Active Perception for Foreground Segmentation: An RGB-D Data-based Background Modelling Method

Yuxiang Sun, Ming Liu, *Senior Member, IEEE* and Max Q.-H. Meng, *Fellow, IEEE*

***Abstract*—Foreground moving-object segmentation is a fundamental problem in many computer vision applications. As a solution for foreground segmentation, background modelling has been intensively studied over past years and many effective algorithms have been developed. However, accurate foreground segmentation is still a difficult problem. Currently, most of the algorithms work solely within the color space, in which the segmentation performance is prone to be degraded by a multitude of challenges, such as illumination changes, shadows, automatic camera adjustments and color camouflage. RGB-D cameras are active visual sensors that provide depth measurements along with color images. We present in this paper an innovative background modelling method by using both the color and depth information from an RGB-D camera. The proposed method is evaluated using a public RGB-D dataset. Various experiments confirm that our method is able to achieve superior performance compared with existing well-known methods.**

***Note to Practitioners*—This paper investigates background modelling for foreground segmentation with active perception. Recent RGB-D cameras that leverage the active perception technology have advanced many computer vision algorithms. In this paper, we develop a background modelling method to achieve superior performance by using an RGB-D camera instead of a color camera. Due to the use of the active sensing technology, the proposed method is characterized by its robustness to common challenges. Our method could be used for improving existing infrastructures, such as visual surveillance systems for parking spaces. Moreover, the simple design of our method allows it to be easily deployed on various computing platforms, which facilitates many practical applications that usually require embedded computing devices. However, our method cannot run real-timely at the current status. We believe that it can be further improved using parallel programming techniques to meet the real-time requirement.**

***Index Terms*—Active Perception, Foreground Segmentation, Background Modelling, RGB-D Camera.**

Manuscript received August 8, 2018; revised November 19, 2018; accepted January 7, 2019. Date of publication January 15, 2019; date of current version January 15, 2019. This paper was recommended for publication by Associate Editor H. Liu upon evaluation of the reviewers' comments. This work was supported in part by the Shenzhen Science and Technology Innovation (SZSTI) project JCYJ20160428154842603 and JCYJ20160401100022706, the Hong Kong Research Grant Council (RGC) project 11210017, 16212815 and 21202816, the National Natural Science Foundation of China project U1713211 awarded to Ming Liu, and in part by the Hong Kong RGC GRF project 14205914 and 14200618, ITC ITF project ITS/236/15, and SZSTI project JCYJ20170413161616163 awarded to Max Q.-H. Meng. *(Corresponding authors: Ming Liu and Max Q.-H. Meng.)*

Yuxiang Sun and Ming Liu are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, China (email: eeyxsun@ust.hk, sun.yuxiang@outlook.com; eelium@ust.hk).

Max Q.-H. Meng is with the Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, China (email: max.meng@cuhk.edu.hk).

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/TASE.2019.2893414

I. Introduction

SEGMENTING foreground moving objects from videos on-line is a fundamental step in many computer vision applications, such as motion tracking [1], obstacle avoidance [2], visual localization [3]. As a solution for foreground segmentation, background modelling has been intensively studied over past decades [4]. It is also known as background subtraction in some literature. As the name suggests, the general idea of background modelling is to pixel-wisely subtract the current frame with the built background model. The subtraction here refers to a kind of pixel-wise comparison. Foreground pixels can be indicated by the comparison results.

Although many effective background modelling algorithms have been developed [4], accurate foreground segmentation is still a hard problem. The reason is typically that most of the existing background modelling algorithms work within the color space. The foreground segmentation performance is prone to suffer from the common color-imagery challenges such as illumination changes, automatic camera adjustments, shadows and color camouflage [5]. The illumination changes and automatic camera adjustments change the pixel values in color images. Particularly, the automatic camera adjustments comprise the white balance adjustments and exposure adjustments. The shadows of moving objects cause artefacts in color images. The color camouflage refers to that the color features of moving objects are similar to those of the background model. These challenges confuse the background modelling and hence cause incorrect foreground segmentation.

Active perception generally refers to retrieving information not as a passive receiver. It can be realized by integrating sensors with physical actuators or intelligent control algorithms, so the problem of active perception becomes the problem of controlling strategies for data acquisition [6]. For instance, Liu *et al.* [7] developed a reinforcement learning-based modality selection algorithm to actively fuse multi-modal sensor information for material perception. Active perception also comprises active sensing, with which sensors are able to retrieve information by actively transmitting signals to the environment and receiving the reflected signals [8]. Active sensors, such as sonar, Lidar and RGB-D cameras, have been widely used in many robotics and automation applications. For example, the Simultaneous Localization and Mapping (SLAM) technology aims to concurrently track a robot itself and reconstruct the traversed environment in 2-D or 3-D models [9]. With Lidar sensors, a robot is able to measure the distances from itself to the surrounding landmarks, and estimate the poses with the measurements.

It is worth noting that the recently developed RGB-D cameras, such as Intel Realsense [10], Microsoft Kinect [11] and Asus Xtion [12], have completely changed the computer vision world [13]. They are active visual sensors [14] which not only get color-imagery information but also measure distances with the actively projected infrared light [15]. Some of them can work in both indoor and outdoor environments [10]. The RGB-D cameras stream registered RGB and depth images at the same time, which provides more information than ordinary color cameras [16]. Thus, they are able to bring opportunities to gain benefits for many computer vision algorithms [17]. As the depth information provided by RGB-D cameras is robust to the common color-imagery challenges, we take this as the major advantage and propose a novel background modelling method through the combination of color and depth information. The major contributions of this paper are summarized as follows:

1) We propose a single-frame initialization scheme just using the first RGB-D frame for background modelling.
2) We develop a depth-based classifier for rough classification and a color-based classifier for refinement.
3) We develop a trimap generation scheme based on morphological operations to bridge the two classifiers.

The remainder of this paper is structured as follows. In section II, related work has been reviewed. In section III, we describe our method in detail. In section IV, experimental results and discussions are presented. Conclusions and future work are drawn in the last section.

II. Related Work

Background modelling algorithms can be generally divided into parametric methods and non-parametric methods according to the mathematical form of the model. They can also be categorized according to the used sensors, such as monocular camera, stereo camera and RGB-D camera. We review some selected algorithms in this section.

A. Parametric Algorithms

The parametric algorithms build a statistical model for each pixel. Foreground can be determined by testing whether a newly observed pixel fits in the statistical model. One of the first parametric algorithms is the Single Gaussian (SG) model proposed by Wren *et al.* [18]. It models the history of pixel values in a single Gaussian distribution. However, the unimodal Gaussian model cannot tackle dynamic background, where the values of each pixel cannot be aggregated into one group. To address this problem, Mixture of Gaussians (MOG) [19] was employed to model the pixel values in Gaussian mixture models. The history of intensity values of each pixel is modelled in a weighted average of a number of Gaussian distributions. Numerous enhancement algorithms for MOG have been proposed to address various challenges [20], however, the MOG algorithm requires tedious parameter tunings. For instance, the number of Gaussians is advised to set according to the variations of pixel values. In addition, large number of Gaussians requires extensive computations, which makes the algorithm impractical for real-time applications.

B. Non-parametric Algorithms

Instead of modelling the history of pixel values in parametric distributions, non-parametric algorithms directly maintain a set of pixel samples as the background model. They are more flexible than parametric algorithms, because the distribution of pixel values does not necessarily follow a known parametric form. Kernel Density Estimation (KDE) was firstly introduced by Elgammal *et al.* [21] as a non-parametric approach in the context of background modelling. The algorithm calculates pixel-wise similarities using kernel functions between the newly observed frame with the background model. Barnich *et al.* [22] proposed a non-parametric algorithm called Visual Background Extractor (ViBe). The main contribution of ViBe is the first use of the random policy in background modelling. The authors applied the random policy in ViBe in three aspects. Firstly, the background model is initialized using randomly selected neighbouring pixels, which provides a fast initialization process just using one frame. Secondly, pixels in the samples are randomly replaced during model update, which ensures an exponential decaying lifespan. Lastly, newly incorporated background pixels are diffused to replace randomly selected neighbouring pixels, which ensures the spatial consistency of the model. Hofmann *et al.* [23] proposed the Pixel-Based Adaptive Segmenter (PBAS) for background modelling. The PBAS algorithm adopts a similar random policy as that of the ViBe algorithm. Different from ViBe, the decision threshold for foreground determination and the learning rate for background update are pixel based. They are dynamically changed according to the proposed tuning mechanisms.

C. Stereo Camera-based Algorithms

Before the advent of RGB-D cameras, stereo cameras were the most popular depth visual sensors. Dense depth maps can be computed using depth estimation algorithms with disparity information provided by stereo cameras. Gordon *et al.* [24] firstly adapted the traditional MOG algorithm using stereo cameras. Each pixel was modelled using a Gaussian distribution with the 4-channel RGB-D data. Recently, Fernandez-Sanchez *et al.* [25] applied the codebook background modelling algorithm with stereo cameras. They proposed an early fusion method called Depth Extended Codebook (DECB) by replacing the color channels with the depth channel. In addition, they proposed a late fusion method combining the results from the traditional codebook algorithm [26] and the DECB algorithm. The major disadvantage for algorithms using stereo cameras is the high dependence on the quality of the disparity maps and the performance of the depth estimation algorithms.

D. RGB-D Camera-based Algorithms

Compared with stereo cameras, RGB-D cameras were developed to provide high quality depth data directly from the hardware. However, as RGB-D cameras become popular just in recent years, research work on background modelling using RGB-D cameras are very limited. Murgia *et al.* [27]

directly extended the traditional codebook algorithm using the 4-channel RGB-D data. Amamra *et al.* [28] integrated the RGB-D data in the classical MOG algorithm and implemented a GPU-accelerated version. These methods simply apply the RGB-D data using traditional background modelling algorithms without considering the noises of the depth data, such as the no-measured depth (*nmd*) points, which usually appear at object boundaries and flickered areas [29]. Camplani *et al.* [30] proposed two pixel-wise classifiers based on color and depth data respectively for background modelling. The two classifiers were combined together to obtain foreground segmentation results. They designed a combination scheme by specially considering the *nmd* points. The combination scheme gave higher weights to the color-based classifier at areas where *nmd* points aggregated. The areas were assumed to be object boundaries which were found using an edge detection algorithm.

III. The Proposed Method

A. Method Overview

The overview of our method is shown in Fig. 1. We initialize the background model merely using the first RGB-D point-cloud frame. Different from most background modelling algorithms, the model we build is a 4-channel RGB-D point-cloud model. In order to ensure a clean background model at the beginning, we require that the first frame does not contain any moving objects.

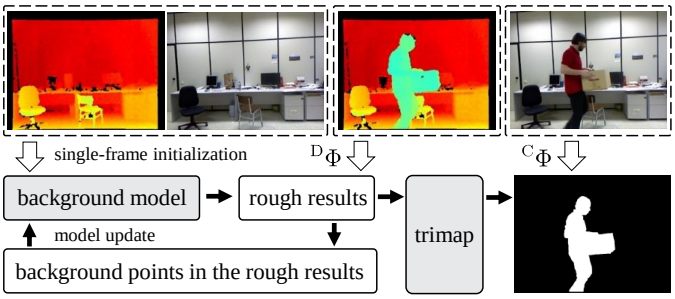


Fig. 1: The schematic overview of the proposed method. The RGB-D image pairs in the left and right dashed boxes are the first frame and a test frame, respectively. The black pixels in the depth images indicate the *nmd* points. The right bottom image shows the final refined segmentation result. The figure is best viewed in color.

As shown in Fig. 1, we firstly develop the depth-based classifier $^{D}\Phi$ to roughly classify the pixels into foreground, background and unknowns. Then, we devise a scheme to generate a trimap based on morphological operations using the rough classification results. Lastly, we develop the color-based classifier $^{C}\Phi$ based on the graph-cut [31] optimization framework to refine the segmentation using the cues provided by the trimap. The two classifiers are sequentially combined and the trimap serves as a bridge between them.

B. Model Initialization

The background model is initialized using the first RGB-D point-cloud frame. We devise a single-frame initialization scheme which is built on the observation that neighbouring RGB-D points share close color and depth values. With this scheme, the proposed method is able to segment the foreground just starting from the second frame. Note that the model we built is a type of non-parametric model. It comprises a number of samples spatially populated from the first frame.

Algorithm 1: Background Model Initialization

Data: $\mathcal{M}$, $^{0}\mathbf{C}$, λ, $\mathbf{w}$, $\mathbf{h}$.

```
for m ← 1 to w do
    for n ← 1 to h do
        for i ← 1 to λ do
            p = R(m)
            q = R(n)
            M_i(m, n) = 0C(p, q)
        end
    end
end
```

It is known that many background modelling algorithms build the initial model using samples from a temporal distribution in a bootstrap sequence. This is because pixel values from the temporal distribution are natural descriptions for pixel variations. Similar to the temporal distribution, we believe that a spatial distribution from neighbouring pixels can also well describe pixel variations. In this paper, we adopt the 8-neighbourhood structure for model population. The number of samples is determined by finding the optimal performance through quantitative evaluations.

Algorithm 1 describes the model initialization scheme. Let $\mathcal{M}$ denote the background model, and λ denote the number of samples in the model $\mathcal{M}$. The points in the RGB-D point-cloud frames can be indexed like images. We populate $\mathcal{M}$ using the first point-cloud frame $^{0}\mathbf{C}$. Let m and n denote the coordinates for a point in a point-cloud frame, and let p and q denote the coordinates of the randomly selected neighbouring point. As aforementioned, we use the 8-neighbourhood structure; thus, the value ranges for p and q are:

$$\begin{aligned} p &\in \{m, m+1, m-1\}, \\ q &\in \{n, n+1, n-1\}. \end{aligned} \tag{1}$$

We use $\mathbf{w}$ and $\mathbf{h}$ to represent the width and height of the point-cloud frames. The function $\mathcal{R}(\cdot)$ obtains a random value from the value ranges of (1). $\mathcal{M}_i(m,n)$ represents the point indexed at (m,n) from the sample i. $^{0}\mathbf{C}(p,q)$ represents the point indexed at (p,q) from the current point-cloud frame $^{0}\mathbf{C}$. As we can see, the initialization process iteratively runs until all the points in $\mathcal{M}$ have been populated with a randomly selected neighbourhood.

C. Foreground Segmentation

1) The depth-based classifier: We develop the depth-based classifier $^{D}\Phi$ to roughly classify the pixels in a testing frame. Firstly, we check whether the depth value of a given point in the testing frame is an *nmd* point. The depth values in the samples at this location are also examined. If the samples at this location contain *nmd* point or the given point is an

Algorithm 2: The depth-based classifier

```
   Data: M, C, λ, w, h, θ, κ.
1  for m ← 1 to w do
2      for n ← 1 to h do
3          ω_{m,n} = 0
4          if φ[D{C(m,n)}] then
5              ω_{m,n} = −1
6          end
7          for i ← 1 to λ do
8              if φ[D{M_i(m,n)}] then
9                  ω_{m,n} = −1
10                 break
11             end
12         end
13         if ω_{m,n} = 0 then
14             compute ω_{m,n} using (2)
15         end
16         if ω_{m,n} > κ then
17             label C(m,n) as background
18         else if 0 ≤ ω_{m,n} ≤ κ then
19             label C(m,n) as foreground
20         else
21             label C(m,n) as unknown
22         end
23     end
24 end
```

nmd point, we label the given point as an unknown point. If the point is not labelled as an unknown point, we then compare the depth value of the given point with those samples at the location. A variable ω is introduced here to measure the similarity between the given point and the corresponding samples:

$$\omega_{m,n} = \sum_{i=1}^{\lambda} \mathcal{H}\Big\{\theta - |\mathcal{D}\{\mathbf{C}(m,n)\} - \mathcal{D}\{\mathcal{M}_i(m,n)\}|\Big\}, \quad (2)$$

where m and n are point indices, $\mathbf{C}$ represents the testing point-cloud frame, $\mathcal{D}(\cdot)$ is a function that retrieves the depth value of a point, $|\cdot|$ represents the absolute value of the difference, θ is a pre-defined threshold, $\mathcal{H}(\cdot)$ is a step function with values 1 and 0 for $x \geq 0$ and $x < 0$ respectively, and x here represents an argument for $\mathcal{H}(\cdot)$. Larger non-negative values of $\omega_{m,n}$ indicate that the point $\mathbf{C}(m,n)$ is more close to the samples. Finally, we classify the point according to the value of $\omega_{m,n}$.

The pseudo-code for ${}^{\mathrm{D}}\Phi$ is presented in Algorithm 2. The statements 4-12 encode the first step. The function $\varphi(\cdot)$ returns true if the argument is an *nmd* point. The statements 4-6 check whether the point $\mathbf{C}(m,n)$ is an *nmd* point. The statements 7-12 check whether there is an *nmd* point in the samples. The statements 13-15 encode the second step. The algorithm compares the point $\mathbf{C}(m,n)$ with the samples at the same location if $\mathbf{C}(m,n)$ is not an *nmd* point and the samples do not contain an *nmd* point. The statements 16-22 encode the last step. The point $\mathbf{C}(m,n)$ is classified according to the value of $\omega_{m,n}$. We introduce a non-negative determination threshold κ here for the closeness. We classify $\mathbf{C}(m,n)$ as background if it is close to the samples. Specially if $\omega_{m,n} = 0$, we consider $\mathbf{C}(m,n)$ as foreground. In this case, there is no point in the samples that is considered close to $\mathbf{C}(m,n)$.

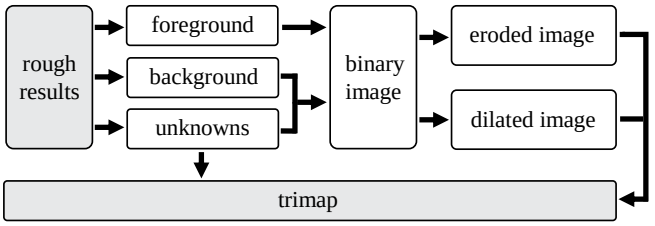


Fig. 2: The pipeline of the trimap generation process. The *rough results* in the grey box represent the rough classification results provided by ${}^{\mathrm{D}}\Phi$. We firstly generate a binary image from the results by painting the foreground as white and the background and the unknowns as black. Then, we apply the erosion and dilation algorithms on the binary image with the unknowns to generate the trimap.

2) The generation of the trimap: The trimap serves as the bridge between the two classifiers. It contains three types of pixels: white pixels for foreground, black pixels for background and grey pixels for unknowns. We generate the trimap using the rough classification results from ${}^{\mathrm{D}}\Phi$. In our method, the color-based classifier ${}^{\mathrm{C}}\Phi$ is built on the Conditional Random Field (CRF) [32] framework. The trimap extracts the foreground and background with high confidence from the rough classification results of ${}^{\mathrm{D}}\Phi$. It provides hard constraints, namely, the segmentation hints for ${}^{\mathrm{C}}\Phi$ to produce the final precise segmentation results.

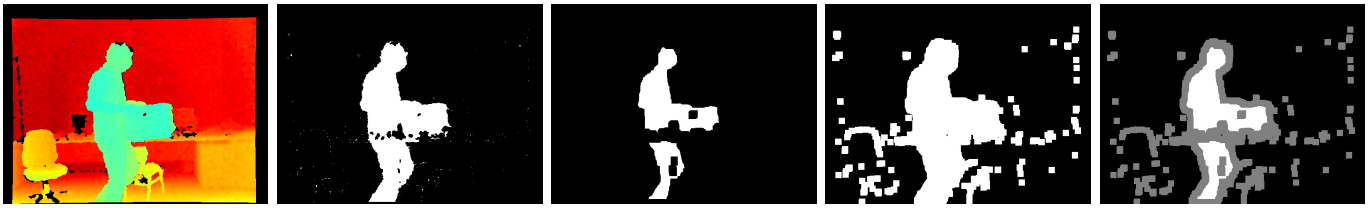
Fig. 2 displays the pipeline of the trimap generation process. We firstly generate a binary image $\mathbf{B}$ using the rough classification results of ${}^{\mathrm{D}}\Phi$. In $\mathbf{B}$, we paint the foreground points as white pixels, and paint the background points and the unknown points as black pixels. The binary image $\mathbf{B}$ is then processed by erosion and dilation morphological algorithms [33] separately. The morphological results and the unknown points are combined to generate the trimap.

Let $\mathcal{F}(\cdot)$ and $\mathcal{B}(\cdot)$ denote the functions that retrieve the foreground and background pixels from an image. Let $\mathcal{I}(\cdot)$ denote a set of pixels. Let $\mathbf{E}$, $\mathbf{D}$, $\mathbf{G}$ and $\mathbf{T}$ denote the eroded image, the dilated image, the gap caused by the morphological operations and the trimap, respectively. We generate the trimap using the formula:

$$\mathcal{I}(\mathbf{T}) = \mathcal{F}(\mathbf{E}) \cup \mathcal{B}(\mathbf{D}) \cup \mathcal{I}(\mathbf{G}). \quad (3)$$

In (3), $\mathcal{F}(\mathbf{E})$, $\mathcal{B}(\mathbf{D})$ and $\mathcal{I}(\mathbf{G})$ provide the foreground pixels, the background pixels and the unknown pixels for the trimap. They are coloured white, black and grey respectively.

The generation scheme used in (3) is mainly based on the observation that the foreground in the eroded image is more likely to be the true foreground and the background in the dilated image is more likely to be the true background. We find that the binary image contains false classifications that are displayed as salt-and-pepper noise. The areas of salt and pepper noise show false positives and false negatives of the



(a) The depth image. (b) The binary image $\mathbf{B}$. (c) The eroded image $\mathbf{E}$. (d) The dilated image $\mathbf{D}$. (e) The trimap $\mathbf{T}$.

Fig. 3: Qualitative demonstration for the trimap generation. The size of the structure elements of the morphological operations is 15. The figure is best viewed in color.

foreground segmentation. The erosion and dilation operations can remove the salt noise and the pepper noise respectively [34]. Thus, the foreground in the eroded image and the background in the dilated image are more trustable compared with those in the binary image $\mathbf{B}$. By tuning the parameter θ in (2), we find that the amount of salt-and-pepper noises tends to increase with increased distances. We think this is caused by the increased depth measurement errors with respective to the increased distances. Larger depth measurement errors lead to more salt-and-pepper noises. Thus, the proposed generation scheme reduces the false labels in the trimap.

We use the white pixels in the eroded image and the black pixels in the dilated image to form the foreground and the background in the trimap, respectively. We label the gap $\mathbf{G}$ between the foreground and the background caused by the morphological operations as unknowns. Note that the pixels in $\mathcal{F}(\mathbf{E})$, $\mathcal{B}(\mathbf{D})$ and $\mathcal{I}(\mathbf{G})$ are mutually exclusive. They form the complete trimap. The reason why we label the pixels in $\mathbf{G}$ as unknowns is because we observed that the gaps tend to appear in areas where the depth measurements are unstable, such as object boundaries. There are *nmd* points at these areas. The points with unstable depth measurements can lead to incorrect classifications of ${}^{\mathrm{D}}\Phi$; thus, conservatively labelling the pixels in $\mathbf{G}$ as unknowns reduces the number of incorrect labels in the trimap. As aforementioned, we use the unknowns from the rough classification results of ${}^{\mathrm{D}}\Phi$ for the trimap generation. In our method, we simply paint these unknown points as grey in the trimap.

It should be noted that we separately apply the erosion and dilation algorithms on the binary image in our method. The motivation of the morphological operations is to obtain the reliable foreground and background labels for the trimap. This is different from most background modelling algorithms that sequentially use erosion and dilation to denoise the foreground segmentation results [35].

Fig. 3 qualitatively demonstrates the trimap generation process. As we can see, the areas of salt noise and pepper noises are removed in the eroded image and the dilated image respectively. In the binary image $\mathbf{B}$, the large holes on the foreground are caused by *nmd* points in the model. The white pixels in $\mathbf{E}$ and the black pixels in $\mathbf{D}$ formulate the foreground and the background, respectively, in the trimap. The grey pixels in the trimap represent the unknown pixels.

3) The color-based classifier: The color-based classifier ${}^{\mathrm{C}}\Phi$ is developed based on the graph-cut optimization framework, which labels each pixel as foreground or background in a given frame. The trimap provides hard constraints for the graph. There are two types of nodes in the graph. One type of node denotes the labels for each pixel. The other type of node represents the foreground and background terminals, which consist of the foreground and background pixels indicated by the hard constraint. The nodes are linked by edges in the graph. There are two types of edges. One type of edge links the labels with the two terminals. The other type of edge links the neighbourhoods of each label. We use the 8-neighbourhood structure in this paper.

Let $\mathbf{L} = \{\ell_1, \ell_2, \cdots, \ell_{|\mathbf{L}|}\}$ denote the set of labels whose entries are the labels for each pixel. $|\mathbf{L}|$ is the cardinality of the set, which can be found by multiplying the width and the height of the given frame. Let $\phi(\cdot)$ denote the energy function for each edge. The energy measures the similarity between two nodes. $\mathcal{E}(\mathbf{L})$ denotes the sum of all the energies. The problem of the image segmentation is to find an optimal label vector $\mathbf{L}^*$ which can minimize the following energy:

$$\mathcal{E}(\mathbf{L}) = \sum_{i \in \mathcal{P}} \phi(\ell_i) + \sum_{i \in \mathcal{P}, j \in \mathcal{N}} \phi(\ell_i, \ell_j), \tag{4}$$

where $\ell_i \in \{0, 1\}$, $\mathcal{P}$ represents all the pixels in the frame, and $\mathcal{N}$ represents the neighbourhoods of the pixel i.

The first term in (4) is called a regional term or data term, which measures the similarity between labels and the terminals. The regional term for an unknown pixel can be found in the formula:

$$\phi(\ell_i) = -\log^{\ell_i}\left({}^{\mathcal{F}}p_i\right)\log^{1-\ell_i}\left({}^{\mathcal{B}}p_i\right), \tag{5}$$

where $\ell_i = 0$ and $\ell_i = 1$ represent that the pixel belongs to the background and the foreground, respectively, and ${}^{\mathcal{B}}p_i$ and ${}^{\mathcal{F}}p_i$ represent the probabilities of the pixel being the background and the foreground, respectively. The probabilities are found by fitting the pixel in the distributions formulated by the background pixels and the foreground pixels indicated by the hard constraints. The regional term energies for the pixels indicated by the hard constraints are constant values:

$$\phi(\ell_i) = \alpha^{\ell_i}\beta^{1-\ell_i}, \tag{6}$$

where α and β are two constants. If pixel i is indicated as a background pixel in the trimap, we set $\alpha = 0$ and β to a non-negative value. If pixel i is indicated as a foreground pixel in the trimap, we set $\beta = 0$ and α to a non-negative value.

Algorithm 3: Background model maintenance

```
Data: M, C, R, λ, w, h.
for m ← 1 to w do
    for n ← 1 to h do
        if ξ[R(m, n)] then
            s = G(λ)
            M_s(m, n) = C(m, n)
        end
    end
end
```

The second term in (4) is called the pairwise term or smoothness term. It measures the similarities between neighbouring pixels. We use a Gaussian kernel to find the value:

$$\phi(\ell_i, \ell_j) = \eta\delta_{i,j} \exp\big[-\frac{\|I_i - I_j\|}{2\sigma^2}\big], \tag{7}$$

where η is a normalization constant, and σ is the standard deviation for the Gaussian kernel. $\delta_{i,j}$ is a function with values of 0 and 1 for the cases of $\ell_i = \ell_j$ and $\ell_i \neq \ell_j$, respectively.

4) Background Model Maintenance: Background model maintenance is to update the background model to adapt the changes in the scene. In our method, the background model is updated immediately after we get the classification results of ${}^{\mathrm{D}}\Phi$. Note that we only use the background points classified by ${}^{\mathrm{D}}\Phi$ to update the background model. The foreground points and the *nmd* points are never used for background maintenance.

Our background model maintenance scheme is presented in Algorithm 3. $\mathbf{R}$ denotes the classification results of ${}^{\mathrm{D}}\Phi$. The function $\xi(\cdot)$ returns true if the given point is classified as background. The function $\mathcal{G}(\cdot)$ gives a random integer number selected from 1 to λ. As we can see, we randomly replace a sample point in the model using the new point from the given frame $\mathbf{C}$. The maintenance scheme avoids *nmd* points in the current frame to be updated in the background model, which reduces the number of unknowns in the trimap. The reason for using this scheme is because incorporating an *nmd* point into the model causes the point at this position to be classified as unknown in the future iterations. The trimap with more unknowns provides less useful hard constraints for ${}^{\mathrm{C}}\Phi$. Note that we do not use the final segmentation results provided by ${}^{\mathrm{C}}\Phi$ to update the background model. The reason for this is that the background points in the final segmentation results may contain points with large errors, such as the boundary points. Incorporating these points into the model will make the future classifications of ${}^{\mathrm{D}}\Phi$ less accurate.

IV. Experimental Results and Discussions

This section presents the experimental results and discussions. We use a public dataset for the evaluation. Firstly, we determine the optimal values of parameters in our method. Then, we compare our method with state-of-the-art algorithms.

A. The Dataset

The public dataset used in this paper was proposed by Camplani *et al.* in [30]. There are four RGB-D sequences with hand-labelled ground truth in this dataset. The sequences are captured using a Microsoft Kinect with the VGA (640×480) resolution at the 30 Hz frame rate. The ground truth is provided every a few frames during the period when moving objects are present in each sequence. There is no moving object at the beginning of each sequence. The four sequences are named as `GenSeq`, `ShSeq`, `ColCamSeq` and `DCamSeq` respectively.

The `GenSeq` sequence is used to test the overall performance of the algorithms in a general case. It was recorded in an office room where common challenges occur. For instance, there are moving-object shadows, color and depth camouflages, noisy depth measurements, illumination changes and automatic camera adjustments. The automatic camera adjustments in this sequence include both the brightness adjustment and the white balance adjustment. The sequence recorded a scenario that a person carries a box into the room and puts it on a table, then goes away. There are 300 frames of RGB-D images and 39 frames of ground truth in this sequence.

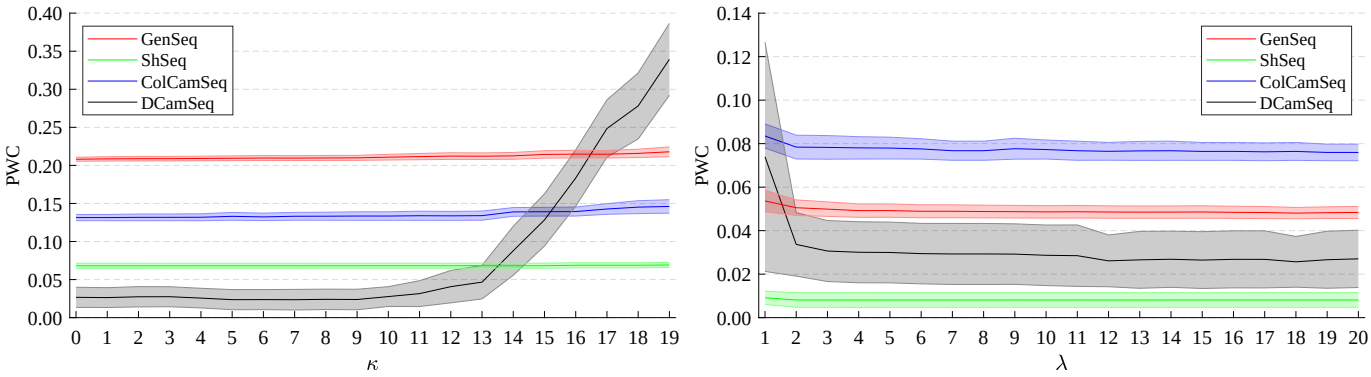
The `ShSeq` sequence is designed to evaluate the influence of shadows on the foreground segmentation performance. In addition to the shadow challenge, depth camouflage also occurs in this sequence. The sequence recorded a scenario that a person repeatedly rotates a carton box on the ground. There are the shadows of the rotating carton box. The depth values at the bottom of the carton box are close to those of the ground. There are 250 frames of RGB-D images and 25 frames of ground truth in this sequence.

The `ColCamSeq` sequence is designed to evaluate the foreground segmentation performance when color camouflage occurs. The sequence recorded a scenario that a person comes into a room with a white carton box in hand, and then repeatedly waves the box before a white panel. The box and the panel share similar colors. There are 360 frames of RGB-D images and 45 frames of ground truth in this sequence. Note that the ground truth is provided within a Region-of-Interest (ROI) in this sequence.

The `DCamSeq` sequence is designed to evaluate the foreground segmentation performance when depth camouflage occurs. Similar to the color camouflage, the depth camouflage refers to that the depth values of the background model are close to those of the foreground. In this sequence, there are close depth values but distinct colors between the foreground and background. The sequence recorded a scenario that a person walks into a room and then waves his arm and hand over a cabinet. The hand, the arm and the cabinet share close depth values. There are 670 frames of RGB-D images and 102 frames of ground truth in this sequence. The ground truth is also provided within a ROI in this sequence.

B. Evaluation Metrics

We employ the widely used metrics False Positive Rate (FPR), False Negative Rate (FNR), Portion of Wrong Classifications (PWC) for the quantitative evaluations [36]. We count the values of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) for a given frame. They are defined as follows:



(a) The κ tuning results.

(b) The λ tuning results.

Fig. 4: The parameter tuning results for the discrimination threshold κ and the number of samples λ. The curve and the shaded error bars encode the mean values and the standard deviations for the sequences. In order to view the curve and the errorbars in the plots clearly, we offset the curves by adding or subtracting some constants to the mean values. For the κ tuning, we set the number of samples λ to 20. For the λ tuning, we set the discrimination threshold κ to 0. Smaller PWC values correspond to better overall performance. From the κ tuning results, we can see that the overall performance of our method becomes worse when we increase the value of κ. From the λ tuning results, we can see that the overall performance of our method becomes better when we increase the value of λ. The figure is best viewed in color.

- TP : The number of foreground points that are correctly classified as foreground;
- TN : The number of background points that are correctly classified as background;
- FP : The number of background points that are wrongly classified as foreground;
- FN : The number of foreground points that are wrongly classified as background.

The metrics FPR and FNR are calculated using the following formulas:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}, \quad \mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP}+\mathrm{FN}}. \tag{8}$$

The metric FPR measures the portion of background points that are wrongly classified as foreground. The metric FNR measures the portion of foreground points that are wrongly classified as background. These two metrics evaluate the performance of algorithms from only one aspect. The metric PWC is able to evaluate the overall performance of algorithms. It is calculated using the formula:

$$\mathrm{PWC} = \frac{\mathrm{FN}+\mathrm{FP}}{\mathrm{TP}+\mathrm{FN}+\mathrm{FP}+\mathrm{TN}}. \tag{9}$$

It measures the portion of total number of wrongly classified points in a given frame. PWC is also named as the Portion of Total Error in some literature.

In this paper, we employ the similarity metric [37] to measure the similarity between the segmented foreground and the ground truth. Let $\mathcal{X}$ and $\mathcal{Y}$ respectively denote the set of foreground pixels determined by a method, and the set of foreground pixels from the ground truth. The similarity S is calculated using the formula:

$$\mathrm{S} = \frac{|\mathcal{X} \cap \mathcal{Y}|}{|\mathcal{X} \cup \mathcal{Y}|}, \tag{10}$$

where $|\cdot|$ represents the cardinality of a set. $|\mathcal{X} \cap \mathcal{Y}|$ is the number of true positives. The value of S falls into the range from 0 to 1. The similarity metric measures the overall segmentation performance. Larger similarity value indicates better performance.

TABLE I: Critical values for the distance threshold θ.

Sequence	GenSeq	ShSeq	ColCamSeq	DCamSeq
Value	4 cm	1 cm	3 cm	3 cm

In order to measure how well a method performs with respect to the other methods, we combine the performance of a method across different metrics and different sequences into a single rank [30]. Let $\mathrm{RM}_{i,c}$ denote the average rank of method i over all the metrics in the sequence c. It is calculated using the formula:

$$\mathrm{RM}_{i,c} = \frac{1}{\mathrm{N_m}} \sum_{k=1}^{\mathrm{N_m}} \pi_i(\mathrm{m}_k, c), \tag{11}$$

where m_k represents the metric k, $\mathrm{N_m}$ is the number of metrics, the number $\mathrm{N_m} = 4$ here, the function $\pi_i(\mathrm{m}_k, c)$ obtains the rank of method i in terms of m_k in the sequence c. Note that $\pi_i(\cdot)$ ranks an ascending order for PWC, FPR, FNR, and a descending order for S. Smaller values of ranks correspond to better performance. $\pi_i(\cdot)$ returns the average rank if more than one value share the same rank. Let RC_i denote the overall rank of method i. It is calculated by taking the average of $\mathrm{RM}_{i,c}$ over all the categories:

$$\mathrm{RC}_i = \frac{1}{\mathrm{N_c}} \sum_{c=1}^{\mathrm{N_c}} \mathrm{RM}_{i,c}, \tag{12}$$

where $\mathrm{N_c}$ is the number of sequences, the number $\mathrm{N_c} = 4$ because we have totally 4 sequences in this paper.

TABLE II: The quantitative comparison results obtained with the GenSeq sequence. Bold font highlights the best results.

Methods	PWC		FPR		FNR		S		RM
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	
Ours	**0.0080**	0.0026	0.0071	0.0026	0.0833	0.2203	0.7621	0.2425	4.50
CL_W	0.0130	0.0042	0.0127	0.0001	0.0149	0.0002	**0.8300**	0.2100	**4.00**
CL_C	0.0238	0.0120	0.0063	0.0002	0.1638	0.0030	0.7200	0.2300	7.25
CL_D	0.0206	0.0048	0.0209	0.0001	0.0177	0.0003	0.7800	0.2100	6.25
MOG_C	0.0284	0.0123	**0.0014**	0.0006	0.4971	0.1591	0.4013	0.1319	10.00
MOG_D	0.0194	0.0101	0.0055	0.0013	0.2156	0.1523	0.6334	0.1712	7.25
$MOG2_C$	0.0952	0.0708	0.1023	0.0777	0.0553	0.1537	0.4240	0.1802	12.00
$MOG2_D$	0.0421	0.0213	0.0444	0.0230	0.0156	0.0209	0.5267	0.1809	9.50
GMG_C	0.0425	0.0203	0.0289	0.0149	0.2683	0.1137	0.4263	0.1212	12.00
GMG_D	0.0595	0.0284	0.0114	0.0064	0.7083	0.1821	0.2030	0.0845	13.75
$ViBe_C$	0.0162	0.0062	0.0029	0.0015	0.3131	0.2079	0.5273	0.1881	7.75
$ViBe_D$	0.0114	0.0042	0.0083	0.0018	0.0692	0.1556	0.7330	0.2048	4.75
$PBAS_C$	0.1448	0.1021	0.1561	0.1133	0.0890	0.1235	0.3218	0.1400	14.00
$PBAS_D$	0.0473	0.0181	0.0508	0.0197	**0.0114**	0.0489	0.5311	0.2111	9.25
$DECOLOR_C$	0.0182	0.0103	0.0044	0.0018	0.2806	0.1932	0.5876	0.2026	7.50
$DECOLOR_D$	0.0159	0.0109	0.0092	0.0024	0.0822	0.1125	0.7084	0.2109	6.25

TABLE III: The quantitative comparison results obtained with the ShSeq sequence. Bold font highlights the best results.

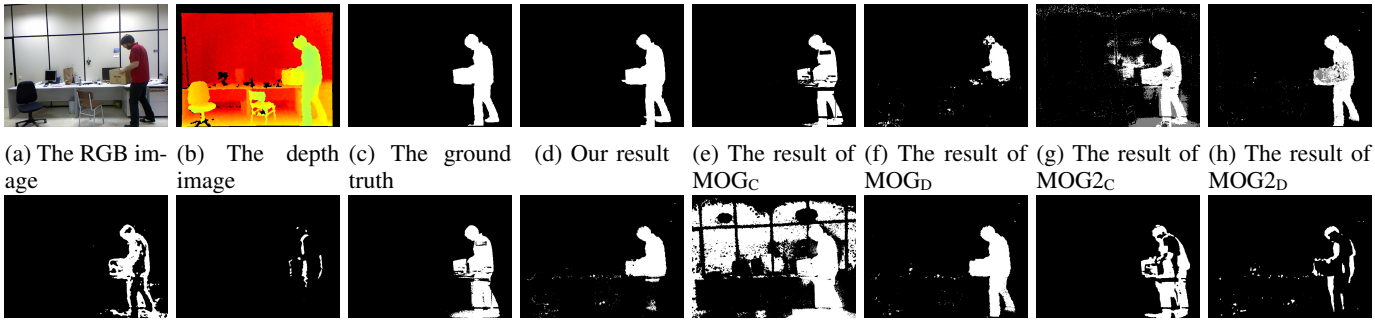
Methods	PWC		FPR		FNR		S		RM
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	
Ours	0.0181	0.0033	0.0188	0.0041	0.0128	0.0040	0.8440	0.0236	5.00
CL_W	**0.0081**	0.0035	0.0068	0.0002	0.0160	0.0005	**0.9400**	0.0400	2.50
CL_C	0.0537	0.0288	0.0323	0.0005	0.1820	0.0058	0.6700	0.1600	9.50
CL_D	0.0098	0.0033	0.0098	0.0002	**0.0095**	0.0004	0.9300	0.0300	**2.25**
MOG_C	0.0526	0.0095	0.0111	0.0022	0.3989	0.0764	0.4887	0.0646	9.13
MOG_D	0.0242	0.0042	0.0050	0.0010	0.1678	0.0362	0.7773	0.0276	5.25
$MOG2_C$	0.2171	0.0764	0.2197	0.1054	0.2109	0.1278	0.2985	0.0276	13.00
$MOG2_D$	0.0480	0.0248	0.0111	0.0015	0.3326	0.2245	0.5675	0.1944	8.38
GMG_C	0.0940	0.0060	0.0380	0.0184	0.5634	0.1106	0.2857	0.0359	12.75
GMG_D	0.1101	0.0115	**0.0036**	0.0011	0.9231	0.0444	0.0710	0.0384	11.50
$ViBe_C$	0.0386	0.0048	0.0223	0.0024	0.1742	0.0472	0.6166	0.0468	8.25
$ViBe_D$	0.0187	0.0022	0.0124	0.0007	0.0644	0.0221	0.8367	0.0141	5.75
$PBAS_C$	0.4596	0.0795	0.5258	0.0890	0.0153	0.0056	0.2196	0.0313	12.50
$PBAS_D$	0.0160	0.0005	0.0167	0.0009	0.0105	0.0051	0.8685	0.0062	4.00
$DECOLOR_C$	0.0987	0.0027	0.0427	0.0093	0.5498	0.0784	0.3065	0.0280	12.50
$DECOLOR_D$	0.1251	0.0064	0.0224	0.0058	0.9025	0.0208	0.0803	0.0141	13.75

C. Parameter Tuning

We have mainly three parameters in our method. They are the number of samples λ that is introduced in Algorithm 1, the distance threshold θ that is introduced in formula (2) and the discrimination threshold κ that is introduced in Algorithm 2. In order to choose optimal values for these parameters, we compare the foreground segmentation performance of our method under different parameter settings.

We observed that small values of θ can produce many unstable false positives in the results of ${}^{D}\Phi$. We conjecture that this is because the value of θ may be smaller than the standard deviations of the depth measurements. The measurement noises that are comparable to θ would increase false classifications. We tune the parameter θ by qualitatively observing the segmentation performance. The false positives increase substantially when θ is set equal to or below a critical value. Tab. I displays the critical values of θ for the four sequences. The critical values vary from scene to scene. For scenes with shorter distances to the camera, such as the ShSeq sequence, the critical value is small. This is because depth measurement noises are relatively small for short distances [29]. For scenes with larger distances, such as the GenSeq sequence, the critical values are larger due to the quadratically increased depth measurement noises [38]. To reduce false classifications caused by depth measurement noises, we prefer to use a large θ. However, too large values of θ lead to the problem of depth camouflage. The foreground and the background could not be discriminated when θ is larger than the size of the moving objects. Thus, a preferred way is to set the value of θ empirically according to the scene and the moving objects. For the DCamSeq sequence, with the consideration of the thickness of the human arm we set θ to 4 cm. This could avoid the arm merging with the cabinet. Since the foreground and the background are not so close in the other sequences, we set θ to 5 cm for them.

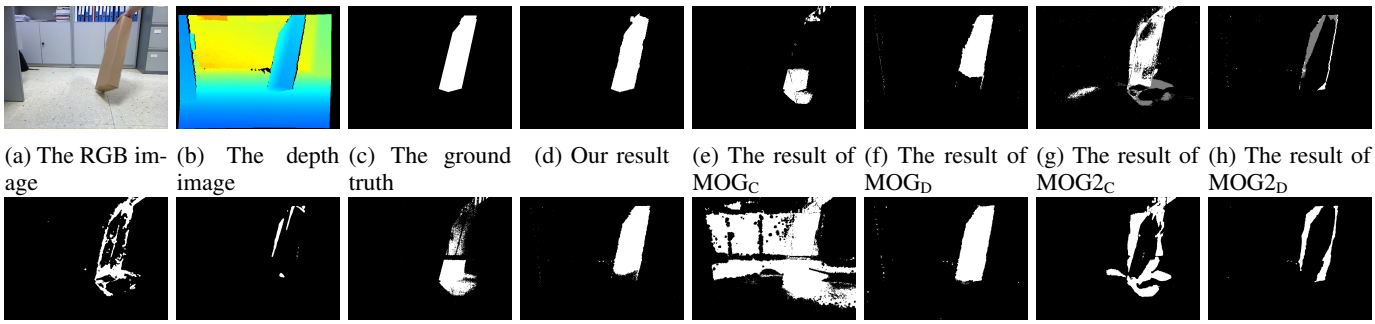
We choose κ and λ by comparing the PWC values under different parameter settings. We firstly set λ to a fixed value and observe PWC according to different values of κ. Fig. 4a



(a) The RGB image (b) The depth image (c) The ground truth (d) Our result (e) The result of MOG_C (f) The result of MOG_D (g) The result of $MOG2_C$ (h) The result of $MOG2_D$

(i) The result of GMG_C (j) The result of GMG_D (k) The result of $ViBe_C$ (l) The result of $ViBe_D$ (m) The result of $PBAS_C$ (n) The result of $PBAS_D$ (o) The result of $DECOLOR_C$ (p) The result of $DECOLOR_D$

Fig. 5: The qualitative comparison results with the `GenSeq` sequence. The gray pixels in (g) and (h) indicate the shadows detected by MOG2. The frame shows that a person is going to put the box on the table. The foreground objects here are the person and the box. As we can see, our method achieves the best segmentation performance.



(a) The RGB image (b) The depth image (c) The ground truth (d) Our result (e) The result of MOG_C (f) The result of MOG_D (g) The result of $MOG2_C$ (h) The result of $MOG2_D$

(i) The result of GMG_C (j) The result of GMG_D (k) The result of $ViBe_C$ (l) The result of $ViBe_D$ (m) The result of $PBAS_C$ (n) The result of $PBAS_D$ (o) The result of $DECOLOR_C$ (p) The result of $DECOLOR_D$

Fig. 6: The qualitative comparison results with the `ShSeq` sequence. The gray pixels in (g) and (h) indicate the shadows detected by MOG2. The frame shows that a person is rotating a box on the table. The foreground object here is the box. As we can see, our method achieves the best segmentation performance.

shows the tuning results for κ. As we can see, the PWC mean and standard deviations tend to increase when we increase the value of κ. This figure suggests that a smaller value of κ is preferred. From qualitative observations, we find that a larger κ causes more false positives in the results of ${}^{D}\Phi$. This would hence lead to more unknown holes in the trimap after the morphological operations, which could reduce hard constraints for the graph-cut algorithm. With less hints for ${}^{C}\Phi$, the segmentation performance would be degraded. From Fig. 4a, we can see that the PWC value is not sensitive to κ especially in the `ShSeq` sequence. We think the reason is that the objects in the scene are close to the camera. Thus, the standard deviations of depth measurements are smaller in this sequence and the variable ω in (2) becomes more concentrated. The value of ω is close to 0 or λ and not prone to be affected by κ. Therefore, the PWC values are stable in such a case.

Fig. 4b illustrates the tuning results for λ. We fix the discrimination threshold κ to 0 and observe PWC with different values of λ. We classify a given point as background as long as there is a sample point close to the given point within the distance threshold. As we can see, PWC tends to increase when we increase the number of samples. This figure suggests that larger values of λ ensure better performance. We think the reason for this is that a model with a larger number of samples could accommodate more pixel variations and hence better describe the distribution of a given point. From Fig. 4b, we find that PWC is not sensitive to λ especially in the `ShSeq` sequence. As aforementioned, the depth measurements are concentrated due to the short distances in the `ShSeq` sequence. More samples could not contribute too much in such a case due to the concentrated values. Thus, the performance is stable in the `ShSeq` sequence.

D. Method Comparison

We compare our method with the well-known background modelling algorithms: MOG [39], MOG2 [40], GMG [41], ViBe [22], PBAS [23] and DECOLOR [42]. It should be noted that these background modelling algorithms are developed using a color camera rather than an RGB-D camera. To ensure fair comparisons, we evaluate these background modelling algorithms using color and depth images separately. In addition, we also compare our method with the RGB-D camera-based method proposed by Camplani *et al.* [30].

For the MOG, MOG2 and GMG algorithms, we use the implementations in OpenCV [43]. For the ViBe and PBAS algorithms, we use the open-source codes from [44] and [45]. For the DECOLOR algorithm, we use the implementation provided by *lrslibrary* [46]. For the Camplani method, we directly

TABLE IV: The quantitative comparison results obtained with the `ColCamSeq` sequence. Bold font highlights the best results.

Methods	PWC		FPR		FNR		S		RM
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	
Ours	**0.0114**	0.0041	0.0145	0.0092	0.0100	0.0067	**0.9411**	0.0323	**2.00**
CL_W	0.0320	0.0277	0.0292	0.0010	0.0352	0.0009	0.8900	0.1500	5.50
CL_C	0.3902	0.2312	0.0227	0.0006	0.8227	0.0111	0.2200	0.2200	11.50
CL_D	0.0247	0.0235	0.0238	0.0010	0.0258	0.0005	0.9100	0.1000	4.00
MOG_C	0.3444	0.0839	**0.0047**	0.0016	0.8675	0.0443	0.1228	0.0403	11.00
MOG_D	0.3523	0.0776	0.0050	0.0037	0.8894	0.0896	0.1031	0.0833	12.00
$MOG2_C$	0.2216	0.0519	0.0864	0.0325	0.3922	0.1470	0.4658	0.1213	9.75
$MOG2_D$	0.1242	0.0730	0.0599	0.0294	0.2034	0.1613	0.6454	0.1578	8.25
GMG_C	0.4210	0.0476	0.2622	0.0640	0.6251	0.1126	0.2308	0.0336	13.00
GMG_D	0.3395	0.1090	0.0073	0.0014	0.8060	0.1680	0.1740	0.1296	9.75
$ViBe_C$	0.3421	0.0806	0.0288	0.0110	0.8286	0.0558	0.1489	0.0465	12.00
$ViBe_D$	0.0141	0.0037	0.0159	0.0077	0.0146	0.0097	0.8934	0.0340	3.25
$PBAS_C$	0.2542	0.0319	0.3376	0.0577	0.1372	0.0734	0.4999	0.0952	9.75
$PBAS_D$	0.0519	0.0226	0.0884	0.0353	**0.0044**	0.0031	0.8658	0.0588	6.25
$DECOLOR_C$	0.2977	0.0787	0.0333	0.0112	0.7077	0.1038	0.2569	0.0881	10.25
$DECOLOR_D$	0.1190	0.0884	0.0371	0.0217	0.2155	0.1615	0.7180	0.1671	7.75

TABLE V: The quantitative comparison results obtained with the `DCamSeq` sequence. Bold font highlights the best results.

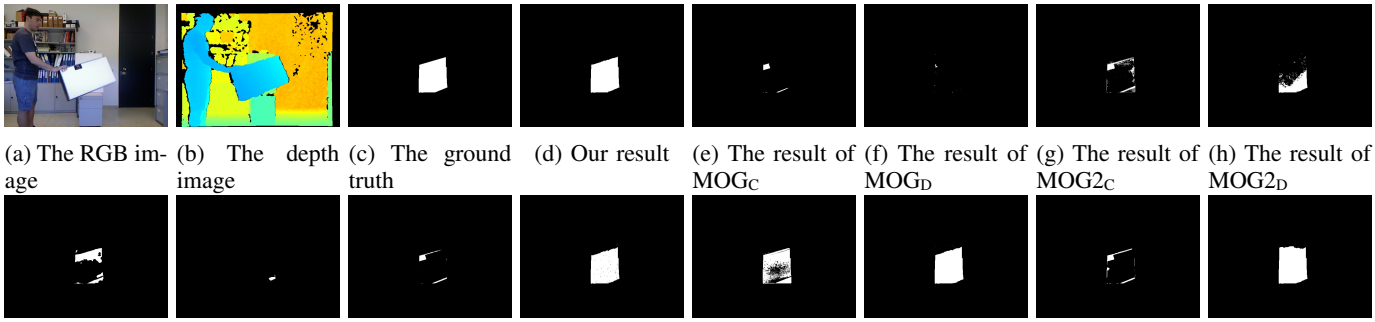
Methods	PWC		FPR		FNR		S		RM
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	
Ours	0.0236	0.0135	0.0231	0.0244	0.0673	0.0496	0.6387	0.0536	5.25
CL_W	0.0246	0.0182	0.0066	0.0001	0.3221	0.0026	0.5500	0.1400	5.25
CL_C	0.0178	0.0147	0.0095	0.0001	0.1560	0.0009	**0.6700**	0.1300	3.25
CL_D	0.0338	0.0219	0.0064	10^{-5}	0.4849	0.0044	0.4000	0.2000	6.75
MOG_C	0.0330	0.0355	0.0014	0.0011	0.5924	0.1147	0.2950	0.0742	7.50
MOG_D	0.0629	0.0536	**0.0008**	0.0007	0.9543	0.0218	0.0430	0.0202	10.75
$MOG2_C$	0.0204	0.0164	0.0120	0.0065	0.1598	0.0270	0.4357	0.0418	5.25
$MOG2_D$	0.0467	0.0254	0.0271	0.0149	0.5902	0.1970	0.1122	0.0251	11.25
GMG_C	0.0521	0.0388	0.0386	0.0342	0.2931	0.0597	0.3773	0.0144	9.75
GMG_D	0.0647	0.0512	0.0136	0.0091	0.8087	0.1113	0.1433	0.0832	11.75
$ViBe_C$	0.0207	0.0179	0.0028	0.0026	0.3897	0.0545	0.3970	0.0374	5.00
$ViBe_D$	0.0351	0.0156	0.0108	0.0161	0.4899	0.2863	0.4067	0.2195	7.00
$PBAS_C$	0.0600	0.0442	0.0651	0.0666	**0.0506**	0.0119	0.4578	0.0677	8.75
$PBAS_D$	0.0573	0.0221	0.0429	0.0202	0.3450	0.2427	0.3800	0.2071	9.75
$DECOLOR_C$	**0.0165**	0.0134	0.0092	0.0092	0.1470	0.0151	0.6080	0.0295	**2.75**
$DECOLOR_D$	0.0622	0.0636	0.0224	0.0379	0.6723	0.1366	0.2306	0.0750	9.75

import the quantitative results from [30] for our comparison. In this paper, we denote the background modelling algorithms using color and depth images as $(\cdot)_C$ and $(\cdot)_D$, respectively. For the Camplani method, we adopt the notations CL_W, CL_C and CL_D from [30] to represent the combined classifier, the color-based classifier and the depth-based classifier, respectively.

1) The `GenSeq` Sequence Results: The quantitative comparison results obtained with the `GenSeq` sequence are reported in Tab. II. The first column of the table shows the method names. The mean and Standard Deviations (S.D.) values are displayed in the Mean and S.D. columns. As we can see, our method guarantees the lowest value of PWC. This demonstrates that our method is able to provide the best overall performance in terms of PWC in such a complex scenario with common challenges. Moreover, our method allows to obtain lower values of FPR and FNR. It is worth noting that the worst performance is obtained by $PBAS_C$. The reason for the poor performance could be that it is sensitive to changes in illumination and color, because the lighting in the room is not stable and there exist automatic brightness and white balance adjustments when the person is walking around in the room.

Fig. 5 qualitatively compares the methods with a selected frame from the `GenSeq` sequence. As we can see, our method presents the best accuracy. By comparing our result with the ground truth, we can find that there are some false positives at the boundaries of the box and the foot. The pixels are mis-classified by the color-based classifier ${}^{C}\Phi$ due to the close color values at these areas. The given frame exhibits automatic camera adjustments. Thus, we can see that many pixels are wrongly classified by $PBAS_C$, while our method is robust to the illumination changes and the automatic camera adjustments.

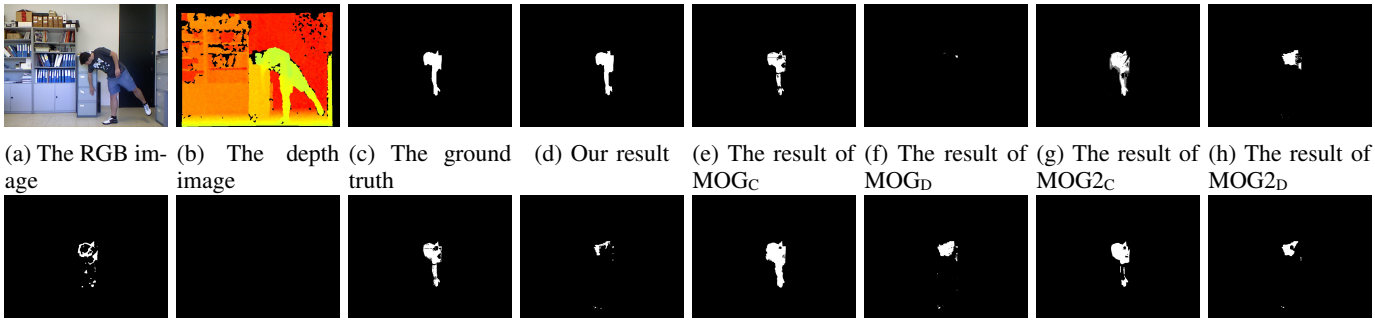
2) The `ShSeq` Sequence Results: Tab. III displays the quantitative results obtained with the `ShSeq` sequence. As we can see, our method allows to obtain the second lowest value of PWC. We find that CL_W provides the best results in terms of PWC and S. It is worth noting that the performance of the DECOLOR methods are worse than most of the other



(a) The RGB image (b) The depth image (c) The ground truth (d) Our result (e) The result of MOG_C (f) The result of MOG_D (g) The result of $MOG2_C$ (h) The result of $MOG2_D$

(i) The result of GMG_C (j) The result of GMG_D (k) The result of $ViBe_C$ (l) The result of $ViBe_D$ (m) The result of $PBAS_C$ (n) The result of $PBAS_D$ (o) The result of $DECOLOR_C$ (p) The result of $DECOLOR_D$

Fig. 7: The qualitative comparison results with the `ColCamSeq` sequence. The gray pixels in (g) and (h) indicate the shadows detected by the MOG2 method. The foreground segmentation results are masked with the provided ROI. The frame shows that a person is waving a white box before the white panel. The foreground object here is the box. As we can see, both our method and the $PBAS_D$ method accurately segment the foreground.



(a) The RGB image (b) The depth image (c) The ground truth (d) Our result (e) The result of MOG_C (f) The result of MOG_D (g) The result of $MOG2_C$ (h) The result of $MOG2_D$

(i) The result of GMG_C (j) The result of GMG_D (k) The result of $ViBe_C$ (l) The result of $ViBe_D$ (m) The result of $PBAS_C$ (n) The result of $PBAS_D$ (o) The result of $DECOLOR_C$ (p) The result of $DECOLOR_D$

Fig. 8: The qualitative comparison results with the `DCamSeq` sequence. The gray pixels in (g) and (h) indicate the shadows detected by the MOG2 method. The foreground segmentation results are masked with the provided ROI. The frame shows that a person is running his hand on the cabinet. The foreground objects here are the hand and the arm. As we can see, our method achieves the best segmentation performance.

methods. In the `ShSeq` sequence, the box is rotated at a fixed point on the ground. The central part of the box becomes motionless due to the little translation movement. This leads to a situation in which the central area cannot be detected as a part of foreground by the DECOLOR algorithm. We believe that this is the main reason for the poor performance of the DECOLOR methods.

Fig. 6 qualitatively compares the methods with a selected frame from the `ShSeq` sequence. As we expected, since the depth information is robust to shadows, all the methods based on depth information are little influenced by the shadow challenge. We can see that the shadows cause false detections for the color-based methods. Similar to the `GenSeq` sequence, the lighting in this sequence is not stable. The camera automatically adjusts the brightness when the box enters the scene. The incorrect classifications in the result of the $PBAS_C$ method show the negative influence caused by the automatic camera adjustment. Compared with the result of the $PBAS_D$ method, the advantage of using depth information is clearly demonstrated. As aforementioned, the bottom of the box is close to the floor. Thus, we can see that there are false classifications in the results of the depth-based methods around this area due to the depth camouflage. For the results of the DECOLOR methods, we can see that the central part of the box is not correctly detected as foreground. Our method outperforms all the other methods, which demonstrates the robustness of our method to the challenges of shadows, automatic camera adjustments and depth camouflage.

3) The `ColCamSeq` Sequence Results: Tab. IV displays the quantitative results obtained with the `ColCamSeq` sequence. As we can see, our method ranks No.1 among all the methods. As this sequence is designed to evaluate the performance of algorithms when color camouflage occurs, it comes as expected that virtually all the color-based methods are inferior to their depth-based ones. The benefit of using depth information is clearly demonstrated here. The RGB-D data-based method CL_W also achieves good results. However, the performance of the corresponding color-based method CL_C is greatly degraded by the color camouflage. We find from the RM values that the combined method CL_W is a little inferior to the depth-based method CL_D. We believe the reason for this is that CL_C weights down the combined method CL_W. It is worth noting that the largest RM difference value is obtained between $ViBe_C$ and $ViBe_D$, which shows that the ViBe algorithm can

TABLE VI: The overall ranking over all the methods across all the sequences. DEC is an abbreviation of DECOLOR. Our method ranks No.1 among all of the methods.

	Ours	CL_W	CL_C	CL_D	MOG_C	MOG_D	$MOG2_C$	$MOG2_D$	GMG_C	GMG_D	$ViBe_C$	$ViBe_D$	$PBAS_C$	$PBAS_D$	DEC_C	DEC_D
RC	**4.19**	4.31	7.88	4.81	9.41	8.81	10.00	9.35	11.88	11.69	8.25	5.19	11.25	7.31	8.25	9.38
Rank	**1.0**	2.0	6.0	3.0	12.0	9.0	13.0	10.0	16.0	15.0	7.5	4.0	14.0	5.0	7.5	11.0

be greatly improved using depth data in such a case.

Fig. 7 shows the qualitative results with a selected frame from the `ColCamSeq` sequence. We can see that our method is robust to the color camouflage challenge. The performance enhancement brought by using depth information can be clearly proven from the results of $ViBe_D$, $PBAS_D$ and $DECOLOR_D$. Moreover, we find that $PBAS_D$ provides a comparative result with that of ours. In the sequence, we note that the lateral surfaces of the box are coloured black. They are distinctive to the camouflaged area. Thus, we can see that the color-based methods are able to roughly outline the boundary of the box.

4) The `DCamSeq` Sequence Results: Tab. V displays the quantitative results obtained with the `DCamSeq` sequence. This sequence is designed to evaluate the performance when depth camouflage occurs. In contrast with the results of the `ColCamSeq` sequence, all the color-based methods are superior to their depth-based ones. The advantage of using color information is clearly demonstrated here. We can see that both our method and CL_W provide good performance in this sequence. The performance of CL_C is weighted down by CL_D. This is similar to the case happened in the `ColCamSeq` sequence. It is worth noting that $DECOLOR_C$ ranks as the best method in this sequence. We conjecture that this is because the underlying assumption of the DECOLOR algorithm is well satisfied in this sequence. The DECOLOR algorithm assumes that the foreground is composed of contiguous pieces of relatively small size. The hand, the arm and the head are contiguous and they are relatively small within the ROI in this sequence.

Fig. 8 qualitatively compares the methods with a selected frame from the `DCamSeq` sequence. We can see that our method achieves the best accuracy for this given frame. Because of the depth camouflage, all the depth-based methods are inferior to the color-based ones. As an extreme example, we can see that GMG_D totally fails to detect true positives.

E. The Overall Ranking

Tab. VI displays the overall ranking over all the methods across all the sequences. We can see that our method ranks No.1 among all of the methods. It comes as no surprise that CL_W ranks second to our method. We believe that the combination mechanism that uses both color and depth data improves the overall performance of CL_W. The superior performance of our method and the CL_W method demonstrate the advantage of using both the color and depth data. It should be noted that the GMG methods rank at the bottom. We think the reason for the unsatisfied performance is that most foreground motions in this dataset are not trivial. This violates the basic assumption of the GMG algorithm, which requires that the region of foreground is significantly smaller than that of the background.

V. Conclusions

We presented here a novel RGB-D data-based background modelling method which sequentially uses the depth and color information provided by an RGB-D camera. As a part of the method, a new single-frame initialization scheme was developed to fast initialize the background model. Like most of the initialization schemes of background modelling algorithms, we assume that there is no moving object during the initialization. For the foreground segmentation, we firstly developed a depth-based classifier to roughly classify the pixels into foreground, background, and unknowns. Then, a trimap generation scheme was devised based on the morphological operations with the rough classification results. Finally, a color-based classifier based on the graph-cut optimization framework was developed to refine the segmentation results. The trimap provides the hard constraints for the graph-cut framework. The experimental results demonstrate that our method is robust to most of common challenges for background modelling algorithms, such as shadows, color and depth camouflages, illumination changes and automatic camera adjustments. We believe that the superior performance was benefited from the active visual perception, specifically, the use of depth information. We quantitatively evaluate our method using a public RGB-D dataset with hand-labelled foreground segmentation ground truth. The results suggest that our method outperforms the current well-known methods. However, our method is unable to run real-timely at the current status. It can only work at around 2Hz on an Intel i5 desktop computer without code optimization. Most of the time was spent on the time-consuming graph-cut optimization framework. We consider it as a major limitation of our method and will accelerate our method with parallel programming techniques. In the future, we would also like to extend our method on freely-moving platforms by encoding and compensating the ego-motion of the camera.

References

[1] Y. Sun and M. Q.-H. Meng, "Multiple moving objects tracking for automated visual surveillance," in *2015 IEEE International Conference on Information and Automation*, Aug 2015, pp. 1617–1621.

[2] T. Nägeli, J. Alonso-Mora, A. Domahidi, D. Rus, and O. Hilliges, "Real-Time Motion Planning for Aerial Videography With Dynamic Obstacle Avoidance and Viewpoint Optimization," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1696–1703, July 2017.

[3] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robotics and Autonomous Systems*, vol. 89, pp. 110–122, 2017.

[4] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Computer Vision and Image Understanding*, vol. 122, pp. 4–21, 2014.

[5] T. Bouwmans, F. Porikli, B. Höferlin, and A. Vacavant, *Background Modeling and Foreground Detection for Video Surveillance*. CRC Press, 2014.

[6] R. Bajcsy and M. Campos, "Active and exploratory perception," *CVGIP: Image Understanding*, vol. 56, no. 1, pp. 31 – 40, 1992, Purposive, Qualitative, Active Vision.

[7] H. Liu, F. Sun, and X. Zhang, "Robotic Material Perception using Active Multi-Modal Fusion," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2018.

[8] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, Aug 1988.

[9] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable RGB-D SLAM in dynamic environments," *Robotics and Autonomous Systems*, vol. 108, pp. 115 – 128, 2018.

[10] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel(R) RealSense(TM) Stereoscopic Depth Cameras," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 1267–1276.

[11] Z. Zhang, "Microsoft Kinect Sensor and Its Effect," *IEEE MultiMedia*, vol. 19, no. 2, pp. 4–10, Feb 2012.

[12] T. Yan, Y. Sun, T. Liu, C.-H. Cheung, and M. Q.-H. Meng, "A Locomotion Recognition System Using Depth Images," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 6766–6772.

[13] Z. Cai, J. Han, L. Liu, and L. Shao, "RGB-D datasets using microsoft kinect or similar sensors: a survey," *Multimedia Tools and Applications*, pp. 1–43, 2016.

[14] X. Ren, D. Fox, and K. Konolige, "Change Their Perception: RGB-D for 3-D Modeling and Recognition," *IEEE Robotics Automation Magazine*, vol. 20, no. 4, pp. 49–59, Dec 2013.

[15] R. Horaud, M. Hansard, G. Evangelidis, and C. Ménier, "An overview of depth cameras and range scanners based on time-of-flight technologies," *Machine Vision and Applications*, vol. 27, no. 7, pp. 1005–1020, Oct 2016.

[16] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal from moving platforms: An RGB-D data-based motion detection, tracking and segmentation approach," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec 2015, pp. 1377–1382.

[17] L. Shao, J. Han, D. Xu, and J. Shotton, "Computer vision for RGB-D sensors: Kinect and its applications [special issue intro.]," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1314–1317, Oct 2013.

[18] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," in *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, Oct 1996, pp. 51–56.

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

[20] M. Shah, J. D. Deng, and B. J. Woodford, "Video background modeling: recent approaches, issues and our proposed techniques," *Machine Vision and Applications*, vol. 25, no. 5, pp. 1105–1119, 2014.

[21] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *European conference on computer vision*. Springer, 2000, pp. 751–767.

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

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

[24] G. Gordon, T. Darrell, M. Harville, and J. Woodfill, "Background estimation and removal based on range and color," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE, 1999.

[25] E. J. Fernandez-Sanchez, J. Diaz, and E. Ros, "Background Subtraction Based on Color and Depth Using Active Sensors," *Sensors*, vol. 13, no. 7, pp. 8895–8915, 2013.

[26] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," *Real-time imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[27] J. Murgia, C. Meurie, and Y. Ruichek, "An Improved Colorimetric Invariants and RGB-Depth-Based Codebook Model for Background Subtraction Using Kinect," in *Mexican International Conference on Artificial Intelligence*. Springer, 2014, pp. 380–392.

[28] A. Amamra, T. Mouats, and N. Aouf, "GPU based GMM segmentation of kinect data," in *Proceedings ELMAR-2014*. IEEE, 2014, pp. 1–4.

[29] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[30] M. Camplani and L. Salgado, "Background foreground segmentation with RGB-D Kinect data: An efficient combination of classifiers," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 122–136, 2014.

[31] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, Sep 2004.

[32] J. Cheng, Y. Sun, and M. Q.-H. Meng, "A dense semantic mapping system based on CRF-RNN network," in *2017 18th International Conference on Advanced Robotics (ICAR)*, July 2017, pp. 589–594.

[33] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE transactions on pattern analysis and machine intelligence*, no. 4, pp. 532–550, 1987.

[34] M. Roushdy, "Comparative study of edge detection algorithms applying on the grayscale noisy image using morphological filter," *GVIP journal*, vol. 6, no. 4, pp. 17–23, 2006.

[35] C. Eveland, K. Konolige, and R. C. Bolles, "Background modeling for segmentation of video-rate stereo sequences," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. IEEE, 1998, pp. 266–271.

[36] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection. net: A new change detection benchmark dataset," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2012, pp. 1–8.

[37] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1459–1472, 2004.

[38] K. Khoshelham and S. O. Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[39] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-based surveillance systems*. Springer, 2002, pp. 135–144.

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

[41] A. B. Godbehere, A. Matsukawa, and K. Goldberg, "Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 4305–4312.

[42] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 597–610, 2013.

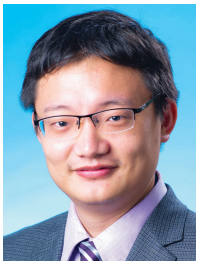
[43] OpenCV Background Modelling Algorithms Website. [Online]. Available: https://docs.opencv.org/3.4.1/d1/dc5/tutorial_background_subtraction.html

[44] ViBe Website. [Online]. Available: http://www.telecom.ulg.ac.be/research/vibe/

[45] PBAS Website. [Online]. Available: https://sites.google.com/site/pbassegmenter/

[46] A. Sobral, T. Bouwmans, and E.-h. Zahzah, "Lrslibrary: Low-rank and sparse tools for background modeling and subtraction in videos," *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*, 2015.

Yuxiang Sun received his Ph.D. degree from The Chinese University of Hong Kong (CUHK), Hong Kong, China, in 2017, the master's degree from University of Science and Technology of China (USTC), Hefei, China, in 2012, and the bachelor's degree from Hefei University of Technology (HFUT), Hefei, China, in 2009. He is now a research associate at the Robotics Institute, Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology (HKUST), Hong Kong, China. His current research interests include mobile robots, autonomous vehicles, deep learning, SLAM and navigation, motion detection, etc. He is a recipient of the Best Student Paper Finalist Award at the IEEE ROBIO 2015.



Ming Liu received the BA degree in Automation at Tongji University in 2005. During his master study at Tongji University, he stayed one year in Erlangen-Nünberg University and Fraunhofer Institute IISB, Germany, as a master visiting scholar. He graduated as a PhD student from the Department of Mechanical and Process Engineering of ETH Zürich in 2013, supervised by Prof Roland Siegwart. He is now affiliated with ECE department, CSE department and Robotics Institute of Hong Kong University of Science and Technology. He is a founding member of Shanghai SWing Automation Ltd. Co. He is also coordinating and involved in NSF projects, and National 863-Hi-Tech-Plan projects in China. As a team member, he won the second place of EMAV'09 (European Micro Aerial Vehicle Competition) and two awards from IARC'14 (International Aerial Robot Competition). He won the Best Student Paper Award as first author for MFI 2012 (IEEE International Conference on Multisensor Fusion and Information Integration), the Best Paper Award in Information for ICIA 2013 (IEEE International Conference on Information and Automation) as first author and Best Paper Award Finalists as co-author, the Best RoboCup Paper Award for IROS 2013 (IEEE/RSJ International Conference on Intelligent Robots and Systems), the Best Conference Paper Award for IEEE-CYBER 2015, Best Student Paper Finalist for RCAR 2015 (IEEE International conference on Real-time Computing and Robotics), Best Student Paper Finalist for ROBIO 2015, Best Student Paper Award for IEEE-ICAR 2017 and Best Paper in Automation Award for IEEE-ICIA 2017. He won twice the innoviation contest Chunhui Cup Winning Award in 2012 and 2013. He won the Wu Weijun AI award in 2016. He was the Program Chair of IEEE-RCAR 2016; the Program Chair of International Robotics Conference in Foshan 2017; He is the Conference Chair of ICVS 2017. Ming Liu's research interests include dynamic environment modeling, deep-learning for robotics, 3D mapping, machine learning and visual control.

Max Q.-H. Meng received his Ph.D. degree in Electrical and Computer Engineering from the University of Victoria, Canada, in 1992. He joined the Chinese University of Hong Kong in 2001 and is currently Professor and Chairman of Department of Electronic Engineering. He was with the Department of Electrical and Computer Engineering at the University of Alberta in Canada, serving as the Director of the Advanced Robotics and Teleoperation Lab and holding the positions of Assistant Professor (1994), Associate Professor (1998), and Professor (2000), respectively. He is affiliated with the State Key Laboratory of Robotics and Systems at Harbin Institute of Technology and the Honorary Dean of the School of Control Science and Engineering at Shandong University, in China. His research interests include robotics, medical robotics and devices, perception, and scenario intelligence. He has published some 600 journal and conference papers and led more than 50 funded research projects to completion as PI. He has served as an editor of several journals and General and Program Chair of many conferences including General Chair of IROS 2005 and General Chair of ICRA 2021 to be held in Xi'an, China. He is an elected member of the Administrative Committee (AdCom) of the IEEE Robotics and Automation Society. He is a recipient of the IEEE Millennium Medal, a Fellow of the Canadian Academy of Engineering, a Fellow of HKIE, and a Fellow of IEEE.