Ph.D. DISSERTATION

Background-Centric Approach for Moving Object Detection in Moving Cameras

동적 카메라에서 동적 물체 탐지를 위한 배경 중심 접근법

BY

Kimin Yun

FEBRUARY 2017

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Abstract

A number of surveillance cameras have been installed for safety and security in actual environments. To achieve a human-level visual intelligence via cameras, there has been much effort to develop many computer vision algorithms realizing the various visual functions from low level to high level. Among them, the moving object detection is a fundamental function because the attention to a moving object is essential to understand its high-level behavior. Most of moving object detection algorithms in a fixed camera adopt the background-centric modeling approach. However, the background-centric approach does not work well in a moving camera because the modeling of moving background in an online way is challengeable. Until now, most algorithms for the object detection in a moving camera have relied on the object-centric approach using appearance-based recognition schemes. However, the object-centric approach suffers from the heavy computational complexity. In this thesis, we propose an efficient and robust scheme based on the background-centric approach to detect moving objects in the dynamic background environments using moving cameras. To tackle the challenges arising from the dynamic background, in this thesis, we deal with four problems: false positives from inaccurate camera motion estimation, sudden scene changes such as illumination, slow moving object relative to camera movement, and motion model limitation in a dashcam video.

To solve the false positives due to motion estimation error, we propose a new scheme to improve the robustness of moving object detection in a moving camera. To lessen the influence of background motion, we adopt a dual-mode kernel model that builds two background models using a grid-based modeling. In addition, to reduce the false detections and the missing of true objects, we introduce an attentional sampling scheme based on spatio-temporal properties of moving objects. From the spatio-temporal properties, we build a foreground probability map and generate a sampling map which selects the candidate pixels to find the actual objects. We apply the background subtraction and model update with attention to only the selected pixels.

To resolve sudden scene changes and slow moving object problems, we propose a situation-aware background learning method that handles dynamic scenes for moving object detection in a moving camera. We suggest new modules that utilizes situation variables and builds a background model adaptively. Our method compensates for camera movement and updates the background model according to the situation variables. The situation-aware scheme enables the algorithm to build a clear background model without contamination by the foreground.

To overcome the limitation of motion model in a dashcam video, we propose a prior-based attentional update scheme to handle dynamic scene changes. Motivated by the center-focused and structure-focused tendencies of human attention, we extend the compensation-based method that focuses on the center changes and neglects minor changes on the important scene structure. The center-focused tendency is implemented by increasing the learning rate of the boundary region through the multiplication of the attention map and the age model. The structure-focused tendency is used to build a robust background model through the model selection after the road and sky region are estimated.

In experiments, the proposed framework shows its efficiency and robustness through qualitative and quantitative comparison evaluation with the state-of-the arts. Through the first scheme, it takes only 4.8 ms in one frame processing without parallel processing. The second scheme enables to adapt rapidly changing scenes while maintaining the performance and speed. Through the third scheme for the driving situation, successful results are shown in background modeling and moving object detection in dachcam videos.

Keywords: moving object detection, background modeling, moving camera, visual surveillance

Student Number: 2010-20847

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Many and various types of surveillance cameras are installed and developed for safety and security in many environments such as a kindergarten, a back street, an airport, and so on. However, human resources for monitoring these videos are limited compared with many installed cameras. For this reason, an intelligent visual surveillance system that helps the monitoring using computer vision technique is one of active research. Although the kind of intelligent visual surveillance system is diverse along the target application, this system mainly includes detection, tracking, and understanding tasks. The detection task is the fundamental task to find objects of interest, which are mostly moving objects. Next, in the tracking task, the detected objects are tracked using motion and appearance information of objects. Finally, in the understanding task, the activity or interaction between objects are modeled, and the outliers from the learned model are detected as unusual/abnormal events. As shown in Figure 1.1, when security guards monitor multiple cameras at the same time, the intelligent visual surveillance system helps them. The region of interest where a moving object appears

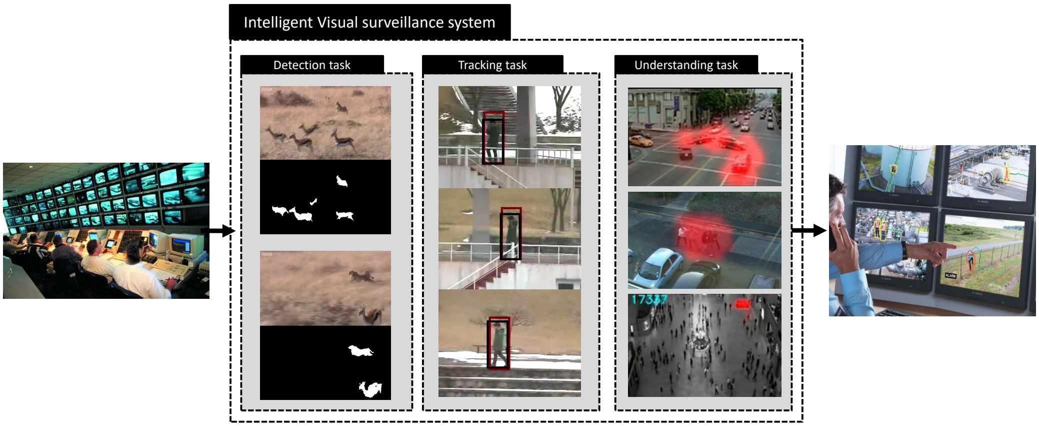


Figure 1.1: The usage of the intelligent visual surveillance system. Because security guards cannot monitor a number of cameras at the same time, the intelligent visual surveillance system helps them. From the low-level task (detection) to the high-level task (understanding), various algorithms help the monitoring to cope with the crime/incident.

is detected, and then this target object is tracked with automatic camera control. Then, when these objects behave suspiciously or unusually, this system informs security guards that an abnormal event occurs automatically to cope with the crime/incident. Although the understanding task is an ultimate objective for an ideal intelligent surveillance, it is limited to several scenarios and is not practical in real-environment until now. However, the detection task is more a general and fundamental work to support other tasks, so we tackle the detection task that finds moving objects in a video.

Before we introduce the related works, we mention the issues of moving object detection for a practical application for visual surveillance. First, the detection algorithm should run in real-time and an online manner. The online algorithm issue means that the detection algorithm generates an output immediately without future observations. Although there are retrieval systems that run in a batch manner using whole observations, the instant response to the crime/incident is important in the visual surveillance system. Second, the algorithm should cope with various unimportant scene changes like sudden illumination changes. Because the surveillance video is influenced by both weather or camera state, and the detection algorithm should be robust to these disturbances. Third, the algorithm should be applicable to various surveillance camera platform. Since the platform of monitoring cameras has been diversified such as drone camera, dashcam on a vehicle, pan-tilt-zoom camera, and a mobile phone camera, the detection algorithm for moving object should be operated well on these videos.

1.2 Related works

Detecting moving objects in a video is a fundamental problem in image processing and computer vision. There are two main approaches for moving object detection: the object-centric approach and the background-centric approach. The object-centric approach [2–11] focuses on modeling the target object using the appearance and motion coherence of the object. Andriluka *et al.* [2] tried to build a motion model for target moving object. Jung and Kim [6] extracted the object using the visual saliency. The other methods [3–5,7–11] are mainly based on the image segmentation techniques. Since semi-supervised segmentation techniques like GrabCut have been achieved success in an image, these methods are extended to find objects in a video. Figure 1.2 shows the example of the object-centric method [9] that preserves a detailed boundary of the moving object. However, the semi-supervised methods need initial seed regions that indicate coarse object regions. Therefore, this approach requires the first object position as an input or makes some assumptions about the target object such as a primary object. The object-centric approach shows good performances in that it preserves the object boundary when the primary object is noticeable. However, this approach performs poorly when target objects are small and numerous. Figure 1.3 shows fail cases of state-of-the-art object-centric methods. As shown in the third row of Figure 1.3, the large regions around the object are falsely detected as a foreground. As shown in the fourth row of Figure 1.3, the algorithm often misses the small object. Also, it is difficult to handle long videos because the appearance of the target object is entirely changed in the scene and only short videos can be tested in the batch manner algorithms. Moreover, it has a high computational complexity, which makes this approach unsuitable for real-time applications. According to the recent benchmark paper, even the fast algorithm [11] takes 12 seconds, and other method [10] takes around three minutes.

The background-centric approach focuses on the modeling of background regions that exclude the target object region. This approach assumes that background pixels have a sim-

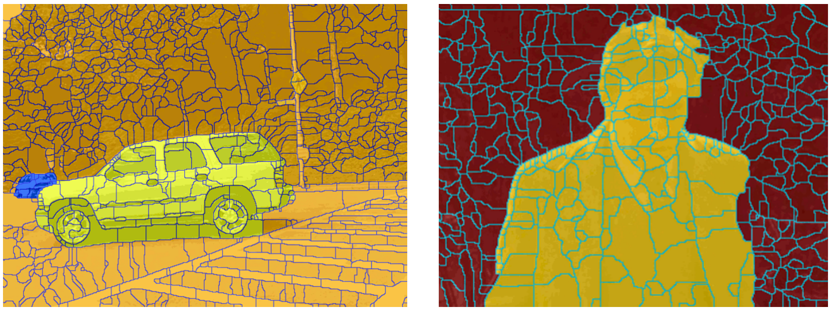


Figure 1.2: The example results from the object-centric method [9].

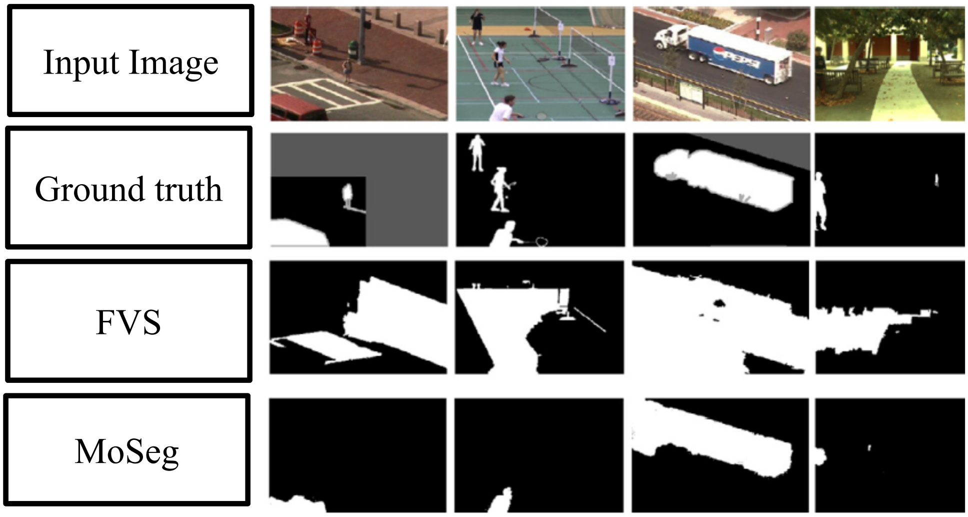
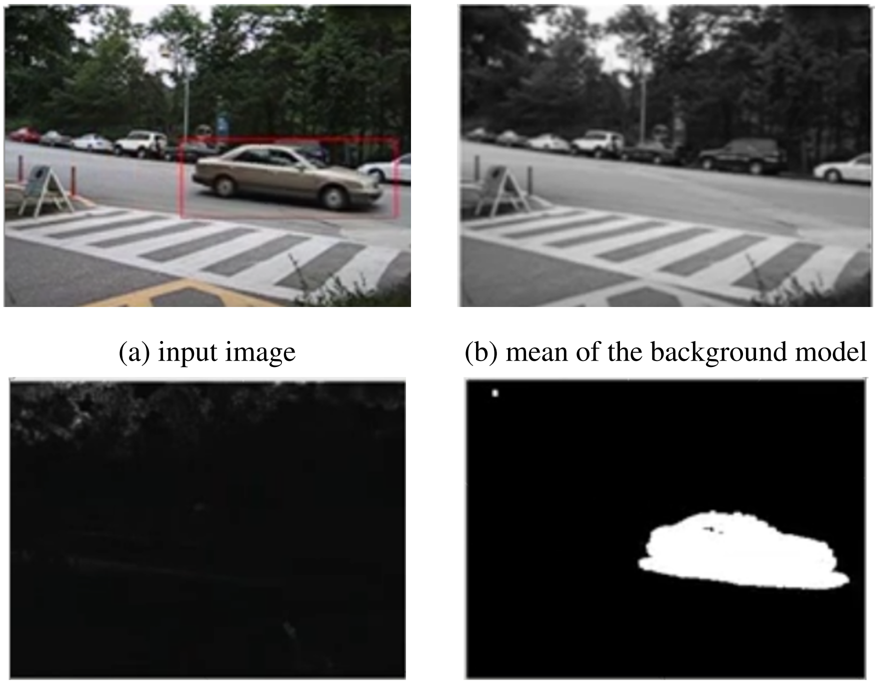


Figure 1.3: The fail cases of state-of-the-art object-centric methods: each row shows input frames, ground truths, FVS results [12], and MoSeg results [7], respectively.



(a) input image

(b) mean of the background model

(c) variance of the background model

(d) foreground result

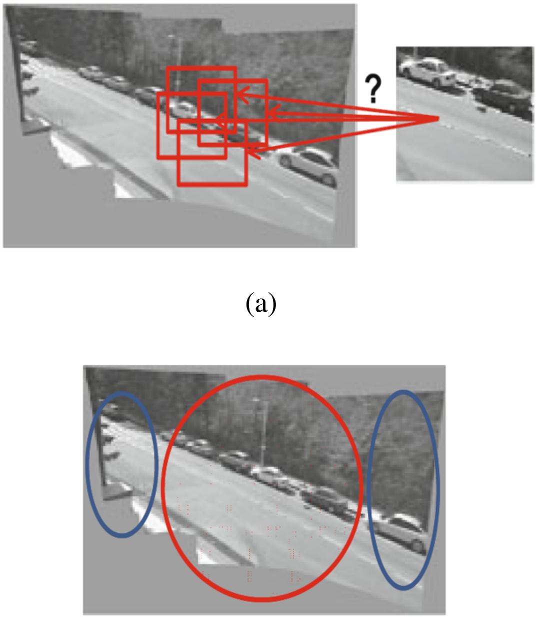
Figure 1.4: The example results of background modeling using the Gaussian mixture model (GMM) [13, 14] in the background-centric approach.

ilar color (or intensity) over time in a fixed camera, and the background model is built on this assumption. The background is abstracted from the input image, and the foreground (moving objects) region is determined by marking the pixels in which a significant difference occurs. This technique is known as background subtraction. Many background subtraction algorithms have been proposed and have achieved success in performance with low computations.

Background modeling schemes can be categorized as those which uses the moving average [15,16], the Gaussian mixture model (GMM) [17–21], kernel density estimation (KDE) [22], the codebook model [23, 24], a network approach [25–29], a low-rank representation [30], and a sample consensus approach [31–34]. Several works [35–37] have evaluated the performances of various background subtraction methods with independent benchmark datasets. Except for the low-rank representation methods that need full frames, most of the background modeling methods run in an online manner and satisfy the real-time requirement. Figure 1.4 shows the example result of the background-centric approach in a fixed camera. The background model is built as shown in Figure 1.4(b)-(c) by an unsupervised method, and the moving object region is obtained as shown in Figure 1.4(d).

However, in a moving camera situation, the assumption of background modeling is broken because the background region also moves due to the camera movement. Hence, in order to build the background model and detect moving objects via a moving camera, an additional process is required to handle the camera motion. As the first approach for the moving camera, a panorama-based approach has been tried by using a stitched panoramic image captured by the moving camera [38–43]. This approach generates a large panorama background covering the entire view of a moving camera and then subtracts the background from an input image. The advantage of this approach is that we can directly utilize the existing background subtraction methods developed for a fixed camera without modification for a moving camera.

This panorama-based approach works well with cameras whose range of view is limited, such as the PTZ camera. However, this approach cannot be applied to cameras mounted



(a)

(b)

Figure 1.5: The problems of the panorama-based approach: (a) Localization problem. (b) Side region distortion after stitching.

on high-mobility units, such as vehicles and drones, because it is impossible to generate a panorama covering the entire range of movement. Moreover, it suffers error accumulation problem caused by stitching errors and needs a considerable amount of memory. Figure 1.5 shows the problems of the panorama-based approach. It has the localization problem to find the matched position in the current view and the stitching distortion problem as shown in the blue circles of Figure 1.5(b).

Recently, to overcome the problems in the panorama-based approach, a compensation-based approach has been proposed to utilize the influence of camera motion for the online learning of the background model without generating a panorama [44–49]. Because camera motion leads to background motion, this approach finds a transformation matrix that indicates the displacements of two consecutive frames caused by the camera movement. The background models are warped to compensate for the camera motion, after which the foreground is detected by subtracting the warped background. This approach requires low computation and a small amount of memory, so it is effective when utilized for moving object detection, even with a moving camera. Because of these strengths, our method is also developed based on the compensation-based approach.

However, the simple combining of existing background subtraction methods with motion compensation is linked to two problems arising from the movement of the camera. The main problem arises from an inaccurate estimation of camera motion from the image sequences, leading to many false positives related to the compensation error. Previous works on the compensation-based approach attempted to reduce false alarms arising from incorrect estimations of camera motions. Kim *et al.* [44] proposed spatio-temporal learning which considers neighboring pixels. Yi *et al.* [45] proposed a block-based dual-mode kernel model which builds acting and standby model using average block intensity. Kim *et al.* [46] used feature clustering and a scatteredness measure to separate the foreground and the background. López-Rubio and López-Rubio [47] proposed a stochastic approximation method which interpolates full covariance matrices of the background model. Hu *et al.* [48] combined multiple object tracking in foreground regions, and Minematsu *et al.* [49] measured the performances according to estimation methods for camera motion. Although these advances can reduce the number of false positives from incorrect estimations of the motion of the camera, several issues remain unsolved. Therefore, in order to solve remaining issues on the compensation-based approach, we propose the unified framework containing the dual-modeling with attentional sampling, situation-aware background learning, and prior-based attentional update for dashcam video.

1.3 Contributions

In this thesis, we propose a new framework for detecting moving objects in a moving camera to tackle the challenges arising from the dynamic background. First, we propose a new scheme to accelerate moving object detection using the dual-mode modeling with attentional sampling that utilizes the spatio-temporal properties of moving objects. We build the foreground probability map which reflects the spatio-temporal properties, then we selectively apply the detection procedure and update the background model corresponding to the selected pixels using the foreground probability. Through this scheme, the algorithm speed is accelerated, and the false positives are reduced effectively.

Second, we propose a moving object detection algorithm that adapts to various scene changes in a moving camera. In the moving camera scene, both backgrounds and objects are moving while the level of illumination in general varies frequently. To handle these scene changes, we propose a situation-aware background learning scheme that adaptively updates the background according to how the scene changes. First, we estimate the three situation variables of background motion, foreground motion and illumination changes for an awareness of situation changes in the moving scene. We then compensate for the camera movement and update the background model in different ways according to the situation changes. Lastly, we propose a new foreground decision method with a foreground likelihood map, two thresholds, and a watershed algorithm to generate a spatially connected foreground region. This situation-aware background learning scheme enables to adapt dynamic scene changes while maintaining the performance and speed.

Third, we propose a moving object detection algorithm for a monocular dashcam mounted on a vehicle. To deal with dynamic changes of the scene from the dashcam, we propose a new scheme inspired by human-attention inclination for change detection. Humans do not build a detailed visual representation and perceive a change of the scene based on the structure of an interesting region. In this perspective, our method focuses on a sky and road region of the scene and builds an abstracted background model, which is updated with a spatially adaptive learning rate according to the center-focused tendency of the human gaze. Through the scheme for the driving situation, we have achieved successful results in background modeling and moving object detection in dachcam videos.

1.4 Contents of Thesis

In chapter 2, as for the problem statements, we explain the background-centric method that finds a moving object in a fixed camera in chapter 2.1, and mention the problems arising when the background-centric methods are applied for a moving camera. Chapter 3 addresses the dual-mode modeling with attentional sampling to make robust and fast moving object detection in a moving camera. In chapter 4, we propose a situation-aware background modeling for adapting to various scene changes in a moving camera. Chapter 5 presents a prior-based attentional update to detect moving objects in a monocular dashcam mounted on a vehicle. In chapter 6, we show the experimental results through both the qualitative and the quantitative comparisons and introduce the application with recognition algorithm. In chapter 7, we summarize the contributions of this thesis and briefly mention the future research directions. Also, we will call our target problem as “moving camera detection” in short or “MCD” as an abbreviated form. That is, the MCD on the algorithm name indicates the moving object detection in a moving camera.

Chapter 2

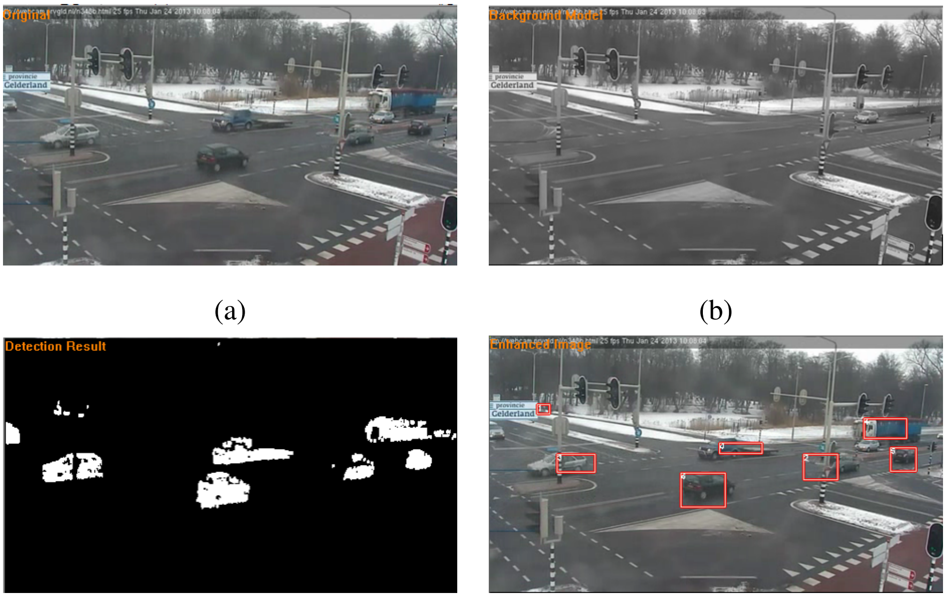
Problem Statements

2.1 Background-centric approach for a fixed camera

We introduce a surveillance scenario that monitors crossroads as shown in Figure 2.1(a). In this case, moving objects are of interest and the rest such as a stationary region is not of interest. Here, the background is defined as a stationary region and the foreground is defined as a moving region or changing regions in a scene. If an intelligent surveillance system builds a background image as shown in Figure 2.1(b), we can obtain the regions of moving objects as shown in Figure 2.1(c) by subtracting the background (Figure 2.1(b)) from the input image (Figure 2.1(a)). After we give a label to each moving region in Figure 2.1(c), we obtain final moving object regions as shown in Figure 2.1(d). Mathematically, the foreground image F is obtained by

$$F(x,y,t) = \begin{cases} 1 & \text{if } |I(x,y,t) - B(x,y,t)| > T \\ 0 & \text{otherwise,} \end{cases} \tag{2.1}$$

where $I(x,y,t)$ and $B(x,y,t)$ are intensities of input image and background on the position (x,y) at time t respectively, and T is a threshold parameter.



(a) (b)

(c) (d)

Figure 2.1: Moving object detection framework using background modeling. When an input frame is given as (a), the background image is built as (b). By subtracting the input frame (a) and the background (b), we can obtain the region containing moving objects like (c). After each moving object is labeled based on (c), we detect the moving object as (d).

Previous works have focused on how to build a background $B(x, y, t)$ automatically in a video. The background image is not fixed but should be adapted to illumination changes (both gradual and sudden), uninteresting changes (such as tree branches), and changes in background geometry (such as parked cars). Also, the background modeling methods should consider the complexity, speed, memory, and accuracy. As a simple approach, when the background $B(x, y, t)$ can be set to a mean of the previous n frames like

$$B(x, y, t) = \frac{1}{n}\sum_{i=0}^{n-1} I(x, y, t-i), \tag{2.2}$$

or a median of the previous n frames like

$$B(x, y, t) = median\{I(x, y, t-i)\}, i \in \{0, ..., n-1\}. \tag{2.3}$$

Although the background B in equation (2.2)-(2.3) can be easily implemented, it requires high memory to keep the previous frames for handling the background changes over time. To reduce the memory requirement, the mean background model can be introduced by a running average as

$$B(x, y, t) = \frac{t-1}{t}B(x, y, t-1) + \frac{1}{t}I(x, y, t), \tag{2.4}$$

or more generally,

$$B(x, y, t) = (1-\alpha)B(x, y, t-1) + \alpha I(x, y, t), \tag{2.5}$$

where α is called as a learning rate. As an initial work, Koller *et al.* [50] suggested a new update form to update only the background region using previous foreground image $F(x, y, t-1)$ as

$$B(x, y, t) = F(x, y, t-1)B(x, y, t-1) + (1-F(x, y, t-1)((1-\alpha)B(x, y, t-1) + \alpha I(x, y, t)). \tag{2.6}$$

When $B(x, y, t)$ indicates the mean background model, we will use $\mu^{(t)}$ instead of $B(x, y, t)$ for convenience in the following.

The memory issue is resolved through the running average, but the issue for threshold remains. The threshold T in equation (2.1) that decides a sensitivity of change should be different along the position. For example, the region containing waving trees should be less sensitive to change because the background color of trees itself fluctuate and so these changes are uninteresting changes. In other words, the sensitivity of the change is decided by the statistical property of background intensity. Therefore, a histogram-based method is first adopted to estimate the statistical distribution of background intensity changes over time. Wren *et al.* [51] fit one Gaussian distribution with mean and variance over the histogram. This distribution gives the probability density function (*pdf*) of a background and is updated through the running average form as

$$\mu_i{}^{(t)} = (1-\alpha)\mu_i^{(t-1)} + \alpha I_i^{(t)}, \tag{2.7}$$

$$\sigma_i{}^{2\,(t)} = (1-\alpha)\sigma_i^{2\,(t-1)} + \alpha(I_i^{(t)} - \mu_i{}^t)^T(I_i^{(t)} - \mu_i{}^t), \tag{2.8}$$

where $I_i^{(t)}$ is an intensity of input image and α is a learning rate. $\mu_i{}^{(t)}$ and $\sigma_i{}^{(t)}$ are a mean and variance of background model of i-th pixel at time t, respectively. To decide a foreground fast, the decision is simply done instead of the probability calculation from the Gaussian *pdf*, that is,

$$F_i{}^{(t)} = \begin{cases} foreground & \text{if } \frac{(I_i^{(t)} - \mu_i^{(t)})^2}{\sigma_i^{2\,(t)}} > T, \\ background & \text{otherwise.} \end{cases} \tag{2.9}$$

If $I_i^{(t)}$ is multi-dimensional features like a color, the variance $\sigma_i^{2\,(t)}$ should be changed to covariance model $\sum_i^{(t)}$. However, most works assume that each channel of input is independent for a computational efficiency, so we simplify the case as the value of $I_i^{(t)}$ is scalar as equation (2.7)-(2.8).

Then, to cope with multimodal background distributions, the background modeling using the mixture of Gaussian models (GMM) is proposed [13]. In the GMM method, the observed features are modeled by a mixture of K Gaussian kernels. For each position in an image, the background probability of an input feature (color) $\mathbf{x}$ is given by

$$P(\mathbf{x}) = \sum_{k=1}^{K} w_k \eta(\mathbf{x}, \mu_k, \Sigma_k) \tag{2.10}$$

$$\eta(\mathbf{x}, \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{n/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu_k)^T {\Sigma_k}^{-1} (\mathbf{x}-\mu_k)} \tag{2.11}$$

where K is the number of Gaussian models, and w_k is the weight of k-th Gaussian model. The k-th model $\eta(\mathbf{x}, \mu_k, \Sigma_k)$ is the Gaussian kernel with μ_k and Σ_k as a mean and a covariance matrix of k-th model, respectively. $P(\mathbf{x})$ is the weighted sum of Gaussian kernels $\eta(\mathbf{x}, \mu_k, \Sigma_k)$. For the computational efficiency, each dimension of features is assumed to be independent to each other and the covariance has a form of

$$\Sigma_k = \sigma_d^2 \mathbf{I}, \tag{2.12}$$

where d indicates each dimension of the feature. Then, by calculating the likelihood of the learned Gaussian models, we update the parameters of the Gaussian model with the highest likelihood. The Gaussian model with brings the highest likelihood is called as the matched Gaussian model. The update for matched Gaussian model is same to equation (2.7)- (2.8). The mixture weight of the k-th Gaussian is adjusted by

$$w_k \leftarrow (1 - \beta_k) w_k + \beta_k (M_k). \tag{2.13}$$

where β_k is the learning rate of the k-th weight and M_k is the matching indicator which has 1 for the matched model and 0 for remaining models. After weights are updated, the weights are normalized.

2.2 Problem statements for a moving camera

The background-centric approach in a fixed camera assumes that the background is more likely to appear, and so mean or median value of previous observations can represent the background model for each position. However, when the camera is moving, the assumption is broken, so we should modify the background model to handle a camera movement. To compensate the motion of the background, we should estimate the camera motion through the scene information from the video. If further information is given from the hardware sensor such as gyro, it can help to calculate the camera movement. In this thesis, we only use video input to estimate the camera movement. The basic idea for camera motion estimation is similar to image stitching [52]. From two consecutive images, we can find corresponding points and stitch two images. In the process of stitching, the relation between two images is represented a transformation matrix, and this matrix can be regarded as the estimated camera motion.

For the mathematical description, the input frame converted to a gray scale image is denoted by $I^{(t)}$ at time t. We calculate the velocity vector of each position based on the brightness constancy assumption that the projection of the same point looks same in every frame. It is called as the brightness constancy assumption expressed as

$$I^{(t+1)}(x+u, y+v) = I^{(t)}(x, y), \tag{2.14}$$

where (u, v) is the displacement/velocity of the position (x, y). Although the velocities of all positions can be calculated by the dense optical flow algorithm [53, 54], it is inefficient due to the large computations. Therefore, we use the Lucas-Kanade tracker [55] on the uniformly sampled positions for velocity estimation. However, these velocities are also obtained in a moving object region, so we assume that the background region is larger than the foreground region. Therefore, we find a transformation matrix to satisfy equation (2.14) as many samples as possible. The camera motion (i.e., background motion) is represented by a projective transform matrix $\mathbf{H}_{t:t-1}$ obtained by

$$[X_1^{(t)}, X_2^{(t)}, ...] = \mathbf{H}_{t:t-1}[X_1^{(t-1)}, X_2^{(t-1)}, ...], \tag{2.15}$$

where

$$X_i^{(t-1)} = (x_i, y_i, 1)^T, X_i^{(t)} = (x_i + u_i, y_i + v_i, 1)^T. \tag{2.16}$$

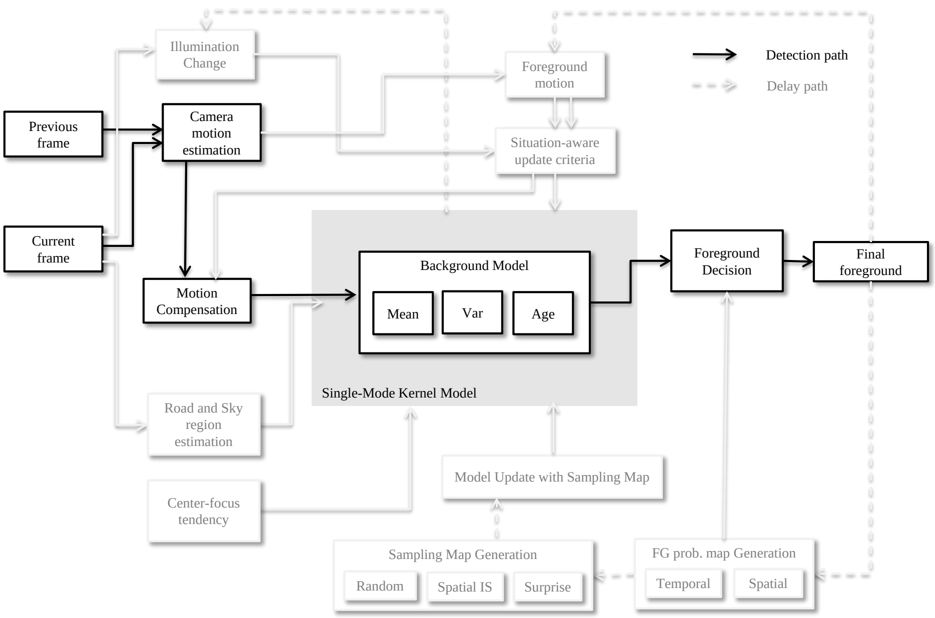


Figure 2.2: The conventional framework of the compensation-based approach. Based on the Single-Mode Kernel background modeling, the camera motion estimation and the motion compensation are newly added for a moving camera.

In solving (2.15), at least four corresponding points are required, and outliers are removed using the RANSAC [56] algorithm.

In conclusion, the $\mathbf{H}_{t:t-1}$ indicates the position mapping between $I^{(t)}$ and $I^{(t-1)}$. In case of a fixed camera, the $\mathbf{H}_{t:t-1}$ becomes the identity matrix. Using the $\mathbf{H}_{t:t-1}$, the background model at time $t-1$ is warped, and the warped background model is used in the background subtraction.

The clear parts in Figure 2.2 shows the conventional baseline method of the compensation-based approach for moving camera moving object detection while the hazy parts are the proposed schemes to cope with the problems stated in the following. In addition to the background subtraction methods for a fixed camera, the baseline method adopts the camera motion estimation and motion compensation modules to cope with the camera move-

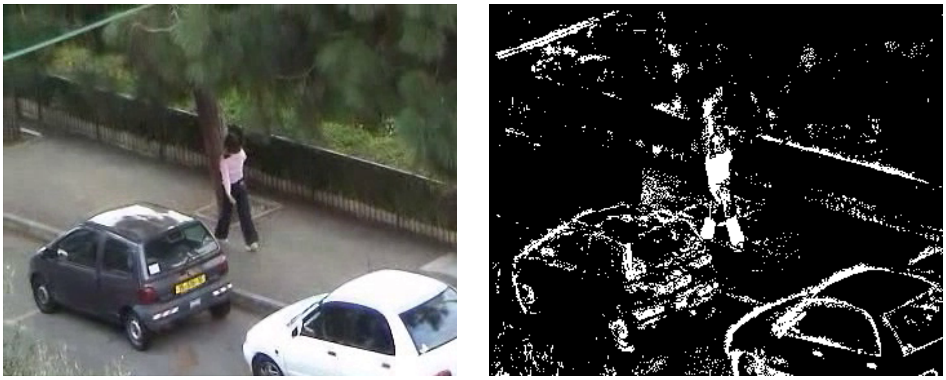


Figure 2.3: The first problem from the naive extension of background subtraction for fixed camera: false positives around edges.

ment. Through this motion compensation, most methods of background subtraction can be extended for a moving object detection in a moving camera.

However, naive extension with background subtraction for moving camera suffers four problems as shown in Figure 2.3–Figure 2.6. The first problem is many false positives because of inaccurate camera motion estimation. As shown in Figure 2.3, many false positives around edges occur. Near the region having strong edges, false positives occur if one pixel is deviated by the compensation error. The second problem is the detection quality degradation from sudden scene changes. For example, a sudden change of illumination occurs more frequently with a moving camera than with a fixed camera as shown in Figure 2.4. In this case, a large portion of the background is falsely detected as foreground during the change. This situation occurs in most digital cameras when using the auto exposure function to control the overall brightness. The third problem arises when an object moves slowly relative to the movement of the camera. The foreground motion is not distinguishable from the background motion, which causes a severe foreground loss as shown in Figure 2.5. The fourth problem is the limitation of the homography model when the video is captured by the complex camera motion such as dashcam. Since the homography motion model cannot represent the forward

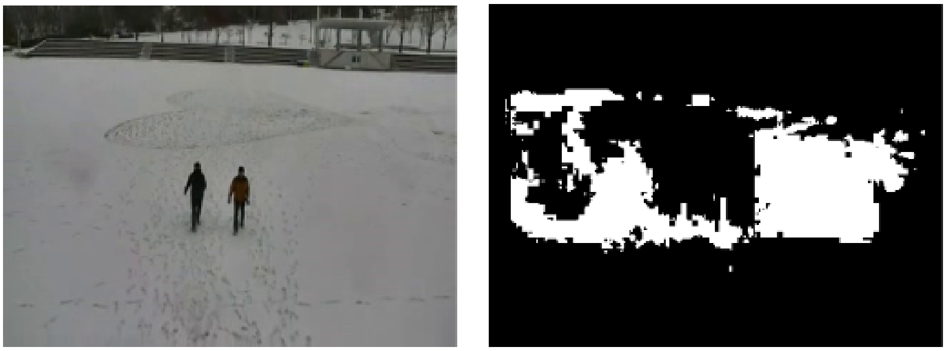


Figure 2.4: The second problem from the naive extension of background subtraction for fixed camera: sudden scene changes such as illumination.

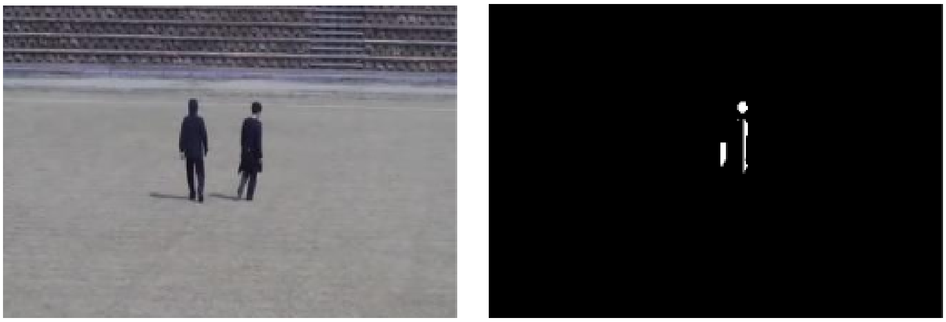


Figure 2.5: The third problem from the naive extension of background subtraction for fixed camera: slow moving object relative to camera movement.

motion well, false positives and foreground missing problems occur as shown in Figure 2.6.

To solve the aforementioned problems, we propose the integrated framework as shown in Figure 2.7. This framework contains three main parts: dual-mode modeling with attentional sampling (red boxes and arrows), situation-aware background learning (green boxes and arrows), and prior-based attentional update for dashcam video (blue boxes and arrows). The dual-mode modeling with attentional sampling is proposed to solve the false positive problem as shown in Figure 2.3 remarked as the first problem. The situation-aware background

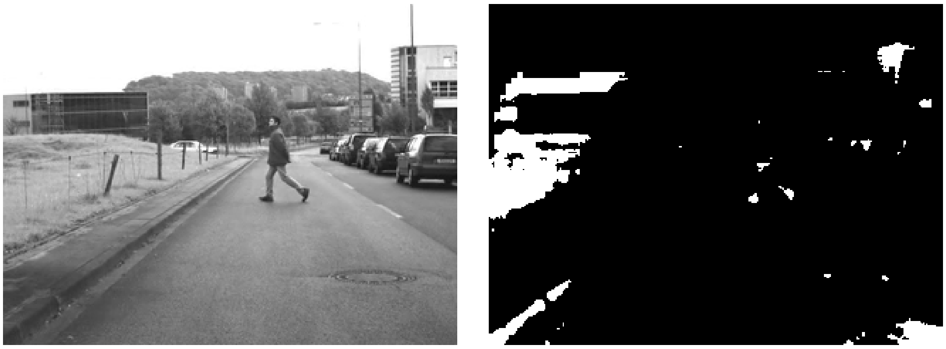


Figure 2.6: The fourth problem from the naive extension of background subtraction for fixed camera: the limitation of homography model in a dashcam.

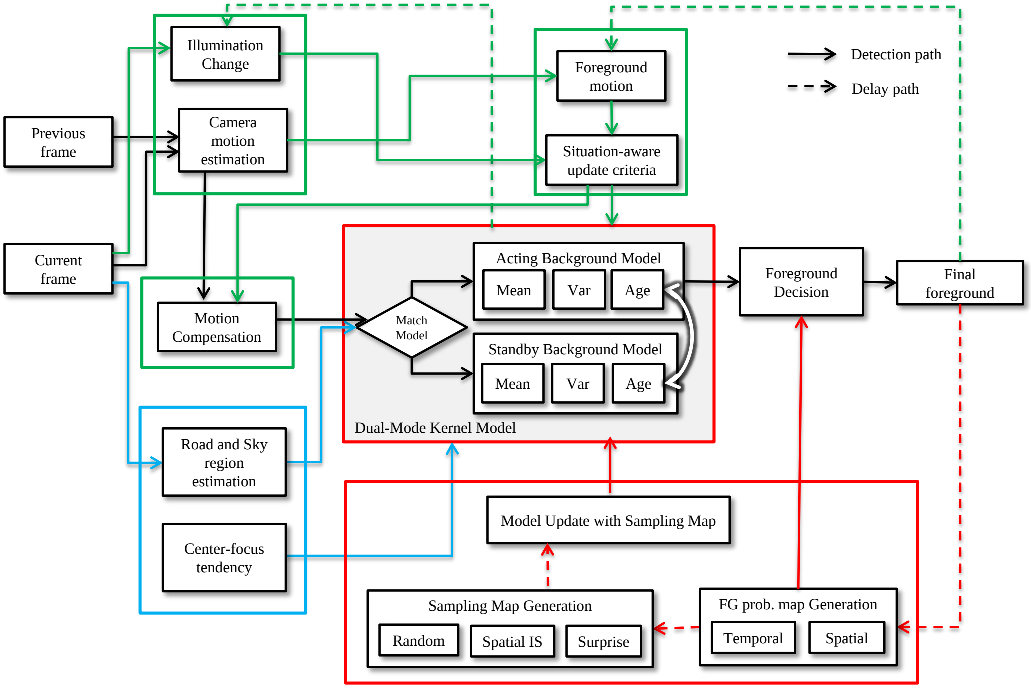


Figure 2.7: The proposed framework of the background-centric method for moving object detection in moving cameras.

learning is suggested to adapt sudden scene changes and handle the slow moving object problem as shown in Figure 2.4 and Figure 2.5. Lastly, the prior-based attentional update is proposed to cope with the problem in a dashcam as shown in Figure 2.6.

Chapter 3

Dual-mode modeling with Attentional Sampling

In the MCD problem, it is important to achieve a computational efficiency as well as detection accuracy. In terms of accuracy, the object-centric methods are recommended, but they take a few seconds per frame and sometimes produce total failures. Therefore, the background-centric approach using motion compensation, which compensates the camera movement to fit the previous model to the current image, is preferred for practical application. However, as mentioned as the first issue in the problem statements, most of the background-based algorithms use a simple camera model because of the computation issues, so many false detections occur at image boundary due to inaccurate estimation of camera movement. While the existing works [44–46] reduced many false detections and achieved real-time performances, they also lose a true object region as a side effect and still show poor performance in drastic frame changes.

In this chapter, we propose a new scheme to improve the robustness of the compensation-based method. This scheme reduces the loss of true object region and the false detections in drastic changes as well as maintaining real-time performance by adopting the grid-based modeling. Our most important insight is that we can figure out the moving objects and the compensation errors through the occurrence pattern of the foreground. While errors and noises flicker in temporal domain and are isolated in the spatial domain, the foreground region of moving objects appears coherently in the spatio-temporal domain. Therefore, our main idea is to use these spatio-temporal properties of moving object occurrence. The proposed scheme is realized by a novel sampling strategy based on the probability of the foreground occurrence using the spatio-temporal properties. From the assumption that the objects move smoothly in consecutive frames, we predict the next positions of objects. To keep the computational efficiency in the prediction, we just use the probability of foreground occurrence that the objects are likely to appear at the spatial and temporal neighbors instead of accurate velocity estimation. Through this probability of foreground occurrence, we can distinguish actual objects and false detections as well as reduce the search space to find the actual positions of objects. The overall scheme is depicted in red shading parts of Figure 3.1. We first explain the dual-mode kernel model and motion compensation step and then introduce the combined method with attentional sampling method.

3.1 Dual-mode modeling for a moving camera

Aforementioned in related works, many state-of-the-arts requires heavy computational loads where they take a few seconds to a few minutes per frame. For the compensation-based methods, they satisfied a real-time requirement but shows many errors and noises that arise from inaccurate motion estimation and compensation. This is a critical reason that we cannot just simply apply background subtraction algorithms for fixed cameras with simple motion compensation techniques. Stationary camera background modeling algorithms usually focus on building an accurate model for each pixel. But for the non-stationary case, we cannot guarantee that the model used to evaluate a pixel is relevant to that pixel. Also, in case of the moving camera, the changes in the scene are numerous by the newly appearing/disappearing region.

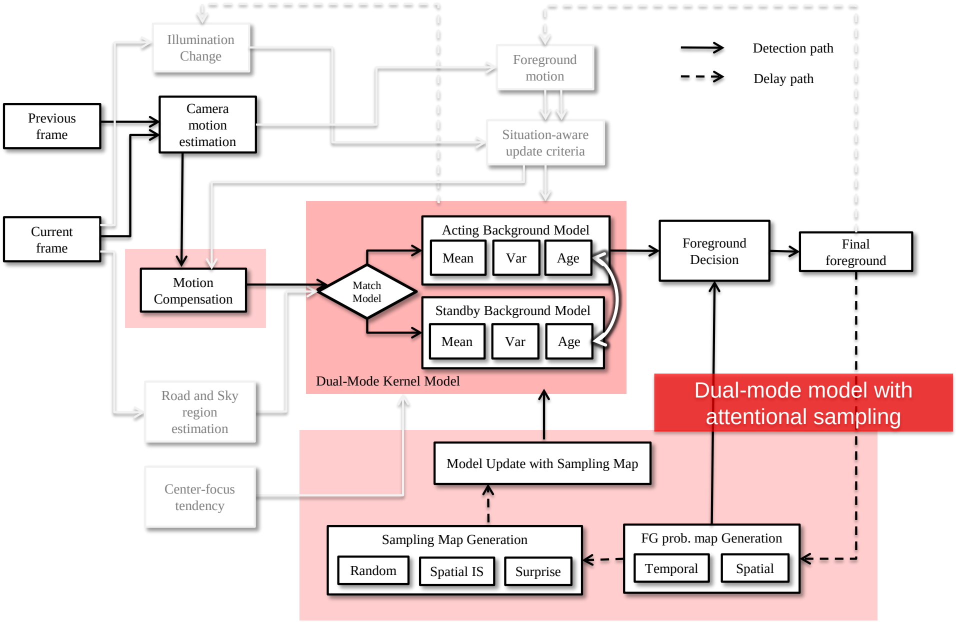


Figure 3.1: The framework for the dual-mode modeling with attentional sampling.

Therefore, we adopt the dual-mode modeling that contains the age model, the grid-based modeling, the dual-mode kernel modeling, and the motion compensation by mixing models. In the following subsections, we explain four modules for adapting the moving background in a moving camera.

3.1.1 Age model for adaptive learning rate

We introduce the adaptive learning rate called as age model. If a fixed learning rate is used, the first observed intensity of the pixel becomes an initial mean value. In case of the stationary camera, incoming observations are not different much to first observation, so the background model becomes mature under a fixed learning rate. However, in case of the moving camera, the first observation of a pixel is not similar to the mean value, and the background models in newly appearing/disappearing regions should be changed quickly. Therefore, we need a different learning rate spatially and temporally. The concept of temporally varying learning rate is similar to equation (2.4), and spatially varying rate is decided by a camera motion $\mathbf{H}_{t:t-1}$ in equation (2.15). So we define *age* of a pixel to define a variable learning rate, where the learning rate is defined as $1/(\mathit{age}+1)$. As shown in Figure 3.2, the *age* indicates how long the region appear by a camera motion, and mathematically the age $\alpha_i^{(t)}$ of i-th pixel at time t is defined as

$$\alpha_i^{(t)} = \tilde{\alpha}_i^{(t-1)} + 1 \tag{3.1}$$

where $\tilde{\alpha}_i^{(t-1)}$ is the compensated age by $\mathbf{H}_{t:t-1}$ (The compensation procedure will be described in chapter 3.1.4). Along with this adaptive learning rate, the mean and variance equation is changed as

$$\mu_i^{(t)} = \frac{\tilde{\alpha}_i^{(t-1)}}{\tilde{\alpha}_i^{(t-1)}+1}\tilde{\mu}_i^{(t-1)} + \frac{1}{\tilde{\alpha}_i^{(t-1)}+1}I_i^{(t)}, \tag{3.2}$$

$$\sigma_i^{2\,(t)} = \frac{\tilde{\alpha}_i^{(t-1)}}{\tilde{\alpha}_i^{(t-1)}+1}\tilde{\sigma}_i^{2\,(t-1)} + \frac{1}{\tilde{\alpha}_i^{(t-1)}+1}\left(\mu_i^{(t)} - I_i^{(t)}\right)^2, \tag{3.3}$$

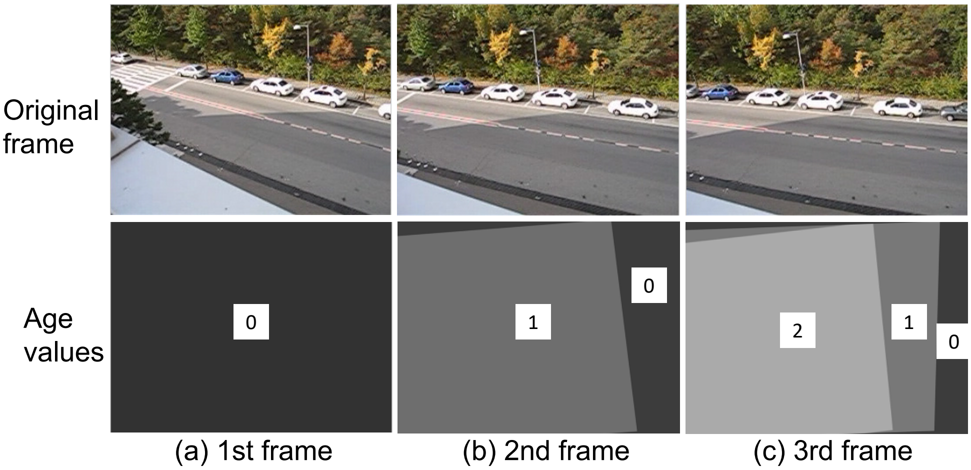


Figure 3.2: The graphical description of the age model. The initial age value is 0. In next frame, the age of newly appearing region becomes 0, and the age of the remaining region is increased by 1.

where $I_i^{(t)}$ is the image intensity of i-th pixel at time t. $\sim$ indicates that the camera motion is compensated.

3.1.2 Grid-based modeling

The homography model for camera motion estimation is very efficient, but it has an inevitable registration error caused by the parallax effect or sub-pixel accuracy. This registration error makes false positives around motionless edge pixels. As an initial try, these false positives are removed by considering neighbor pixels in the decision step. Figure 3.3 shows the process considering neighbor pixels. Originally, the pixel X_t is matched to the pixel X_b by the homography. Because there is the registration error, we locally find the best-matched pixel $\hat{X}_b$. In decision and update, $\hat{X}_b$ is used instead of X_b for the corresponding point of X_t. This procedure effectively reduces the false positives from the registration error, but the foreground loss also occurs as shown in Figure 3.4.

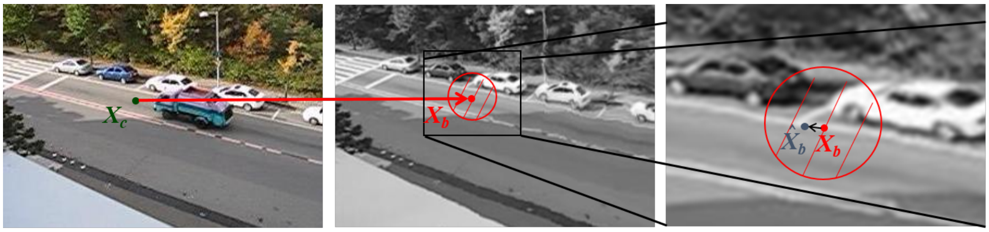
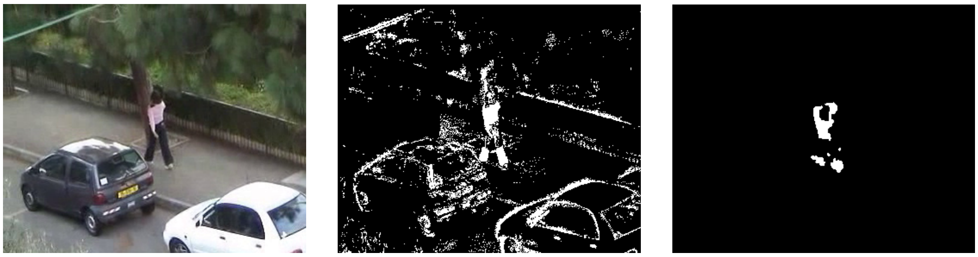


Figure 3.3: False positive removal by considering neighbor pixels in the foreground decision.



(a) (b) (c)

Figure 3.4: The effect of neighbor pixel consideration. The naive combination of motion compensation and background modeling produces many false positives as shown in (b). By considering the neighbor pixels, the false positives can be reduced, but the foreground region is also eroded as shown in (c).

Therefore, we adopt the grid-based modeling that builds a coarse and spatially coherent background model. According to the scene representation of human vision [57], human builds a volatile background that contains overall scene structures. While existing background modeling builds the pixel-wise detailed background, a compact background is enough to detect the foreground region. In other words, each grid has a background model and the pixels in the same grid shares the same background model. It is computationally efficient and robust to the registration error. First, the input image divided into a same grid of size $N \times N$. If the group of pixels in i-th grid at time t is denoted as $\mathbf{G}_i^{(t)}$, the number of pixels in $\mathbf{G}_i^{(t)}$ as $\left|\mathbf{G}_i^{(t)}\right|$, and the intensity of a j-th pixel at time t as $I_j^{(t)}$, then the mean $\mu_i^{(t)}$ is updated using the average of pixel intensities in a grid as

$$\mu_i^{(t)} = \frac{\tilde{\alpha}_i^{(t-1)}}{\tilde{\alpha}_i^{(t-1)}+1}\tilde{\mu}_i^{(t-1)} + \frac{1}{\tilde{\alpha}_i^{(t-1)}+1}M_i^{(t)}, \tag{3.4}$$

where $M_i^{(t)}$ is defined as

$$M_i^{(t)} = \frac{1}{|\mathbf{G}_i|}, \sum_{j\in\mathbf{G}_i} I_j^{(t)} \tag{3.5}$$

and $\tilde{\mu}_i^{(t-1)}$ indicates the mean model of time $t-1$ compensated for use in time t. In case of the variance, the observation of variance is approximately calculated using the maximum of the squared deviation from the grid mean as

$$\sigma_i^{2\ (t)} = \frac{\tilde{\alpha}_i^{(t-1)}}{\tilde{\alpha}_i^{(t-1)}+1}\tilde{\sigma}_i^{2\ (t-1)} + \frac{1}{\tilde{\alpha}_i^{(t-1)}+1}V_i^{(t)}, \tag{3.6}$$

where $V_i^{(t)}$ is

$$V_i^{(t)} = \max_{j\in\mathbf{G}_i}.\left(\mu_i^{(t)} - I_j^{(t)}\right)^2 \tag{3.7}$$

This grid-based model considers the spatial coherence of the pixels and makes an abstracted background model. Although this model is no longer a Gaussian model, the mean and variance model play a similar role to a Gaussian model with low computational loads.

3.1.3 Dual-mode kernel modeling

The background modeling method finds the foreground as the outliers from the trained model. While the background model becomes mature as time goes on in case of fixed camera, the background of moving camera changes rapidly for adapting different scenes using high learning rate. When high learning rates for fast adaptation are used, the data from foreground pixels can be included, and the background model is contaminated. To solve this problem, we use a dual-mode kernel model that has a pair of acting and standby model in each grid. When the observation is given, only one model of a dual-mode kernel model is selected and updated. The model selection is made by comparing the distance between the average intensity and the mean value of the model. In the update step, the selected model is updated continually, and its age is increased. Hence, the model with a relatively large age is defined as the acting model for the grid background, and the other model is defined as a standby model. However, when a temporal object appears at the grid, the other model is selected and updated to prevent a corruption due to temporal changes.

Let $\{\mu_{A,i}^{(t)}, \sigma_{A,i}^{2\,(t)}, \alpha_{A,i}^{(t)}\}$ be the mean, variance, and age for the acting model and $\{\mu_{S,i}^{(t)}, \sigma_{S,i}^{2\,(t)}, \alpha_{S,i}^{(t)}\}$ be the mean, variance, and age for the standby model of the i-th grid. Based on the squared difference between the observed mean $M_i^{(t)}$ and $\mu_{A,i}^{(t)}$, each grid is assigned to one of three categories. First, i-th grid selects the acting model if the squared difference is less than a threshold with respect to $\sigma_{A,i}^{2\,(t)}$, i.e.,

$$\left(M_i^{(t)} - \mu_{A,i}^{(t)}\right)^2 < \theta_s \sigma_{A,i}^{2\,(t)}, \tag{3.8}$$

where θ_s is a threshold parameter. In this case, we update the acting model $\{\mu_{A,i}^{(t)}, \sigma_{A,i}^{2\,(t)}, \alpha_{A,i}^{(t)}\}$ according to equation (3.4), equation (3.6), and equation (3.1). And i-th grid selects the standby model if the above condition does not hold and if the observed mean matches the standby background model,

$$\left(M_i^{(t)} - \mu_{S,i}^{(t)}\right)^2 < \theta_s \sigma_{S,i}^{2\,(t)}, \tag{3.9}$$

then we update the standby model $\{\mu_{S,i}^{(t)}, \sigma_{S,i}^{2\,(t)}, \alpha_{S,i}^{(t)}\}$ according to equation (3.4), equation (3.6), and equation (3.1). If none of the conditions hold, the standby background model is initialized with the current observation. The age value of the acting and standby model indicates the number of selection, if the standby model's age exceeds the age model's age, two models are swapped as

$$\alpha_{S,i}^{(t)} > \alpha_{A,i}^{(t)}, \tag{3.10}$$

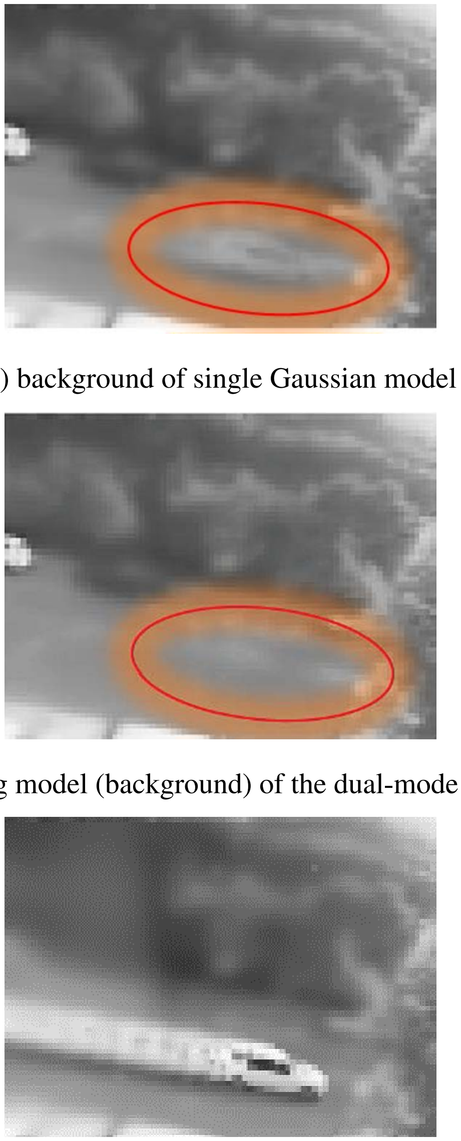
and the standby background model is initialized after swapping.

The reason why this swap procedure is used can be explained through the following example. When a car is parked on the road, the parked-car region mainly receives the car's color information and sometimes gets crossing-pedestrian information in front of the car. The model having the car's color becomes the acting model, and the model containing the pedestrian information becomes the standby model. If the parked car starts to move, new parking lot information appears, and updates the standby model for a long time, whereas the acting model stops the update. When the age of the standby model (parking lot) exceeds that of the acting model (car), the standby model becomes the acting model representing the new background (parking lot). Figure 3.5 shows the effect of the dual-mode kernel modeling. When the conventional single Gaussian model is used, the background is contaminated by the moving object as shown in Figure 3.5(a). From the dual-mode kernel modeling, the foreground and background are separated as shown in Figure 3.5(b)-(c).

In the decision step, each pixel is labeled using only the acting model as

$$l(j) = \begin{cases} foreground & \text{if } \frac{\left(I_j^{(t)} - \mu_{A,i}^{(t)}\right)^2}{\sigma_{A,i}^{2\,(t)}} > \theta_l, \\ background & \text{otherwise}, \end{cases} \tag{3.11}$$

where j is the pixel index, i is the grid index containing the pixel j, and θ_l is a thresholding parameter for the foreground decision.



(a) background of single Gaussian model

(b) acting model (background) of the dual-mode model

(c) standby model (foreground) of the dual-mode model.

Figure 3.5: The effect of the dual-mode kernel modeling. When the single Gaussian model is used, the background model is contaminated by the moving object in (a). When the dual-mode kernel model is used, the background and foreground are separated as (b) and (c).

3.1.4 Motion compensation by mixing models

From the motion compensation, the background model is warped to align the current frame. Since the corresponding location at frame t, warped by $\mathbf{H}_{t:t-1}$ from the location at frame t-1, can be a floating value, the warped background model at the frame t may not be matched at the frame $t-1$ as shown in Figure 3.6. In this case, the background can be warped by the nearest neighbor mapping, but the warping by the bilinear interpolation gives an accurate warped background. For example, we assume that the center location with the grid position $(10, 10)$ at time t is mapped to the center location with the grid position $(5.3, 4.7)$ at time $t-1$ by the $\mathbf{H}_{t:t-1}$. When the nearest neighbor warping is used, the model at grid $(5, 5)$ is used. When the bilinear interpolation is used, the background model of the position $(10, 10)$ at time t is calculated by the weighted sum of the background model of the neighboring grids $(5, 4)$, $(5, 5)$, $(6, 4)$, and $(6, 5)$ at time $t-1$.

Using equation (2.15) and the matrix inversion, the inverse mapping is obtained by

$$X_i^{(t-1)} = \mathbf{H}_{t:t-1}^{-1} X_i^{(t)}. \tag{3.12}$$

Let (x, y) be the position of the inversely mapped location at time $t-1$ from the i-th pixel at time t. The black dot in Figure 3.6 shows an example of (x, y). The set of block indices covering the black dot is defined by

$\mathbf{R}_i = \{(\lfloor x \rfloor, \lfloor y \rfloor), (\lfloor x \rfloor, \lceil y \rceil), (\lceil x \rceil, \lfloor y \rfloor), (\lceil x \rceil, \lceil y \rceil)\}$. The warped mean, variance, and age of the i-th background model at time t are obtained by the weighted sum of those of the four points in $\mathbf{R}_i$ as

$$\tilde{\mu}_i^{(t-1)} = \sum_{k \in \mathbf{R}_i} w_k {\mu_k}^{(t-1)}, \tag{3.13}$$

$$\tilde{\sigma}_i^{2\,(t-1)} = \sum_{k \in \mathbf{R}_i} w_k {\sigma_k}^{2\,(t-1)}, \tag{3.14}$$

$$\tilde{\alpha}_i^{(t-1)} = \sum_{k \in \mathbf{R}_i} w_k {\alpha_k}^{(t-1)}, \tag{3.15}$$

where the weight w_k is defined using the rectangle's area determined by the black dot and $\mathbf{R}_i$'s points as

$$\begin{aligned} w_1 &= \frac{(\lceil x \rceil - x)(\lceil y \rceil - y)}{(\lceil x \rceil - \lfloor x \rfloor)(\lceil y \rceil - \lfloor y \rfloor)}, w_2 = \frac{(x - \lfloor x \rfloor)(\lceil y \rceil - y)}{(\lceil x \rceil - \lfloor x \rfloor)(\lceil y \rceil - \lfloor y \rfloor)}, \\ w_3 &= \frac{(\lceil x \rceil - x)(y - \lfloor y \rfloor)}{(\lceil x \rceil - \lfloor x \rfloor)(\lceil y \rceil - \lfloor y \rfloor)}, w_4 = \frac{(x - \lfloor x \rfloor)(y - \lfloor y \rfloor)}{(\lceil x \rceil - \lfloor x \rfloor)(\lceil y \rceil - \lfloor y \rfloor)}. \end{aligned} \tag{3.16}$$

The variance is increased at the large gradient region because most false positives arise due to a misaligned edge from the compensation error. In other words, the warped variance $\tilde{\sigma}_i^{2\,(t-1)}$ is additionally increased by a squared deviation from the interpolated mean $\tilde{\mu}_i^{(t-1)}$ as

$$\tilde{\sigma}_i^{2\,(t-1)} = \tilde{\sigma}_i^{2\,(t-1)} + \sum_{k \in \mathbf{R}_i} w_k (\tilde{\mu}_i^{(t-1)} - {\mu_k}^{(t-1)})^2 \tag{3.17}$$

Figure 3.7 shows the effect of motion compensation by mixing models. When the nearest neighbor model is selected for a warping the background model, the warped background becomes a non-smooth background as shown in Figure 3.7(a). As a result, the foreground result has many false positives due to compensation error as shown in Figure 3.7(b). Through the motion compensation by mixing models, we can get the smooth background as shown in Figure 3.7(c) and a clear foreground result as shown in Figure 3.7(d).

3.2 Dual-mode modeling with Attentional sampling

Based on the concept of selective attention [58] for background subtraction in a stationary camera, we learn the spatio-temporal properties of objects. From the assumption that the objects appear at the neighbors of the previous detections, we build the probability map of foreground occurrence as shown in Figure 3.8(e). Then, we restrict the search space using the sampling map as shown in Figure 3.8(c) obtained from the probability occurrence, and detect the moving objects as shown in Figure 3.8(d). Lastly, we refine the object region using foreground probability as shown in Figure 3.8(f), and update the background model and the next probability of foreground occurrence. In short, our work defines the foreground

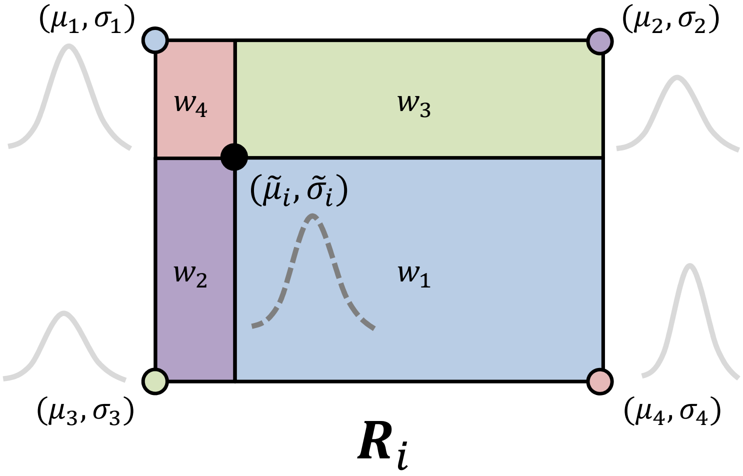
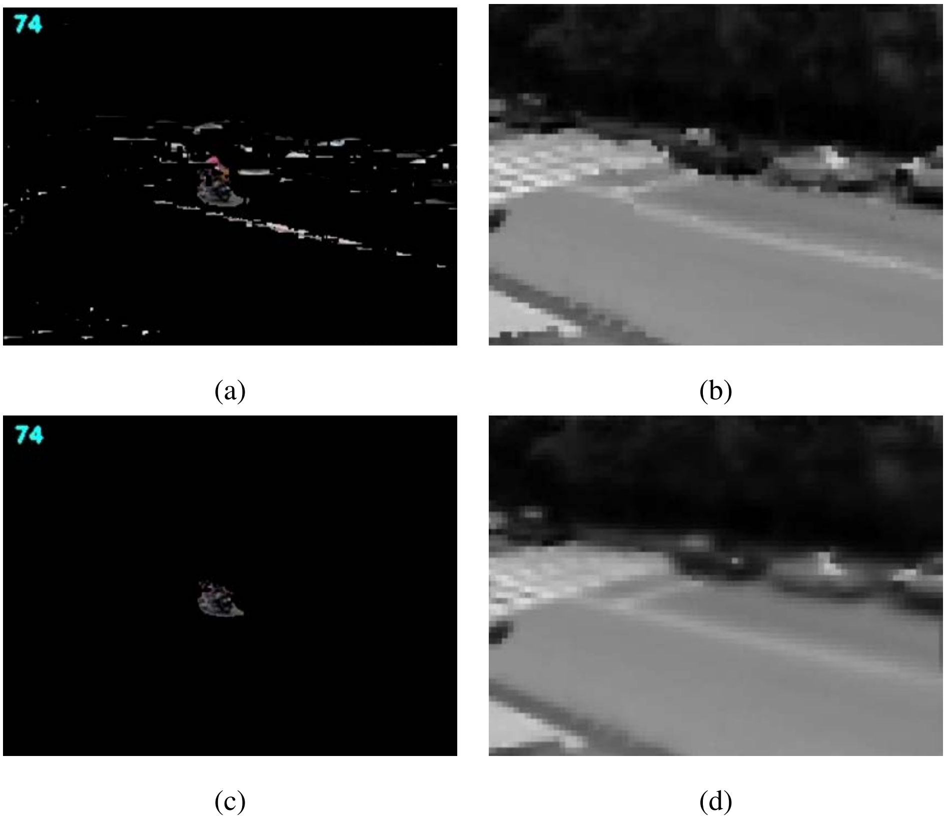


Figure 3.6: Bilinear interpolation for a warping background model. The mean and variance model of the black dot are calculated by a weighted sum of the neighbor models of quantized four points. Each weight is proportional to the rectangle area as a bilinear interpolation.

probability based on the occurrence frequency, and utilizes this probability to reduce false positives and speed up the algorithm through the attentional sampling method.

3.2.1 Foreground probability map based on occurrence

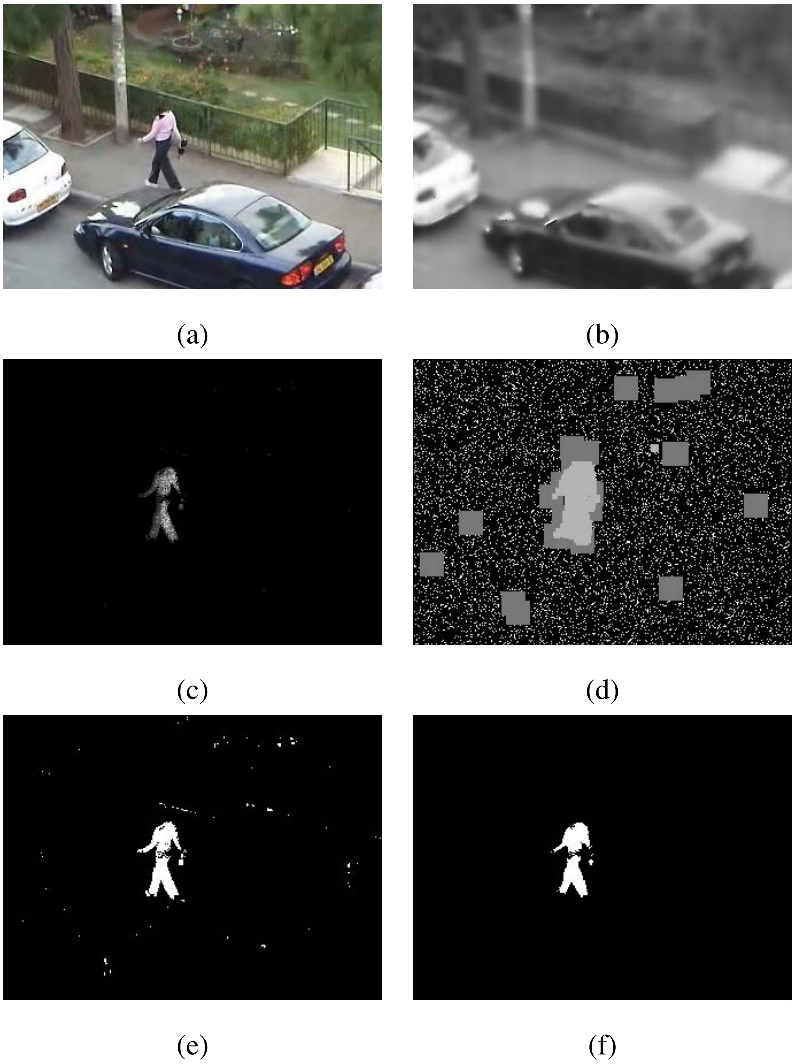
To build a foreground probability map, our assumption is that object movements are smooth spatially and temporally. According to [58], the foreground pixels tend to have three properties: temporal, spatial, and frequency properties. The temporal property means that a pixel is more likely to a foreground if that pixel has been a foreground at the previous time. The spatial property means that a pixel is highly probable to being a foreground if the neighbor pixels are a foreground. The frequency property means that if a pixel label is changed too frequently, this pixel is more like to a background. This frequency property is used to remove the inconsistent pixels which are changing periodically However, in a moving camera, the periodic noise issue is not critical, and this frequency property is redundant compared to other



(a) (b)

(c) (d)

Figure 3.7: The effect of the motion compensation by mixing models. When the nearest neighbor match is used for a warping background model, we obtain the inaccurate background model as (a) and the noisy foreground as (b). When the bilinear interpolation is used and additionally variance is increased, we get better background model as (c) and clean foreground as (d).

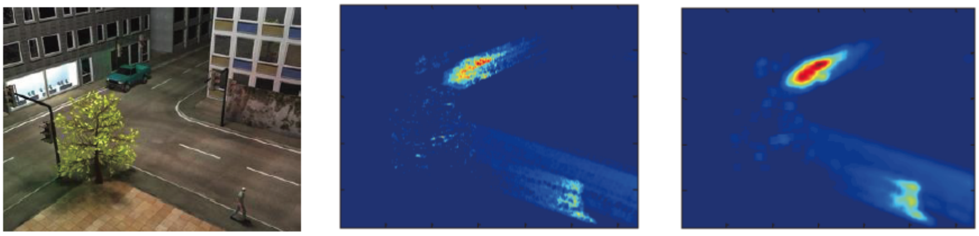


(a) (b)

(c) (d)

(e) (f)

Figure 3.8: Example images in the proposed procedure for moving object detection. The background model (b) is selectively updated by using the sampling map (d) which is determined by considering the foreground probability map (c). The foreground probability map is estimated from the previous detection results. The current initial foreground (e) is obtained by using the previous background model and sampling map. The final foreground (f) is fine-tuned by the foreground probability map.



(a) (b) (c)

Figure 3.9: The illustration of the foreground properties. When a moving car and pedestrian appear as (a), the temporal property map M_T is obtained as (b) and the spatial property map M_S is obtained as (c).

two properties. Therefore, we adopt temporal and spatial properties among three properties to express our assumption of moving objects.

Temporal property M_T is defined as a recent history of the foreground at each pixel position as

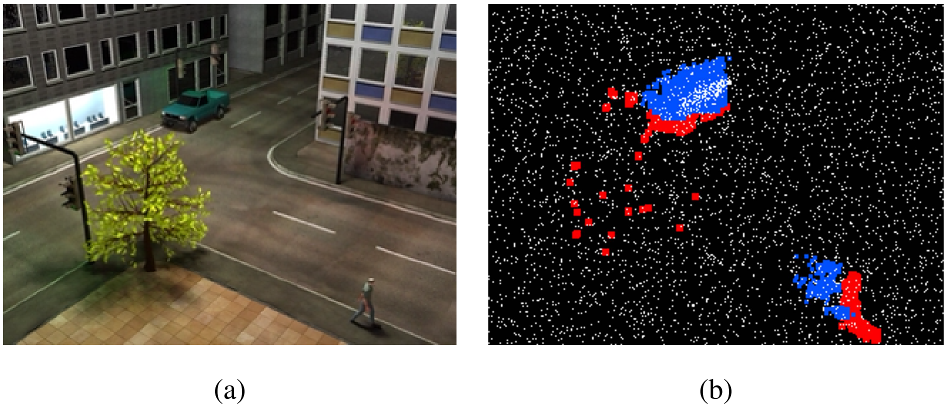
$$M_T^t(n) = (1 - \alpha_T) M_T^{t-1}(n) + \alpha_T D^t(n), \tag{3.18}$$

where t is time index and α_T is temporal learning rate. $D^t(n)$ is binary detection map which means that $D^t(n) = 1$ if pixel n belongs to foreground and $D^t(n) = 0$ if pixel n belongs to background at time t. As shown in Figure 3.9(b), the moving object region has a high value by accumulating the foreground occurrence through equation (3.18).

The spatial property measures the coherency of nearby pixels of the foreground as

$$M_S^t(n) = (1 - \alpha_S) M_S^{t-1}(n) + \alpha_S \frac{1}{w^2} \sum_{i \in N(n)} D^t(i), \tag{3.19}$$

where α_S is spatial learning rate, $N(n)$ denotes a spatial neighborhood around pixel n, and w^2 is the area of the neighborhood. As shown in Figure 3.9(c), the neighborhood of moving object region has a high value in the spatial property map through equation (3.19). Then, the foreground probability $P_{FG}^t(n)$ is defined as multiplication of temporal and spatial proper-



(a) (b)

Figure 3.10: The illustration of three sampling mask. When an input image is given as shown in (a), sampling mask is obtained like (b) where white region indicates the M^t_{RS}, blue region indicates M^t_{SEI}, and red region indicates M^t_{SP}.

ties, *i.e.*,

$$P^t_{FG}(n) = M^t_T(n) \times M^t_S(n). \tag{3.20}$$

3.2.2 Sampling Map Generation

Because we learn the temporal and spatial properties of the foreground, the additional computational loads are inevitable. To keep the efficiency even in the additional loads, we try to restrict the search space based on the foreground probability without loss of detection performance. According to the attentional sampling [58], we extract the candidate pixel positions to run the background subtraction and model update. The candidate pixel positions are obtained by combining three sampling masks via a pixel-wise 'OR ($\oplus$)' operation as

$$M^t = M^t_{RS} \oplus M^t_{SEI} \oplus M^t_{SP}, \tag{3.21}$$

where M^t_{RS}, M^t_{SEI}, and M^t_{SP} are sampling masks from randomly scattered sampling, spatially expanding importance sampling, and surprise pixel sampling, respectively.

The randomly scattered sampling means that about 5% of entire pixels are extracted and tested. The resulting mask from the randomly scattered sampling looks like a salt and pepper noise as shown in the white region of Figure 3.10(b). Obviously, since only this random sampling mask is not enough to detect a foreground due to its sparsity, the spatially expanding importance sampling is introduced. The basic idea for the spatially expanding importance sampling is that if a sampled position has a high foreground probability, we also extract the neighbor pixels of a sampled position. Also, the neighborhood area is proportional to the foreground probability. Mathematically, for each random sampled pixel (i.e., $M^t_{RS}(i) = 1$),

$$M^t_{SEI}(j) = \begin{cases} 1 & \text{if } j \in N(i), \\ 0 & \text{otherwise,} \end{cases} \tag{3.22}$$

where $N(i)$ is a rectangular region centered at pixel i with size $\varsigma^t(i) \times \varsigma^t(i)$. This $\varsigma^t(i)$ is determined using the probability of foreground occurrence in equation (3.20) as

$$\varsigma^t(i) = round(P^t_{FG}(i) \times w_e), \tag{3.23}$$

where w_e is an expanding parameter. As a result, the blue region in Figure 3.10(b) is obtained through the spatially expanding importance sampling.

Although these two sampling methods reduce the search space efficiently, it is a highly probable to miss a newly appearing object. If new moving object appears in a scene, this region should be attended in a short time. Therefore, the neighbor of the newly detected region are sampled to catch the newly appearing foreground, and it is called as the surprise pixel sampling. The surprise pixels are decided when a pixel belongs to a foreground among the randomly sampled pixels with low probability of foreground occurrence as

$$\xi^{(t)}(i) = \begin{cases} 1 & \text{if } (l^t(i) = foreground) \& (M^t_{RS}(i) = 1) \& (P^{t-1}_{FG}(i) < \theta_{sp}), \\ 0 & \text{otherwise,} \end{cases} \tag{3.24}$$

For each surprise pixel i (i.e., $\xi^{(t)}(i) = 1$), we widen the sampling area to cover neighboring pixels as

$$M_{SP}^t(j) = \begin{cases} 1 & \text{if } j \in N(i), \\ 0 & \text{otherwise}, \end{cases} \tag{3.25}$$

where $N(i)$ is a rectangular region centered at pixel i with size $w_s \times w_s$. Then, we widen the sampling area around the surprise pixel as shown in the red region of Figure 3.10(c).

3.2.3 Model update with sampling map

We use the grid-based modeling and the dual-model kernel method as a baseline, and modify the updating part by utilizing the sampling map. The mean $\mu_i^{(t)}$ and variance $\sigma_i^{2\,(t)}$ of a grid i at time t are updated by the weight sum of previous model $\{\mu_i^{(t-1)}, \sigma_i^{2\,(t-1)}\}$ and current observation $\{M_i^{(t)}, V_i^{(t)}\}$ as

$$\mu_i^{(t)} = \frac{\tilde{\alpha}_i^{(t-1)}}{\tilde{\alpha}_i^{(t-1)} + 1}\tilde{\mu}_i^{(t-1)} + \frac{1}{\tilde{\alpha}_i^{(t-1)} + 1}M_i^{(t)}, \tag{3.26}$$

$$\sigma_i^{2\,(t)} = \frac{\tilde{\alpha}_i^{(t-1)}}{\tilde{\alpha}_i^{(t-1)} + 1}\tilde{\sigma}_i^{2\,(t-1)} + \frac{1}{\tilde{\alpha}_i^{(t-1)} + 1}V_i^{(t)}, \tag{3.27}$$

where $1/(\tilde{\alpha}_i^{(t-1)} + 1)$ is time-varying learning rate at time $t-1$ from the age model.

In our scheme, background subtraction is applied to only a small portion selected by the sampling map. That is, we modify the updating rules and the observation $\{M_i^{(t)}, V_i^{(t)}\}$ to use only sampled pixels. When a grid contains selected pixels, the mean and variance observation of the model on the corresponding to the grid, $M_i^{(t)}$ and $V_i^{(t)}$ are calculated as

$$M_i^{(t)} = \frac{1}{|\mathbf{G_s}(i)|}\sum_{j \in \mathbf{G_s}(i)} I_j^{(t)}, \tag{3.28}$$

$$V_i^{(t)} = \max_{j \in \mathbf{G_s}(i)} (\mu_i^{(t)} - I_j^{(t)})^2 \tag{3.29}$$

where i, j, $I^{(t)}$ denote grid index, pixel index, and intensity map of image at time t respectively, whereas $\mathbf{G_s}(i)$ denotes the group of selected pixels in the i-th grid. In other words, we calculate the mean and variance observations by using only the selected pixels in a grid.

On the other hands, when a grid does not contain any selected pixels, we keep the mean unchanged and initialize the variance to a high value. If the camera is static, we can just keep the previous model, but, in case of a non-stationary camera, we get many false detections when the previous models are kept. Because pixel intensity changes drastically in a non-stationary camera due to rapid illumination change, we initialize the variance to a high value for a fast model adaptation.

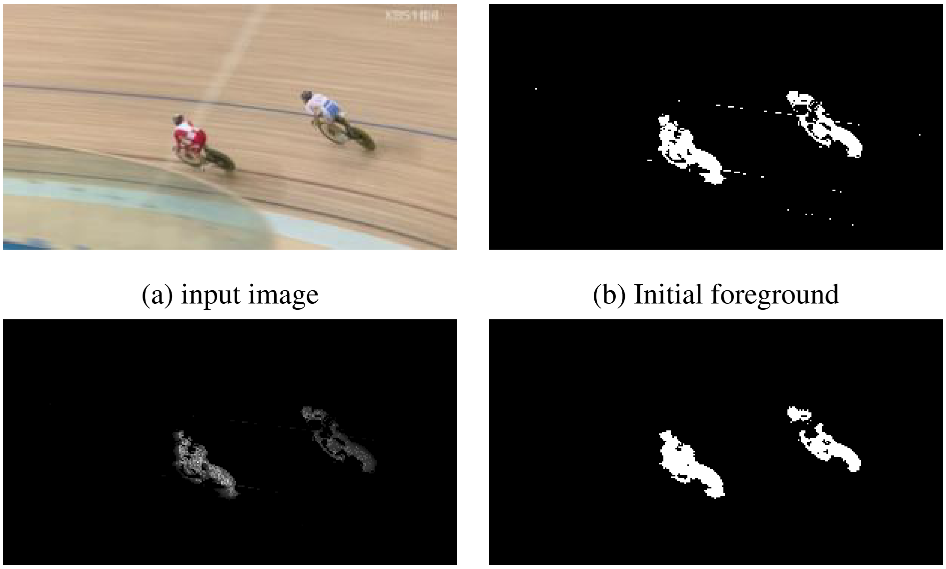
In conclusion, the sampling method reduces the computation time and also reduces transient false positives from a compensation error. In addition, for this combination, the pixel-based model is not suitable, but the grid-based method is suitable. Because the grid-based model already tries to build a rough background model, observations from only sampled pixels are enough to build a background model. However, in case of the pixel-based model, many not updated regions are generated from the sampling. Although the naive solution such as keeping the previous background is valid in a stationary camera, this solution is not effective in a moving camera due to the severe background changes in a moving camera.

3.2.4 Probabilistic Foreground Decision

The initial foreground region is decided by comparing current observation and acting model $\{\mu_{A,i}^{(t)}, \sigma_{A,i}^{2\,(t)}\}$ in a dual-mode kernel model as

$$l_{init}(j) = \begin{cases} 1 & \text{if } (I_j^{(t)} - \mu_{A,i}^{(t)})^2 > \theta_l \sigma_{A,i}^{2\,(t)}, \\ 0 & \text{otherwise,} \end{cases} \tag{3.30}$$

where j is the pixel index, i is the grid index containing the pixel j, and θ_l is a thresholding parameter for the foreground decision. When the foreground decision relies on only the background, many false detections occur due to illumination change and inaccurate estimation of camera movement as shown in Figure 3.8(e). However, we can refine the foreground using foreground occurrence probability in section 3.2.1. First, we multiply the foreground probability map to the initial foreground obtained by the background subtraction. We can de-



(a) input image

(b) Initial foreground

(c) Foreground occurrence probability

(d) Final foreground

Figure 3.11: Effect of probabilistic foreground decision.

termine the detection map by a simple thresholding method to the multiplied map. However, in this case, foreground regions include inner holes and noisy detection regions. To cope with this problem, we use the watershed algorithm [59] which effectively segments the foreground regions. We cut the foreground probability map to a high threshold, and then apply the watershed algorithm with the seed points remaining after thresholding. This refinement reduces false detections and fills the foreground clearly with low computation.

3.3 Benefits

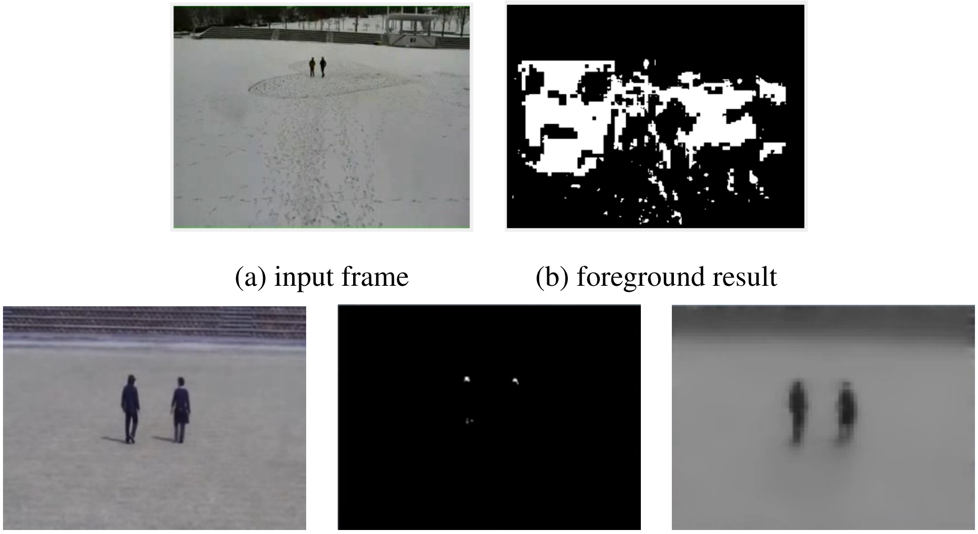
By adopting the spatio-temporal properties of moving objects, the dual-mode model with attentional sampling reduces the loss of true objects and the false detections, We build a foreground probability map and generate a sampling map which selects the candidate pixels to find the actual objects. Then, by applying the background subtraction and model update to only the selected pixels, the algorithm speed is accelerated, and the false positives are reduced. In the experiment chapter, it is verified that the proposed scheme can solve the raised issues and outperforms the state-of-the-art methods in the detection quality and speed.

Chapter 4

Situation-aware Background Learning

In this chapter, we focus on the developing of a moving object detection algorithm adapting to various scene changes in a moving camera. The compensation-based method with camera motion compensation is linked to three problems arising from the movement of the camera. The first problem arises from an inaccurate estimation of camera motion from the image sequences, leading to many false positives related to the compensation error. The second problem arises when an object moves slowly relative to the movement of the camera. In this situation, the foreground motion is not distinguishable from the background motion, which causes a contaminated background model. This contaminated background model results in severe problems, such as foreground loss. Lastly, because a sudden change of illumination occurs more frequently with a moving camera than with a fixed camera, a large portion of the background is falsely detected as foreground during the change. This situation occurs in most digital cameras when using the auto exposure function to control the overall brightness.

Previous works on the compensation-based approach [44–48] mainly focused on the first problem and attempted to reduce false alarms arising from incorrect estimations of camera motions. Although these advances can reduce the number of false positives from incorrect



(a) input frame (b) foreground result

(c) input frame (d) foreground result (e) background mean

Figure 4.1: The failure examples of previous works that do not consider the scene situation. The first row shows that the existing method has a weakness in the illumination change. The second row shows the missing problem when the object moves slowly relative to the movement of the camera.

estimations of the motion of the camera, they overlook the remaining two problems, i.e., foreground loss caused by a slow foreground and false positives caused by illumination changes. Figure 4.1 shows the failure examples of previous works. The first row of Figure 4.1 shows that the existing method has a weakness in the illumination change. The second row of Figure 4.1 shows the missing problem when the object moves slowly relative to the movement of the camera. Because the foreground motion is not distinguishable from the background motion, the foreground cannot be detected as Figure 4.1(d) and the background is contaminated as Figure 4.1(e).

In this chapter, to solve all of three problems at the same time, we propose a situation-aware background learning that updates the background model adaptively depending on the situation in the video scenes. To be aware of situations, we define three situation variables: background motion, foreground motion, and illumination changes. The situation variables are estimated at each frame and utilized for the situation-aware warping and updating of the background model. When warping the background model, we compensate for the background model with the background motion. Depending on the situation variables, the variance model of the background is additionally modified to reduce false positives (the details are described in section 4.2.1). In the updating of the background model, the mean model is additionally adjusted using the illumination change variable, and the background model is updated differently depending on the foreground motion variable. With the new background model, we calculate a foreground likelihood map based on the Gaussian distribution with the mean and variance of the background model. By thresholding the foreground likelihood map with a high threshold and a low threshold, we obtain an initial foreground region and expand the initial foreground region using the watershed segmentation method. To evaluate the performance of the proposed method, we test the robustness of the method using ten videos with various scene situations. Our method qualitatively and quantitatively outperforms the state-of-the-art algorithms in this test.

Figure 4.2 depicts the overall scheme of the proposed method. From a video captured

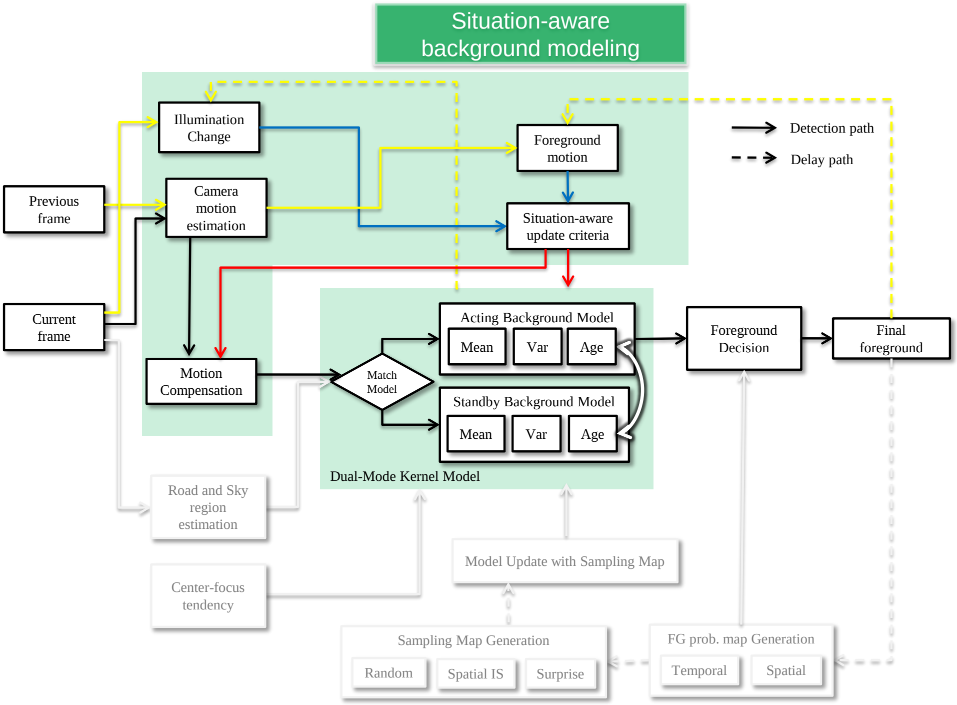


Figure 4.2: The overall scheme of the proposed method. First, we estimate the situation variables: background motion, foreground motion, and illumination change (yellow lines). Then, we adaptively update the moving background model differently (red lines) depending on situations determined by the situation variables (blue lines). Finally, we decide the foreground region through the post processing based on the foreground likelihood map.

by a moving camera, we estimate the situation variables to indicate the scene status. We measure three properties of the scene: background motion caused by the camera motion, foreground motion in the scenes, and illumination change between consecutive frames. We propose the situation-aware background learning method which adaptively updates the background model according to the situation. To compensate for the background motion, the background model is warped to align itself with the current frame using the background motion and foreground motion. In the background update, the warped background model is adaptively updated by the input image based on the foreground motion and illumination changes. The new input frame is compared to the updated background model, and then the foreground likelihood map is calculated. We find an initial foreground region by thresholding the foreground likelihood map, and generate the final foreground region by connecting the initial foreground region and the foreground candidate region through the watershed segmentation [60] method.

4.1 Situation Variable Estimation

In our situation-aware background learning method, the situation is determined by the three situation variables. In the following, we describe how to estimate the situation variables.

4.1.1 Background Motion Estimation

Using the assumption that the background region is larger than the foreground region, we can estimate the background motion using the velocities of local feature points except for outliers in velocities. Although recent methods using multiple motion models have been proposed to estimate the background motion [61–63], the computational complexity is heavy, and the estimation becomes inaccurate in a texture-less region. For fast computation, in this paper, we use a single projective model to represent the background motion.

First, the input frame is converted to a grayscale image which is denoted by $I^{(t)}$ at time

t. The corresponding locations between $I^{(t-1)}$ and $I^{(t)}$ are found by using the KLT [55] algorithm. For computational efficiency, we sparsely sample the center pixels at 10×10 grids. Letting (x_i, y_i) be the i-th grid center point, its velocity (u_i, v_i) is calculated by

$$I^{(t)}(x_i + u_i, y_i + v_i) = I^{(t-1)}(x_i, y_i), \tag{4.1}$$

under the assumption that the intensity does not change between the consecutive frames. Then, the background motion is represented by a projective transform matrix $\mathbf{H}_{t:t-1}$ obtained by

$$[X_1^{(t)}, X_2^{(t)}, ...] = \mathbf{H}_{t:t-1}[X_1^{(t-1)}, X_2^{(t-1)}, ...], \tag{4.2}$$

where

$$X_i^{(t-1)} = (x_i, y_i, 1)^T, X_i^{(t)} = (x_i + u_i, y_i + v_i, 1)^T. \tag{4.3}$$

In solving equation (4.2), outliers are removed using the RANSAC [56] algorithm.

4.1.2 Foreground Motion Estimation

To estimate the foreground motion variable without additional computation, we use $\mathbf{H}_{t:t-1}$ obtained in (4.2). The foreground regions are not fitted with the background motion model, so the i-th grid point which does not satisfy $X_i^{(t)} = \mathbf{H}_{t:t-1}X_i^{(t-1)}$ becomes the foreground pixel. The i-th grid velocity $(\hat{u}_i, \hat{v}_i)$ relative to the background is obtained by subtracting the warped image from the current image as

$$(\hat{u}_i, \hat{v}_i, 1) = X_i^{(t)} - \mathbf{H}_{t:t-1}X_i^{(t-1)}. \tag{4.4}$$

When the i-th grid point is background, the velocity $(\hat{u}_i, \hat{v}_i)$ becomes 0 from equation (4.2), otherwise $(\hat{u}_i, \hat{v}_i)$ is the foreground velocity. Figure 4.3(b) shows an example of a foreground velocity from the input frame in Figure 4.3(a). The color indicates the moving direction along a color circle in Figure 4.3(b). As shown in Figure 4.3(b), the foreground velocity

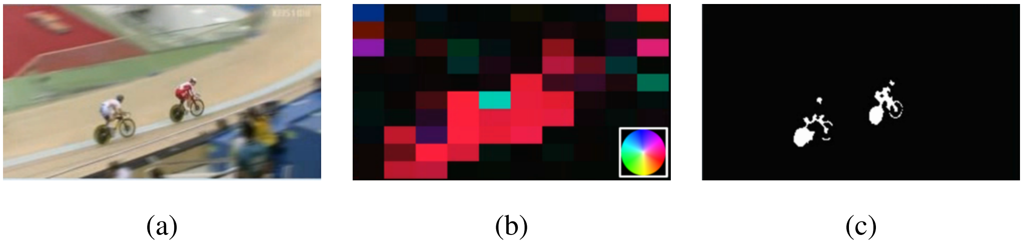


Figure 4.3: Foreground velocity estimation. (a) Input image. (b) Foreground velocity map. Color indicates the direction of foreground velocity according to the right-bottom color circle. (c) Foreground region in the previous frame.

map includes noises arising from the grid-based rough estimation. To remove the influence of noisy regions, we mask the foreground velocity map using the previous foreground map $FG^{(t-1)}$. That is, the average foreground speed $s^{(t)}$ at time t is estimated by

$$s^{(t)} = \frac{1}{P} \sum_{p \in FG^{(t-1)}} \sqrt{\hat{u}_p^2 + \hat{v}_p^2}. \tag{4.5}$$

4.1.3 Illumination Change Estimation

The basic idea to estimate illumination change is to measure the difference between the mean intensity of the background model and the average intensity of the current frame. That is, the illumination change $b^{(t)}$ is obtained by

$$b^{(t)} = \frac{1}{N} \sum_{j=1}^{N} I_j^{(t)} - \frac{1}{M} \sum_{i=1}^{M} \tilde{\mu}_i^{(t)}, \tag{4.6}$$

where N is the number of pixels, and M is the number of grids. This $b^{(t)}$ is used in updating the background model in section 4.2.2.

4.2 Situation-Aware Background Learning

As mentioned in the introduction, the critical situation is the case in which the target object (foreground) is moving slowly relative to the camera movement (background motion). In our grid-wise modeling, the target moving less than a grid size during a frame period is not distinguishable from the background grid. In this situation, the grid-wise learning yields a contaminated background model. To avoid this contamination, a scheme is required to be aware of this situation and to stop the updating of the background model. It can be a situation-aware scheme to check whether the foreground speed $s(t)$ in equation (4.5) is less than the grid size B or not. However, if the foreground always moves less slowly than the grid size B per each frame, the background model may not be updated for a long time, which leads to a performance degradation from under-learning of the background. To solve this problem, we define a count variable $c^{(t)}$ which increases by one whenever the situation occurs. The final situation-aware scheme is to check whether $c^{(t)} \cdot s^{(t)}$ is less than grid size B. The count variable $c^{(t)}$ is initialized by one. When $c^{(t)} \cdot s^{(t)}$ is smaller than the grid size B, $c^{(t+1)}$ is increased by one; otherwise, $c^{(t+1)}$ is reinitialized, i.e.,

$$c^{(t+1)} = \begin{cases} c^{(t)} + 1 & \text{if } c^{(t)} \cdot s^{(t)} < B, \\ 1 & \text{otherwise.} \end{cases} \tag{4.7}$$

As to be described in the following, the warping of the previous background model and the update of the current background model will be performed in different ways depending on the value of situation-aware variable $c^{(t)} \cdot s^{(t)}$.

4.2.1 Situation-Aware Warping of the Background Model

From section 4.1, we obtain the background motion $\mathbf{H}_{t:t-1}$ which gives the location mapping between frame $t-1$ and frame t. The mean $\mu^{(t-1)}$ and variance $\sigma^{(t-1)}$ of the background model are warped using $\mathbf{H}_{t:t-1}$. Since the warped location corresponding to each grid can be a floating value, the warped background model of a grid at the frame t may not be matched to a grid at the frame $t-1$. Therefore, we use the bilinear interpolation to warp the background model. The warped mean and variance of the i-th background model at time t are obtained by the weighted sum of those of the nearest four points as

$$\tilde{\mu}_i^{(t-1)} = \sum_{k \in \mathbf{R}_i} w_k {\mu_k}^{(t-1)}, \tag{4.8}$$

$$\tilde{\sigma}_i^{2\,(t-1)} = \sum_{k \in \mathbf{R}_i} w_k {\sigma_k}^{2\,(t-1)}, \tag{4.9}$$

where the $\mathbf{R}_i$ indicates the set of nearest four points after warping and weight w_k is the coefficient of bilinear interpolation.

Most false positives arise at the region having a large gradient due to a misaligned edge from the compensation error. We can reduce these false positives by increasing the variance at the large gradient region for dulling the background probability distribution (see Section 4.3 for details). Since the foreground region also has a large gradient, we increase the variance only when the background is not contaminated by the foreground. In other words, the warped variance $\tilde{\sigma}_i^{(t-1)}$ in equation (4.9) is additionally modified depending on the situation-aware variable $c^{(t)} \cdot s^{(t)}$. When the situation-aware variable $c^{(t)} \cdot s^{(t)}$ is smaller than the grid size B, we maintain the variance to prevent foreground loss. When $c^{(t)} \cdot s^{(t)} > B$, the variance is increased as much as the weighted sum of squared differences between the warped mean $\tilde{\mu}_i^{(t-1)}$ and the means of its neighboring models, i.e., ${\mu_k}^{(t-1)}$, $k \in \mathbf{R}_i$. That is, the final variance is obtained by

$$\tilde{\sigma}_i^{2\,(t-1)} = \begin{cases} \tilde{\sigma}_i^{2\,(t-1)} & \text{if } c^{(t)} \cdot s^{(t)} < B, \\ \tilde{\sigma}_i^{2\,(t-1)} + \sum\limits_{k \in \mathbf{R}_i} w_k (\tilde{\mu}_i^{(t-1)} - {\mu_k}^{(t-1)})^2 & \text{otherwise.} \end{cases} \tag{4.10}$$

4.2.2 Situation-Aware Update of the Background Model

The mean and variance of the background model are updated using the situation-aware variable $c^{(t)} \cdot s^{(t)}$, the warped mean and variance of the $(t-1)$-th background model, the intensity of the t-th frame image, and the illumination change $b^{(t)}$. Therefore, when the situation-aware variable $c^{(t)} \cdot s^{(t)}$ is smaller than the grid size B, the current background model adopts the warped mean and variance of the previous background model until $c^{(t)} \cdot s^{(t)}$ becomes larger than B. To adapt to the illumination change, the mean model is additionally adjusted by $b^{(t)}$ as given in equation (4.11). When the situation-aware variable becomes larger than the grid size B, the mean and variance are updated using the new intensity information of the current frame. This situation-aware update formula is given by

$$\mu_i{}^{(t)} = \begin{cases} \tilde{\mu}_i^{(t-1)} + b^{(t)} & \text{if } c^{(t)} \cdot s^{(t)} < B, \\ \frac{\alpha_i^{(t-1)}}{\alpha_i^{(t-1)}+1}(\tilde{\mu}_i^{(t-1)} + b^{(t)}) + \frac{1}{\alpha_i^{(t-1)}+1} M_i^{(t)} & \text{otherwise,} \end{cases} \tag{4.11}$$

$$\sigma_i^{2\ (t)} = \begin{cases} \tilde{\sigma}_i^{2\ (t-1)} & \text{if } c^{(t)} \cdot s^{(t)} < B, \\ \frac{\alpha_i^{(t-1)}}{\alpha_i^{(t-1)}+1} \tilde{\sigma}_i^{2\ (t-1)} + \frac{1}{\alpha_i^{(t-1)}+1} V_i^{(t)} & \text{otherwise,} \end{cases} \tag{4.12}$$

$$\alpha_i^{(t)} = \alpha_i^{(t-1)} + 1, \tag{4.13}$$

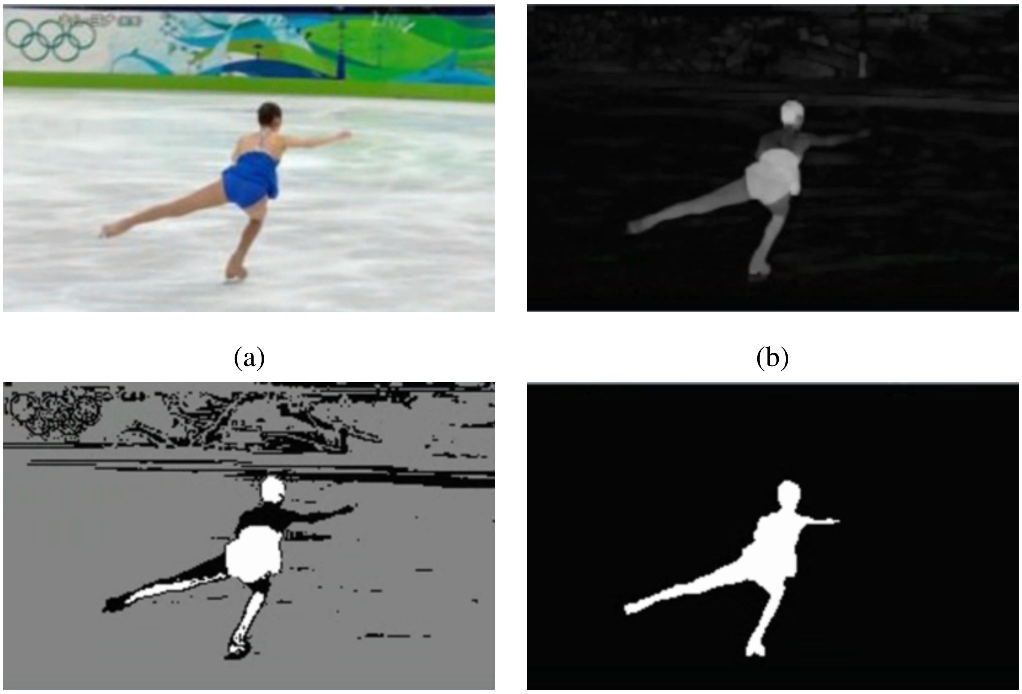
where $M_i^{(t)}$ is the average intensity of the i-th grid $\mathbf{G}_i$ given by

$$M_i^{(t)} = \frac{1}{|\mathbf{G}_i|} \sum_{j \in \mathbf{G}_i} I_j^{(t)}, \tag{4.14}$$

and $V_i^{(t)}$ is defined as

$$V_i^{(t)} = \max_{j \in \mathbf{G}_i} (\mu_i{}^{(t)} - I_j^{(t)})^2. \tag{4.15}$$

The $\alpha_i^{(t)}$ is a parameter for time-varying learning rate where it is set to one as an initial value for the newly appearing region and is increased by one in every updating. The learning rate is designed as $1/(\alpha_i^{(t)} + 1)$, which drives fast adaptation for the newly appearing region and low adaptation for the old region. This $\alpha_i^{(t)}$ has a maximum limit α_{max} to preserve a minimum learning rate.



(a) (b)

(c) (d)

Figure 4.4: Probabilistic foreground decision: (a) Input image (b) Foreground likelihood map (c) Thresholded image (d) Final foreground.

4.3 Foreground Decision

Using the updated background model with mean $\mu^{(t)}$ and variance $\sigma^{(t)}$, each pixel of the input image is decided whether it belongs to background or foreground. The background probability is given by

$$P_{BG}(j) = \frac{1}{\sqrt{2\pi\sigma_i^{2\ (t)}}} \exp\left(-\frac{1}{2}\frac{\left(I_j^{(t)} - \mu_i^{(t)}\right)^2}{\sigma_i^{2\ (t)}}\right), \tag{4.16}$$

where j is the pixel index and i is the grid index containing the pixel j. To calculate the exact background probability in equation (4.16), many computations are needed, such as the square-root and exponential function. Hence, the log-likelihood of the probability is used for simplicity. The background likelihood is proportional to the minus of the normalized distance between the input intensity and mean of the background model, i.e.,

$$L_{BG}(j) = -\frac{\left(I_j^{(t)} - \mu_i^{(t)}\right)^2}{\sigma_i^{2\ (t)}}. \tag{4.17}$$

Then the foreground likelihood is proportional to the minus of the background likelihood. In our algorithm, hence, the foreground likelihood map is obtained by

$$L_{FG}(j) = \frac{\left(I_j^{(t)} - \mu_i^{(t)}\right)^2}{\sigma_i^{2\ (t)}}. \tag{4.18}$$

In our scheme, to consider spatial connectivity within foreground and background regions, additional processing is introduced. Like the canny edge detection [64] method, two thresholds are used to decide the foreground and background. The initial foreground/background labels are determined by using high threshold T_{high} and low threshold T_{low} as

$$l_{init}(j) = \begin{cases} Background & \text{if } L_{FG}(j) \leqslant T_{low}, \\ Candidate & \text{if } T_{low} \leqslant L_{FG}(j) \leqslant T_{high}, \\ Foreground & \text{if } L_{FG}(j) \geqslant T_{high}. \end{cases} \tag{4.19}$$

Then, for the candidate region, a spatial connectivity to the initial foreground is checked with the watershed segmentation method [60] and the final foreground is determined by propagating the foreground label according to the connectivity. Figure 4.4 shows the intermediate result of our foreground decision method. The foreground likelihood map L_{FG} is obtained as Figure 4.4(b), and the initial label is obtained using two threshold parameters. In Figure 4.4(c), the white region is the initial foreground region, the black region is the candidate region, and the gray region is the initial background region. Using the watershed algorithm with l_{init} as an input, the final foreground map l_{final} is obtained as Figure 4.4(d) which shows a clear foreground.

4.4 Benefits

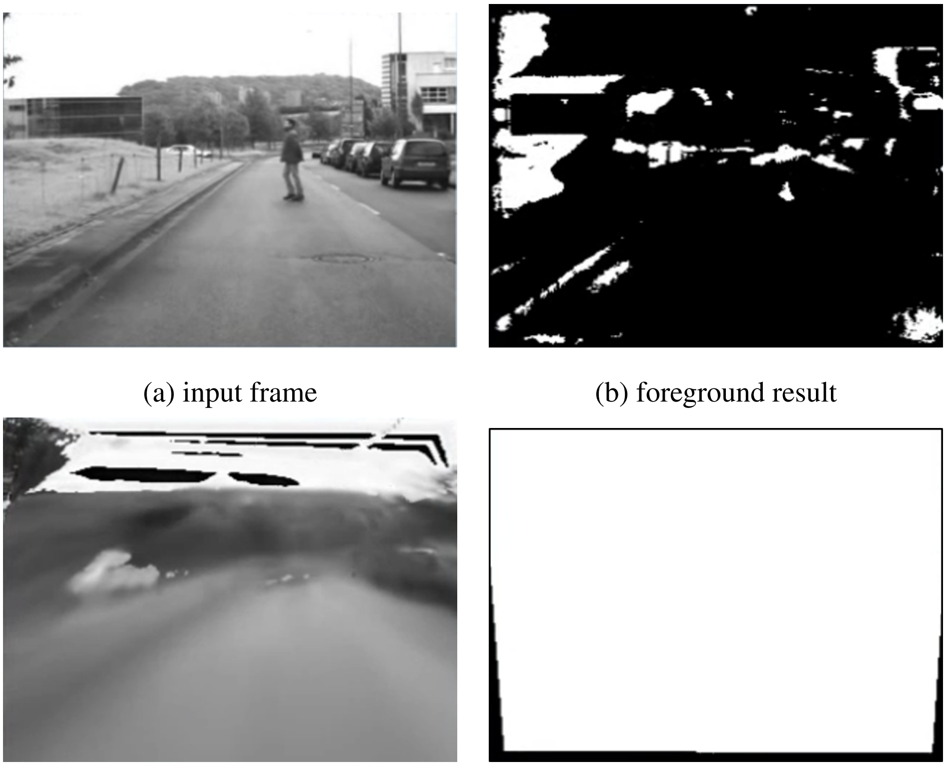
Through the proposed situation-aware background learning method, the background modeling for moving object detection can handle dynamic scenes captured by a moving camera. Because this method compensates for camera movement and updates the background model according to the situation variables, it enables the algorithm to build a clear background model without contamination by the foreground. In addition, a new foreground segmentation method helps to obtain more accurate foreground region by the two thresholds. In the experiment chapter, it is verified that the proposed scheme can cope with the raised problems and is effective compared to the state-of-the-art methods on various moving camera videos.

Chapter 5

Prior-based Attentional Update for dashcam video

Recently, the demand for detection in a moving camera has increased since camera sensors are starting to be mounted on vehicles or drones for smart mobility. This smart mobility, such as detection of unexpected or abruptly moving objects for driverless cars, can provide great benefits to people. In this case, panorama-based approach [40–42] cannot be applied to cameras mounted on high-mobility units, such as vehicles and drones, because it is impossible to generate a panorama covering the entire range of movement. On the other hand, the compensation-based approach is more suitable because it keeps the current background model updated together with camera motion compensation in an online and real-time manner [44, 45, 47, 65]. Until now, this approach has been efficient for application to systems with high mobility because it has a low computational complexity and carries a small scene model relative to the registered panoramic model. However, this approach still has many research issues that need to be overcome for actual application to a system with high mobility where it is extremely difficult to get an estimation of camera motion. Figure 5.1 shows the problems of the previous methods when using a dashcam for the moving object detection. As shown in Figure 5.1(b), the performance of the conventional foreground detection methods in a dashcam is not satisfactory because the built background model cannot represent the actual background. This problem is caused by the limitation of the homography model. Figure 5.1(d) shows the model warping map that indicates the camera motion by displaying the newly appearing region as the black region. Due to the plane assumption of the homography model, we cannot infer this motion between the forward motion and the upward motion. This limitation makes it difficult to detect moving objects in a dashcam video.

In this chapter, in order to cope with issues arising when detecting moving objects using a monocular dashcam with high mobility, we propose an attention-inspired approach by extending the compensation-based approach. In biological studies [57, 66, 67], human beings tend to build an abstract representation of scenes and perceive changes in a region of interest. The human-attention inclination comprises two innate characteristics: center-focused and structure-focused tendencies. First, the center-focused tendency [68, 69] can be easily observed in an experiment where vehicle drivers concentrate on the road around the center of their view. Inspired by this tendency, we propose a scheme to emphasize changes in the center area more than the boundary region by controlling the learning rates depending on the region. The structure-focused tendency means that humans focus on an overall composition of the scene (sky, road, etc.) and neglect detailed information such as the texture of the road. To reflect this tendency to our method, the sky and road regions are estimated as the important structures in dashcam scenes, and these regions are assigned to apparent background regions in order to prevent false detection caused by minor changes in the sky and road regions. In addition, the final detection map is refined to emboss the foreground object region by combining the detection result with the foreground detected from the median-filtered image. While the foreground detected from median-filtered image can create false alarms due to strong edges in the background, the median filter can remove small noises, so it yields a clear foreground map. This combining procedure would reduce false alarms and enhance the clearness of the detection region.



(a) input frame

(b) foreground result

(c) background

(d) model warping map

Figure 5.1: The problems caused by the difficult camera motion in a dashcam. When an input frame is given as (a), the foreground result is obtained as (b). The background model does not represent the actual background well as shown in (c) due to the limitation of the homography model. The black area in model warp map as shown in (d) indicates the newly appearing region.

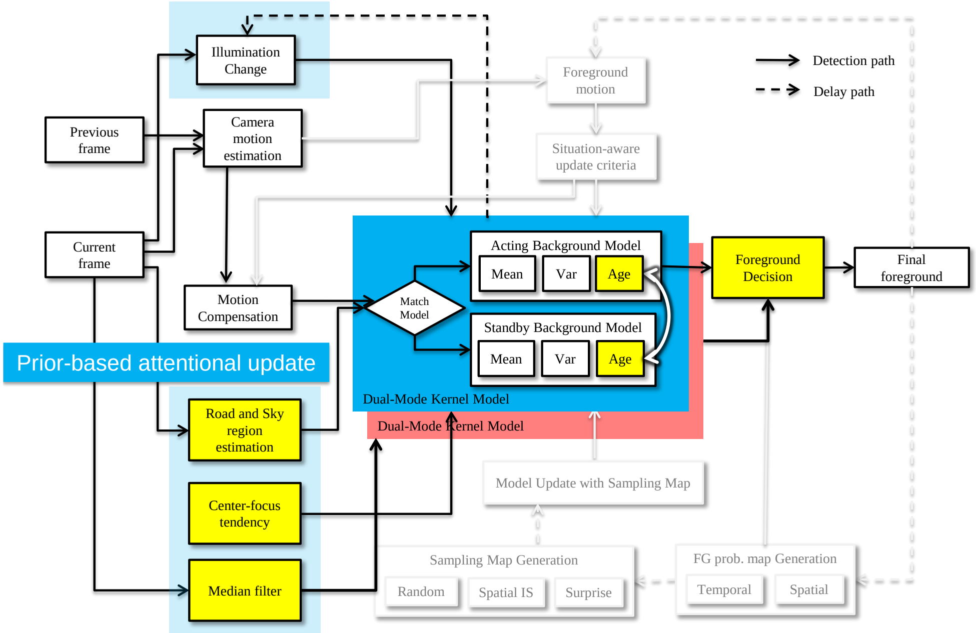


Figure 5.2: Functional scheme of the proposed method. The block with yellow color is newly added or modified in order to cope with dashcam video.

To detect a moving object in a dashcam video, we extend the dual-mode kernel model, which consists of the acting model and the standby model–in which the acting model is trained to contain clear background information, and the standby model is trained to take in other information, such as foreground or image noise. As illustrated in Figure 5.2 (yellow-colored boxes indicate the new or modified modules), the structure estimation module is newly introduced, and the background model update module is modified in the age adaptation scheme as well as foreground decision scheme. In each frame, the sky and road regions are estimated for the important scene structure in dashcam videos. The regions are used as important clues in selecting the model (acting or standby model) to be updated in the dual-mode model. Depending on the selection, we control the learning rate of each pixel to update the background model accordingly. Our algorithm works in twin processes where one is applied to the original image and the other to the median-filtered image. To acquire the final foreground pixels, we combine the foreground detection results from the original image and the median-filtered image, which is described in section 5.4.

5.1 Camera Motion Estimation

Let $I^{(t)}$ denote the single channel input image at time t. Using the KLT [55] algorithm, the corresponding locations are found between the previous frame $I^{(t-1)}$ and the current frame $I^{(t)}$. Let (x_i, y_i) be the i-th sampled point at time $t-1$ and $({x_i}', {y_i}')$ be the corresponding point at time t. The camera motion is represented by a projective transform matrix $\mathbf{H}_{t:t-1}$ obtained by

$$[X_1^{(t)}, X_2^{(t)}, ...] = \mathbf{H}_{t:t-1}[X_1^{(t-1)}, X_2^{(t-1)}, ...], \tag{5.1}$$

where

$$X_i^{(t-1)} = (x_i, y_i, 1)^T, X_i^{(t)} = ({x_i}', {y_i}', 1)^T. \tag{5.2}$$

To estimate the camera motion matrix $\mathbf{H}_{t:t-1}$, the RANSAC [56] is used for robust estimation.

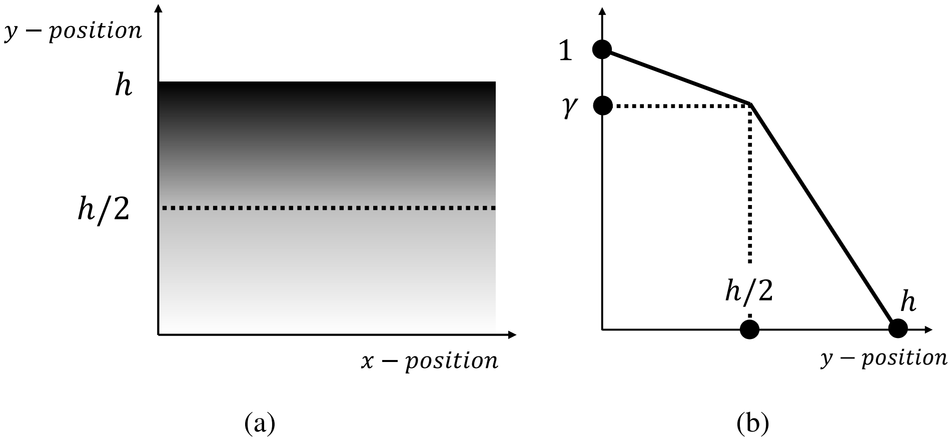
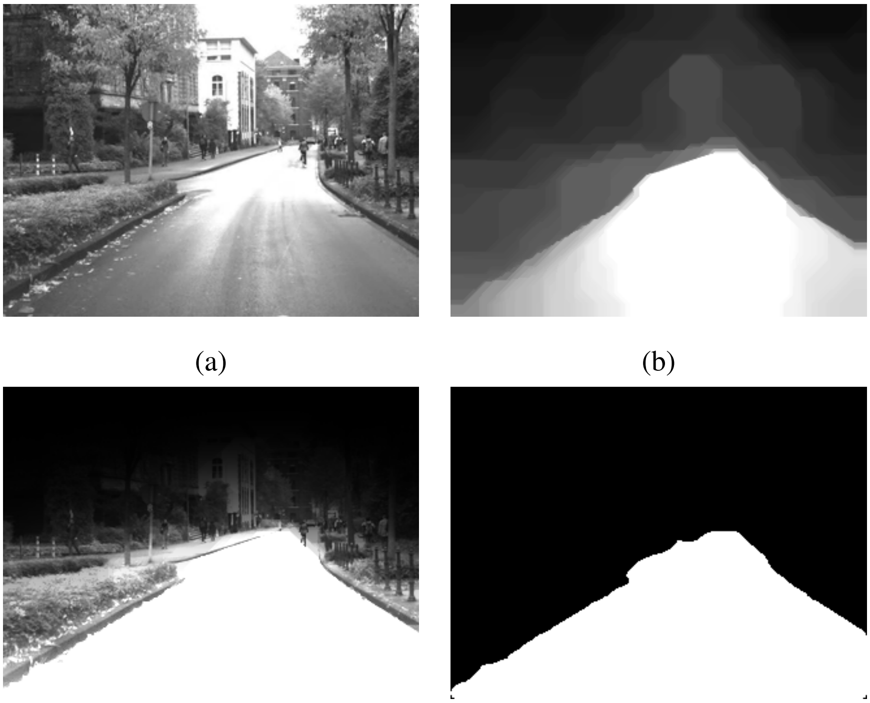


Figure 5.3: Road estimation cue where h refers to the vertical maximum position of image, (a) road confidence map R, (b) road confidence function versus y.

5.2 Road and Sky region estimation

In our algorithm, road and sky regions are estimated with the geometric cue. Without using complex methods [70,71], we propose a simple and effective method based on location priors of road and sky. The estimated region in this step is passed to the model selection module, and pixels in this region are assigned to the acting model for updating since sky and roads must be in the background.

Road region estimation. In a dashcam, the road is generally below the horizon line, which means that the road is predominantly detected at the lower part of the scene. Although there is a variation in the location of the horizon line, we assume that the horizon line is located at the middle position of the image, which is reasonable in a dashcam scene. A pixel-wise road confidence map R is generated as Figure 5.3(a), and the value of road confidence map R is determined along the y-position as Figure 5.3(b). The confidence value is designed to increase linearly as it goes to the bottom part of the image, i.e., inversely proportional to the y-position. Different slopes are used in the upper and lower parts, respectively, where we



(a) (b)

(c) (d)

Figure 5.4: Road region estimation: (a) input image, (b) blurred image after multiplying road confidence map R and original frame, (c) overlaid image with blurred image and input, (d) final road region.

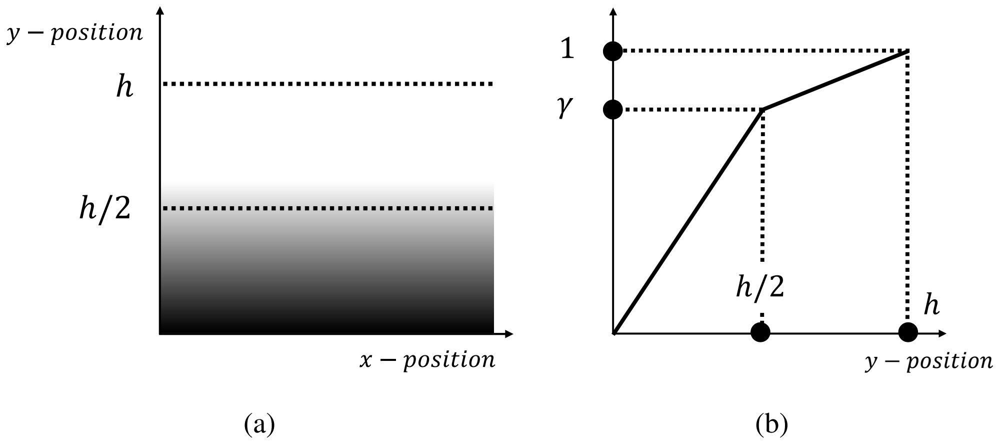


Figure 5.5: Sky estimation cue where h refers to the maximum vertical position of image, (a) Sky confidence map S, (b) sky confidence function versus y.

use a relatively large slope above the horizon line to decrease the road confidence rapidly. The load confidence at the horizon line is set to γ which has been set to 0.75 in our experiment.

To make the pixels in the lower part have a higher probability of road, we multiply this road confidence map R to the input image. Then, a blurring filter with a large window is applied in order to eliminate the details and to leave only structures, as shown in Figure 5.4(b). For the blurring filter, opening and closing function are used in morphological processing [72]. Because this blurring process affects the overall shapes, we overlay this blurred image on the original input image, as shown in Figure 5.4(c). Then, the final road region is obtained by segmenting this overlaid image through thresholding, as shown in Figure 5.4(d).

Sky region estimation. In opposition to the road, the values of sky confidence map S are increased along the y-position. Using the same idea in the road case, we design a pixel-wise sky confidence map S using the same γ as in Figure 5.5. Similar to a road region estimation, we multiply this sky confidence map S to the input image, and then apply a blurring filter. Contrary to road detection, the blurring filter has a small window that is applied because the sky region has a low variance compared to other regions. After the blurred result is obtained,

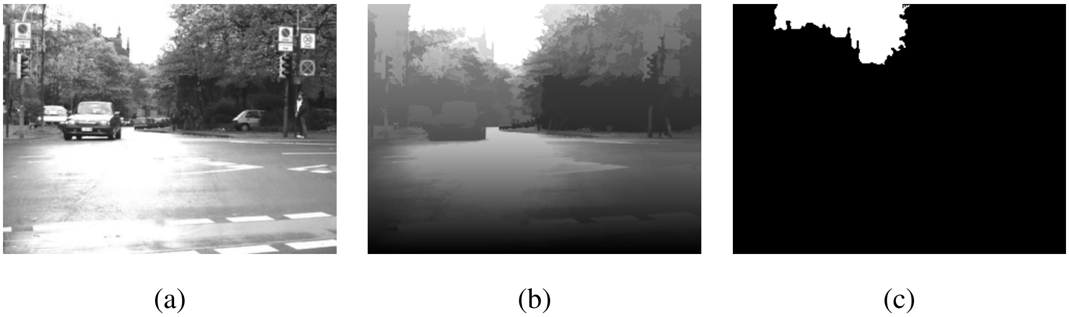


Figure 5.6: Sky region estimation to the input image: (a) input image (b) blurred image after multiplying sky confidence map S, (c) final sky region.

as shown in Figure 5.5(b), the final sky region is obtained by thresholding this filtered image, as shown in Figure 5.5(c).

5.3 Background learning

As mentioned previously, our approach has twin processes where one is for the original image and the other is for median-filtered image as shown in Figure 5.2. The twin processes have the same procedure to each other except the input image and so we describe the procedure without any distinction between them in the following. The background learning is performed to update the Gaussian model with image intensity and an age model for learning rates. The mean and variance of the Gaussian model represent the temporal average and deviation of the image intensity after camera motion compensation. The age model indicates the lifetime that determines a learning rate as $1/(age + 1)$ for fast learning in a newly appearing region. The baseline method uses grid-based modeling that a modeling unit is a grid with multiple pixels. Since this modeling makes neighboring pixels have similar models, it enables us to reduce false detections from minor changes such as alignment errors in motion compensation. Except for the foreground decision based on a pixel unit, the model updating is conducted on each grid.

To build a background model, only one of the dual models in each grid is selected and updated depending on the average intensity of the corresponding grid. The model selection is made by comparing the distance between the average intensity and the mean value of the model. In the update step, the selected model is updated continually, and its age is increased. Hence, the model with a relatively large age is defined as the acting model for the grid background, and the other model is defined as a standby model. However, when a temporal object appears at the grid, the other model is selected and updated to prevent a corruption due to temporal changes. The foreground is decided by the acting model only which contains pure background information.

Motion Compensation. To deal with vehicle motion of dashcam, the motion compensation between the background model and the current frame is required. The mean, variance, and age of both model and standby models at time $t-1$ are warped by $\mathbf{H}_{t:t-1}$ to align with the current frame $I^{(t)}$. Since the center of the warped model usually does not move in a grid unit and is not located at the center of any grid of the new frame, each model in the new grid is interpolated with the neighboring warped models using bilinear interpolation. In the case of the newly appearing region, we set the age value of this region to 0.

Model Selection. Let $\{\mu_{A,i}^{(t)}, \sigma_{A,i}^{(t)}, \alpha_{A,i}^{(t)}\}$ be the mean, variance, and age for the acting model and $\{\mu_{S,i}^{(t)}, \sigma_{S,i}^{(t)}, \alpha_{S,i}^{(t)}\}$ be the mean, variance, and age for the standby model of the i-th grid. Our goal is for an acting model to contain apparent background information, and a standby model to contain latent background information. In the original dual-mode modeling, the appropriate model is selected between the acting model and the standby model using the likelihood of each Gaussian. Since the sky and road regions are apparent backgrounds in a dashcam video, these regions are first assigned to update the acting model. Figure 5.7 shows the model selection process using the scene structure information. From this process, the unimportant changes are reduced by prior semantic cues such as illumination changes on the road and variations of the sky regions.

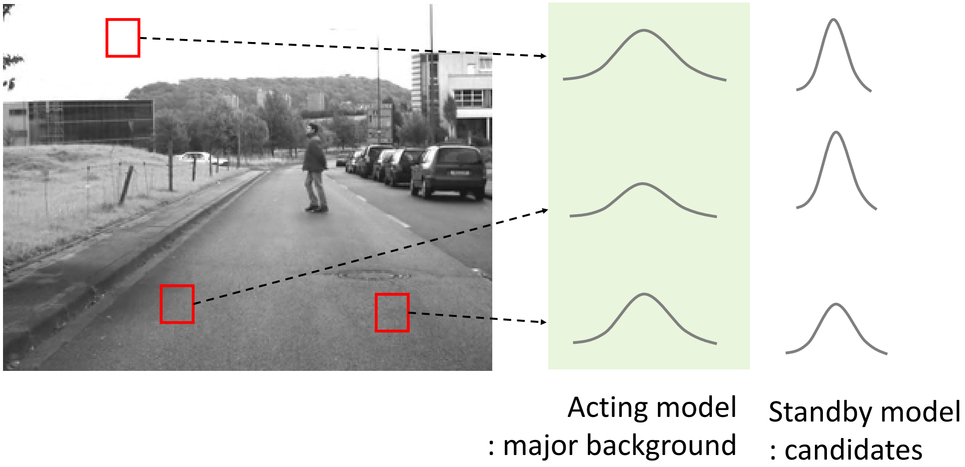


Figure 5.7: The model selection using scene structure information of a dashcam video. Because the sky and road regions are apparent backgrounds, these regions are assigned to the acting model directly.

Let $M_i^{(t)}$ be the average intensity of the i-th grid $\mathbf{G}_i$ given by

$$M_i^{(t)} = \frac{1}{|\mathbf{G}_i|} \sum_{j \in \mathbf{G}_i} I_j^{(t)}, \tag{5.3}$$

where $I_j^{(t)}$ is the input image's intensity of pixel j. Based on a normalized distance by each mean and variance, each grid is assigned to one of the three categories; the i-th grid selects the acting model if

$$\frac{(M_i^{(t)} - \mu_{A,i}^{(t)})^2}{\sigma_{A,i}^{2\,(t)}} < \theta_m, \tag{5.4}$$

and selects the standby model else if

$$\frac{(M_i^{(t)} - \mu_{S,i}^{(t)})^2}{\sigma_{S,i}^{2\,(t)}} < \theta_m, \tag{5.5}$$

otherwise, the standby model is initialized. Here θ_m is a thresholding parameter for the model selection.

If the current i-th grid selects the acting model, $\{\mu_{A,i}^{(t)}, \sigma_{A,i}^{2\,(t)}, \alpha_{A,i}^{(t)}\}$ are updated according to equation (5.6)-(5.9). If it selects the standby model, $\{\mu_{S,i}^{(t)}, \sigma_{S,i}^{2\,(t)}, \alpha_{S,i}^{(t)}\}$ are updated according to equation (5.6)-(5.9). If it does not select both the acting and standby model, the standby model is initialized with current i-th grid intensity.

Model Update. The baseline method does not consider the large illumination change due to the auto-exposure control in the camera. This happens frequently in a moving camera and needs to be solved. Therefore, the overall change of illumination is additionally compensated in the mean update step. Let $\mu_i^{(t-1)}$, $\sigma_i^{2\,(t-1)}$, and $\alpha_i^{(t-1)}$ be the mean, variance, and age model at $(t-1)$ frame of the i-th grid after the camera motion is compensated. The update formula of ${\mu_i}^{(t)}$ is given by

$$ {\mu_i}^{(t)} = \frac{\alpha_i^{(t-1)}}{\alpha_i^{(t-1)}+1}(\mu_i^{(t-1)} + b^{(t)}) + \frac{1}{\alpha_i^{(t-1)}+1}M_i^{(t)}, \tag{5.6} $$

where $M_i^{(t)}$ is the average intensity of the i-th grid in equation (5.3) and $b^{(t)}$ is the overall change of scene illumination defined as

$$ b^{(t)} = \frac{1}{N}\sum_{j=1}^{N} I_j^{(t)} - \frac{1}{M}\sum_{i=1}^{M} \mu_i^{(t-1)}. \tag{5.7} $$

In (5.7), N is the number of pixels and M is the number of grids. Through the $b^{(t)}$, the mean model is adjusted to compensate for an illumination change. $\sigma_i^{2\,(t)}$, and $\alpha_i^{(t)}$ are updated by

$$ \sigma_i^{2\,(t)} = \frac{\alpha_i^{(t-1)}}{\alpha_i^{(t-1)}+1}\sigma_i^{2\,(t-1)} + \frac{1}{\alpha_i^{(t-1)}+1}V_i^{(t)}, \tag{5.8} $$

$$ \alpha_i^{(t)} = \alpha_i^{(t-1)} + 1, \tag{5.9} $$

where $V_i^{(t)}$ is defined as

$$ V_i^{(t)} = \max_{j \in \mathbf{G}_i} ({\mu_i}^{(t)} - I_j^{(t)})^2. \tag{5.10} $$

In (5.6) and (5.8), the learning rate is designed as $1/(\alpha_i^{(t-1)} + 1)$, which drives rapid adaptation for the newly appearing region and slow adaptation for the old region.

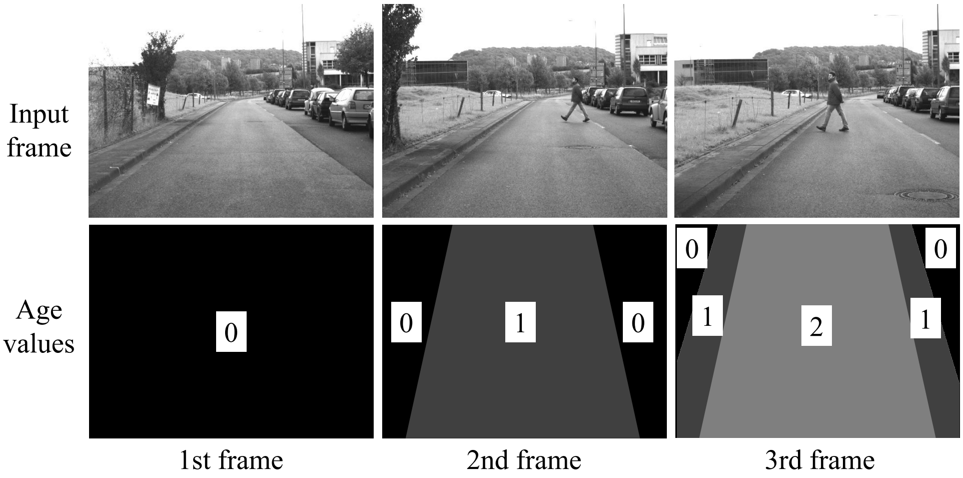
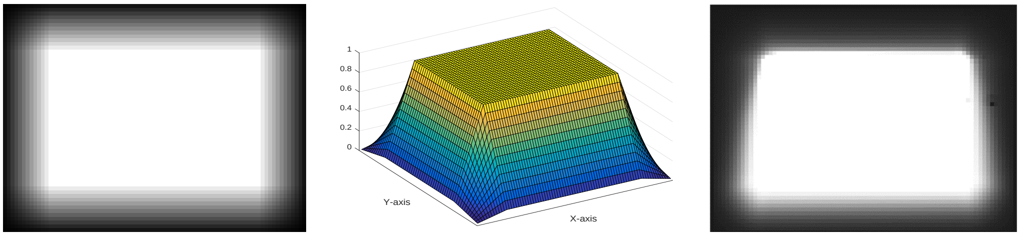


Figure 5.8: The change of acting model age when camera moves forward.

In the case of the dashcam video, however, age values of the entire region are consistently increased when the camera moves forward, as shown in Figure 5.8. It makes for a very small learning rate and obstructs the adaptation of scene change by a moving camera. In our method, the age model is modified to accommodate the change at the boundary region inspired by human attention, since humans usually do not recognize the details around the boundary because they tend to pay attention to the center. To mimic this scheme, the learning rate (age) is designed to increase (decrease) for the rapid learning of the changes around the boundary region. To design this scheme, the attention map A is utilized to decrease the age by multiplying the attention map to the age map $\alpha^{(t)}$. As a result, the rapidly trained background model tends to accept the various changes at the boundary region and reduce the false foreground detections even various changes at the boundary region. The attention map is designed as a clipped-off pyramid, as shown in Figure 5.9(b). The attention map A is decomposed into x-axis attention A_x and y-axis attention as

$$A(p, q) = A_x(p) \times A_y(q), \tag{5.11}$$



(a) (b) (c)

Figure 5.9: Age modification using attention map A: (a) 2D view of A, (b) 3D view of A, (c) modified age map after multiplication with A.

where

$$A_x(p) = \begin{cases} p/b_x & \text{if } 0 \le p \le b_x, \\ 1 & \text{if } b_x \le p \le w - b_x, \\ (w-p)/b_x & \text{if } w - b_x \le p \le w, \end{cases} \tag{5.12}$$

$$A_y(q) = \begin{cases} q/b_y & \text{if } 0 \le q \le b_y, \\ 1 & \text{if } b_y \le q \le h - b_y, \\ (h-q)/b_y & \text{if } h - b_y \le q \le h, \end{cases} \tag{5.13}$$

where (w, h) is the image size, and (b_x, b_y) is a design parameter indicating the length of the boundary in each axis. Then, the maximum age α_{max} is set to guarantee a minimum learning rate to adapt a background change as

$$\alpha^{(t)} \leftarrow \min(A \cdot \alpha^{(t)}, \alpha_{max}), \tag{5.14}$$

where $\cdot$ and $\min$ are both grid-wise operation.

Model Switching. If the age of the standby model exceeds that of the acting model, the standby model becomes a new acting model as

$$\{\mu_{A,i}^{(t)}, \sigma_{A,i}^{2\,(t)}, \alpha_{A,i}^{(t)}\} \leftarrow \{\mu_{S,i}^{(t)}, \sigma_{S,i}^{2\,(t)}, \alpha_{S,i}^{(t)}\}, \tag{5.15}$$

and the new standby model is initialized as

$$\alpha_{S,i}^{(t)} \leftarrow 0. \tag{5.16}$$

Foreground Decision. To decide the background and foreground regions, we calculate the background probability of each pixel through the acting model as

$$P_{BG}(j) = \frac{1}{\sqrt{2\pi\sigma_{A,i}^{2\,(t)}}} \exp\left(-\frac{1}{2} \frac{\left(I_j^{(t)} - \mu_{A,i}^{(t)}\right)^2}{\sigma_{A,i}^{2\,(t)}} \right), \tag{5.17}$$

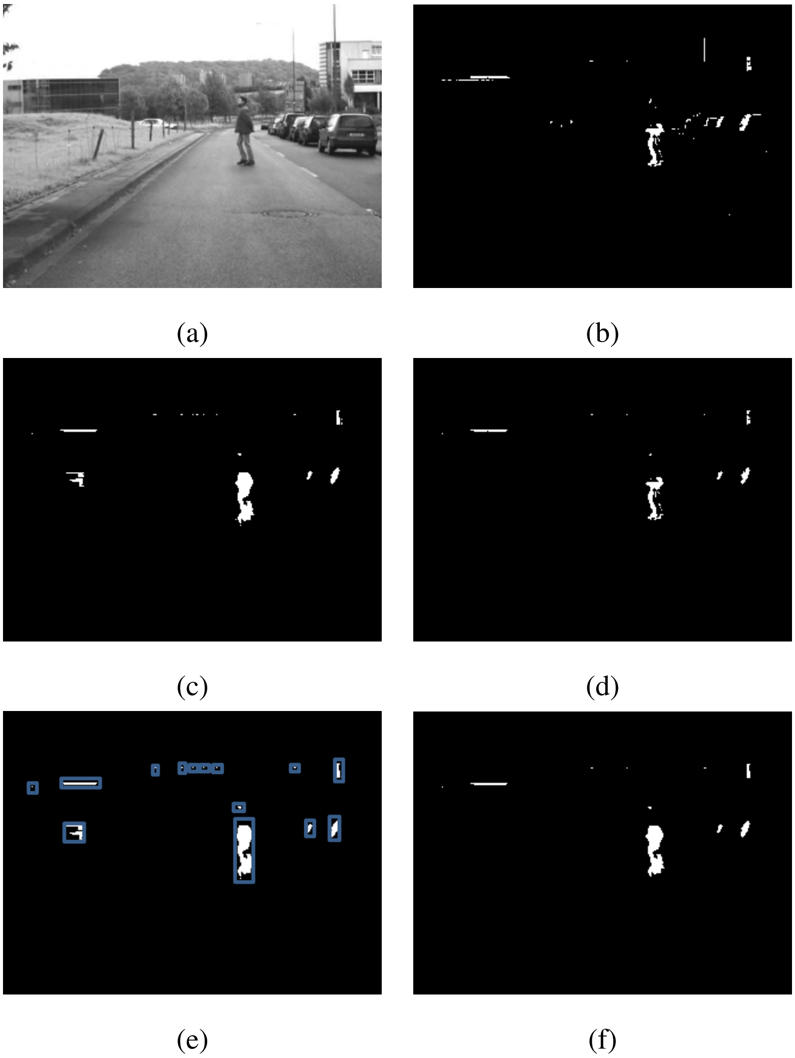
where j is the pixel index and i is the grid index containing the pixel j. To reduce computations for calculating the equation (5.17), we simply decide the label through thresholding the normalized distance as

$$l(j) = \begin{cases} foreground & \text{if } \frac{\left(I_j^{(t)} - \mu_{A,i}^{(t)}\right)^2}{\sigma_{A,i}^{2\,(t)}} > \theta_l, \\ background & \text{otherwise,} \end{cases} \tag{5.18}$$

where θ_l is a thresholding parameter for the foreground decision.

5.4 Foreground Result Combining

Our model has twin processes using an original image and a median-filtered image. The median filter helps detect an entire foreground without loss of parts, but it creates false alarms near the background edges. Therefore, the final foreground is obtained combining the foreground results from the twin processes. Figure 5.10 shows the intermediate results of the foreground combining. Two foreground maps are obtained from (5.18); l_{orig} is the foreground map from the original image as shown in Figure 5.10(b), and l_{med} is that from the median-filtered image as shown in Figure 5.10(c). First, two foregrounds are multiplied, and the multiplied result is used as seed regions to get a reliable foreground, as shown in Figure 5.10(d). As in Figure 5.10(e), the connected blobs in l_{med} are generated by the blob analysis [72]. Among these blobs, we remove the blobs not containing the obtained seed regions. Last, the noises in the road and sky region are removed in the final foreground.



(a) (b)

(c) (d)

(e) (f)

Figure 5.10: Foreground result combining: (a) Input image, (b) Foreground l_{orig} from input image, (c) Foreground l_{med} from median-filtered image, (d) Multiplied image by l_{orig} and l_{med}, (e) l_{med} with connected blobs, and (f) Refined foreground using removing noisy blobs.

5.5 Benefits

In this chapter, a human attention-inspired background learning method is proposed to handle dynamic scene changes for moving object detection in a dashcam. Motivated by the center-focused and structure-focused tendencies, the prior-based attentional update method focuses on the center changes and neglects minor changes on the important scene structure. As a result, the unimportant noise such as road crack and light reflection is removed, and the background model fastly adapts itself to the side changes. In the experiment chapter, it is verified that the proposed scheme is robust against to the various changes of a dashcam compared to the existing state-of-the-art methods. Also, it is shown that our method can be used as an efficient algorithm for the object proposal through the combination with the recognition algorithm.

Chapter 6

Experiments

In this chapter, the experimental results and the effect of each proposed scheme are shown via the qualitative and quantitative results. Then the unified method covering all issues are verified through the comparison with the state-of-the-art methods. In addition, the combined work of recognition and our method is introduced for the future application of the moving object detection.

We tested our method with 22 video sequences captured by moving cameras on various platforms. Moving object detection in these videos is a challenging problem due to sudden camera movements, slowly moving objects, large illumination changes, dashcam videos, and the other issues. The information of each video is described in Table 6.1. In case of dashcam sequences, although we extracted ten sequences for the quantitative result, we tested a long single video having $12,243$ frames in the Daimler dataset [73].

To evaluate the performance quantitatively, we created the binary ground truth mask by hand at a constant sampling rate. Figure 6.1 shows the examples of the ground truth mask. Based on the ground truth mask G and the final foreground map, we measured the pixel-wise

Table 6.1: Characteristics of the employed videos. All videos were acquired at 30 FPS.

no.	name	# of frame	Description
1	$walking$	332	a fast moving object
2	$skating$	185	a dynamic object
3	$woman$	596	illumination change
4	$woman2$	557	a large object
5	$mountainbike$	228	illumination change
6	$cycle$	300	fast moving multiple objects
7	$fence$	1,309	various object speeds
8	$ground1$	1,466	various object sizes
9	$ground2$	1,666	various object sizes
10	$ground3$	933	a slow object
11	$ground4$	428	a small object, illumination change
12	$ground5$	781	illumination change
13	$sequence - 1$	31	a pedestrian in dashcam video
14	$sequence - 2$	21	a vehicle in dashcam video
15	$sequence - 3$	22	a vehicle in dashcam video
16	$sequence - 4$	18	a vehicle in dashcam video
17	$sequence - 5$	16	a vehicle in dashcam video
18	$sequence - 6$	26	a pedestrian in dashcam video
19	$sequence - 7$	31	a pedestrian in dashcam video
20	$sequence - 8$	26	a pedestrian in dashcam video
21	$sequence - 9$	36	a pedestrian in dashcam video
22	$sequence - 10$	31	a pedestrian in dashcam video

Note: The $skating$ sequence is from [3], the $woman$ is from [74], and the $woman2$ is from [47]. The $mountainbike$ is from [75], and the ten sequences of a dashcam video are from the Daimler dataset [73]. The other sequences are our videos.

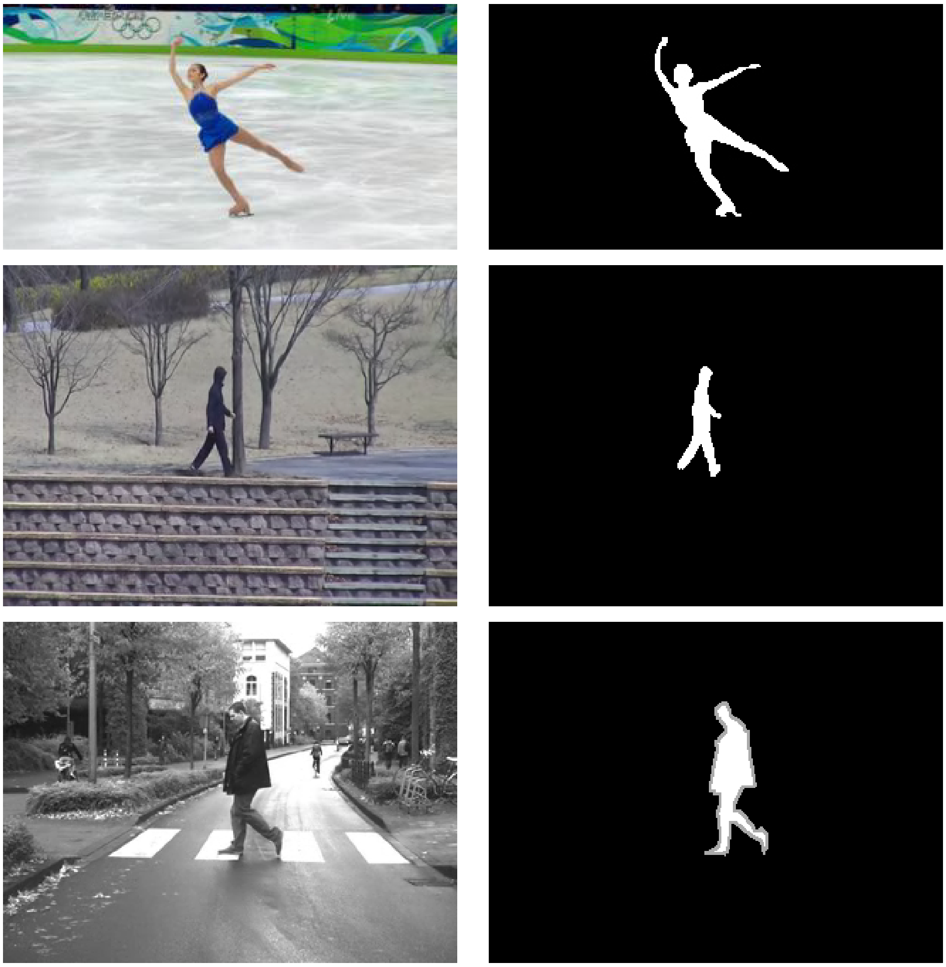


Figure 6.1: The examples of the dataset image and the ground truth mask. First row shows the example of $skating$ video and its ground truth mask. Second row shows the example of $fence$ video and its ground truth mask. Third row shows the example of $sequence-6$ video and its ground truth mask.

precision and *recall*. The $precision$ and $recall$ are calculated at

$$precision = \frac{TP}{TP + FP},\ recall = \frac{TP}{TP + FN} \tag{6.1}$$

where TP is the number of true positives, FP is the number of false positives (false alarms), and FN is the number of false negatives (missing) in each video. As for an overall performance measure, we used the F-measure representing a harmonic mean of *precision* and *recall*.

The proposed method was implemented using C++ and the OpenCV library. For the parameters, the grid size B is 4, α_{max} is 100, T_{high} is 3.5, and T_{low} is 1.0. For the experiments, we fixed parameters that the size of grid $\mathbf{G}_i$ is 4×4, the road & sky confidence map parameter $\gamma = 0.75$, the threshold for model selection $\theta_m = 3$, the threshold for label decision $\theta_l = 4$, the boundary length for attention map $(b_x, b_y) = (40, 40)$, and the maximum age $\alpha_{max} = 10$.

6.1 Qualitative Comparisons

6.1.1 Dual modeling with attentional sampling

We compared our method to the state-of-the-art methods: segmentation-based method [3] and compensation-based methods [31, 44, 45]. For ViBe [31] that is originally designed for a stationary camera, we added the motion compensation for non-stationary camera as shown in the author's websites.[1]

Figure 6.2 shows the qualitative results of the compared methods. General BS [3], as a segmentation-based method, does not assume the specific camera model, so they can remove false detections and shows the best performance in the *cycle* sequence. However, the resulting detection region contains large neighbor backgrounds, like the case of *woman* and *walking* sequences. Moreover, General BS sometimes miss the object completely as shown in the *Mountain bike* sequence when a foreground is not distinguished from a complex back-

[1] http://www2.ulg.ac.be/telecom/research/vibe/

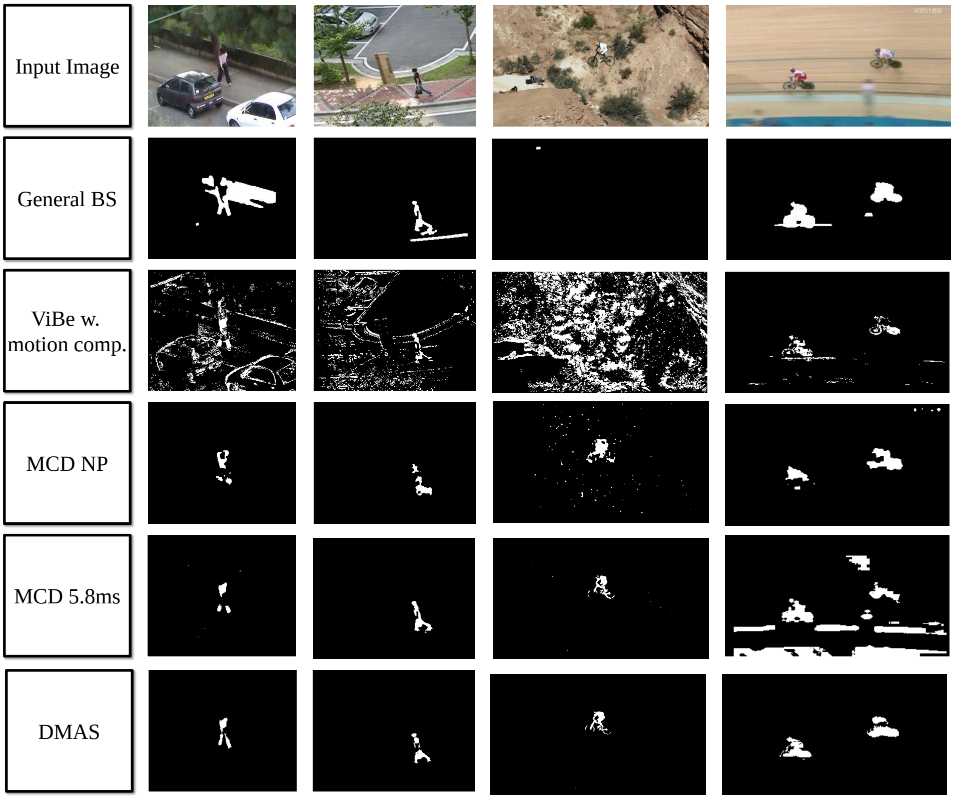


Figure 6.2: Qualitative results by the dual-mode model with attentional sampling. From left to right: *woman*, *walking*, *mountain bike*, and *cycle*. First row shows input images, and the other rows show the results of the compared methods: General BS [3], ViBe [31] with motion compensation, MCD NP [44], MCD 5.8ms [45], and dual mode with attentional sampling (DMAS). All datasets shown are non-commercial and publicly available.

ground. In [31] as shown in Figure 6.2(c), many false detections arise in the image edge because they do not consider the inaccurate estimation of camera movements. Non-panoramic moving object detection in moving camera (MCD NP) [44] produces an incomplete foreground with inner hole and noise in Figure 6.2(d). Our method detects the objects clearly without foreground missing (*woman* sequence) and drastic noise (*cycle* sequence) unlike the result in MCD 5.8ms [45]. We uploaded a supplementary video to YouTube to illustrate the distinctive comparison on the compared methods[2].

6.1.2 Situation-aware background learning

We qualitatively compared our method with the following state-of-the-art methods for moving object detection in a moving camera: MCD NP [44], MCD 5.8ms [45], and Stochastic approx [47]. Figure 6.3 and Figure 6.4 show the qualitative results on critical frames from a couple of video sequences.

The videos as shown in Figure 6.3 contain large illumination changes and a fast moving object. As shown in this figure, MCD NP reduces false positives, but the silhouette of the moving object is exaggerated near the edges. MCD 5.8ms is vulnerable to illumination changes showing many false positives over a wide area. Although Stochastic approx. detects the moving object well, false positives occur near the object, as shown in the second row. Our method shows a clear and robust foreground result against illumination changes. The video sequences as shown in Figure 6.4 contain slowly moving people. In this case, MCD NP, MCD 5.8ms, and Stochastic approx. cannot handle the slow motion of the foreground effectively. Compared with other methods, our method can segment clear foregrounds as shown in the figure. The video results for all methods are provided on YouTube.[3].

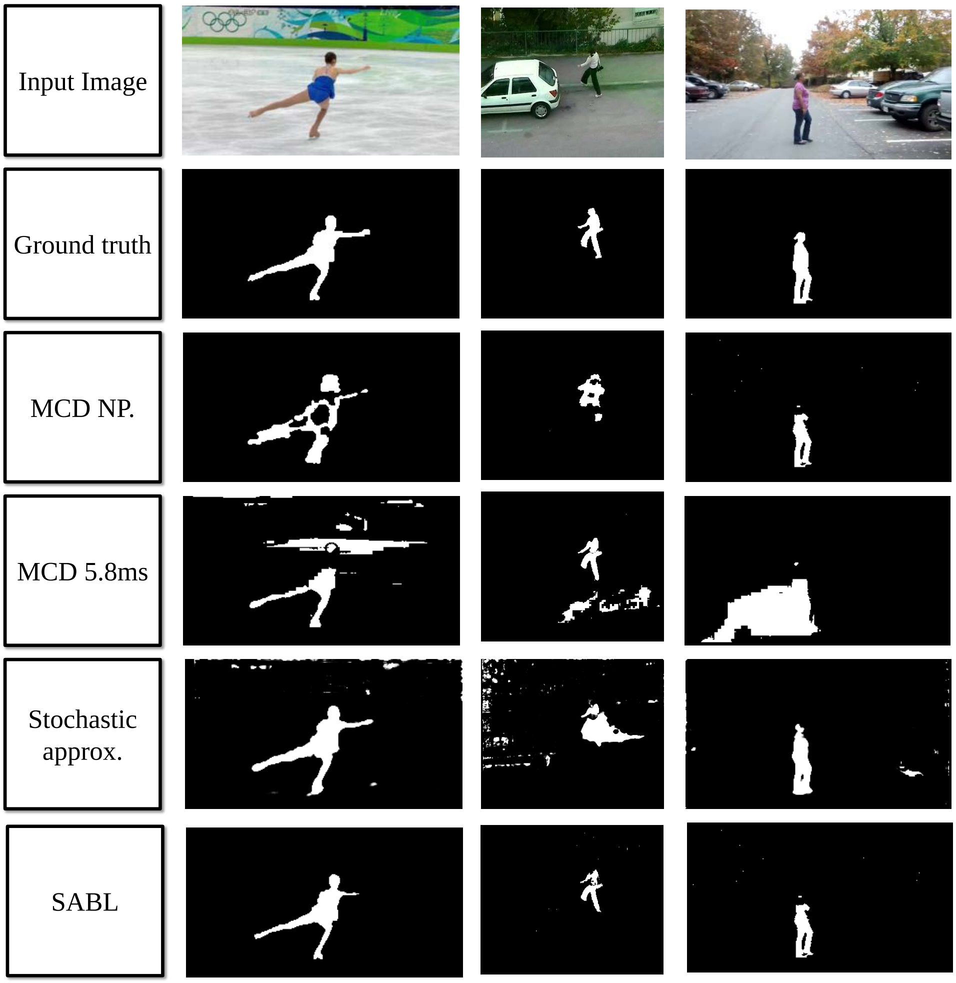


Figure 6.3: Qualitative results on critical frames by the situation-aware background learning. From left to right: *skating*, *woman*, *woman2*. The first row shows input images, and the other rows show the results of the compared methods: input image, ground truth, MCD NP [44], MCD5.8ms [45], Stochastic approx [47], and the situation-aware background learning (SABL).

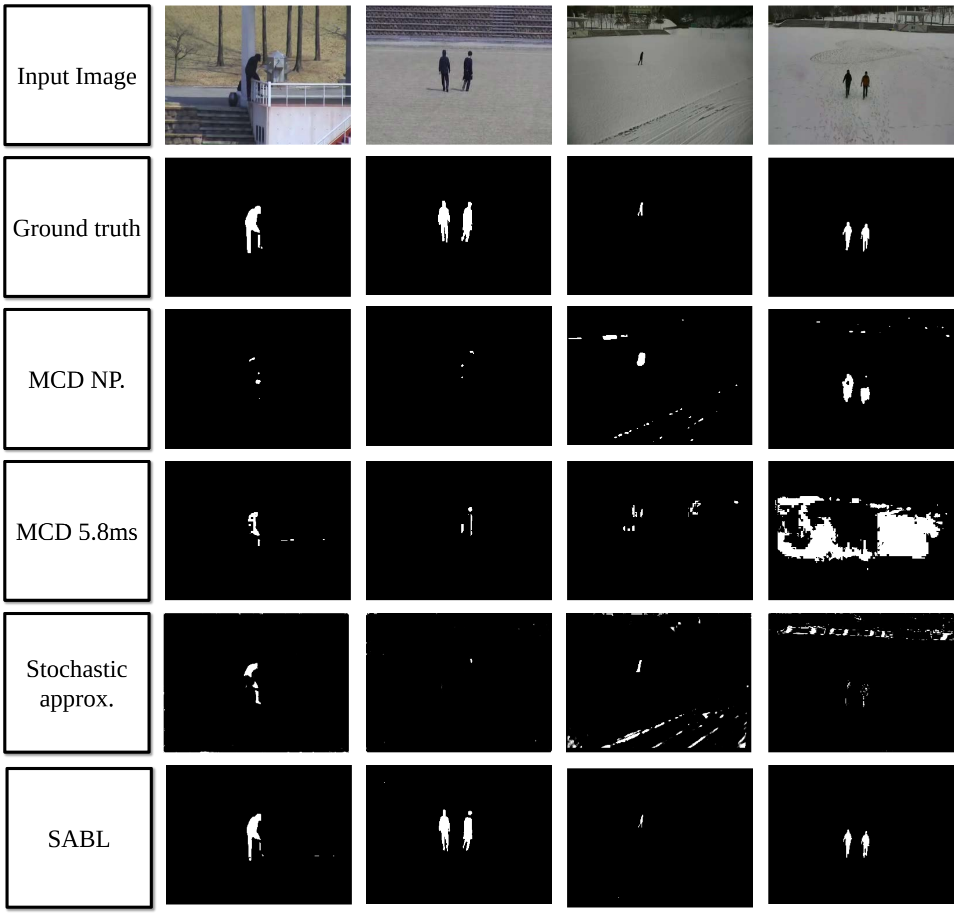


Figure 6.4: Qualitative results on critical frames by the situation-aware background learning. From left to right: *fence*, *ground3*, *ground4* and *ground5*. The first row shows input images, and the other rows show the results of the compared methods: input image, ground truth, MCD NP [44], MCD 5.8ms [45], Stochastic approx [47], and the situation-aware background learning (SABL).

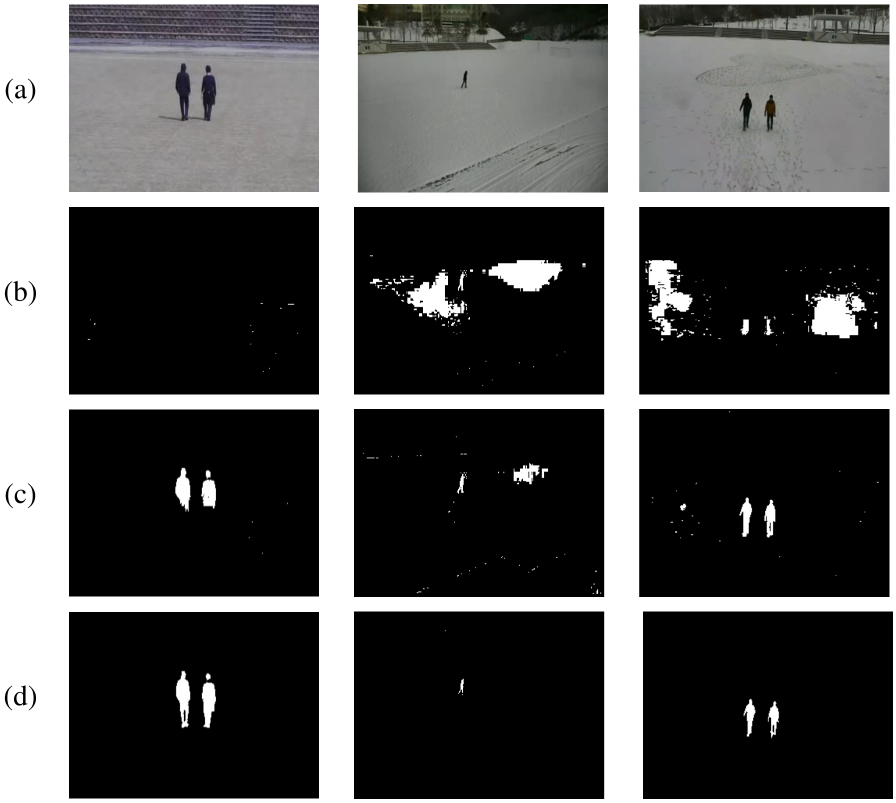


Figure 6.5: The qualitative effect of situation-aware background learning: first row shows input frames in (a), the other rows show the result of baseline method in (b), baseline + SABL in (c), and baseline + SABL + WS in (d), respectively.

6.1.2.1 Effect of Situation-aware Background Learning

Figure 6.5 shows the effect of situation-aware background learning (SABL) as introduced in section 4.2 and the watershed segmentation (WS) described in section 4.3. The result for the baseline in Figure 6.5(b) performs poorly when objects move slowly, and when illumination changes occur. By applying SABL, the detection results are enhanced as shown in Figure 6.5(c). With WS as a post-processing step, small instances of noise are removed, and foreground losses are restored.

6.1.3 Prior-based attentional update

To validate the prior-based attentional update, we compared our method with four state-of-the art methods in the compensation-based approach; MCD NP [44], MCD 5.8ms [45], FP Sampling [65], and Stochastic approx [47]. MCD NP [44] uses spatio-temporal learning, which considers neighboring pixels to reduce false positives. MCD 5.8ms [45] is the baseline method that builds a dual-mode background model consisting of the acting model and the standby model. FP Sampling [65] uses a sampling method to focus on the apparent foreground region based on foreground probability. Stochastic approx [47] builds the full covariance matrices of the background model using the multi-channel feature.

Figure 6.6 and Figure 6.7 show the qualitative results on critical frames. MCD NP has false positives around image boundaries and road textures. MCD 5.8ms reduces these false positives but loses the foreground as shown in the top row. FP Sampling also suffers from the foreground loss problem because FP sampling focuses on detecting a reliable foreground. Stochastic approx has some fail cases when their background is reinitialized due to complex camera motion. From Figure 6.6 and Figure 6.7, the result from the prior-based attentional update outperforms the compared methods for segmentation of foregrounds in a dashcam video. The comparison video for all 12,243 frames can be seen on YouTube.[4].

[2] http://youtu.be/2UOu4OuBYUs

[3] https://youtu.be/ZNFZzhQgjkc

[4] https://youtu.be/tk3m7z0S8vE

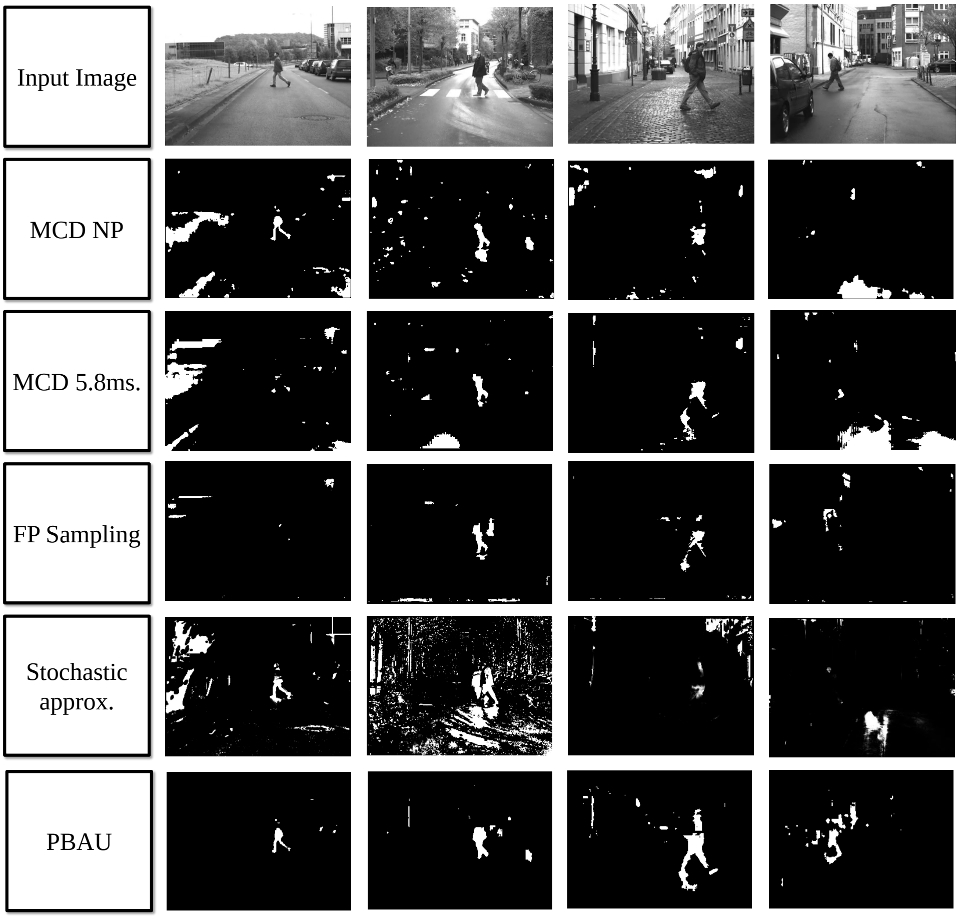


Figure 6.6: Qualitative results on moving pedestrian detection of Daimler dashcam video. The first row shows input images, and the other rows show the results of the compared methods: input image, MCD NP [44], MCD 5.8ms [45], FP Sampling [65], Stochastic approx [47], and the prior-based attentional update (PBAU).

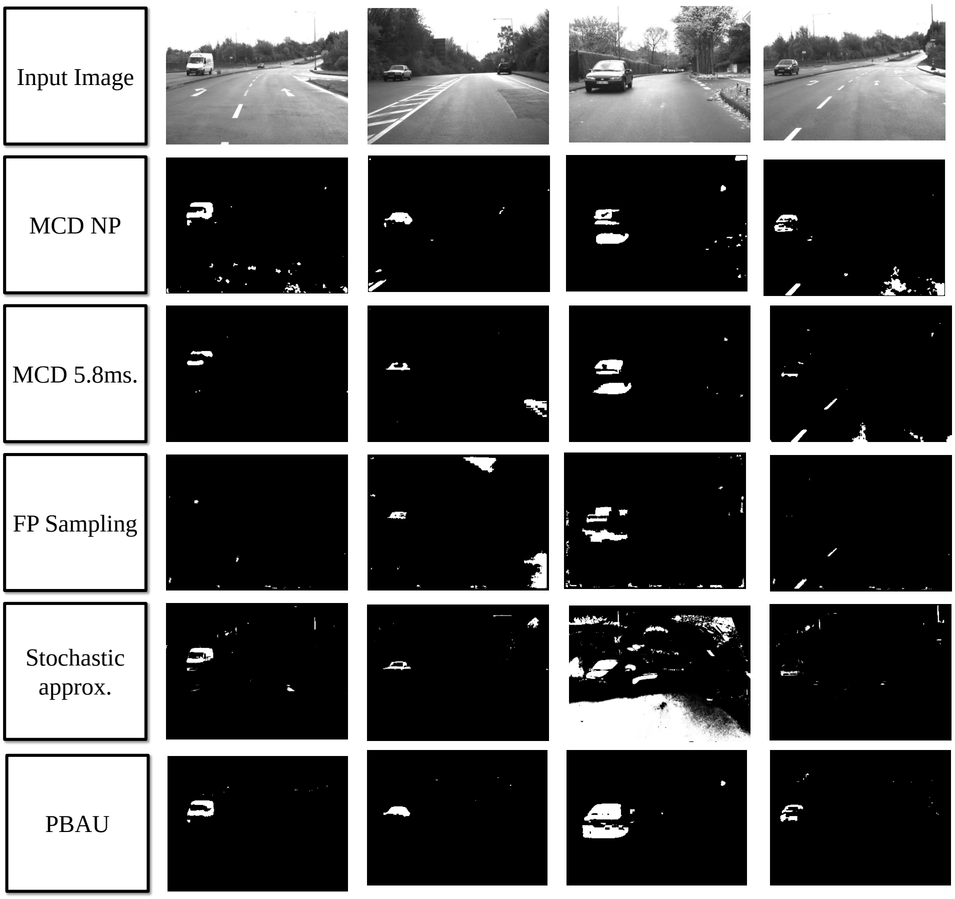


Figure 6.7: Qualitative results on moving vehicle detection of Daimler dashcam video. The first row shows input images, and the other rows show the results of the compared methods: input image, MCD NP [44], MCD 5.8ms [45], FP Sampling [65], Stochastic approx [47], and the prior-based attentional update (PBAU).

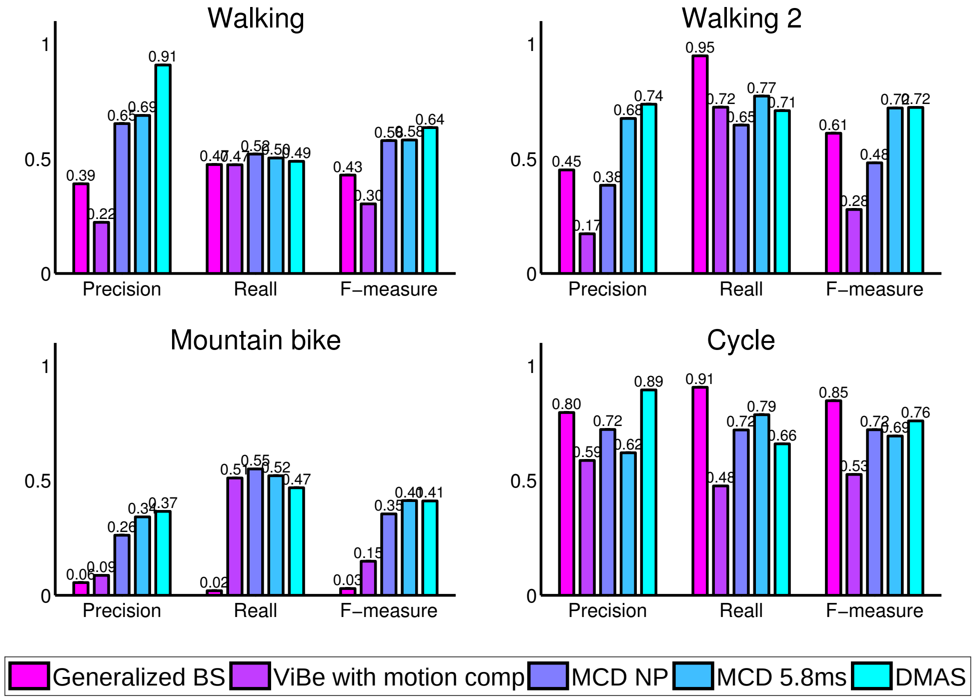


Figure 6.8: Quantitative results for the dual-mode modeling with attentional sampling (DMAS) using pixel-wise *precision*, *recall*, F-measure.

6.2 Quantitative Comparisons

6.2.1 Dual modeling with attentional sampling

Figure 6.8 shows the quantitative results of dual modeling with attentional sampling. Since the false positives are reduced significantly through the attentional sampling, the *precision* values of the proposed scheme are highest over all sequences. Although Generalized BP [3] marked best performance in $cycle$ sequence, it marked the worst in $mountain\ bike$ sequence.

6.2.2 Situation-aware background learning

We quantitatively compared our method with the following state-of-the-art methods: SOBS [25], ViBe [31], FIC [21], BMRI-ViBe [33], MCD NP [44], MCD 5.8ms [45], and Stochastic approx [47]. SOBS and ViBe are standard background subtraction (BS) methods, and FIC and

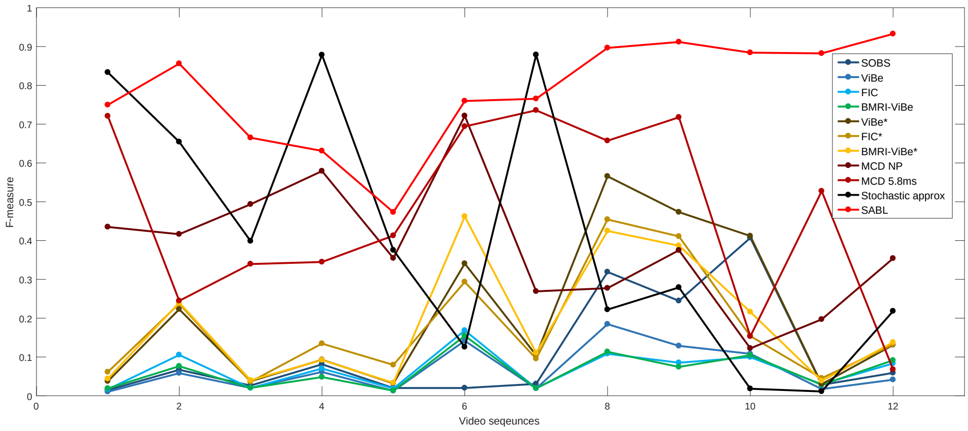


Figure 6.9: The pixel-wise F-measure results for the first 12 videos in Table 6.1, where x-axis indicates the video sequence no. in Table 6.1. * indicates that the background model was warped to compensate the camera motion for the algorithm which was designed for the stationary camera. Since SOBS was provided as a binary file, we could not attach the motion compensation.

BMRI-ViBe are BS methods modified to cope with sudden illumination changes. However, because these methods are designed for a fixed camera, we also used a simple motion compensation procedure to adapt them to a moving camera. In Figure 6.9, the asterisk(*) indicates the combined method with motion compensation. MCD NP, MCD5.8ms, and Stochastic approx were proposed for a moving camera.

Figure 6.9 shows the performance of each method on each video. It is clear that the existing methods for a fixed camera perform poorly in a moving camera video. Although the performances improve slightly through motion compensation, these methods remain inferior to methods developed for a moving camera. Specifically, FIC* and BMRI-ViBe*, which were designed to deal with illumination changes, were ineffective because illumination change tendencies which arise with a moving camera differ from those with a fixed camera. While the performances of the existing methods were unstable and degraded depending on the situations in the test video sequences, the proposed method showed robust performance in such situations. Our method outperforms state-of-the-arts in tested videos with situation changes

Table 6.2: The pixel-wise average $precision$, $recall$, and F-measure results for the contributions of each module.

Method	$precision$	$recall$	F-measure
baseline	0.5943	0.5068	0.4394
baseline + SABL	0.6757	0.6598	0.6339
baseline + SABL + WS	0.7572	0.9134	0.8173

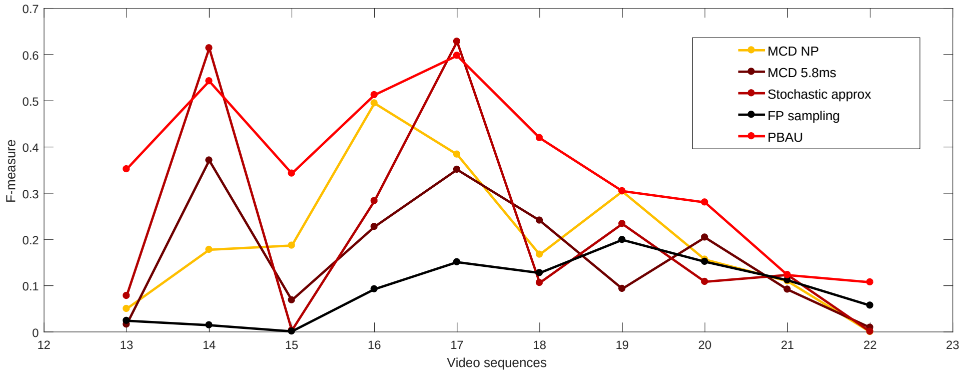


Figure 6.10: Quantitative results for dashcam videos listed, where x-axis indicates the video sequence no. in Table 6.1 using pixel-wise F-measure.

and is comparable to Stochastic approx with typical videos.

Table 6.2 shows the contributions of SABL and WS for the average $precision$, $recall$, and F-measure. Through SABL and WS, the overall performances are improved.

6.2.3 Prior-based attentional update (PBAU)

In the F-measure graph in Figure 6.10, the proposed PBAU outperforms state-of-the-art methods in ten dashcam videos listed in Table 6.1. For the $sequence$-2 and $sequence$-4, 'Stochastic approx' shows a good performance because it builds the detailed background model. However, when the scene changes dynamically, this method fails to detect the foreground. On the contrary, our method shows a robust performance against changes in a dashcam video.

Table 6.3: The average computational loads of each algorithm. * indicates that the background model was warped to compensate the camera motion for the algorithm which was designed for the stationary camera. DMAS means the dual-mode with attentional sampling, SABL means the situation-aware background learning, and PBAU means the prior-based attentional update.

Methods	Time per frame	frame/section
Generalized BP [3]	35.3s	0.028 fps
ViBe* [31]	11.23ms	89.05fps
MCD NP [44]	16.08ms	62 fps
MCD 5.8ms [45]	5.74ms	174 fps
DMAS	4.80ms	208 fps
SABL	7.36 ms	136 fps
PBAU	16.58 ms	60.3 fps

6.2.4 Runtime evaluation

To evaluate the computational efficiency, we measured the computation loads of the compared methods on Intel Core i5-3570 3.4GHz PC with 320 × 240 image without parallel processing. Table 6.3 shows the run-time comparisons using average computation time. Generalized BP [3] takes about 30 seconds to per one frame. Moreover, it also needs the optical flow calculation. DMAS is the fastest algorithm among the compared methods owing to the attentional sampling method. SABL also shows high efficiency in terms of the computation time comparable to MCD5.8ms. In case of PBAU, although it slightly takes more time than MCD NP [44] (16.08ms) and MCD 5.8ms [45] (5.74ms) due to the scene structure estimation and the twin processes, it still has a high efficiency in computation.

6.2.5 Unified framework

The unified framework covering all of the proposed schemes is evaluated over 22 video sequences described in Table 6.1, comparing it with the proposed individual schemes and the existing state of the arts. Figure 6.11 shows the pixel-wise F-measure for all videos, and Fig-

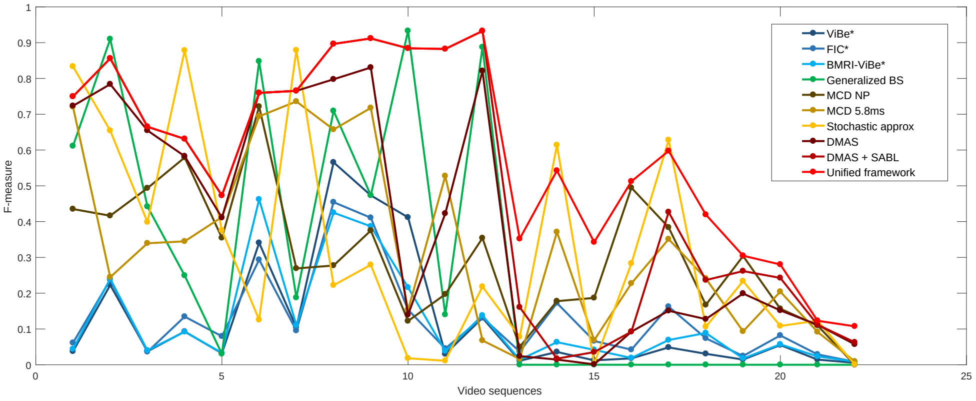


Figure 6.11: The pixel-wise F-measure results for all of 22 videos where x-axis indicates the video sequence no. listed in Table 6.1. * indicates that the background model was warped to compensate the camera motion for the algorithm which was designed for the stationary camera. DMAS means the dual-mode model with attentional sampling, and SABL means the situation-aware background learning.

ure 6.12 shows the averaged pixel-wise F-measure of each algorithm. Our approach showed good performance compared to the state-of-the-art methods, and the performance is improved with the addition of the proposed schemes (DMAS and SABL).

For the better visualization, we categorized 10 methods into four groups and drew the average F-measure plot for each group as shown in Figure 6.13: background subtraction for stationary camera + motion compensation (ViBE* [31], FIC* [21], BMRI-ViBe [33]), Object-centric approach (Generalized BS [3]), Background-centric methods (MCD NP [44], MCD 5.8ms [45], Stochastic approx [47]), and proposed methods (DMAS, DMAS+SABL, Unified framework). Although the object-centric approach (green line) show the best performance for three videos, this approach showed the unstable result in other videos. However, our unified framework showed the robust performance along the various videos (red line). The first three methods which belong to stationary + motion compensation showed poor performances because they did not consider the issues of the moving camera. The object-centric method showed good performance slightly, but it needs many computations and several assumptions for the target object. Our approach showed robust performance on various videos

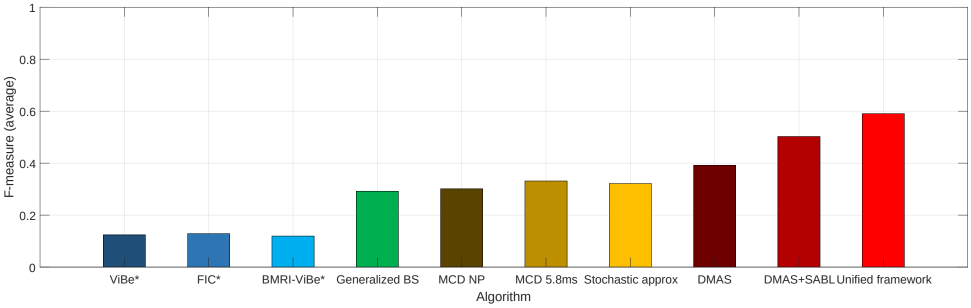


Figure 6.12: The average F-measure of each algorithm: ViBe* [31], FIC* [21], BMRI-ViBe [33], Generalized BS [3], MCD NP [44], MCD 5.8ms [45], Stochastic approx [47], Dual-mode model with attentional sampling (DAS), Situation-aware background learning (SABL), and Unified framework.

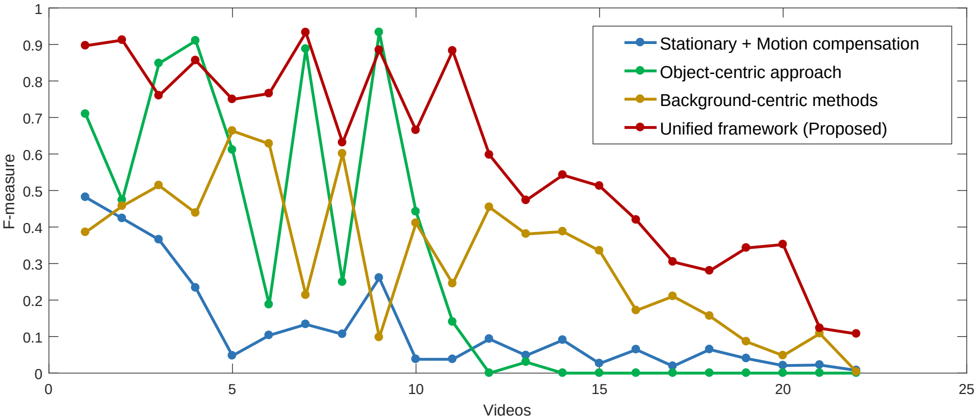


Figure 6.13: The average F-measure of each group: Stationary + motion compensation (ViBe* [31], FIC* [21], BMRI-ViBe* [33]), Object-centric approach (Generalized BS [3]), Background-centric methods (MCD NP [44], MCD 5.8ms [45], Stochastic approx [47]), and the unified framework.

compared to the state-of-the-art methods.

6.3 Application: combining with recognition algorithm

In the research for object detection and recognition, the numerous methods for extracting object candidates have been proposed to reduce a search space and computations [76–81]. This approach is based on a single image, and at least $1,000$ candidates are used. In a video case, however, these methods are ineffective because the redundant candidates can be effectively removed by using the temporal clue. Therefore, we combine the recognition algorithm with our foreground result, which enables to recognize a moving object with only a few candidates (about 10). Figure 6.14 shows the framework for moving object recognition. From the final foreground, we generate several boxes as the moving object proposals. The image patch of each box is tested by a pre-trained recognition network. Finally, this network gives both the class label and corresponding segmentation.

Before extracting patches in the foreground, we first apply the blob analysis [72] and remain large blobs whose area is greater than 100. Then, we extract the patch of which area is 1.5 times bigger than its original size to cover a missing region. Extracted patches are rescaled as a 224×224 image to fit an input size of the pre-trained network. In the choice of recognition network, we choose the fully convolutional network (FCN) [82] that produces both the class label and the segmentation result. Although there are so many famous networks for the recognition such as AlexNet [83], VGGNet [1], and ResNet [84], they do not generate a realistic recognition because they are trained using $1,000$ classes. Figure 6.15 shows the recognition result through VGGNet. Although we expect a pedestrian or man for the input image in Figure 6.15(a), the algorithm recognizes this input as a duck. As shown in Figure 6.15(b) is tested, the network generates wrong classes like a device though it looks like a small blur or noise. Fortunately, the FCN adopted in our experiment is the fine-tuned network using more practical classes such as a bicycle, boat, car, person, dog, etc. Also, the FCN class can remove the noise or non-object because FCN is fine-tuned using the *'background'* class. Therefore, the inevitable noises from the foreground can be filtered out in the

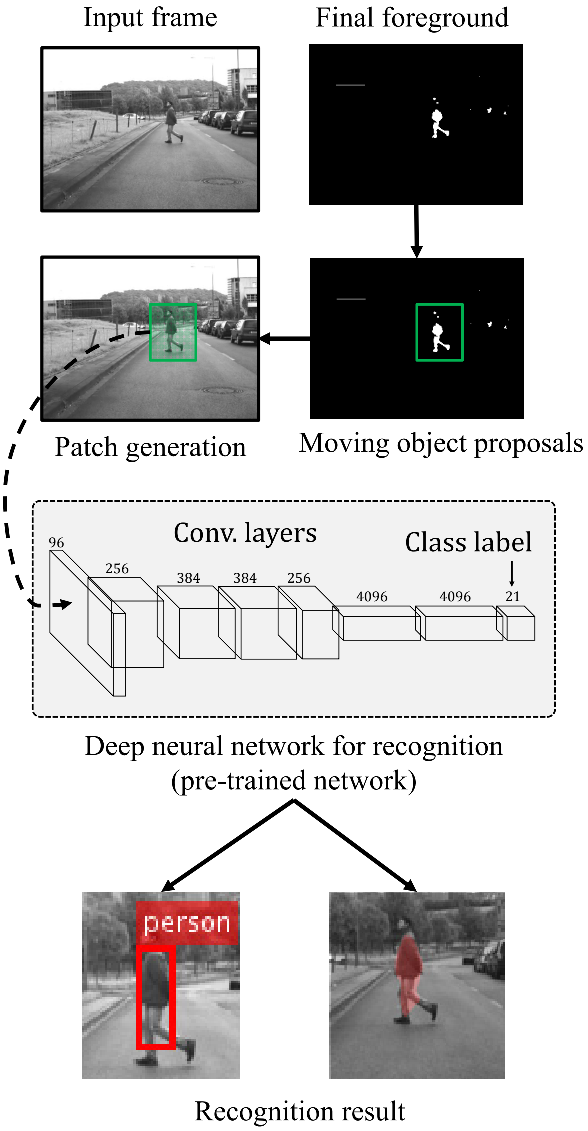
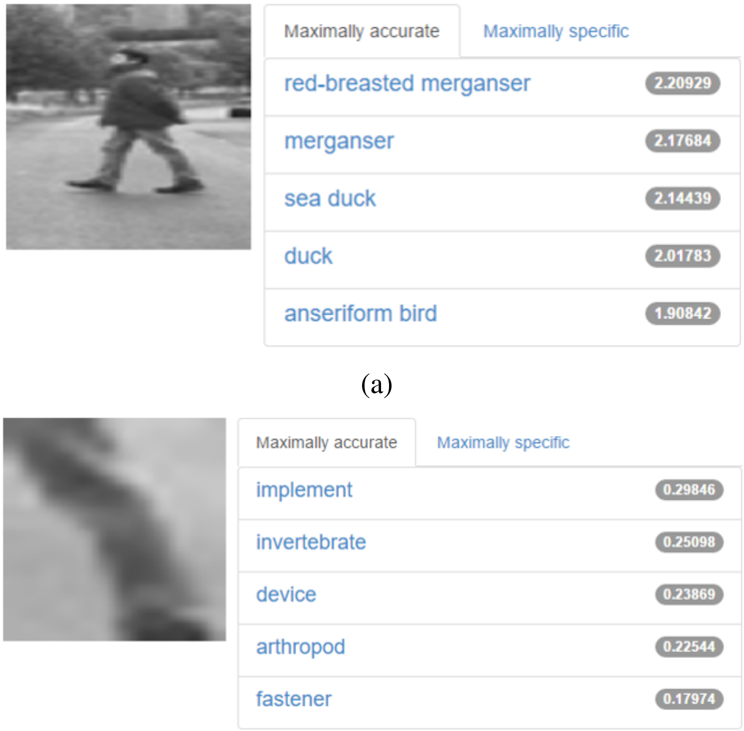


Figure 6.14: The framework for moving object recognition in a dashcam video. From the foreground result, the candidate regions of moving object are extracted. Then, patches on the candidate regions are tested by the pre-trained deep neural network.



red-breasted merganser	2.20929
merganser	2.17684
sea duck	2.14439
duck	2.01783
anseriform bird	1.90842

(a)

implement	0.29846
invertebrate	0.25098
device	0.23869
arthropod	0.22544
fastener	0.17974

(b)

Figure 6.15: The recognition result from the VGGNet [1]. Although the VGGNet showed a good performance in the ImageNet classification with 1000 class, it is far from the realistic classification.

recognition step.

Figure 6.16 and Figure 6.17 show the qualitative results of the recognition based on our foreground result. Contrary to the pre-trained networks based on ImageNet, the FCN produces the realistic class label as a person or car as shown in Figure 6.16. As shown in Figure 6.17, the network does not find a precise boundary such as a pedestrian leg, but it finds a quite reasonable region according to the class. Lastly, we compare the baseline FCN that uses an entire image to the FCN with a few candidates from our foreground. Figure 6.18 shows the comparison result. Of course, the baseline FCN finds a static object such as a parked car in Figure 6.18(a). However, the baseline misses small objects because the image

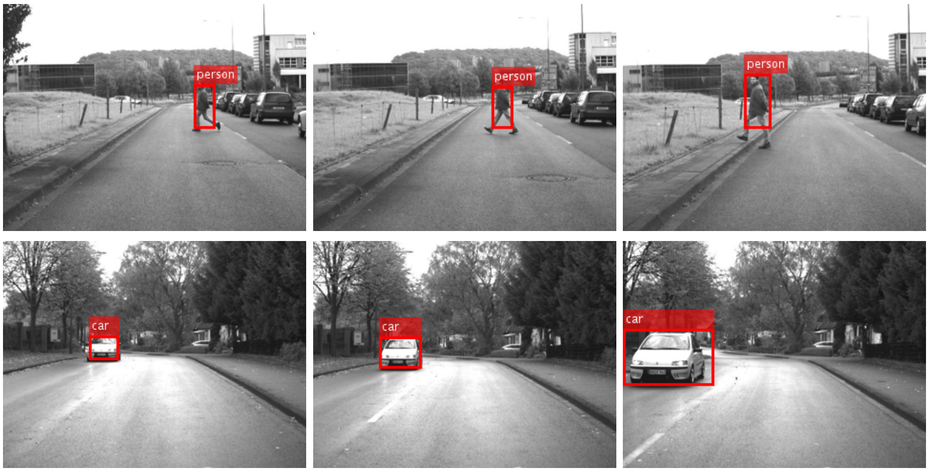


Figure 6.16: The recognition result of FCN combined with our foreground through the red box with a class label.

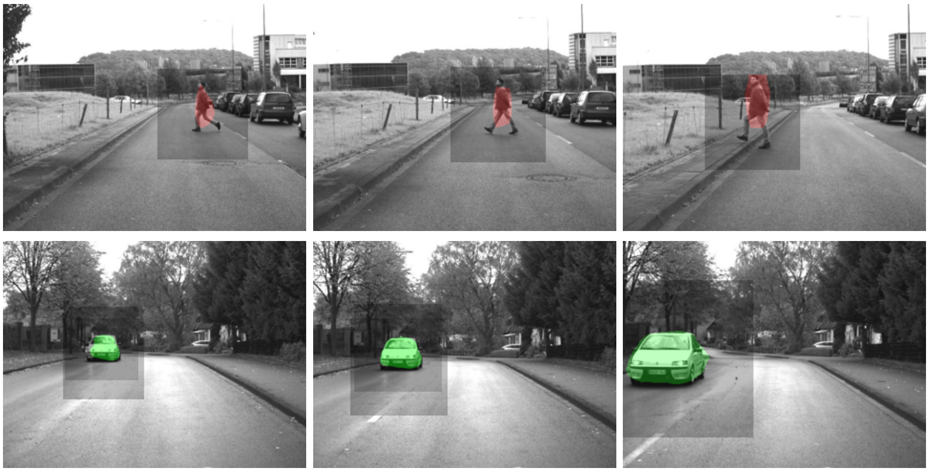
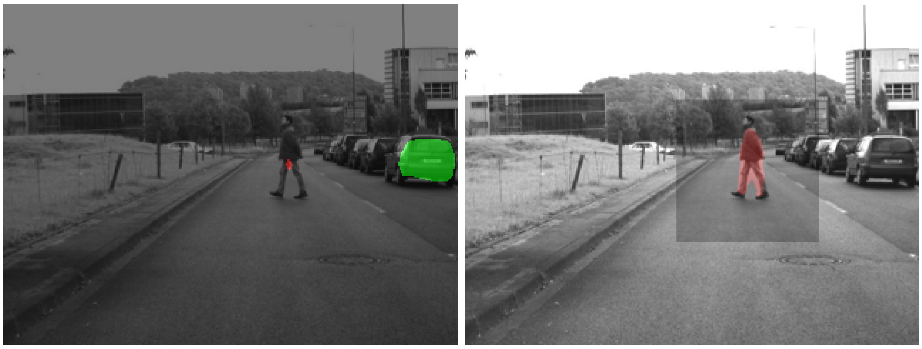


Figure 6.17: The recognition result of FCN combined with our foreground with segmentation. Along the class, the color of the segmented region is changed.



(a) baseline FCN (b) FCN with our foreground

Figure 6.18: The comparison result with baseline FCN and FCN based on our foreground.

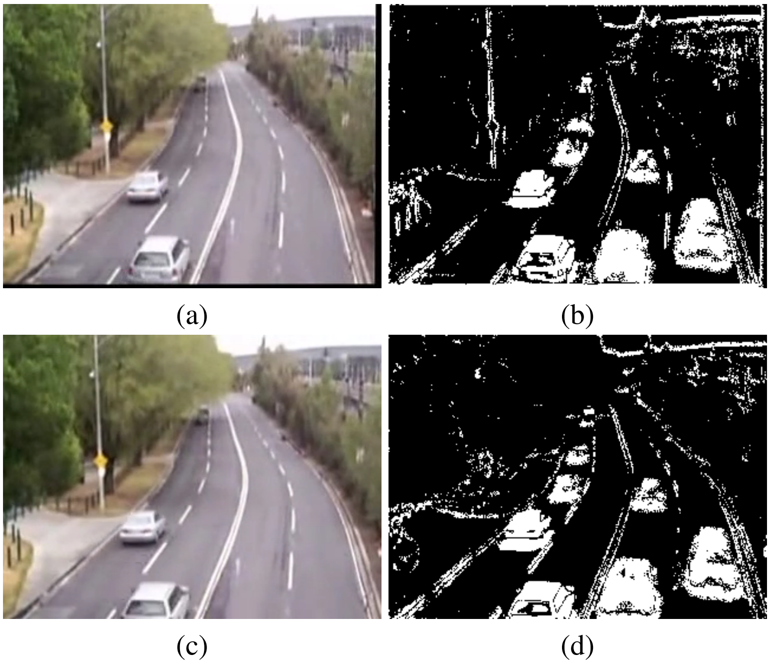
data used in training contain a large object near the image center. On the contrary, the FCN with our foreground detects and recognizes the moving objects because the object proposals of our method enable to focus on the object of interest.

6.4 Discussion

The proposed framework showed a good performance compared to the state-of-the-art methods and needed the low computations. However, the videos capture by moving cameras are numerous, and our framework cannot guarantee the good performance for entire cases. Therefore, we consider several issues and summarize the strength and weakness of our framework in this section.

6.4.1 Issues

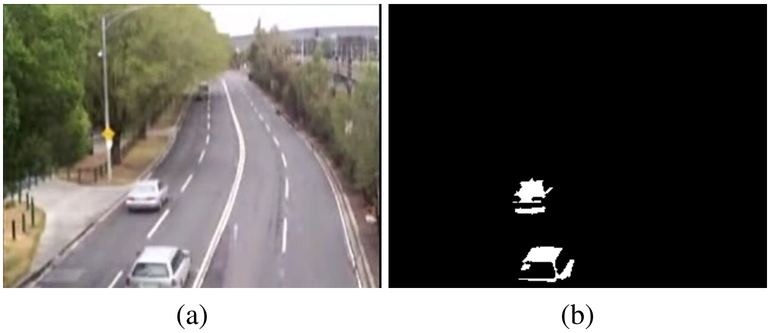
As a naive approach, if the moving video is converted to the stationary video through the video stabilization, we can think that the background subtraction method for the stationary camera is enough to find the moving object. However, the video stabilization also has a limitation to remove the camera motion effect. Figure 6.19 shows the moving object detection result when the background subtraction for the stationary camera is applied to the stabilized video. These detection results have many false positives around edges because the stabiliza-



(a) (b)

(c) (d)

Figure 6.19: The detection result when stationary background subtraction is applied to the stabilized video: (a) stabilized frame by [85], (b) detection result of (a), (c) stabilized frame by [86], (d) detection result of (c).



(a) (b)

Figure 6.20: The detection result when the proposed method applied without stabilization: (a) input frame (b) detection result.

tion algorithm does not remove the effect of camera motion. On the contrary, the proposed framework works well as shown in Figure 6.20 because it is designed by considering the properties of the moving videos.

The other issues are the unimportant moving pixels. Our framework detects the moving pixels as the counterpart of the built background. Therefore, the foreground from our framework contains the unimportant moving objects such as the rain and the windscreen wipers in a dashcam. Several predefined rules can be applied to remove these unimportant moving pixels, but it is impossible to define the rules for the numerous videos. Instead, we think that the combining with recognition algorithm is a good solution as described in Section 6.3. When a human perceives a moving object, human first recognizes that something is moving in a very short time. Then human understands what is the object and pays attention to it when the object is interesting. Our framework can operate as fast detection of the change and give the location cues to recognize the important moving object.

6.4.2 Strength

Our framework consists of three main schemes: dual-mode modeling with attentional sampling, situation-aware background learning, and prior-based attentional update. Although we showed the effect of each scheme in the qualitative results, we mention the situations to produce the good result for each scheme empirically. The dual-mode modeling with attentional sampling is well operated when the target object is determined such as tracking situation. Because the spatio-temporal properties are used to build the sampling map, these properties are based on the foreground occurrence. Since the camera is moved to focus the target object in the tracking video, the probability of foreground occurrence is well established.

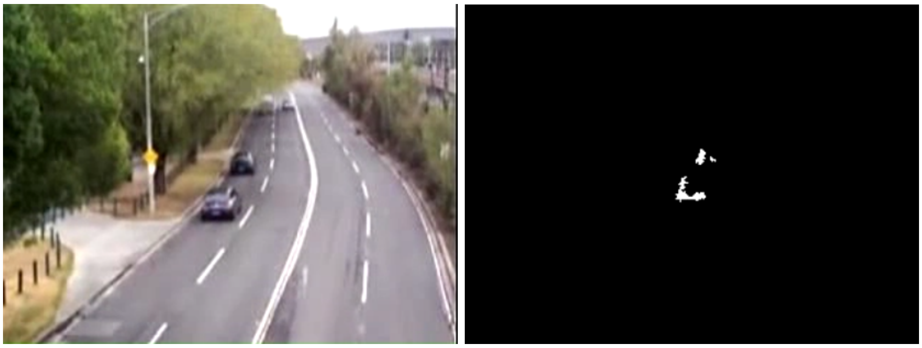
Second, the situation-aware background learning is effective when the camera monitors the wide area. When the camera monitors the wide area such as PTZ camera, the overall brightness of scene is changed by the auto exposure function and the situation-aware background learning is useful. When the object is moved vertically, the situation-aware background learning is effective. In the perspective of background modeling, since the vertical movement produces small change region, it can be ignored by the noise. However, the situation-aware background learning controls the update frequency to discriminate the noise and small moving object and enable to detect the moving object.

Third, the prior-attentional update is obviously effective to dashcam videos. Although it is designed to find the sky and road region, the detection result is not degraded when there are no sky or road region in the scene. The center-focus tendency is more powerful to build the background for dashcam video, and the road and sky region play a role to reduce the unimportant changes in those regions.

6.4.3 Limitation

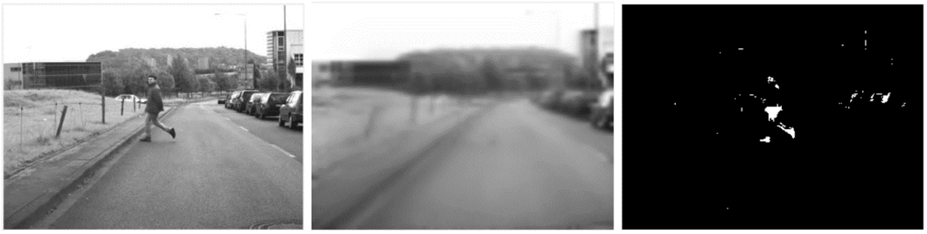
Since our method is based on the background-centric approach, the background region should be larger than the foreground region. In case of the webcam, if a person occupies more than half of the scene, our method fails to model the background because the motion estimation is severely affected by the foreground. This problem occupies in case of other vision algorithms that contain the camera motion estimation step. If we utilize other sensors such as gyro as well as the vision sensor, the accurate camera motion can be estimated, and this problem can be effectively solved.

Another limitation is the foreground loss problem caused by the grid-based modeling and intensity modeling. Although the grid-based modeling enables to build a coarse background fast, it sometimes misses the small moving pixels when the moving distance is smaller than a grid size. Figure 6.21 shows the limitation of the grid-based modeling. When the car moves far away, the observed moving distance becomes short and the small part of the foreground is detected. The intensity modeling also causes the foreground loss as shown in Figure 6.22. Though the background in Figure 6.22(b) does not contain the pedestrian, the foreground suffers from the part loss problem as shown in Figure 6.22(c). This problem occurs because the intensity is not discriminative in finding the difference between the input image and back-



(a) (b)

Figure 6.21: The limitation caused by the grid-based modeling. By the perspective of the image, the moving distance on the image is short when the object moves from a distance. Thus, the changed area becomes small, so only a part of the object is detected.



(a) input frame (b) background (c) detection result

Figure 6.22: The limitation caused by the intensity modeling. Even if the background is built clearly as shown in (b), if the intensity of the object is similar to the intensity of the background, the loss of object region occurs as shown in (c).

ground. In other words, the intensities of pedestrian's upper part of the input image are similar to that of the corresponding location in the background. To solve this foreground loss problem, we can consider the combination with the high-level knowledge for the object. As shown in section 6.3, the high-level knowledge like the recognition can be the solution for the imperfect foreground result. As human perceives the changes on the scene and recognizes the moving object, the ultimate moving object detection should be combined with the recognition based on the high-level knowledge. We will describe this future approach in the chapter 7.

Chapter 7

Concluding remarks and Future works

In this thesis, we tackled the detection problems and developed efficient and robust methods for finding moving objects in a moving camera video. In chapter 3, based on the spatio-temporal properties of moving object, we accelerated the algorithm and reduced false positives using the attentional sampling. After the spatio-temporal properties was modeled by the foreground occurrence probability, a sampling map that selects the candidate pixels to find the actual objects was generated based on this probability. The background subtraction and dual-mode model update were applied to only the selected pixels. Lastly, the foreground was refined to reduce false detections and fill the foreground hole clearly using the foreground occurrence probability.

In chapter 4, the situation-aware background modeling was proposed to adaptively update the background model along the scene situation. In this a new scheme, the situation variables was estimated and the background model was adaptively updated. As well as the background update, the compensation of camera movement was modified according to the situation variables. Through the situation-aware scheme, a clear background model was obtained without contamination by the foreground. Also, a new foreground segmentation method was proposed using the watershed segmentation with two thresholds. This situation-aware scheme showed robust performance under various situations such as sudden illumination changes.

In chapter 5, the modified background modeling was proposed for dashcam video that has received the attention for smart mobility like an autonomous vehicle. From the motivation of the center-focused and structure-focused tendencies of human attention, the background modeling was extended to focus on the center changes and neglect minor changes on the scene structure. By increasing the learning rate of the boundary region, the center-focused tendency was implemented through the multiplication of the attention map and the age model. To implement the structure-focused tendency, the road and sky region which give scene structure information in a dashcam were estimated. Then, estimated scene information was used to build a robust background model through the model selection. Lastly, the final foreground was obtained by combining two foregrounds from twin processes using original image and the median-filtered image to emboss the foreground region. Also, through the combination with the recognition algorithm using the deep neural network, we showed the application of our method for the efficient moving object proposal method.

The unified framework, even covering all of the proposed schemes, satisfies the real-time performance which is important to an actual application for a pre-processing. Moreover, since our framework run in an unsupervised and online manner, it is a less constrained and less sensitive than other methods. Due to the usability of our methods, our detection method was used in the PTZ tracking [87]. We expect that our methods can be widely utilized of many applications in the moving camera.

Even though the proposed methods showed good performances in videos captured from various moving cameras, there are still many future works to improve the performance. First of all, because only the vision-based camera motion estimation has an inevitable estimations error, this motion estimation can be improved by other sensors. That is, additional sensor information to estimate camera motion will help to overcome the limitation of the homography model. Although we keep the homography model due to the algorithm efficiency, the forward motion in a dashcam and the rise motion in a drone cannot be represented well by the single homography model. However, if the additional cues of camera motion from other sensors are given, we can use more suitable motion model for representing the camera motion and reduce the false detections as a result.

Another future approach is the combination with a deep neural network. In chapter 6.3, we showed that our foreground could be used as a moving object proposal for the recognition algorithm. In this combination, the detection module and recognition module of the moving object were independently operated. If an integrated algorithm is developed to detect and recognize the moving object simultaneously, it produces the successful synergy. In other words, the current moving object detection is based on the pixel-wise detection that does not use the high-level information such as object class and scene structure. Through the deep neural network which extracts and builds the high-level knowledge, many problems on the moving object detection can be solved. That is, the false positives from the compensation error can be easily removed because the trained neural network discriminates the objectness using the high-level knowledge. Although the issues for the network design and the dataset with ground truths, the integrated work of detection and recognition will be a good future work.

Bibliography

[1] Karen Simonyan and Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *International Conference on Learning Representations*, Sept. 2014.

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele, “People-tracking-by-detection and people-detection-by-tracking,” in *Computer Vision and Pattern Recognition (CVPR)*, 2008.

[3] S Kwak, T Lim, W Nam, B Han, and J H Han, “Generalized background subtraction based on hybrid inference by belief propagation and Bayesian filtering,” in *International Conference on Computer Vision (ICCV)*, 2011.

[4] Ali Elqursh and Ahmed M Elgammal, “Online Moving Camera Background Subtraction.,” in *European Conference on Computer Vision (ECCV)*, 2012.

[5] Ajay K Mishra, Yiannis Aloimonos, Loong-Fah Cheong, and Ashraf Kassim, “Active Visual Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 639–653, 2012.

[6] C Jung and C Kim, “A unified spectral-domain approach for saliency detection and its application to automatic object segmentation,” *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1272–1283, 2012.

[7] Peter Ochs, Jitendra Malik, and Thomas Brox, “Segmentation of Moving Objects by Long Term Video Analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1187–1200, May 2014.

[8] Won-Dong Jang, Chulwoo Lee, and Chang-Su Kim, “Primary Object Segmentation in Videos via Alternate Convex Optimization of Foreground and Background Distributions,” in *Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 696–704.

[9] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi, “Video segmentation by tracing discontinuities in a trajectory embedding,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1846–1853.

[10] D Zhang, O Javed, and M Shah, “Video object segmentation through spatially accurate and temporally dense extraction of primary object regions,” in *Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 628–635, IEEE.

[11] A Faktor and M Irani, “Video Segmentation by Non-Local Consensus voting.,” in *British Machine Vision Conference (BMVC)*, 2014.

[12] Anestis Papazoglou and Vittorio Ferrari, “Fast object segmentation in unconstrained video,” in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1777–1784.

[13] Chris Stauffer and W E L Grimson, “Adaptive background mixture models for real-time tracking,” in *Computer Vision and Pattern Recognition (CVPR)*, 1999.

[14] Hyung Jin Chang, Kwang Moo Yi, Shimin Yin, Soo Wan Kim, Young Min Baek, Ho Seok Ahn, and Jin Young Choi, “PIL-EYE: Integrated System for Sustainable Development of Intelligent Visual Surveillance Algorithms,” in *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. 2011, pp. 231–236, IEEE.

[15] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, July 1997.

[16] S C Huang, "An advanced motion detection algorithm with video quality analysis for video surveillance systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 1, pp. 1–14, 2011.

[17] Chris Stauffer and W Eric L Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition (CVPR)*, 1999.

[18] Y Sheikh and M Shah, "Bayesian object detection in dynamic scenes," in *Computer Vision and Pattern Recognition (CVPR)*, 2005.

[19] Dar-Shyang Lee, "Effective Gaussian Mixture Learning for Video Background Subtraction.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 827–832, 2005.

[20] Teresa Ko, Stefano Soatto, and Deborah Estrin, "Warping background subtraction.," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[21] JinMin Choi, Hyung Jin Chang, Yung Jun Yoo, and Jin Young Choi, "Robust moving object detection against fast illumination change," *Computer Vision and Image Understanding*, vol. 116, no. 2, Feb. 2012.

[22] Ahmed M Elgammal, David Harwood, and Larry S Davis, "Non-parametric Model for Background Subtraction.," in *European Conference on Computer Vision (ECCV)*, 2000.

[23] Kyungnam Kim, Thanarat H Chalidabhongse, David Harwood, and Larry S Davis, "Real-time foreground-background segmentation using codebook model.," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[24] Jing-Ming Guo, Chih-Hsien Hsia, Yun-Fu Liu, Min-Hsiung Shih, Cheng-Hsin Chang, and Jing-Yu Wu, "Fast Background Subtraction Based on a Multilayer Codebook Model for Moving Object Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 10, pp. 1809–1821, Sept. 2013.

[25] L Maddalena and A Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1168–1177, 2008.

[26] B H Chen and S C Huang, "An advanced moving object detection algorithm for automatic traffic monitoring in real-world limited bandwidth networks," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 837–847, 2014.

[27] Bo-Hao Chen and Shih-Chia Huang, "Probabilistic neural networks based moving vehicles extraction algorithm for intelligent traffic surveillance systems," *Information Sciences: an International Journal*, vol. 299, no. C, Apr. 2015.

[28] S C Huang and B H Chen, "Highly accurate moving object detection in variable bit rate video-based traffic monitoring systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 12, pp. 1920–1931, 2013.

[29] S C Huang and B H Chen, "Automatic moving object extraction through a real-world variable-bandwidth network for traffic monitoring systems," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 4, pp. 2099–2112, 2014.

[30] Xiaowei Zhou, Can Yang, and Weichuan 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, Mar. 2013.

[31] Olivier Barnich and Marc Van Droogenbroeck, "ViBe: A Universal Background Subtraction Algorithm for Video Sequences.," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2011.

[32] P St-Charles, G Bilodeau, and R Bergevin, "SuBSENSE: A Universal Change Detection Method With Local Adaptive Sensitivity," *Image Processing, IEEE Transactions on*, vol. 24, no. 1, pp. 359–373, 2015.

[33] F C Cheng, B H Chen, and S C Huang, "A background model re-initialization method based on sudden luminance change detection," *Engineering Applications of Artificial Intelligence*, 2015.

[34] Fan-Chieh Cheng, Bo-Hao Chen, and Shih-Chia Huang, "A Hybrid Background Subtraction Method with Background and Foreground Candidates Detection," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 7, no. 1, Oct. 2015.

[35] Donatello Conte, Pasquale Foggia, Gennaro Percannella, Francesco Tufano, and Mario Vento, "An experimental evaluation of foreground detection algorithms in real scenes," *EURASIP Journal on Advances in Signal Processing*, 2010.

[36] Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012.

[37] Yi Wang, Pierre-Marc Jodoin, Fatih Murat Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar, "CDnet 2014 - An Expanded Change Detection Benchmark Dataset.," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014.

[38] Kiran S Bhat, Mahesh Saptharishi, and Pradeep K Khosla, "Motion Detection and Segmentation using Image Mosaics.," in *IEEE International Conference on Multimedia and Expo*, 2000.

[39] Anurag Mittal and Daniel P Huttenlocher, "Scene Modeling for Wide Area Surveillance and Image Synthesis.," in *Computer Vision and Pattern Recognition (CVPR)*, 2000.

[40] Rita Cucchiara, Andrea Prati, and Roberto Vezzani, "Advanced video surveillance with pan tilt zoom cameras," in *Proc. of the 6th IEEE International Workshop on Visual Surveillance on ECCV*, 2006.

[41] Constant Guillot, Maxime Taron, Patrick Sayd, Quoc Cuong Pham, Christophe Tilmant, and Jean-Marc Lavest, "Background subtraction adapted to PTZ cameras by keypoint density estimation," in *British Machine Vision Conference (BMVC)*, 2006.

[42] Lionel Robinault, Stéphane Bres, and Serge Miguet, "Real Time Foreground Object Detection using PTZ Camera.," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2009.

[43] Slim Amri, Walid Barhoumi, and Ezzeddine Zagrouba, "A robust framework for joint background/foreground segmentation of complex video scenes filmed with freely moving camera," *Multimedia tools and applications*, vol. 46, no. 2-3, Jan. 2010.

[44] Soo Wan Kim, Kimin Yun, Kwang Moo Yi, Sun Jung Kim, and Jin Young Choi, "Detection of moving objects with a moving camera using non-panoramic background model," *Machine Vision and Applications*, 2012.

[45] Kwang Moo Yi, Kimin Yun, Soo Wan Kim, Hyung Jin Chang, Hawook Jeong, and Jin Young Choi, "Detection of Moving Objects with Non-stationary Cameras in 5.8ms: Bringing Motion Detection to Your Mobile Device," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013.

[46] Jiman Kim, Xiaofei Wang, Hai Wang, Chunsheng Zhu, and Daijin Kim, "Fast moving object detection with non-stationary background," *Multimedia tools and applications*, vol. 67, no. 1, pp. 311–335, 2013.

[47] Francisco Javier López-Rubio and Ezequiel López-Rubio, "Foreground detection for moving cameras with stochastic approximation," *Pattern Recognition Letters*, vol. 68, pp. 161–168, 2015.

[48] Wu-Chih Hu, Chao-Ho Chen, Tsong-Yi Chen, Deng-Yuan Huang, and Zong-Che Wu, "Moving object detection and tracking from video captured by moving camera," *Journal of Visual Communication and Image Representation*, vol. 30, no. C, July 2015.

[49] Tsubasa Minematsu, Hideaki Uchiyama, Atsushi Shimada, Hajime Nagahara, and Rinichiro Taniguchi, "Evaluation of foreground detection methodology for a moving camera," in *Frontiers of Computer Vision (FCV), 2015 21st Korea-Japan Joint Workshop on*, 2015.

[50] Dieter Koller, Joseph Weber, and Jitendra Malik, "Robust multiple car tracking with occlusion reasoning," in *European Conference on Computer Vision*. Springer, 1994, pp. 189–196.

[51] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[52] Richard Szeliski, "Image alignment and stitching: A tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104, 2006.

[53] C Liu, *Beyond pixels: exploring new representations and applications for motion analysis*, Doctoral thesis, Massachusetts Institute of Technology, 2009.

[54] T Brox and J Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[55] Carlo Tomasi and Takeo Kanade, “Detection and tracking of point features,” Tech. Rep., IJCV, 1991.

[56] Martin A Fischler and Robert C Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[57] R A Rensink, “Change detection,” *Annual review of psychology*, 2002.

[58] Hyung Jin Chang, Hawook Jeong, and Jin Young Choi, “Active attentional sampling for speed-up of background subtraction,” in *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[59] Fernand Meyer, “Topographic distance and watershed lines,” *Signal Processing*, vol. 38, no. 1, pp. 113–125, July 1994.

[60] Fernand Meyer, “Topographic distance and watershed lines,” *Signal Processing*, vol. 38, no. 1, pp. 113–125, July 1994.

[61] F Liu, M Gleicher, H Jin, and A Agarwala, “Content-preserving warps for 3D video stabilization,” *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 44, 2009.

[62] Junhong Gao, Seon Joo Kim, and Michael S Brown, “Constructing image panoramas using dual-homography warping,” in *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[63] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun, “Bundled camera paths for video stabilization,” *ACM Trans Graph*, vol. 32, no. 4, pp. 1, July 2013.

[64] John Canny, “A Computational Approach to Edge Detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[65] Kimin Yun and Jin Young Choi, "Robust and fast moving object detection in a non-stationary camera via foreground probability based sampling.," in *International Conference on Image Processing (ICIP)*. 2015, pp. 4897–4901, IEEE.

[66] Ronald A Rensink, J Kevin O'Regan, and James J Clark, "To See or not to See: The Need for Attention to Perceive Changes in Scenes," *Psychological science*, vol. 8, no. 5, pp. 368–373, Sept. 1997.

[67] B W Tatler, "The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions," *Journal of Vision*, vol. 7, no. 14, pp. 4–4, Nov. 2007.

[68] Dario D Salvucci, "A Multitasking General Executive for Compound Continuous Tasks.," *Cognitive Science*, vol. 29, no. 3, pp. 457–492, 2005.

[69] Tilke Judd, Krista A Ehinger, Frédo Durand, and Antonio Torralba, "Learning to predict where humans look.," in *International Conference on Computer Vision (ICCV)*. 2009, pp. 2106–2113, IEEE.

[70] José Manuel Alvarez, Theo Gevers, and Antonio M Lopez, "3D Scene priors for road detection.," in *Computer Vision and Pattern Recognition (CVPR)*. 2010, pp. 57–64, IEEE.

[71] José Manuel Alvarez, Theo Gevers, Yann LeCun, and Antonio M Lopez, "Road Scene Segmentation from a Single Image.," in *European Conference on Computer Vision (ECCV)*, Berlin, Heidelberg, 2012, pp. 376–389, Springer Berlin Heidelberg.

[72] Robert M. Haralick and Linda G. Shapiro, *Computer and Robot Vision*, vol. 1, Addison-Wesley, 1992.

[73] M Enzweiler and D M Gavrila, "Monocular Pedestrian Detection: Survey and Experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2179–2195, Oct. 2009.

[74] Amit Adam, Ehud Rivlin, and Ilan Shimshoni, "Robust Fragments-based Tracking using the Integral Histogram," in *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[75] Martin Godec, Peter M Roth, and Horst Bischof, "Hough-based tracking of non-rigid objects," in *International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 81–88.

[76] Hawook Jeong, Sangdoo Yun, Kwang Moo Yi, and Jin Young Choi, "Category Attentional Search for Fast Object Detection by Mimicking Human Visual Perception," in *Winter Conference on Applications of Computer Vision (WACV)*. 2015, pp. 829–836, IEEE.

[77] S Manen, M Guillaumin, and L Van Gool, "Prime object proposals with randomized Prim's algorithm," in *International Conference on Computer Vision (ICCV)*. 2013, pp. 2536–2543, IEEE.

[78] C . Lawrence Zitnick and Piotr Dollar, "Edge Boxes: Locating Object Proposals from Edges.," in *European Conference on Computer Vision (ECCV)*, Cham, 2014, pp. 391–405, Springer International Publishing.

[79] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip H S Torr, "BING: Binarized Normed Gradients for Objectness Estimation at 300fps.," in *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[80] JRR Uijlings, KEA van de Sande, and T Gevers, "Selective search for object recognition," *International Journal of Computer Vision*, 2013.

[81] Sangdoo Yun, Hawook Jeong, Soo Wan Kim, and Jin Young Choi, "Voting-based 3D object cuboid detection robust to partial occlusion from RGB-D images," in *Winter Conference on Applications of Computer Vision (WACV)*. 2016, pp. 1–8, IEEE.

[82] J Long, E Shelhamer, and T Darrell, "Fully convolutional networks for semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440, IEEE.

[83] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[84] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," in *Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[85] Kimin Yun, Soo Wan Kim, and Jin Young Choi, "Probabilistic Approach with Three Hierarchies of Motion Estimation for Video Stabilization," in *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. Nov. 2011, pp. 262–267, IEEE.

[86] Matthias Grundmann, Vivek Kwatra, and Irfan Essa, "Auto-directed video stabilization with robust l1 optimal camera paths," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 225–232.

[87] Byeongju Lee, Kimin Yun, Jongwon Choi, and Jin Young Choi, "Robust pan-tilt-zoom tracking via optimization combining motion features and appearance correlations.," in *Advanced Video and Signal-Based Surveillance (AVSS)*. 2015, pp. 1–6, IEEE.

초록

본 연구에서는 다양한 동적 카메라에서 촬영된 영상에서 동적 물체 탐지를 위한 배경 중심 접근의 프레임워크를 제안한다. 이 방법은 기존의 객체 중심 접근법의 큰 문제인 연산량 문제를 크게 개선하여, 실용적으로 사용할 수 없던 기존의 방법들에 비해 효율적이고 성능도 우수한 새로운 방법이다. 기존의 정지 카메라에서 사용되어온 배경 중심 접근 방법은 부정확한 카메라 움직임 추정으로 인해 오탐지 문제, 카메라 시야의 변화로 인한 조명변화와 같은 급격한 변화, 카메라 움직임과 물체의 움직임을 구별 하기 어려운 상황에서 놓침 문제, 그리고 자동차에 설치된 카메라에서의 문제 등 다양한 문제가 발생한다. 이러한 어려움들을 극복하기 위하여, 크게 세가지 부분으로 이루어진 하나의 프레임워크가 제안되었고 이를 통해 다양한 움직이는 카메라 영상에서 효율적이고 강인하게 물체를 탐지할 수 있게 하였다.

첫 번째로 카메라 움직임 보상과정에서 부정확한 카메라 움직임 추정으로 인해 발생하는 오탐지를 줄이기 위해서, 듀얼 모델링과 선택적 영역 탐지 알고리즘을 이용해 알고리즘을 가속화 하고 전경 영역의 훼손을 막는 모듈을 제안하였다. 듀얼 모델링은 전경 정보의 배경 모델 간섭을 방지하는 효과를 내고, 선택적 영역 탐지 알고리즘은 알고리즘을 더욱 가속화시켜준다. 선택적 영역 알고리즘은 영상에서 움직이는 물체가 나타나면 다음 시간에는 물체가 나타난 주변에서 물체가 나타날 것이라는 특성을 이용하여 물체가 나타날 확률 맵을 정의하고, 이를 기반으로 하여 선택적 탐색 방법을 적용하여 매우 빠른 탐지가 가능하도록 하였다.

두 번째로는 영상의 급격한 변화나 카메라의 움직과 객체 움직임을 구별하기 어려운 경우와 같이 특정 상황에서 성능이 급격하게 떨어지는 현상을 개선하기 위한 방법을 제안하였다. 이 방법은 장면의 변화에 따라서 스스로 상황을 인지하여, 이에 맞게 가변적으로 배경을 학습함으로서 움직이는 물체를 안정적으로 탐지할 수 있게 한다. 현재 영상에서 나타나는 상황을 인지하기 위해서 먼저 상황 변수라고 정의된

카메라 움직임, 전경 움직임, 그리고 조명 변화를 추정하고, 이에 따라 배경을 가변적으로 업데이트 해 나간다. 이를 통해 기존의 성능과 속도는 유지하면서, 변화가 급격한 영상에서도 눈에 띄게 성능이 개선되는 결과를 얻었다.

마지막으로 기존 프레임워크가 자동차에 부착된 카메라에서 잘 동작하지 못하는 점을 해결하기 위한 사전정보 기반의 선택적 업데이트 방법을 제안하였다. 카메라 움직임 모델의 한계로 인해서 블랙박스 영상 등에서는 움직이는 물체 탐지가 쉽지 않은데, 본 방법에서는 자동차 카메라에서 나타나는 사전정보를 이용하여 이 한계를 극복하고자 하였다. 주행 상황에서 사람이 영상의 전체적인 구조를 파악하고 중심부분에 집중하여 상황을 인지하는 것에서 아이디어를 얻어 배경 모델링을 개선하고 영상의 측면 부분에서 발생하는 급격한 변화를 대응하는 방법론을 제안하였다. 이를 통해, 기존 대부분의 방법이 주행 영상에서 배경 모델링과 움직이는 물체 탐지에 실패한 것과 달리 본 방법은 주행 영상에서도 배경 모델링과 움직이는 물체 탐지에서 성공적인 결과를 보여주었으며, 또한 인식 방법과의 결합을 통해 자율주행을 위한 활용처로서의 가능성도 확인하였다.

이 통합된 프레임워크는 움직이는 카메라 영상에서 발생하는 다양한 문제를 해결하면서 동작하게 설계되었고 또한 매우 빠르게 동작하여 실제 최신 방법들과의 정성적, 정량적 비교 평가를 통해 그 효율성을 검증하였다. 첫 번째 모듈들로 인해서 한 프레임 처리에 4.8ms밖에 걸리지 않는 효율성을 보여주었고, 두 번째 모듈들로 인해 기존의 일반적인 상황에서의 성능과 속도는 유지하면서, 변화가 급격한 영상에서도 눈에 띄게 성능이 개선되는 결과를 얻었다. 마지막 주행 상황 영상을 위한 모듈들을 통해 주행 영상에서도 배경 모델링과 움직이는 물체 탐지에서 성공적인 결과를 보여주었으며, 또한 인식 방법과의 결합을 통해 자율주행을 위한 활용처로서의 가능성도 확인하였다.

주요어: 움직이는 물체 탐지, 움직이는 카메라, 배경 모델링, 영상 감시

학번: 2010-20847