the pixels in s_i into homogeneous and boundary sets, respectively. The pixels of the boundary set s_i^B in s_i are the pixels that spatially connect to the pixels in the neighbor superpixel s_j, where $i \neq j$, as follows:

$$s_i^B = \big\{p_k \,|\, d(p_k, p_l) = 1, p_k \in s_i, p_l \in s_j \;\; \forall i \neq j\big\} \tag{6}$$

where $d(p_k, p_l)$ is defined as follows:

$$d(p_k, p_l) = \big\{\|\mathbf{p}_k - \mathbf{p}_l\| \,|\, p_k \in s_i, p_l \in s_j \;\; \forall i \neq j\big\} \tag{7}$$

where $\mathbf{p}_k$ and $\mathbf{p}_l$ are the 2-D image positions of p_k and p_l, respectively. The homogeneous set s_i^H is then defined as the pixels are in s_i but are not in the boundary set s_i^B as

$$s_i^H = s_i - s_i^B. \tag{8}$$

With s_i^H and s_i^B, we define a content-adaptive mapping function $\mathbf{H}_{t,s_i}(p_k)$ to retrieve pixels of the homogeneous set of s_i as follows:

$$\mathbf{H}_{t,s_i}(p_k) = \begin{cases} \mathbf{I}_t(p_k), & p_k \in s_i^H \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{9}$$

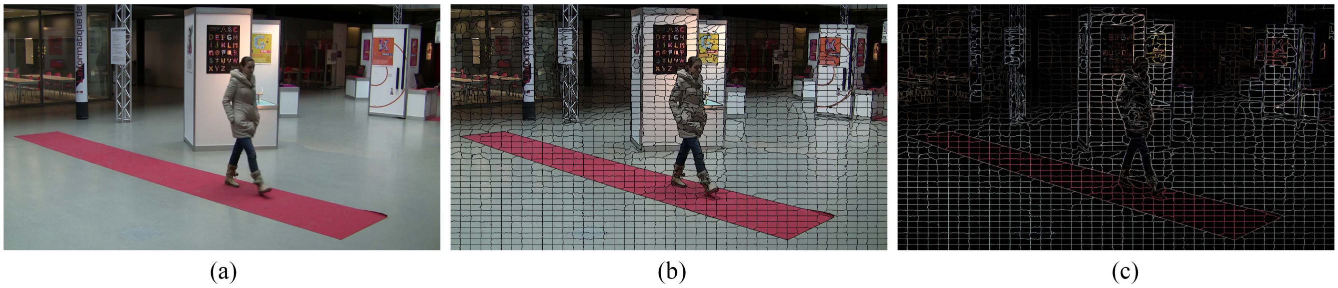


Fig. 2. Visualization results of matrices of (a) $\mathbf{I}_t$, (b) $\mathbf{H}_t$, and (c) $\mathbf{B}_t$ of the surveillance video frame. Please note that the boundaries shown in $\mathbf{H}_t$ and $\mathbf{B}_t$ are content adaptive and adhere to true objects boundaries in the frame.

where $\mathbf{0} = [0\,0\,0]^\top$ represents the black color. Similarly, $\mathbf{B}_{t,s_i}(p_k)$ is defined as follows:

$$\mathbf{B}_{t,s_i}(p_k) = \begin{cases} \mathbf{I}_t(p_k), & p_k \in s_i^B \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{10}$$

which represents the pixels in the boundary set of s_i. Thus, $\mathbf{I}_{t,s_i}(p_k) = \mathbf{H}_{t,s_i}(p_k) + \mathbf{B}_{t,s_i}(p_k)$ is conducted, where $\mathbf{I}_{t,s_i}(p_k)$ represents $\mathbf{I}_t(p_k)$ of p_k in s_i. By using [15], each $\mathbf{I}_t$ can be segmented to $U' \times V'$ superpixels, i.e., the union of superpixels represents $\mathbf{I}_t$. For each superpixel s_i, we retrieve s_i^B and s_i^H by using (6) and (8), respectively. Thus, s_i is the union of s_i^B and s_i^H. Because $\mathbf{B}_t$ and $\mathbf{H}_t$ are constructed based on all of s_i^B and s_i^H, respectively, the union of $\mathbf{B}_t$ and $\mathbf{H}_t$ is then equivalent to $\mathbf{I}_t$ which is constructed based on all of s_i. As a result, $\mathbf{I}_t$ can be separated as $\mathbf{B}_t$ and $\mathbf{H}_t$ as shown in (1).

Fig. 2(a)–(c) shows the visualization results of the matrices $\mathbf{I}_t$, $\mathbf{H}_t$, and $\mathbf{B}_t$, respectively. As shown in Fig. 2(b), the homogeneous sets of each superpixel contain visually similar pixels. For example, pixels of the gray floor are assigned to the superpixels, which also contain pixels of the floor. Similarly, pixels of the red carpet are assigned to the superpixels, which contain pixels of the red carpet. Thus, different objects are adaptively separated by using superpixels based on the content. The boundaries between neighbor superpixels are then embedded and preserved in the homogeneous matrix $\mathbf{H}_t$ and boundary matrix $\mathbf{B}_t$ as shown in Fig. 2(c). In the following, we will introduce how to use $\mathbf{H}_t$ and $\mathbf{B}_t$ to boost the computation speed of BM methods and preserve the quality of the upsampled foreground segmentation results.

C. Content-Adaptive Resizing Framework

As shown in Fig. 1, our CARF contains two layers for boosting the computation speed of BM methods. The first one is the downsampling layer, which is used to downsample f_t to a low-resolution frame. To achieve the goal, we propose a novel content-adaptive downsampling interpolation function $F_t^D(\cdot)$ with respect to the content of f_t to efficiently and effectively obtain the low-resolution frame. Then, the low-resolution frame is processed by using BM methods to reduce the computation time and obtain low-resolution foreground segmentation results. Finally, the foreground segmentation results are upsampled by using the upsampling layer which incorporates the content-adaptive upsampling interpolation function $F_t^U(\cdot)$ with the homogeneous matrix and boundary matrix of f_t to obtain high-resolution foreground segmentation results.

One of the most important properties of superpixels is that superpixels adhere to boundaries of objects in f_t, i.e., the boundary information is encoded in superpixels as shown in Fig. 2(c) without pretraining. Moreover, each superpixel contains visually similar pixels as shown in Fig. 2(b). If the pixels of the downsampled frame are computed from the homogeneous regions of superpixels, the blur effect caused by the interpolation of edge and non-edge pixels can be avoided. Because of these properties of superpixels, each superpixel is represented as a unit to compute a new downsampled pixel of the low-resolution frame, i.e., the number of superpixels equals to the number of pixels of the low-resolution frame. Let two neighbor superpixels s_i and s_j belong to different foreground objects. If the downsampled pixels are constructed from s_i and s_j, respectively, the boundaries between s_i and s_j will naturally be preserved between pixels of the low-resolution frame. Instead of considering time-consuming optimization methods to solve the interpolation problem in (5), we propose a novel content-adaptive downsampling interpolation function $F_t^D(\cdot)$ for each f_t to transfer superpixels to downsampled pixels of the low-resolution frame. In the following, we will introduce how to impose the superpixel information to construct downsampled pixels of the low-resolution frame in a very effective and efficient way.

In our approach, we set the number of superpixels equal to the number of pixels of the low-resolution frame $\mathbf{I}_t^D$. Each superpixel s_i is then corresponding to a pixel p_i^D of $\mathbf{I}_t^D$. The content-adaptive downsampling interpolation function $F_t^D(\cdot)$ aims to compute the color vector of p_i^D based on the corresponding superpixel s_i. The homogeneous set s_i^H represents the pixels of content of s_i, while the boundary set s_i^B contains the pixels of the boundaries of s_i. To avoid the effects of gradual changes of color vectors of pixels in the boundary set, we use the color vectors of pixels in s_i^H to compute the color vector of p_i^D. Thus, $F_t^D(\cdot)$ is designed to obtain the average color vector of s_i^H, and the average color vector is used as the color vector of p_i^D. $F_t^D(\cdot)$ is defined as follows:

$$F_t^D(s_i) = \frac{\sum_{p_k \in s_i} \{\mathbf{H}_{t,s_i}(p_k)\,|\forall p_k \in s_i^H\}}{\sum_{p_k \in s_i} \{1\,|\forall p_k \in s_i^H\}} \tag{11}$$

where the numerator is the summation of colors of pixels in s_i^H and the denominator is the number of pixels in s_i^H. By using (11), the color vector $\mathbf{I}_t^D(p_i^D)$ of the pixel p_i^D is then obtained as follows:

$$\mathbf{I}_t^D\big(p_i^D\big) = F_t^D(s_i). \tag{12}$$

Because s_i^H contains homogeneous pixels of s_i, the obtained $\mathbf{I}_t^D(p_i^D)$ is also visually similar to the colors of the pixels of s_i. Given two neighbor superpixels s_i and s_j containing different colors from different foreground objects, the colors $\mathbf{I}_t^D(p_i^D)$ and $\mathbf{I}_t^D(p_j^D)$ of $\mathbf{I}_t^D$ are computed by using (12), respectively. Since $\mathbf{I}_t^D(p_i^D)$ and $\mathbf{I}_t^D(p_j^D)$ are computed from homogeneous regions of superpixels of different objects, the color vectors of p_i^D and p_j^D will be different. In this way, the boundary between p_i^D and p_j^D is then visually visible in $\mathbf{I}_t^D$. As a result, the boundary between s_i and s_j remains existing between p_i^D and p_j^D of the low-resolution frame which implies that the designed downsampling interpolation function $F_t^D(\cdot)$ can effectively preserve the boundaries of objects during downsampling based on the content of each f_t. In this way, we can obtain the high-quality low-resolution video, which contains clear boundaries of foreground objects to avoid the degradation of the results of BM methods. Here, s_i^H and s_i^B of each superpixel are reserved for boundary information and be used to help achieve high-quality upsampling.

After the computation of the downsampling layer, the low-resolution frame $\mathbf{I}_t^D$ of f_t is obtained. $\mathbf{I}_t^D$ then serves as the input of a BM method. With the low-resolution frames, the proposed method can be applied to state-of-the-art BM methods to boost the computation speed of these BM methods and it is not necessary to perform the modifications of these BM methods. By subtracting the frames and backgrounds provided by BM methods, low-resolution foreground segmentation results are obtained. Without loss of generality, we treat the BM process as a mapping function $\mathbf{M}_t^D(\mathbf{I}_t^D(p_i^D))$ to obtain foreground segmentation results as follows:

$$\mathbf{M}_t^D\big(\mathbf{I}_t^D\big(p_i^D\big)\big) = \begin{cases} 255, & \text{detected foregrounds} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Here, the mapping function $\mathbf{M}_t^D(\cdot)$ is varied with respect to different BM methods, such as GMM [17], Vibe [25], and SuBSENSE [28]. In the experiments, we will also evaluate the effectiveness of our method with respect to different BM methods.

To obtain the high-resolution foreground segmentation results, the upsampling layer is adopted. Because the upsampled boundaries of foreground objects $\mathbf{I}_t^U$ should be as similar as those of $\mathbf{I}_t$, the boundaries of upsampled foreground segmentation results should also be consistent of those of the objects in the original resolution. However, the upsampling problem is an ill-posed problem in general. It is very hard to reconstruct unknown pixels because the boundary information is usually lost during downsampling. Thus, most of the upsampling methods attempt to interpolate the low-resolution mask $\mathbf{I}_t^D$ by exploiting the information in $\mathbf{I}_t^D$. However, such methods have some performance limitations as mentioned above and are not designed for the scenario of boosting BM methods.

TABLE I
INFORMATION OF EVALUATION VIDEOS

Methods	# of Frames	Resolution	Scene
Highway [31]	1700	320×240	Outdoor
PETS2006 [31]	1200	720×576	Indoor
Walking [32]	400	1920×1080	Indoor
Dropping [32]	400	1920×1080	Indoor

Let the upsampled foreground segmentation frame be $\mathbf{I}_t^U$ which has the same resolution $U \times V$ as $\mathbf{I}_t$. The question is how to effectively and efficiently upsample the low-resolution foreground segmentation results $\mathbf{M}_t^D(\mathbf{I}_t(p_i^D))$ to $\mathbf{I}_t^U$. During downsampling, a superpixel s_i is corresponding to a pixel p_i^D. If p_i^D belongs to a foreground object based on $\mathbf{M}_t^D(\cdot)$, its corresponding superpixel s_i should also belong to a foreground object. Because each superpixel adheres to the boundaries of objects, pixels in s_i should also belong to the same foreground object. In contrast, if p_i^D belongs to a background object, its corresponding superpixels s_i and pixels in s_i should also belong to a background object. The content-adaptive upsampling interpolation function $F_t^U(\cdot)$ is designed to decide if a pixel $p_k \in s_i$ belongs to a foreground object or a background object based on the low-resolution foreground segmentation results of p_i^D. Thus, $F_t^U(\cdot)$ is defined as follows:

$$F_t^U\big(\mathbf{I}_t\big(p_i^D\big), p_k\big) = \big\{\mathbf{M}_t^D\big(\mathbf{I}_t^D\big(p_i^D\big)\big)\, |\forall p_k \in s_i\big\}. \tag{14}$$

Then, $\mathbf{I}_t^U(p_k)$ is obtained as

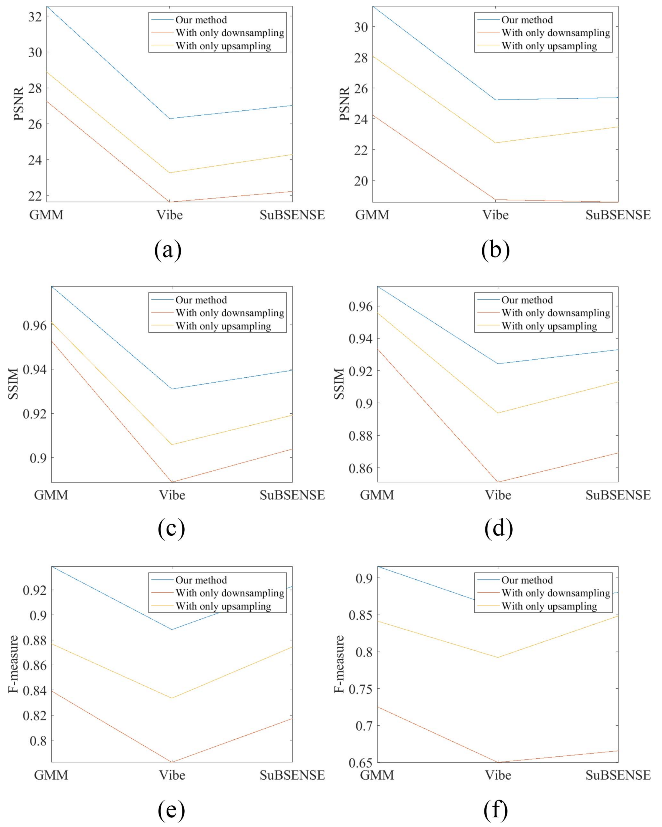
$$\mathbf{I}_t^U(p_k) = F_t^U\big(\mathbf{I}_t\big(p_i^D\big), p_k\big). \tag{15}$$

By using (14) and (15), the value $\mathbf{I}_t^U(p_k)$ of the pixel p_k of $\mathbf{I}_t^U$ can be efficiently computed based on the low-resolution foreground segmentation results. Moreover, the upsampled results can adhere to boundaries of objects and achieve better results. Please note that we do not need to interpolate foreground pixels and directly fill the labels by using the superpixel information, which means that no uncertain values exist. Our method takes no thresholds and is content adaptive. In (15), pixels of the same superpixels of the upsampled frame will have consistent foreground labels. Moreover, the labels of pixels between boundaries will be different based on the superpixel information. Thus, the upsampled foreground segmentation results can also maintain the original boundaries of foreground objects. In addition, the time complexity of our method is low in both of the downsampling layer and upsampling layer, because only pixel-level value assignment is performed based on the content-adaptive downsampling interpolation function and upsampling interpolation function.

IV. Experimental Results

A. Experimental Settings

1) Datasets: In the experiments, to evaluate the performance of the proposed method, we applied four surveillance videos with different resolutions, including the highway video [31], the PETS2006 video [31], the dropping video [32], and the walking video [32]. The highway and PETS2006 videos were used to evaluate the proposed method in low-resolution videos. The dropping and walking videos [32] are full HD surveillance videos for evaluating the performance of the proposed method in high-resolution videos. The detailed information of the videos is shown in



(a) (b)

(c) (d)

(e) (f)

Fig. 3. Average results of the ablation study of the proposed method. (a) PSNR of $s = 2$. (b) PSNR of $s = 4$. (c) SSIM of $s = 2$. (d) SSIM of $s = 4$. (e) F-measure of $s = 2$. (f) F-measure of $s = 4$.

Table I. For the evaluation of the computation time, an Intel i7 3.6-GHz computer with 32-GB RAM and GTX-1080 GPU on Windows 10 is used in the following experiments. Please note that GPU is only used for competing deep learning methods.

2) Comparative Baselines and Evaluation Metrics: Our CARF aims to boost different kinds of BMs methods. To evaluate the generalization ability of the proposed method with respect to different BM methods, three state-of-the-art BM methods are applied, including GMM [17], Vibe [25], and SuBSENSE [28]. The frames of the original resolutions of the evaluation videos are processed by these three methods to obtain the ground-truth (GT) foreground segmentation results. During the experiments, we performed the downsampling factor $s = 2$ and $s = 4$ to obtain the downsampled low-resolution videos. The frames of the downsampled low-resolution videos are processed by these three BM methods to obtain the low-resolution foreground segmentation results, i.e., these BM methods provide $\mathbf{M}_t^D(\cdot)$ for evaluation. Then, the low-resolution results are upsampled by state-of-the-art image upsampling or super-resolution methods.

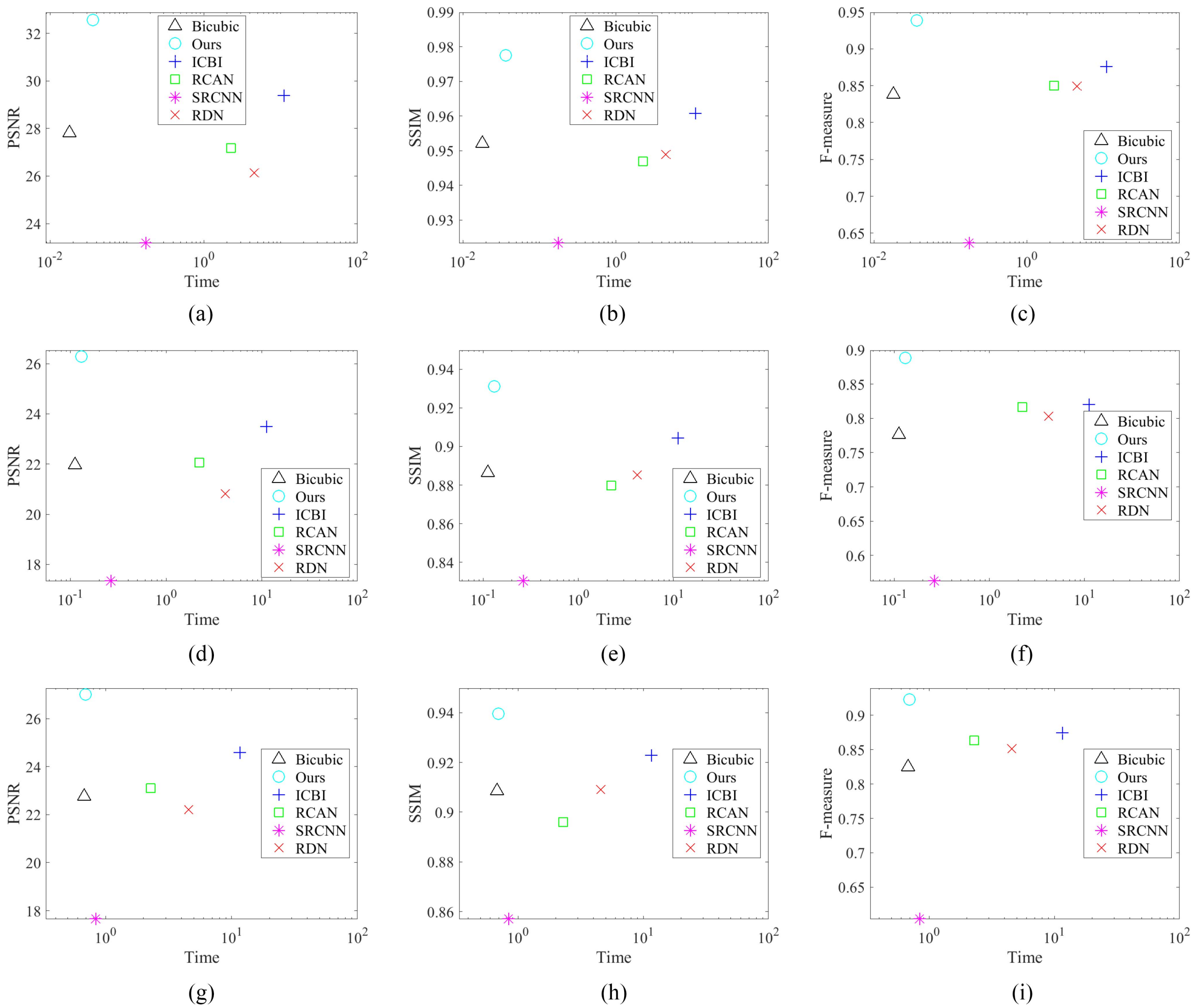
To the best of our knowledge, the proposed method is the first method aiming to boost the computation speed of BM methods by considering frame downsampling and upsampling simultaneously. Thus, we compared our method with traditional upsampling methods, including the Bicubic interpolation [10] and the iterative curvature-based interpolation (ICBI) [11]. By considering the most recent advance of deep learning-based upsampling methods, we also compared our method with SRCNN [33], RDN [34], and RCAN [35] for upsampling quality comparisons. The upsampled foreground segmentation results are compared with the GT to compute PSNR, structural similarity (SSIM) [36], and F-measure values [31].

B. Ablation Study

Our method contains two layers, including the downsampling layer and the upsampling layer. Ablation studies were performed to evaluate the necessity of these two layers. The obtained metrics are the average results of four evaluation videos. With only downsampling layer means that we replace the upsampling layer by using bicubic interpolation. With only upsampling layer means that we replace the downsampling layer by using bicubic interpolation. Fig. 3(a) and (b) shows the PSNR values of $s = 2$ and $s = 4$ with respect to GMM, Vibe, and SuBSENSE, respectively. With only upsampling layer owns better PSNR values compared with only downsampling layer with respect to all BM methods. With the content-adaptive boundary information which is used in our upsampling layer, the upsampled foreground segmentation results can truly adhere to the boundaries of foreground objects and thus, leads to better results. Nevertheless, combining both layers achieves the best results. The SSIM and F-measure values of Fig. 3(c)–(f) also reveal the same results as the PSNR values. As a result, combining both layers are necessary in our method.

C. Quantitative Results

The average PSNR, SSIM, and F-measure values of upsampled foreground segmentation results obtained by BM methods for each evaluation video with respect to $s = 2$ and $s = 4$ are shown in Figs. 4 and 5, respectively. The x-axis represents the average computation time of each method. The y-axis represents the average metrics of each method. Fig. 4(a) shows the PSNR values of our method and competing methods, including Bicubic [10], ICBI [11], SRCNN [33], RDN [34], and RCAN [35] with respect to GMM and $s = 2$. Traditional interpolation methods, such as Bicubic and ICBI are hard to achieve good PSNR values because of the information loss during downsampling. Nevertheless, pixel-based Bicubic and ICBI own better results compared with deep learning methods because GMM detects noisy foregrounds. When noisy foregrounds are detected, they are easily to be enlarged by patch-based deep learning methods. Because RCAN and RDN have deeper network structure, the noise is less enlarged by their networks. Compared with competing methods, our method owns the best PSNR value. It is also significantly faster than ICBI, SRCNN, RCAN, and RDN. By considering content-adaptive superpixels, our method can better obtain the boundary information in the downsampling layer and apply it in the upsampling layer. Fig. 4(b) and (c) also shows the same results with respect to SSIM and F-measure values, respectively. Fig. 4(d)–(f) shows the average PSNR, SSIM, and F-measure values with respect to Vibe, respectively. Because



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 4. Quantitative result of $s = 2$. (a) PSNR, (b) SSIM, and (c) F-measure of GMM; (d) PSNR, (e) SSIM, and (f) F-measure of Vibe; and (g) PSNR, (h) SSIM, and (i) F-measure of SubSENSE.

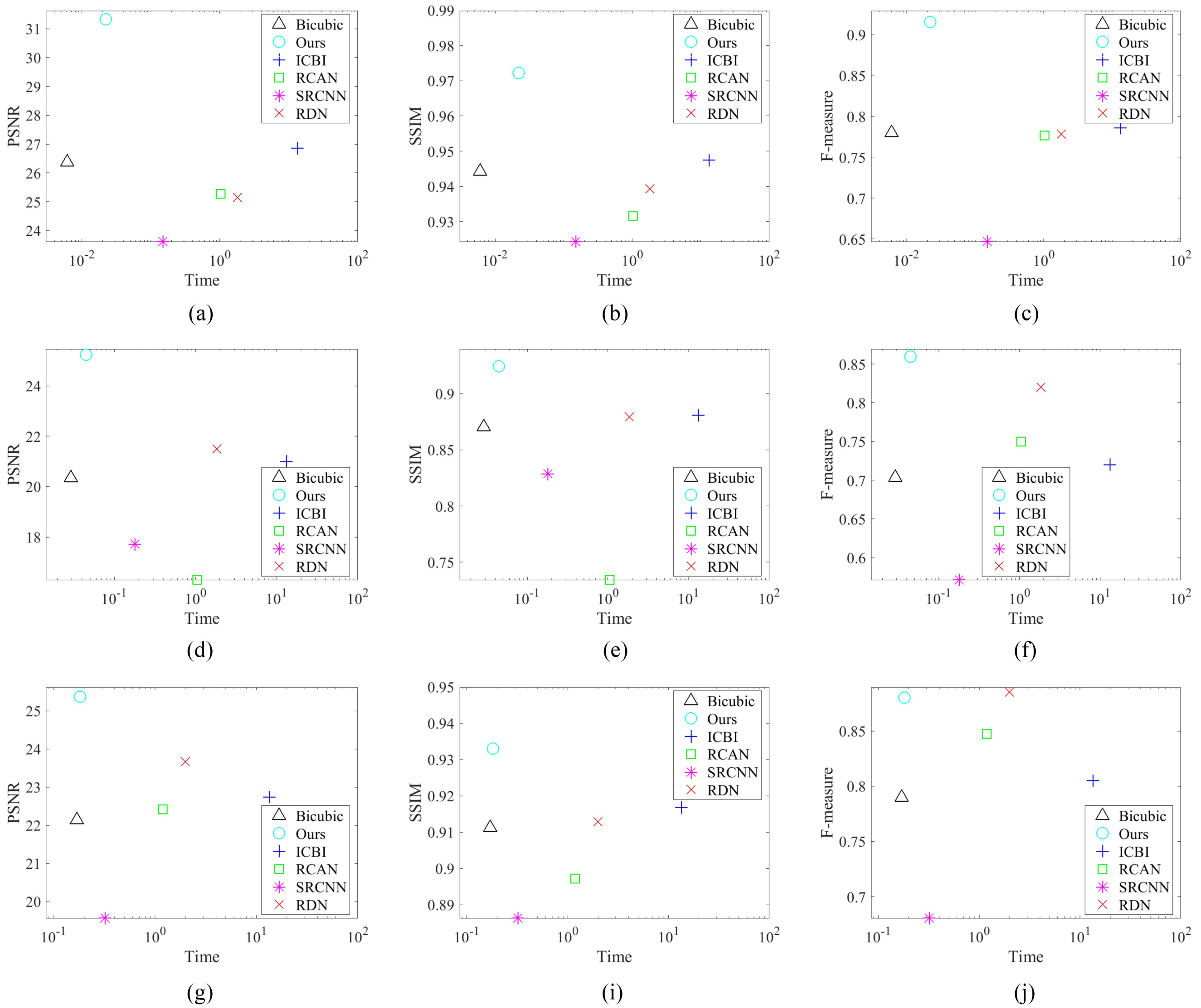
Vibe misdetects fewer backgrounds, deep learning methods, such as RCAN and RDN can then successfully obtain better upsampling results as shown in Fig. 4(d)–(f) compared with the traditional methods. Nevertheless, our method is still the best one. Similar results can be observed in Fig. 4(g)–(i) with respect to SubSENSE.

Also shown in Fig. 5 for $s = 4$ with respect to different background methods, our method achieves the best PSNR, SSIM, and F-measure values. Such results show that downsampling frames by using the proposed downsampling layer and recovering boundaries of the foreground segmentation results by using the proposed upsampling layer are important. Moreover, the boundaries obtained from the superpixels are content adaptive for each frame. That is, even if the BM results are not fit with the boundaries of objects, our method can still recover the foregrounds based on the boundary information. In addition, the results of $s = 4$ are worse than those of $s = 2$ for all of the methods because the information loss of $s = 4$ is more than that of $s = 2$.

TABLE II
AVERAGE COMPUTATION TIME WITH RESPECT TO DATASETS

Scale	Videos	GMM	Vibe	SuBSENSE
s = 1	Highway	0.0041	0.0309	0.1809
	PETS2006	0.0247	0.1588	0.9912
	Walking	0.1161	0.7958	4.9534
	Dropping	0.1154	0.8013	4.9559
s = 2	Highway	0.0038	0.0092	0.0444
	PETS2006	0.0136	0.0473	0.2440
	Walking	0.0611	0.2295	1.2354
	Dropping	0.0661	0.2344	1.2496
s = 4	Highway	0.0019	0.0036	0.0117
	PETS2006	0.0088	0.0174	0.0650
	Walking	0.0385	0.0774	0.3218
	Dropping	0.0391	0.0806	0.3269

Table II shows the average computation time of combining the proposed method with different BM methods with respect to each dataset. In our experiments, the computation time includes the time of obtaining the low-resolution frame, low-resolution foreground segmentation results based on each BM



(a) (b) (c)

(d) (e) (f)

(g) (i) (j)

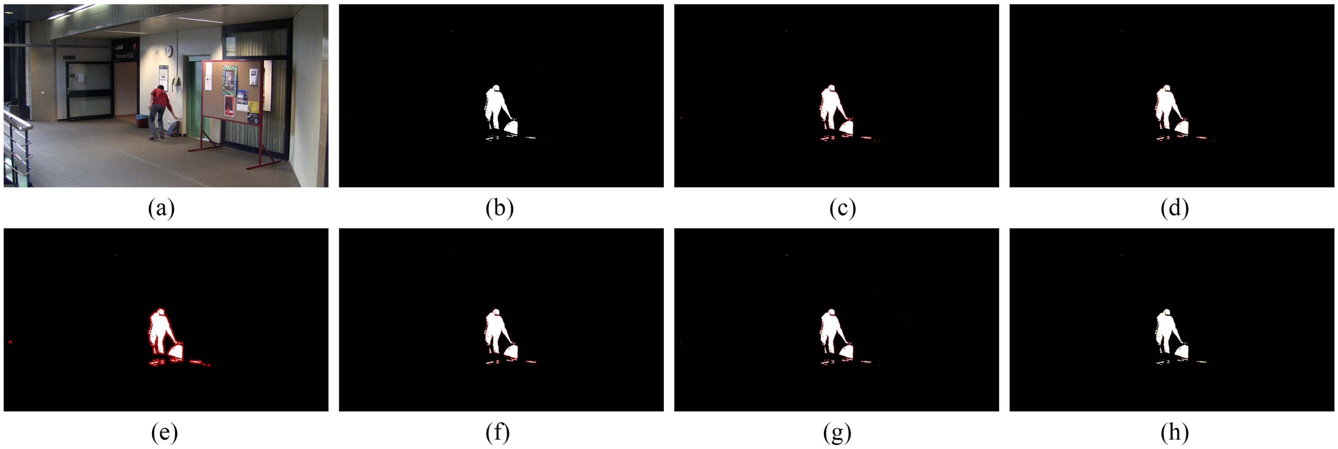
Fig. 5. Quantitative result of $s = 4$. (a) PSNR, (b) SSIM, and (c) F-measure of GMM; (d) PSNR, (e) SSIM, and (f) F-measure of Vibe; and (g) PSNR, (h) SSIM, and (i) F-measure of SubSENSE.

TABLE III
AVERAGE COMPUTATION TIME OF COMPETING METHODS IN SECONDS

Scale	Methods	GMM	Vibe	SuBSENSE
s = 1	Original	0.0651	0.4467	2.7704
s = 2	Bicubic	0.0178	0.1114	0.6760
	ICBI	11.0400	11.1335	11.6981
	SRCNN	0.1762	0.2634	0.8340
	RDN	4.5641	4.1699	4.5641
	RCAN	2.2734	2.2253	2.2802
	Ours	0.0362	0.1301	0.6934
s = 4	Bicubic	0.0061	0.0290	0.1692
	ICBI	13.2106	13.2334	13.3736
	SRCNN	0.1489	0.1794	0.3204
	RDN	1.8179	1.8405	1.9771
	RCAN	1.0314	1.0541	1.1907
	Ours	0.0221	0.0447	0.1814

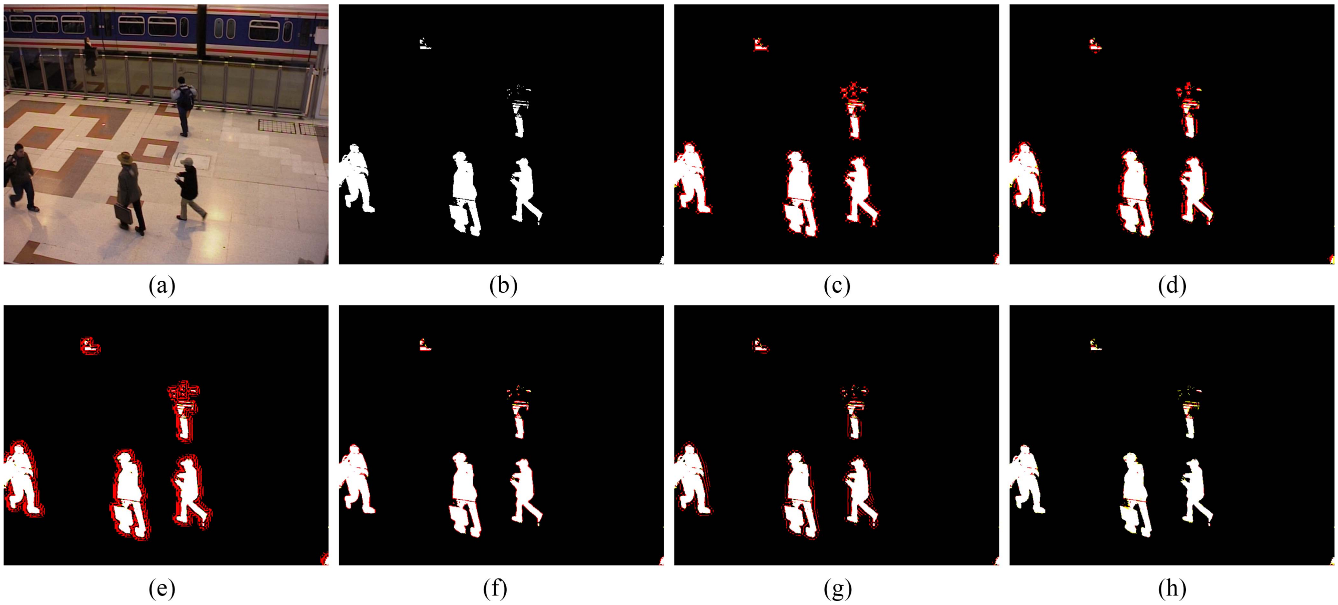
method, and high-resolution foreground segmentation results based on the upsampling method. When GMM is applied to obtain foreground segmentation results from the original videos ($s = 1$), it can achieve real-time performance in low-resolution videos (Highway and PET2006). However, when GMM is applied to full HD videos (Walking and Dropping), its computation time significantly increases. In contrast, when applying the proposed method ($s = 2$ and $s = 4$) with GMM, the computation time of GMM is significantly less than that of GMM in the original resolutions ($s = 1$). Because Vibe and SuBSENSE are more complicated methods compared with GMM, both methods require much more computation time for full HD videos ($s = 1$). By using the proposed method ($s = 2$ and $s = 4$) with Vibe and SuBSENSE, the computation time of both BM methods can also be significantly reduced. The results demonstrate the usefulness of the proposed framework in boosting the computation of BM methods.

Table III shows the average computation time of our method and competing methods under different scale factors in seconds for four evaluation videos. Original means that the average computation time of BM methods performed on the evaluation videos in the original resolutions. As shown in Table III, the computation time of Vibe and SuBSENSE is higher than that of GMM which indicates that Vibe


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6. Qualitative results of $s = 2$ for GMM. (a) Original frame. (b) GT. (c) Bicubic. (d) ICBI. (e) SRCNN. (f) RDN. (g) RCAN. (h) Our method.


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 7. Qualitative results of $s = 4$ for GMM. (a) Original frame. (b) GT. (c) Bicubic. (d) ICBI. (e) SRCNN. (f) RDN. (g) RCAN. (h) Our method.

and SuBSENSE are hard to achieve real-time performance for high-resolution videos. Thus, the computation bottleneck shows the problem of practical usage of these background models and points out the importance of developing a resizing framework as our method to boost the computation speed of these BM methods. Among all of the methods, Bicubic is the fastest one. However, its upsampling results are not good. Without GPU accelerations, our method is the second fast method which is only slightly slower than Bicubic. Nevertheless, our method has the best-upsampled foreground segmentation results. Because SRCNN, RDN, and RCAN have GPU accelerations, they are much faster than ICBI. In addition, even with GPU accelerations, the computation time of SRCNN, RDN, and RCAN is even longer than that of GMM and Vibe in the original resolution. Thus, these deep learning methods are hard to be used for real-time BM applications. In summary, our method can achieve both boosting the computation speed of BM methods and good performance of foreground segmentation results.

D. *Qualitative Results*

In the following, to visualize the error pixels between the results of each method and the GT, we draw the error pixels between the results of each method and the GT by using red pixels and yellow pixels. The red pixels indicate that the upsampling method generates additional foreground pixels that are not generated by the BM method in the original resolution. The yellow pixels indicate that the upsampling method misses foreground pixels that are generated by the BM method in the original resolution. Please note that the following foreground segmentation results are generated by each competing method without any post-processing for fair comparison.

Fig. 6 shows the upsampled qualitative results of GMM with respect to $s = 2$ for the dropping video. Fig. 6(a) and (b) shows the original frame and the GT obtained by GMM in the original resolution. Fig. 6(c)–(h) shows the results of Bicubic, ICBI, SRCNN, RDN, RCAN, and our method, respectively. As shown in Fig. 6(c) and (d), ICBI considers edge information

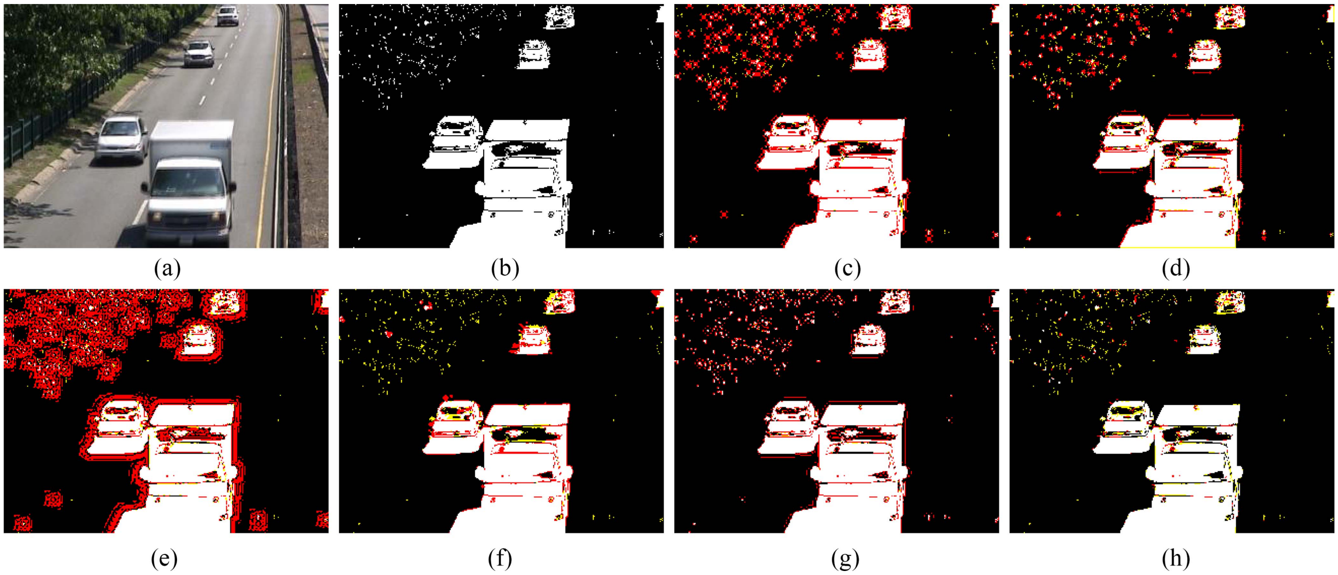


Fig. 8. Qualitative results of $s = 2$ for Vibe. (a) Original frame. (b) GT. (c) Bicubic. (d) ICBI. (e) SRCNN. (f) RDN. (g) RCAN. (h) Our method.

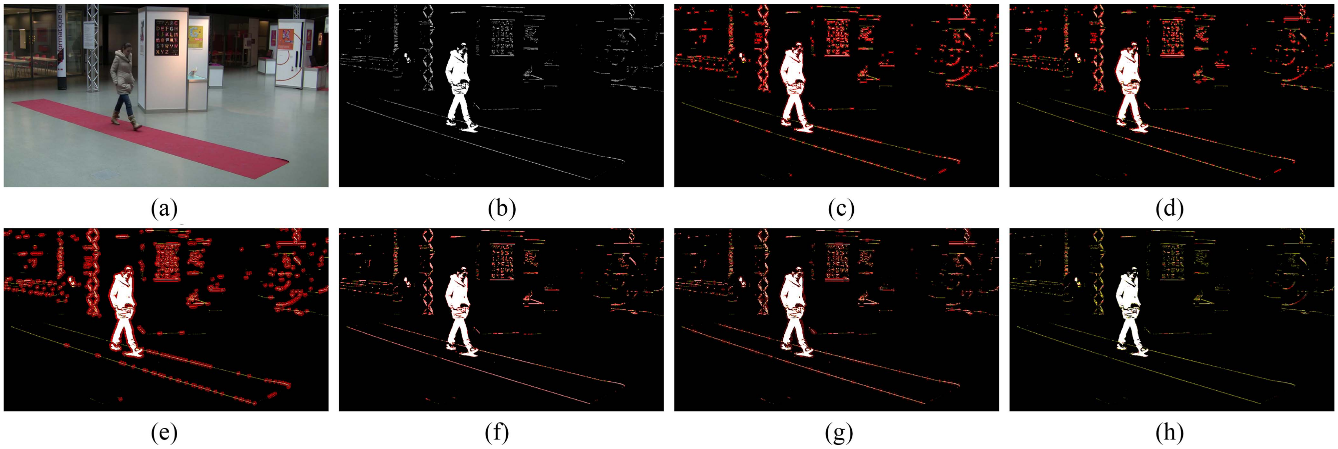


Fig. 9. Qualitative results of $s = 4$ for Vibe. (a) Original frame. (b) GT. (c) Bicubic. (d) ICBI. (e) SRCNN. (f) RDN. (g) RCAN. (h) Our method.

for upsampling, and thus achieves better results compared with Bicubic. Both of the bicubic interpolation and ICBI enlarge the noise of the foreground segmentation results obtained by GMM. Compared with the results of ICBI, SRCNN can also upsample the foreground segmentation results to the original resolution by pretrained models. Since the pretrained models are not able to be online modified to fit the content of surveillance videos, the upsampled results of SRCNN then easily contain more error pixels as shown in Fig. 6(e). Moreover, the noise is further enlarged by SRCNN, because SRCNN performs upsampling based on image patches. When a patch contains noise, SRCNN may incorrectly reconstruct the patch. Nevertheless, with deeper and complicated network structure as RDN and RCAN, the upsampled error pixels can be significantly reduced as shown in Fig. 6(f) and (g), respectively. Fig. 6(h) shows the results of our method. The upsampled results of the proposed method are visually similar to the GT of GMM. Because the proposed CARF is based on superpixels, it can then better represent the details of the content and achieve fewer error pixels. Moreover, our method does not require any pretraining process on collected data which facilitates the practical usage of our method.

Fig. 7 shows the upsampled qualitative results of GMM with respect to $s = 4$ for the PETS2006 video. When the scale factor becomes larger, the noise is much easier to be enlarged as shown in Fig. 7(c)–(e), respectively. Again, as shown in Fig. 7(f) and (g), RDN and RCAN achieve better results compared with SRCNN. Nevertheless, because the trained models of RDN and RCAN do not contain the boundaries of the foreground objects of PETS2006 videos, the upsampled boundaries are still different from the results of the GT. In comparison, the upsampled results of our method shown in Fig. 7(h) successfully fit the boundaries of foreground objects of GTs because of the proposed CARF. Such results indicate the importance of the content-adaptive properties when boosting the computation speed of BM methods.

Besides the evaluation of the content-adaptive properties of our method, we also evaluate the generalization ability of our method with respect to different state-of-the-art BM

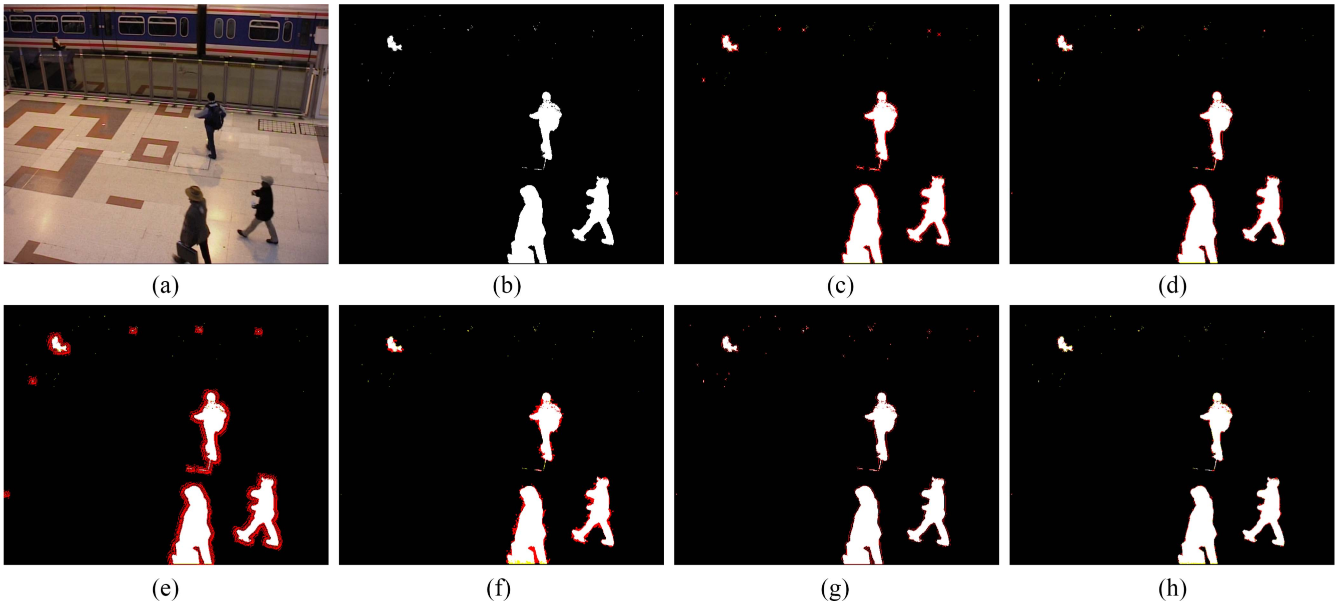


Fig. 10. Qualitative results of $s = 2$ for SuBSENSE. (a) Original frame. (b) GT. (c) Bicubic. (d) ICBI. (e) SRCNN. (f) RDN. (g) RCAN. (h) Our method.

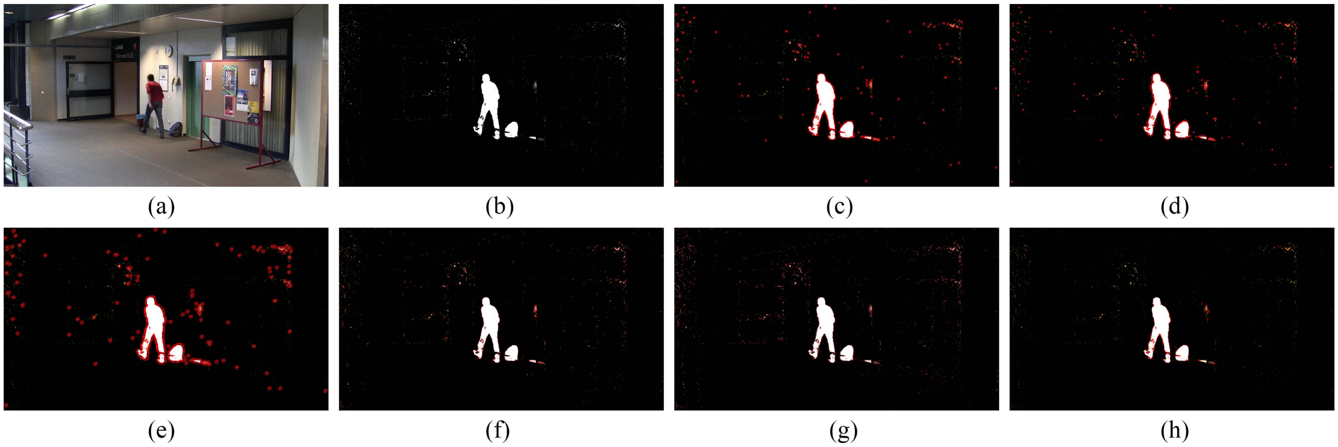


Fig. 11. Qualitative results of $s = 4$ for SubSENSE. (a) Original frame. (b) GT. (c) Bicubic. (d) ICBI. (e) SRCNN. (f) RDN. (g) RCAN. (h) Our method.

methods. Fig. 8 shows the upsampled foreground object segmentation results of the highway video obtained by Vibe with respect to $s = 2$. Due to the shaking of trees, misdetected foregrounds can be found in the top-left corner as shown in Fig. 8(b). Similar to the upsampling results of GMM shown in Figs. 6 and 7, Bicubic, ICBI, SRCNN, and RCAN significantly enlarge the misdetected foregrounds. In contrast, RDN can remove the misdetected foregrounds due to better representation of training data with respect to noise. Compared with competing methods, our method can also remove misdetected foregrounds because the boundary of the misdetected foregrounds cannot fill the region generated by boundary information of objects. Such results again indicate that the content-adaptive properties are important when performing upsampling.

Fig. 9 shows the upsampled foreground object segmentation results of the walking video obtained by Vibe with respect to $s = 4$. Although only the pedestrian is the true foreground, Vibe misdetects background pixels as foreground pixels as shown in Fig. 9(b). Similar to the results shown in Fig. 8, the results of the proposed method are significantly better than those of competing methods in both containing fewer additional foreground pixels and removing misdetected background pixels based on content-adaptive properties in both downsampling and upsampling layers. Figs. 10 and 11 show the upsampled foreground object segmentation results obtained by SuBSENSE with respect to $s = 2$ and $s = 4$ for the PETS2006 and dropping videos, respectively. Again, our results are significantly better than those of competing methods and can adhere to the true boundaries of foreground objects. The qualitative results show the content-adaptive properties and generalization ability of the proposed method in boosting different kinds of surveillance videos under different BM methods. Due to limited space, the experimental results, including the comparative baselines and the proposed method are available at https://www.youtube.com/playlist?list=PLeFabaAzO2xwAr_Ya9ui8hWEtFpAieTYR.

V. Conclusion

In this article, we proposed a novel CARF to boost the computation speed of BM methods in high-resolution videos. Different from state-of-the-art methods, our method is derived from superpixels which are computed adaptively for the content of each frame. Moreover, the downsampling layer preserves the adaptive boundary information of each frame and helps the upsampling layer to upsample low-resolution foreground segmentation results to high-resolution ones. The proposed downsampling and upsampling layers without GPU accelerations have been shown their computational efficiency and qualitative performance in the experimental results compared with recent deep learning-based upsampling methods. Moreover, the proposed method can be generally applied to different BM methods for high-resolution surveillance videos without modifications of BM methods.

Because the proposed CARF reduces the resolutions of video frames and achieves high-quality upsampling results, it can also be applied to various video processing applications for real-time computation. For example, we can replace the BM and subtraction methods shown in Fig. 1 to optical flow methods [37], [38]. By using CARF, boosting the computation speed and obtaining high-quality results of recent optical flow methods without additional hardware implementation [39] can be achieved. Our method can also be cooperated with video saliency detection methods [40], [41] to boost the computation. In the future, we will extend the proposed method to boost the aforementioned video processing applications in high-resolution videos without the need of hardware accelerations.

References

[1] C. Fan, Y. Wang, and C. Huang, "Heterogeneous information fusion and visualization for a large-scale intelligent video surveillance system," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 4, pp. 593–604, Apr. 2017.

[2] B. Yu, Y. Liu, and Q. Sun, "A content-adaptively sparse reconstruction method for abnormal events detection with low-rank property," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 4, pp. 704–716, Apr. 2017.

[3] W.-C. Wang, C.-Y. Chiou, C.-R. Huang, P.-C. Chung, and W.-Y. Huang, "Spatiotemporal coherence-based annotation placement for surveillance videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 787–801, Mar. 2018.

[4] C.-R. Huang, P.-C. J. Chung, D.-K. Yang, H.-C. Chen, and G.-J. Huang, "Maximum a posteriori probability estimation for online surveillance video synopsis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 8, pp. 1417–1429, Aug. 2014.

[5] J. Zhu, S. Liao, and S. Z. Li, "Multicamera joint video synopsis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 6, pp. 1058–1069, Jun. 2016.

[6] A. Kamel, B. Sheng, P. Yang, P. Li, R. Shen, and D. D. Feng, "Deep convolutional neural networks for human action recognition using depth maps and postures," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 9, pp. 1806–1819, Sep. 2019.

[7] J. Yu and J. Sun, "Multiactivity 3-D human pose tracking in incorporated motion model with transition bridges," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 8, pp. 1389–1402, Aug. 2018.

[8] T. Lindeberg, "Scale-space for discrete signals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 3, pp. 234–254, Mar. 1990.

[9] Y. Zhang, D. Zhao, J. Zhang, R. Xiong, and W. Gao, "Interpolation-dependent image downsampling," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3291–3296, Nov. 2011.

[10] J. Shi and S. E. Reichenbach, "Image interpolation by two-dimensional parametric cubic convolution," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1857–1870, Jul. 2006.

[11] A. Giachetti and N. Asuni, "Real-time artifact-free image upscaling," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2760–2768, Oct. 2011.

[12] J.-J. Huang, W.-C. Siu, and T.-R. Liu, "Fast image interpolation via random forests," *IEEE Trans Image Process.*, vol. 24, no. 10, pp. 3232–3245, Oct. 2015.

[13] W. Yang, T. Yuan, W. Wang, F. Zhou, and Q. Liao, "Single-image super-resolution by subdictionary coding and kernel regression," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 9, pp. 2478–2488, Sep. 2017.

[14] Y. Zhang *et al.*, "Collaborative representation cascade for single-image super-resolution," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 5, pp. 845–860, May 2019.

[15] C.-R. Huang, W.-C. Wang, W.-A. Wang, S.-Y. Lin, and Y.-Y. Lin, "USEAQ: Ultra-fast superpixel extraction via adaptive sampling from quantized regions," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4916–4931, Oct. 2018.

[16] S. Popa, D. Crookes, and P. Miller, "Hardware acceleration of background modeling in the compressed domain," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 10, pp. 1562–1574, Oct. 2013.

[17] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Fort Collins, CO, USA, 1999, pp. 246–252.

[18] V. Pham, P. Vo, V. T. Hung, and L. H. Bac, "GPU implementation of extended Gaussian mixture model for background subtraction," in *Proc. Int. Conf. Comput. Commun. Technol. Res. Innovat. Vis. Future*, Hanoi, Vietnam, 2010, pp. 1–4.

[19] X. Ye and W. Wan, "Fast background modeling using GMM on GPU," in *Proc. Int. Conf. Audio Lang. Image Process.*, Shanghai, China, 2014, pp. 937–941.

[20] R. Boghdady, C. Salama, and A. Wahba, "GPU-accelerated real-time video background subtraction," in *Proc. 10th Int. Conf. Comput. Eng. Syst.*, Cairo, Egypt, 2015, pp. 34–39.

[21] P. Kumar, A. Singhal, S. Mehta, and A. Mittal, "Real-time moving object detection algorithm on high-resolution videos using GPUs," *J. Real Time Image Process.*, vol. 11, no. 1, pp. 93–109, Jan. 2016.

[22] M. Poremba, Y. Xie, and M. Wolf, "Accelerating adaptive background subtraction with GPU and CBEA architecture," in *Proc. IEEE Workshop Signal Process. Syst.*, San Francisco, CA, USA, 2010, pp. 305–310.

[23] B. Wilson and A. Tavakkoli, "An efficient non-parametric background modeling technique with CUDA heterogeneous parallel architecture," in *Proc. Int. Symp. Vis. Comput.*, 2015, pp. 210–220.

[24] L. Qin, B. Sheng, W. Lin, W. Wu, and R. Shen, "GPU-accelerated video background subtraction using Gabor detector," *J. Vis. Commun. Image Represent.*, vol. 32, pp. 1–9, Oct. 2015.

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

[26] W. Song, Y. Tian, S. Fong, K. Cho, W. Wang, and W. Zhang, "GPU-accelerated foreground segmentation and labeling for real-time video surveillance," *Sustainability*, vol. 8, no. 10, pp. 1–20, Oct. 2016.

[27] D. Pawar, "GPU based background subtraction using CUDA: State of the art," in *Proc. Int. Conf. Wireless Commun. Signal Process. Netw.*, Chennai, India, 2017, pp. 1201–1204.

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

[29] M.-H. Yang, C.-R. Huang, W.-C. Liu, S.-Z. Lin, and K.-T. Chuang, "Binary descriptor based nonparametric background modeling for foreground extraction by using detection theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 4, pp. 595–608, Apr. 2015.

[30] Y. Lin, Y. Tong, Y. Cao, Y. Zhou, and S. Wang, "Visual-attention-based background modeling for detecting infrequently moving objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 6, pp. 1208–1221, Jun. 2017.

[31] Y. Wang, P. M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Columbus, OH, USA, 2014, pp. 393–400.

[32] P. Korshunov and T. Ebrahimi, "PEViD: Privacy evaluation video dataset," in *Proc. SPIE Appl. Digit. Image Process. XXXVI*, vol. 8856, 2013, Art. no. 88561S.

[33] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[34] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 2472–2481.

[35] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 294–310.

[36] W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[37] S. Cai, Y. Huang, B. Ye, and C. Xu, "Dynamic illumination optical flow computing for sensing multiple mobile robots from a drone," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 8, pp. 1370–1382, Aug. 2018.

[38] R. Ke, Z. Li, J. Tang, Z. Pan, and Y. Wang, "Real-time traffic flow parameter estimation from UAV video based on ensemble classifier and optical flow," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 54–64, Jan. 2019.

[39] K. Seyid, A. Richaud, R. Capoccia, and Y. Leblebici, "FPGA-based hardware implementation of real-time optical flow calculation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 1, pp. 206–216, Jan. 2018.

[40] C.-R. Huang, Y.-J. Chang, Z.-X. Yang, and Y.-Y. Lin, "Video saliency map detection by dominant camera motion removal," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 8, pp. 1336–1349, Aug. 2014.

[41] K. Zhang and Z. Chen, "Video saliency prediction based on spatial-temporal two-stream network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 12, pp. 3544–3557, Dec. 2019.

Chun-Rong Huang (Senior Member, IEEE) received the bachelor's and Doctor of Philosophy degrees from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, in 1999 and 2005, respectively.

In 2005, he joined the Institute of Information Science, Academia Sinica, Taipei, Taiwan, as a Postdoctoral Fellow. He joined the Department of Computer Science and Engineering, National Chung Hsing University, Taichung, Taiwan, in 2010, where he became a Full Professor in 2019. His research interests include computer vision, computer graphics, multimedia signal processing, image processing, and medical image processing.

Prof. Huang is a member of the IEEE Circuits and Systems Society, the IEEE Signal Processing Society, the IEEE Computational Intelligence Society, and the Phi Tau Phi Honor Society.

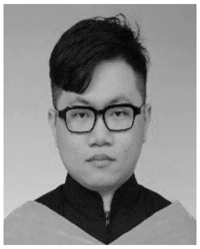
Wei-Yun Huang received the bachelor's and master's degrees from the Department of Computer Science and Engineering, National Chung Hsing University, Taichung, Taiwan, in 2014 and 2016, respectively.

Yi-Sheng Liao received the bachelor's and master's degrees from the Department of Computer Science and Engineering, National Chung Hsing University, Taichung, Taiwan, in 2018 and 2020, respectively.

Chien-Cheng Lee (Member, IEEE) received the Ph.D. degree from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, in 2003.

He is currently an Assistant Professor with the Department of Electrical Engineering, Yuan Ze University, Taoyuan, Taiwan. He was a Visiting Researcher with Telcordia Inc. (formerly, Bellcore), Piscataway, NJ, USA, from October 2007 to January 2008. His research interests include image processing, pattern recognition, and machine learning.

Dr. Lee is one of the guest editors for a special issue on Signal Processing for Applications in Healthcare Systems for *EURASIP Journal on Advances in Signal Processing* in 2008.



Yu-Wei Yeh received the bachelor's and master's degrees from the Department of Communications Engineering, Yuan Ze University, Taoyuan, Taiwan, in 2015 and 2017, respectively.