ABSTRACT

ADVANCED TRAFFIC VIDEO ANALYTICS FOR ROBUST TRAFFIC ACCIDENT DETECTION

by
Hadi Ghahremannezhad

Automatic traffic accident detection is an important task in traffic video analysis due to its key applications in developing intelligent transportation systems. Reducing the time delay between the occurrence of an accident and the dispatch of the first responders to the scene may help lower the mortality rate and save lives. Since 1980, many approaches have been presented for the automatic detection of incidents in traffic videos. In this dissertation, some challenging problems for accident detection in traffic videos are discussed and a new framework is presented in order to automatically detect single-vehicle and intersection traffic accidents in real-time.

First, a new foreground detection method is applied in order to detect the moving vehicles and subtract the ever-changing background in the traffic video frames captured by static or non-stationary cameras. For the traffic videos captured during day-time, the cast shadows degrade the performance of the foreground detection and road segmentation. A novel cast shadow detection method is therefore presented to detect and remove the shadows cast by moving vehicles and also the shadows cast by static objects on the road.

Second, a new method is presented to detect the region of interest (ROI), which applies the location of the moving vehicles and the initial road samples and extracts the discriminating features to segment the road region. After detecting the ROI, the moving direction of the traffic is estimated based on the rationale that the crashed vehicles often make rapid change of direction. Lastly, single-vehicle traffic accidents and trajectory conflicts are detected using the first-order logic decision-making system.

The experimental results using publicly available videos and a dataset provided by the New Jersey Department of Transportation (NJDOT) demonstrate the feasibility of the proposed methods. Additionally, the main challenges and future directions are discussed regarding (i) improving the performance of the foreground segmentation, (ii) reducing the computational complexity, and (iii) detecting other types of traffic accidents.

ADVANCED TRAFFIC VIDEO ANALYTICS
FOR ROBUST TRAFFIC ACCIDENT DETECTION

by
Hadi Ghahremannezhad

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Department of Computer Science

August 2023

APPROVAL PAGE

ADVANCED TRAFFIC VIDEO ANALYTICS FOR ROBUST TRAFFIC ACCIDENT DETECTION

Hadi Ghahremannezhad

Dr. Chengjun Liu, Dissertation Advisor — Date
Professor of Computer Science, NJIT

Dr. Ali Mili, Committee Member — Date
Professor of Computer Science, NJIT

Dr. Zhi Wei, Committee Member — Date
Professor of Computer Science, NJIT

Dr. Taro Narahara, Committee Member — Date
Associate Professor of Architecture and Design, NJIT

Dr. Ming-Ching Chang, Committee Member — Date
Assistant Professor of Computer Science, University of Albany, Albany, New York

BIOGRAPHICAL SKETCH

Author: Hadi Ghahremannezhad

Degree: Doctor of Philosophy

Date: August 2023

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Science,
  New Jersey institute of Technology, Newark, NJ, 2023
- Master of Science in Software Engineering,
  Shahid Beheshti University, Tehran, Iran, 2017
- Bachelor of Science in Software Engineering,
  Khajeh Nasir Toosi University of Technology, Tehran, Iran, 2014

Major: Computer Science

Presentations and Publications:

Hadi Ghahremannezhad, Chengjun Liu, and Hang Shi. Intelligent Traffic Video Analytics. *Intelligent Video Analytics: Clustering and Classification Applications*, CRC Press, Taylor & Francis Group, Boca Raton, FL, U.S.A., 2023.

Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. Object Detection in Traffic Videos: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):6780-6799, 2023.

Hadi Ghahremannezhad, Chengjun Liu, and Hang Shi. Traffic Surveillance Video Analytics: A Concise Survey. In *International Conference on Machine Learning and Data Mining*, July 16-21, 2022.

Hadi Ghahremannezhad, Chengjun Liu, and Hang Shi. Ammunition Component Classification Using Deep Learning. In *International Conference on Machine Learning and Data Mining*, July 16-21, 2022.

Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. Real-Time Hysteresis Foreground Detection in Video Captured by Moving Cameras. In *IEEE International Conference on Imaging Systems and Techniques*, June 21-23, 2022.

Hang Shi, **Hadi Ghahremannezhad**, and Chengjun Liu. Unsupervised Anomaly Detection in Traffic Surveillance Based on Global Foreground Modeling. In *IEEE International Conference on Imaging Systems and Techniques*, June 21-23, 2022.

Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. Real-Time Accident Detection in Traffic Surveillance Using Deep Learning. In *IEEE International Conference on Imaging Systems and Techniques*, June 21-23, 2022.

Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. Illumination-Aware Image Segmentation for Real-Time Moving Cast Shadow Suppression. In *IEEE International Conference on Imaging Systems and Techniques*, June 21-23, 2022.

Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. A New Online Approach for Moving Cast Shadow Suppression in Traffic Videos. In *IEEE International Conference on Intelligent Transportation Systems*, September 19-22, 2021.

Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. Anomalous Driving Detection for Traffic Surveillance Video Analysis. In *IEEE International Conference on Imaging Systems and Techniques*, August 24-26, 2021.

Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. Automatic Road Detection in Traffic Videos. In *IEEE International Conference on Big Data and Cloud Computing*, December 17-19, 2020.

Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. A New Adaptive Bidirectional Region-of-Interest Detection Method for Intelligent Traffic Video Analysis. In *IEEE International Conference on Artificial Intelligence and Knowledge Engineering*, December 10-12, 2020.

Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. Robust Road Region Extraction in Video Under Various Illumination and Weather Conditions. In *IEEE International Conference on Image Processing, Applications and Systems*, December 9-11, 2020.

Hang Shi, **Hadi Ghahremannezhad**, and Chengjun Liu. A Statistical Modeling Method for Road Recognition in Traffic Video Analytics. In *IEEE International Conference on Cognitive Infocommunications*, September 23-25, 2020.

Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. A Real Time Accident Detection Framework for Traffic Video Analysis. In *International Conference on Machine Learning and Data Mining*, July 18-23, 2020.

M. Faruque, **Hadi Ghahremannezhad**, and Chengjun Liu. Vehicle Classification in Video Using Deep Learning. In *International Conference on Machine Learning and Data Mining*, July 13-18, 2019.

To My Beloved Parents

ACKNOWLEDGMENT

Throughout the writing of this dissertation, I have received a great deal of support and assistance. First, I would like to thank my supervisor, Dr. Chengjun Liu, for his guidance and invaluable advice in formulating research questions and methodologies.

I would like to express my appreciation to Dr. Ali Mili, Dr. James Mchugh, Dr. Taro Narahara, Dr. Zhi Wei, and Dr. Ming-Ching Chang for taking the time to serve as committee members and for their insightful feedback.

My thanks to the Department of Computer Science for providing me with financial support. Also, to the New Jersey Department of Transportation (NSF grant 1647170) for funding our research.

In addition, I appreciate the kind assistance and academic advice provided by Dr. Reza Curtmola, Dr. Baruch Schieber, Ms. Angel Butler, and Ms. Kathy Thompson in the Computer Science department.

I am also thankful for all the support I received from my fellow graduate students, Hang Shi and Mohammad Omar Faruque.

Most importantly, I would like to express my sincere gratitude to my parents, Ms. Parizad Bahardoost, and Mr. Mohammad Ghahremannezhad, for their constant support and patience.

TABLE OF CONTENTS

Chapter Page

TABLE OF CONTENTS
(Continued)

Chapter Page

TABLE OF CONTENTS
(Continued)

Chapter Page

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

Figure Page

LIST OF FIGURES
(Continued)

Figure Page

LIST OF FIGURES
(Continued)

Figure Page

CHAPTER 1

INTRODUCTION

The ever-escalating demand for land transportation, along with the continuous growth in the number of motor vehicles and other road users, has given rise to several traffic-related issues, including congestion, safety, commuting delays, increased energy consumption, and negative environmental impacts. There is an inevitable need to monitor road traffic and develop strategies for enabling safer roadways, limiting environmental impacts and enhancing the mobility of transport networks. Due to the ineluctable requirement for smart traffic management along with the accelerated advancements in the fields of electronics, sensors, communication, information, and computers, has led to the development of intelligent transportation systems (ITSs).

The functioning capacity of intelligent transportation systems is heavily affected by the competence of traffic data collection via sensors and the performance of the algorithms designed for automatic data processing. This is why traffic surveillance cameras have become one of the most popular sensors used in ITS applications [61]. Traffic cameras are, on one hand, one of the most cost-effective sensor technologies due to their simple installation, provision of a rich source of visual data, and a vast area of coverage. On the other hand, the revolutionary breakthroughs that have emerged in the world of artificial intelligence (AI), especially in the field of computer vision, have enabled modern traffic management systems to effectively process the footage obtained from traffic cameras automatically. The data provided by the traffic surveillance cameras is used for a wide variety of applications, including vehicle counting, road-user classification, anomaly detection, traffic flow estimation, speed estimation, and incident detection.

The architecture of traffic video analytics systems involves several hierarchical steps that are taken to process the raw data and generate useful information. The main steps in the process of analyzing traffic surveillance videos are camera calibration; locating objects of interest; object tracking; region-of-interest (ROI) determination; and incident detection. Figure 1.1 illustrates the general architecture of intelligent traffic monitoring systems. As seen in the figure, among the core components of intelligent traffic video analytics, locating the objects is the most important step, as it serves as the basis for most of the other steps [167].

One of the most important applications in traffic management systems is traffic accident detection. Despite all the improvements in road and vehicle safety, car accidents have been one of the leading causes of fatalities in the world. Automatic detection and notification of traffic collisions can help reduce the accident response time and, consequently, decrease the number of fatalities. Since videos captured by camera sensors provide a large amount of information at a relatively low cost, they have been the focus of many vision-based traffic accident detection methods throughout the previous years [46].

As mentioned before, there are several integral components to traffic video analysis, including background subtraction, moving vehicle detection, vehicle tracking, and object classification. Statistical methods are more applicable in real-time systems due to their computational efficiency and generalizability. Background subtraction and object tracking are the core components of statistical video analysis methods. In order to detect the moving vehicles in traffic videos, most approaches tend to segment the moving foreground from the stationary background. Each video frame is compared with the background model, and the pixels with significantly different values are classified as foreground. Background subtraction is a prerequisite of many video analysis applications and has been studied intensely over the past decades [47, 53, 125, 128, 132, 172]. Among the foreground segmentation techniques, statistical approaches based on Gaussian mixture models (GMM) are widely used for their good performance and low computational cost. Specifically, in real-time traffic video analysis, GMM has proven to be one of the best methods for subtracting the background and detecting moving vehicles. Here, we have applied a new

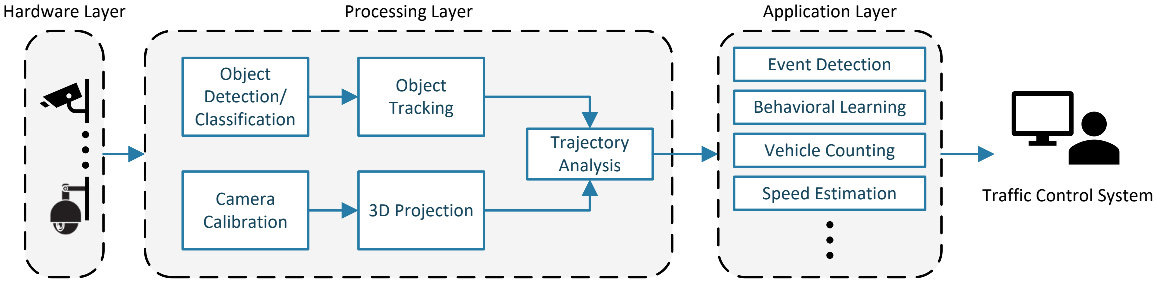


Figure 1.1 The general architecture of intelligent traffic monitoring systems. Locating objects of interest is a core component in the pipeline of these systems and the performance of the further steps is heavily reliant on this task.

foreground detection method, which is based on GMM. This approach is robust in dealing with moving cameras, stopped objects, and low-quality videos, which are common issues in the case of traffic video analysis.

Another challenge in traffic videos captured during daytime is the shadows cast by the static and moving objects [45, 48]. The shadows cast by moving vehicles are often classified as foreground due to their similar motion patterns to their corresponding objects. Moving cast shadows deteriorate the performance of the following video analysis tasks by linking different objects together or increasing the location estimation error. Also, the shadows cast by static objects on the scene cause the performance of image segmentation to reduce significantly, which in turn causes issues for the region of interest (ROI) determination. Thus, the shadows should be detected and removed from the foreground prior to taking further steps.

In terms of object tracking, there have been a large number of studies over the past years [90]. Generally, for the purpose of traffic video analysis, several vehicles are present in each frame, and all the vehicles should be tracked simultaneously. Therefore, multiple object tracking methods are preferred to process traffic videos. Tracking multiple objects in videos at the same time involves the detection of objects in each video frame and the association of the detected objects across multiple consecutive frames. With the improvements in object detection methods in recent years, tracking by detection has been the most studied approach in multi-target tracking. Some methods depend on the information from previous and future frames at the same time to deal with detection errors and improve the tracking performance [134]. Nevertheless, multiple object tracking (MOT) methods based on batch-wise strategies cannot be applied in real-world applications with no information about the future frames of videos. Another way to approach the tracking problem is to use only the information gained up to the current frame. The so-called online tracking strategies associate the detected objects in the frame and estimate the trajectories based on current and previous frames and can be utilized in real-world applications. There have been many attempts to improve the performance of the MOT methods both from the aspect of object detection and object association [11]. Some studies have targeted the MOT problem by improving the performance of the object detection step [67]. For the sake of computational efficiency, we have applied the simple blob-tracking method [18] that tracks each vehicle based on the distance between its centroid and the blob centroids in the previous frame.

The focus of this study is to develop an accident detection framework in traffic videos by automatically determining the region of interest and monitoring the motion behavior of vehicles in order to detect single-vehicle and intersection accidents. Specifically, an innovative real-time foreground detection method is presented that models the foreground and the background simultaneously and works for both moving and stationary cameras. In particular, first, each input video frame is partitioned into a number of blocks. Then, assuming the background takes the majority of each video frame, the iterative pyramidal implementation of the Lucas-Kanade optical flow approach is applied to the centers of the background blocks in order to estimate the global motion and compensate for the camera movements. Subsequently, each block in the background is modeled by a mixture of Gaussian distributions, and a separate Gaussian mixture model is constructed for the foreground in order to enhance the classification. However, the errors in motion compensation can contaminate the foreground model with background values. The novel idea of the proposed method is to match a set of background samples to their corresponding blocks for the most recent frames in order to avoid contaminating the foreground model with background samples. The input values that do not fit into either the statistical or the sample-based background models are used to update the foreground model. Finally, the foreground is detected by applying the Bayes classification technique to the major components in the background and foreground models, which removes the false positives caused by the hysteresis effect.

After background subtraction, the cast shadows are detected and removed from the foreground by a novel shadow removal method. The potential shadow pixels are identified by considering the physical properties of reflection and comparing the changes in luminance values in the corresponding background and foreground locations. The integrated features extracted from the RGB and HSV color spaces for each pixel are modeled by a mixture of Gaussian distributions to classify the foreground pixels into shadows and objects. The classified shadow and object pixels are clustered to detect the shadow regions and improve the results of the classification.

Furthermore, a new adaptive road detection method for determining the region of interest is presented. The initial road samples are obtained from the subtracted background model in the location of the moving vehicles. The integrated features extracted from both the grayscale and the RGB and HSV color spaces are further applied to construct several probability maps, which are then combined in order to estimate a more accurate road region map. The robust road mask is derived by integrating the initially estimated road region and the regions located by the flood-fill algorithm. Lastly, the moving direction of the traffic is estimated and traffic accidents are detected using the first-order logic decision-making system. Experimental results using real traffic video data show the feasibility of the proposed method. In particular, traffic accidents are detected in real time in the traffic videos without any false alarms.

This study is organized in the following manner: Chapter 2 outlines the previous related work that has approached the problem from various angles of view and compares the differences to our proposed method. Chapter 3 presents the main steps of the proposed foreground detection framework. Chapter 4 describes the new cast shadow detection and removal method, which is applied in order to remove cast shadows in the object detection step and to enhance the performance of the further steps. Chapter 5 contains details on initial road recognition and refining the extracted road region by using temporal and color features. Chapters 6 and 7 demonstrate the steps of the proposed method for traffic accident detection along with experimental results. Chapter 8 concludes and summarizes the work and outlines some future research directions.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Foreground Detection Methods

The main advantage of video data over images is the additional motion information provided by the temporal features that allow incorporating many signal processing techniques in the procedures of video analytics. In the case of surveillance applications, as an instance, exploiting the temporal features can be very useful due to the significant motion associated with the usual objects of interest compared to the stationary background. Therefore, identifying the pixel locations that are associated with considerable motion has been the focus of some of the closely related tasks in computer vision, such as background subtraction, foreground segmentation, change detection, and motion segmentation. As opposed to videos captured by in-vehicle cameras where the camera is mounted on a moving platform, traffic surveillance videos are usually captured by stationary camera sensors, and the objects of interest, which are usually moving vehicles, can be distinguished from the background solely based on the motion information. In addition to that, the input video data for traffic surveillance applications in intelligent transportation systems is live feeds from mostly low-quality camera sensors overlooking roads, highways, intersections, and other urban traffic environments with wide fields of view and few visual details of the target objects are available. Hence, motion segmentation techniques have proven to be more practical in real-time applications than image-based object detection methods due to their generalization and computational efficiency. Also, motion segmentation can be applied to generate hypotheses about object locations, followed by a feature extraction and classification step to improve the detection performance [66].

Most motion segmentation techniques applied in traffic video analytics consider the camera to be static and the target objects to have significant motion. There is large variety of mathematical [125], machine learning [35], signal processing [32], and classification [109] techniques proposed for background subtraction [40]. In spite of recent advances in this field, most real-world systems tend to apply relatively older techniques, such as MoG [131], AGMM [171], Codebook [71], Multi-Cue [107], PAWCS [130], PBAS [55], and ViBe [8] due to the limitations in computational capacity and the lack of collaboration between researchers and the industry [151]. Nonetheless, among various approaches for motion segmentation in the case of static cameras that are applied to traffic videos, frame differencing, optical flow, and statistical background modeling have been applied to traffic videos the most.

Frame differencing is the simplest motion estimation method in which the locations of the moving objects are estimated by calculating the absolute value of intensity difference between adjacent frames and applying a threshold to the results. Several studies have applied frame differencing to detect moving traffic objects such as vehicles [70]. Although this method is simple and fast, it is prone to errors and its performance suffers in many challenging scenarios, such as changes in illumination. One of the main drawbacks of this approach is the blank holes that appear in the foreground mask of objects due to the slow movement or relatively large parts of the object with uniform intensities. A number of studies have attempted to solve these issues by using three [75] or five [65] consecutive video frames.

Another approach for estimating the location of moving objects is to use the correlation between adjacent frames and find corresponding points so as to calculate the optical flow vector of the moving object, which describes the instantaneous velocity of a certain point in the image. The optical flow algorithm has been applied in the applications of traffic video analytics for various purposes, including motion-based object localization [21]. In the study conducted by Chen and Wu [21], the pyramid model of the Luas-Kanade optical flow algorithm is applied to a set of feature points that are extracted from the edges of the image. The feature points are clustered using the weighted Kmeans method in order to detect moving vehicles. The methods that are based on optical flow are not computationally as efficient as statistical modeling or temporal differencing; therefore, limiting the calculations to a lower frequency or a smaller reference region will help with achieving real-time performance.

In the applications of traffic video surveillance where the data is captured by static cameras, background modeling is by far the most popular approach for locating moving objects due to its compromise between efficiency and performance [40]. These methods benefit from the higher frequency in the intensity values corresponding to the stationary objects in the temporal domain compared to the moving objects in order to construct a background model. Each video frame is compared with the established background model and the spatial locations of the video frame with considerably different values from the current background model are classified as foreground, which represents the location of moving objects. In general, there are five groups of background subtraction methods, namely, basic, non-parametric, fuzzy, neural networks, and statistical methods [108]. The variations in the video quality and hardware capacity among video surveillance systems bring about an important requirement for background subtraction methods to be concurrently generalizable, robust, and efficient. This requirement has resulted in the methods based on statistical modeling being the most popular among background subtraction methods in real-time surveillance applications.

Most statistical background subtraction methods have attempted to establish the background model by the use of frame averaging [70], single Gaussian [106, 152], or a mixture of Gaussian distributions [173] with the majority tending to use Gaussian models. In the earlier studies each pixel was modeled with a single Gaussian distribution [152] and later the Gaussian Mixture Model (GMM) was proposed to model each pixel with a mixture of K Gaussian distributions in order to better deal with the effects of noise, camera jitter, and background texture [131]. Further improvements upon the GMM method were achieved by efficient parameter updating in adaptive GMM (AGMM) [171, 173] and other innovative techniques [125]. There are other representative background modeling methods, such as Vibe [8], PBAS [55], and Codebook [71] that have been applied in surveillance applications.

Background subtraction methods have been applied to traffic videos in a large number of studies [40, 122]. Shi and Liu [125] construct twelve-dimensional feature vectors from the values in the RGB, YIQ, and YCbCr color-spaces, the horizontal and vertical Haar wavelets, and the temporal difference, and establish a global foreground model along with the local background model in order to improve the discrimination and classification performance of the MoG method for vehicle detection. In the study done by Chetouane et al. [29], Gaussian Mixture Model (GMM), GMM-Kalman filter, Optical Flow, and Aggregate Channel Features (ACF) [34] methods are applied in order to detect vehicles in urban and highway traffic videos.

2.1.1 Challenging scenarios faced by motion-based methods

Despite all the benefits in terms of generalization and computational efficiency, locating objects based on motion information comes with its own set of challenging problems, such as illumination changes, camera jitter, multi-modal backgrounds, detection of small objects, cast shadows, low frame-rate, and dynamic backgrounds [40]. Figure 2.1 demonstrates sample video frames and the corresponding foreground masks extracted by popular motion segmentation methods in challenging situations.

Moving cast shadows Cast shadows are specifically a problem for traffic surveillance videos due to the abundance of their occurrence during the daytime and the consecutive effects they have on further tasks, such as vehicle tracking and classification. Cast shadows are mostly classified as foreground because of the similarities in the motion patterns among the moving objects and their shadows. Therefore, in order to avoid deteriorating the performance of video analytics, many studies have attempted to suppress the cast shadows in motion segmentation algorithms [60, 113, 120]. Statistical methods [58, 95, 103, 149], DCNN-based approaches [170], or various features such as color [3, 33], texture [51, 141], or other information such as shape, size, and direction [20,56] have been utilized in order to detect cast shadows [118].

In the case of traffic surveillance applications, due to a strict requirement for real-time performance the computational complexity of shadow removal algorithms should be as light as possible. A few considerations are worth noting for cast shadow detection in traffic-related videos. One observation is that the sun is the single major environmental light source in outdoor scenes, which brings along the possibility of applying heuristics based on light direction and assuming a contiguous region for the shadow cast by each object. Another observation is the uniformity of the road region in terms os texture and color which results in the pixels corresponding to shadowed regions, which are mostly on the road, exhibiting homogeneous features. These observations and other physics-based properties of shadows have been the basis of many algorithms developed for shadow removal in traffic videos. Hang and Liu [126] developed a hierarchical cast shadow detection framework by integrating a set of chromatic criteria in the HSV color-space, a region-based clustering technique, and a statistical global shadow modeling method in order to detect and remove moving cast shadows in traffic surveillance videos. Russell et al. [117] scan each video frame in horizontal lines in the opposite direction to the illumination direction and utilize intensity measurements in the neighboring pixels to classify foreground pixels into objects and shadows. Phan et al. [112], employ gradient features to discriminate between vehicles and their shadows for a real-time shadow removal method in traffic surveillance videos.

Non-stationary cameras In addition to the case of videos captured by stationary cameras, there are many studies addressing the problem of motion segmentation in dynamic cameras. In many modern surveillance systems, remote control pan-tilt-zoom (PTZ) cameras are utilized in order to give the operators the ability to move the cameras remotely and direct attention to a specific event or survey a different area. Since the assumption of a static camera does not hold, motion segmentation methods applied for applications of static cameras cannot

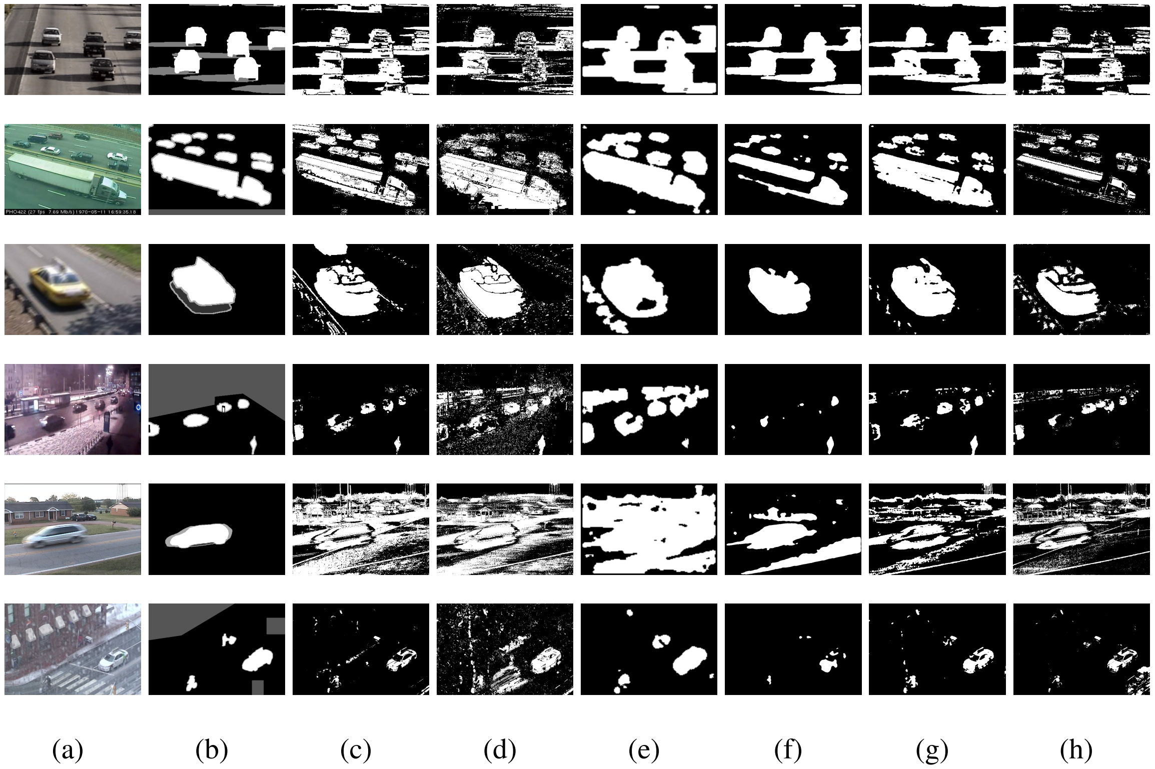


Figure 2.1 The qualitative performance of popular motion segmentation methods in challenging scenarios tested on ATON [120] and CDnet [150] datasets. From top to bottom the rows represent a challenging situation with cast shadows, low frame-rate, camera jitter, night time, continuous pan, and adverse weather conditions, respectively. From left to right the columns show a sample video frame and the results of different methods: (a) original frame, (b) ground truth, (c) AGMM [171], (d) Codebook [71], (e) Multi-Cue [107], (f) PAWCS [130], (g) PBAS [55], (h) ViBe [8].

be directly used in the case of dynamic cameras. Therefore, motion segmentation studies are generally grouped into two categories based on their application and use of static or dynamic cameras.

A dynamic camera can refer to a freely moving camera, such as a handheld, drone, smartphone, or dashcam, which can have unrestricted movements, or a constrained moving camera, such as pan-tilt-zoom (PTZ) cameras, which can have a restricted type of motion. When it comes to the applications of traffic video analytics, both types of dynamic cameras are typically used with in-vehicle cameras, such as dashcams, being considered as freely moving and in-road cameras, such as PTZ, being considered as constrained moving cameras.

Motion segmentation methods in the case of stationary cameras rely heavily on the assumption that the static objects of the scene are captured at a spatially stable location in the video frame. This strong assumption is due to the fixed viewing angle and distance of the stationary cameras, and even if there are variations in the background intensity values, they are associated with changes in illumination, shadows, small motions, or camera jitter. However, this assumption does not hold in the case of dynamic cameras, where the static objects also appear to have a so-called ego-motion, and therefore, the same methods cannot be directly applied for segmenting the foreground.

There are several studies, especially in recent years, that address motion segmentation in the case of a moving camera [19,162]. In the case of in-vehicle cameras, most studies tend to apply object detection or image segmentation algorithms rather than motion segmentation. However, most in-road surveillance cameras are stationary with PTZ capabilities that capture videos with lower resolution, so motion segmentation methods are more practical. Nevertheless, it is worth considering motion segmentation methods in the case of dynamic cameras for the applications of traffic analytics.

In general, studies concerning motion segmentation in videos captured by dynamic cameras can be categorized into two groups. One group of studies focus on statistically modeling and subtracting the dynamic background and reporting the values that do not fit into the model as the segmented foreground. These methods vary mainly based on the approach to background representation. The other group of studies tend to distinguish the moving objects from the background based on the differences in the motion patterns. This group of methods is more computationally expensive than the first group as they require more detailed steps and a greater number of calculations.

Most of the motion segmentation methods in the first group are based on ideas that are inspired by the algorithms used in the case of static cameras. Several techniques have been utilized in these studies in order to adapt to a dynamic scene and distinguish the motion of the objects from the motion of the camera. To name a few of these techniques, we can refer to panoramic background subtraction, superpixel segmentation, motion compensation, low rank matrix decomposition, and block-based splitting of the video frames.

Motion compensation is the simplest and most efficient approach used in motion segmentation methods in videos captured by dynamic cameras. As in the case of stationary cameras, one of the most common approaches for subtracting the background in videos captured by dynamic cameras is to benefit from the high frequency of data points in the temporal domain in order to model the background. In these methods, a set of beginning frames is first used to initialize a parametric or non-parametric model for each local representation of the background image. Since the entire scene seems to be moving in the eyes of the dynamic camera, the camera motion should be compensated for for the background modeling to function.

To estimate the motion of the camera, a set of feature points or uniformly distributed points are selected and the corresponding points in the new video frame are found in order to calculate a homography matrix and warp all the pixels in the new frame to corresponding pixels in the previous frame through an inverse perspective transformation. This is assumed to be the movement of the background compensated after applying the two-dimensional parametric transformation. After motion compensation, the background model is registered with the current video frame and can be updated and used for foreground segmentation. Since the set of selected points includes the feature points of the foreground objects, there are some registration errors after the transformation estimation which often results in false positives in the foreground segmentation step. Therefore, the registration is usually followed by a refinement step. Some methods repeat this process a number of times until a condition is reached which results in extracting multiple planes where each plane corresponds to a dynamically homogeneous group of pixels [5]. One of the popular techniques is dividing each video frame into a number of blocks with a pre-defined size or using superpixel segmentation in order to simultaneously reduce the computational complexity and improve the performance by taking spatial relations into account. Motion compensation can be applied to the entire image or separately to each block.

Another common approach is the use of low rank and sparse decomposition for the task of motion segmentation. An optimization process is carried out to form an observation matrix using a set of video frames and Principal Component Pursuit (PCP) [15] is applied in order to construct low-rank and sparse representations. Similar to the background modeling techniques, a global motion compensation technique is first applied to obtain a transformation matrix and align the background before matrix decomposition. The static objects are coherent in terms of relative motion to the camera, but the moving objects exhibit different dynamic behavior. Therefore, the low-rank matrix is assumed to represent the background while the sparse matrix contains the outliers and is considered to represent the moving objects. In spite of the effectiveness of this group of algorithms in motion segmentation, the requirement for collecting a pre-defined number of frames before being able to apply them imposes limitations on their applicability in real-time systems.

Some studies attempt to stitch the images captured by the moving camera together in order to construct a panorama or mosaic, which is a bigger image that represents the entire background. This panorama is constructed by frame to frame, frame to mosaic, or mosaic to frame alignment, depending on the desire to use a fixed coordinate system. The background is modeled based on the constructed panorama, and moving objects are detected by applying one of the background subtraction methods used in the case of fixed cameras.

The second group of studies has taken a different approach by attempting to track the trajectories of the feature points or uniformly distributed points that represent the displacements in a sequence of adjacent frames and applying clustering techniques to classify the trajectories and extract the foreground from the dynamic background. Modeling the background values is not required as the motion segmentation only relies on the differences between the trajectories of the moving and static objects in the eyes of the camera. This group of methods relies heavily on the precision of trajectory calculation and dense segmentation of moving objects, which is a common problem [162].

Stopped objects Most foreground detection methods fail to keep detecting the moving objects after they stop. In parametric modeling methods such as GMM, the stopped objects are absorbed by the background model shortly after they stop moving. This is specifically problematic for traffic surveillance systems, as road users may stop regularly at intersections. On the other hand, stopped vehicles are considered a threat to highway and road traffic and should be marked as anomalies. In order to locate the stalled vehicles and report them as anomalies, many studies have attempted a combination of motion-based and appearance-based methods [153, 169]. These studies assume the stopped vehicles are merged into the background model and they can be located by applying an object detection method, such as Faster R-CNN or YOLO on the background image. However, there are studies conducted on locating the stopped vehicles solely based on the motion information [123]. Among the regular motion segmentation methods, the LBAdaptiveSOM [92] and adaptive background learning techniques have shown better performance in detecting stopped objects [129].

Weather and illumination variations Traffic surveillance systems are required to work day and night under adverse weather conditions and illumination changes in the presence of large shadows and reflections. These variations can lead to sizable drops in the performance of motion-based object locating methods. Although there are studies that have attempted to solve these issues by applying motion-based features [6, 159], most studies tend to rely on appearance-based features as they are more robust to illumination changes.

Occlusions Locating objects of interest solely based on motion information is prone to severe performance drops in the case of object occlusions. Since every connected component in the foreground mask is considered to be an object, two or more nearby objects can easily fall into the same component. This is specifically problematic for traffic surveillance applications where moving vehicles can be occluded by each other. This problem has been addressed in many studies [30, 93] who have attempted to handle the occlusions by various heuristics. However, these techniques are limited to specific scenarios and cannot be considered as a general solution.

2.2 Shadow Detection Methods

In video analysis applications, shadows cast by moving objects are often classified as foreground due to their similar motion patterns to the moving objects. Since object detection is one of the fundamental steps, this misclassification causes several issues in the subsequent operations. In order to solve this problem, many methods have been proposed throughout the previous years [120]. Most methods assume similar chromaticity values among the background and shadows while darker illumination for the shadows [50, 120]. Therefore, several color-spaces such as HSV, HSI, C1C2C3, and YUV are examined along with RGB to separate the luminance and chromaticity components with the goal of detecting shadows [33].

Many shadow detection methods operate at a pixel level. McKenna et al. [98] made the popular assumption that shadows change the intensity but not the chromaticity. The chromaticity values and gradient information of the pixels are modeled, and the foreground pixels are classified as background if they match the background in terms of chromaticity and gradient. Cucchiara et al. [33] convert the image from RGB to HSV color-space, expecting the shadows to darken the pixel values in the luminance component while preserving the hue and saturation components. The main problem with these types of methods is their need for empirical parameter tuning and their weak performance in the case of achromatic shadows where the ambient component of the light is strong.

Another group of studies approaches the problem of cast shadow detection in a statistical manner [58, 94, 103, 149]. Martel-Brisson and Zaccarin [94] examine the stability of different states in the mixture of Gaussian distributions of each pixel and detect shadows based on the assumption that states corresponding to shadows are more frequent than those corresponding to foreground objects. In a later study [95], they proposed a non-parametric method for modeling the changes in pixels while they under shadow. A single direction in the RGB space is determined in which the shadowed pixels reside.

As opposed to pixel-level methods, some studies tend to exploit region-based strategies [139, 160]. Toth et al. [139] applied the mean-shift image segmentation technique and used the segmented regions as a reference for analyzing the constancy of the intensity ratios over the neighboring area. Yang et al. [160] exploit multiple cues, such as color, shading, texture, neighborhood, and temporal consistency in order to detect the shadows. The reliance of these methods on texture information makes them computationally expensive and limits their generalization capability.

Over recent years, machine learning algorithms and methods based on deep learning have grown to be some of the most popular techniques for detecting shadows [28, 77, 141, 168, 170]. Lee et al. [78] generate several super-pixels by over-segmenting the image and learn shadow features through a convolutional deep neural network consisting of seven layers. Chen et al. [24] present a multi-task mean teacher model for semi-supervised shadow detection by using unlabeled data and learning shadow regions, edges, and counts. Le and Samaras [76] set a number of physics-based constraints in order to train an adversarial network using only patches cropped from the images. Although these methods achieve high performance, they often require supervision and large datasets of shadowed and non-shadowed images, which are difficult to obtain. Therefore, in real-time applications of video analysis, statistical methods are more feasible.

2.3 Road Detection Methods

Automatic Region of Interest (RoI) detection is an important task in many traffic video analysis applications and can be used in road management, driver assistance systems, automatic driving, intelligent traffic surveillance, robot and car navigation systems, etc. In recent years, many automatic RoI detection methods have been proposed in order to reduce manual work in urban and highway traffic monitoring applications. Some methods have tried to utilize various features in order to segment the road region from the remaining parts of the image. In the paper written by Santos et al. [121], a feature vector of gray amount, texture homogeneity, traffic motion, and the horizontal line are fed to a support vector machine to classify each superpixel into road or non-road. Helala et al. [54] use the contours of superpixel blocks to generate a large number of edges, which are organized into clusters of co-linearly similar sets, and the best clusters are chosen according to a confidence level assigned to each cluster. In the end, the top-ranked pair of clusters are selected as road boundaries. Almazan et al. [2] combine a spatial prior with the vanishing point and horizontal line estimators in order to adapt to new weather conditions. Cheng et al. [26] propose a road segmentation method by applying the Gaussian mixture model to color features and fusing them with the geometric cues within a Bayesian framework.

Some studies approach the task of roadway detection by using temporal features and extracting the active traffic regions. Lee and Ran [79], extract the moving parts of the scene in videos of bidirectional traffic as difference images between two consecutive frames and accumulate them to form a road map. Then a center line is used to divide the roadway into two segments, each of which corresponds to one of the two major traffic directions. Similarly, Tsai et al. [140] accumulate the difference between two consecutive frames to obtain a map of the road where the motion vectors are used to separate the roadway into two regions in order to represent two major traffic directions. The performance of background subtraction and tracking methods utilized in these techniques has a large influence on the results of the road segmentation process.

Most recent studies tend to propose illumination-invariant methods to deal with strong shadows and benefit from the recent advances in deep learning models to segment the road in a supervised manner. Li et al. [82] propose a bidirectional fusion network (BiFNet) consisting of a dense space transformation module and a context-base feature fusion module in order to fuse the image and the bird's eye view of the point cloud. Tong et al. [138], calculate an effective projection angle in the logarithmic domain to extract the intrinsic images with a weakened shadow effect and adopt to different directions of the camera view. Li et al. [84] propose a road segmentation by estimating the spatial structure of the road and using the color and edge features of the intrinsic image, which is extracted based on regression analysis. Cheng et al. [27] propose a novel adaptation method to generalize road segmentation to new illumination situations and viewing geometries by training a fully-convolutional network for road segmentation. The learned geometric prior is anchored by estimating the vanishing point of the road and is used to extract road regions that are utilized as ground-truth data to adapt the network to the target domain. Wang et al. [146] generate an illumination invariant image and a manual triangular area is used as the color sample to obtain a number of probability maps which are used to segment the road, which is further refined by taking the extracted road boundaries into consideration. Junaid et al. [68], extract multiple abstract features from the explicitly derived representations of the video frames and feed them to a shallow convolutional neural network. Most of the new studies benefit from supervised learning methods, which limits their ability to adapt to new videos. Here, we proposed an unsupervised statistical method which can be applied in real-time applications.

2.4 Accident Detection Methods

Over the past several decades, there have been some studies addressing the issue of vision-based accident detection on roads and highways. Zu et al. [174] use a Gaussian Mixture Model to detect the moving vehicles and the mean shift method for tracking them. In this study, three main motion features, namely, velocity, acceleration, and orientation, are derived from the trajectories of the tracked vehicles. When all these values exceed the predefined thresholds, an accident is reported. Since the videos are from the viewpoint of a driver, the motion features are more reliable than those captured by the CCTV cameras overlooking the highway from above. Note that this method may cause false alarms when the pattern of traffic flow varies in a short period of time. Besides, rapid changes in motion features do not always result in an accident. Ren et al. [116] use a modified Gaussian Mixture Model to extract the moving vehicles in aerial videos and, after detecting the lanes and dividing each lane into a cluster of cells, some traffic features are extracted for each cell based on the tracking information. Finally, a support vector machine is trained to detect incident points. Traffic parameters include flow rate, average travel speed, and average space occupancy. This method is reliable and fast, but it is for generally detecting traffic incidents and is not specifically for accident detection. Also, it relies on straight road lanes, whereas in our case, accidents usually occur in the curved lanes of the road. Xia et al. [154] propose a close-to-real-time approach that divides each frame into non-overlapping blocks for each of which an average velocity magnitude is calculated and the low-rank matrix approximation is utilized to detect the increase in approximation error. Although this method is more generalizable to different situations, it can result in some false alarms. On the other hand, the method can be computationally expensive for higher resolution videos. Maaloul et al. [91] use the Farneback optical flow to extract motion and a statistical heuristic approach to select thresholds and adaptively model traffic flow for accident detection. This method is effective in various scenarios of traffic videos (e.g. highways and expressways) and requires a low amount of training data for motion modeling. Nevertheless, the use of optical flow makes this approach not suitable for real-time applications.

Some other studies use more complex methods to detect abnormalities in traffic flow. Thomas et al. [136] formulate vehicle incident analysis as an optimization problem. An optimal summarization framework is proposed that relies on the salient features of the moving vehicles. This method achieves comparatively good results. However, it suffers from errors in segmentation techniques. Ahmadi et al. [1] use a group sparse topical coding-based technique to model the normal traffic motion using the Lukas-Kanade's optical flow vectors in a document of words. In this model, each word corresponds to velocities in a specific range of orientations, and when the computed words do not match the model, it means some abnormal motion has happened. This approach is focused mostly on abnormal movement detection and is not specific to a type of accident. Arceda and Riveros [4] present a three-stage approach to detect car crash incidents. First, cars are detected using the You Only Look Once (YOLO) deep learning model. Then, after tracking each detected car, the Violent Flow (ViF) descriptor is used alongside an SVM to detect car crashes. This approach is not real-time, and there can be some false alarms. Xu et al. [157] present a model for anomaly detection in road traffic by analyzing vehicle motion patterns in static and dynamic modes. In the static mode, the background is subtracted and fed into a Faster R-CNN model for detecting stopped vehicles. In the dynamic mode, the trajectories of vehicles are tracked to find an abnormal trajectory that is aberrant from the dominant motion patterns. This method ranked first place in the NVIDIA AI City Challenge [105]. However, it has some limitations due to the use of a supervised deep learning model and is also not very specific about the type of detected abnormality.

There have been more studies for vision-based traffic accident detection with the use of deep convolutional networks in recent years. Batanina et al. [9] use a video game to generate synthetic data due to the lack of real videos of car crashes. After training a three-dimensional (3D) deep convolutional neural network on the synthetic rendered videos, domain adaptation is used to adapt the model to real videos. Huang et al. [59] propose an integrated two-stream convolutional network architecture to detect and track vehicles in real time and also detect near-accidents in videos from overhead cameras. Appearance and motion features from the two networks are incorporated to detect near accidents. Most of these studies are generally designed to detect abnormal traffic motion, which can include stopped vehicles, head-to-head collisions, unexpected congestion, etc. and they are not specific to the type of anomaly. Some methods cannot be applied in real time due to computational complexity. Also, many of the existing methods rely on supervised data to train a prediction model before they can be applied. In this study, we present a novel real-time traffic accident detection framework to detect two types of traffic accidents, namely, single-vehicle traffic accidents and trajectory conflicts at intersections.

CHAPTER 3

FOREGROUND DETECTION

3.1 Introduction

Detecting the location of interesting objects has been intensively studied in the field of computer vision. Generally speaking, the current techniques for locating objects of interest can be categorized into two groups: appearance-based and motion-based methods. Motion-based methods are applicable to video data and tend to perform a binary classification on the pixel locations in each video frame. In many applications of video analytics systems, the objects of interest (aka the foreground) have a dynamic pattern different from the rest of the scene, namely the background. This difference has been exploited by many studies in order to segment the foreground from the background and subsequently locate the objects of interest.

Foreground segmentation has specifically been applied to intelligent surveillance systems [16], traffic monitoring [37, 41–44, 48, 88, 123, 125], gesture recognition [64], and robot vision [96]. The input video data used in the majority of these applications is captured by stationary cameras, which causes the foreground to have significant motion compared to the background. A large number of studies have attempted various approaches to subtract the relatively static background from the changing foreground in order to detect the location of the moving objects [40]. The strong presumption that the camera is stationary or only has jittering movements is common among all these studies and substantially affects their strategies to the point that they become ineffective in the event that the camera has considerable movements. However, in real-world applications, camera movements are common and can happen in restricted forms, such as panning, tilting, or zooming in the case of PTZ cameras used in video surveillance, and freely moving cameras, such as handheld cameras, smartphones, drones, or dashcams, in which case the camera is mounted on a moving platform. In all these scenarios, the camera is non-stationary with regard to the captured scene, and therefore, everything seems to be moving in reference to the camera. Consequently, there is a need to implement foreground segmentation methods that are capable of dealing with camera motion and quickly adapting to the changes in the background. When relying solely on motion information to segment the foreground from the background in video frames captured by non-stationary cameras, the only heuristic lies in the differences between the dynamic patterns of the moving objects and the background (Figure 3.1). Many approaches have been proposed to take these differences into account and locate the objects of interest in videos captured by non-stationary cameras [19, 162].

The real-world applicability of the current methods suffers from high requirements for computational resources and/or low performance in classifying foreground and background. Here we apply spatial and temporal features for statistical modeling of the background and the foreground separately in order to classify them in real-time. Each block of the background is modeled using a mixture of Gaussian distributions (MOG) and a set of values sampled randomly in spatial and temporal domains. At each video frame, the Lucas-Kanade optical flow method is applied to the block centers in order to estimate the camera motion and find the corresponding locations between two adjacent frames. The global motion is then compensated by updating the background models of each block according to the values of its corresponding location in the previous frame. On the other hand, the foreground is modeled by another MOG, which is updated by the input values that do not fit into the background models. The final classification is performed by comparing the input super-pixel intensity values with the major components in the statistical background and foreground models. The remainder of this chapter is organized as follows: In Section 3.2 the main steps of the proposed framework are described in order. Section 3.3 contains experimental evaluations of the method's performance, and the conclusions are summarized in Section 3.4.

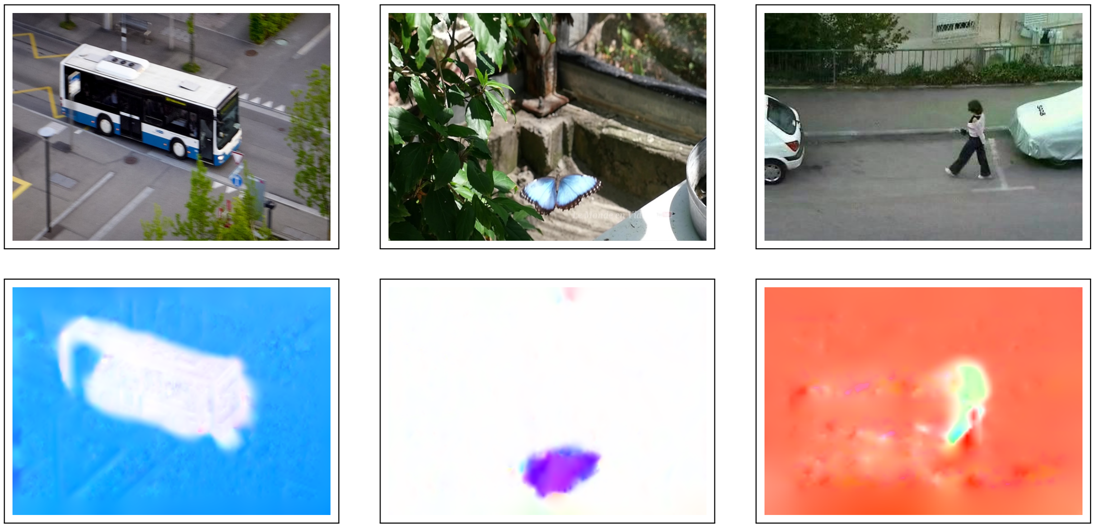


Figure 3.1 Optical flow field calculated by applying the UnFlow method [99]. The direction is indicated by hue and the velocity is represented by saturation.

3.2 The Proposed Foreground Segmentation Method

The first observation in videos obtained by moving cameras is that the entire captured scene appears to be moving from the camera's perspective. However, by assuming the background to occupy the majority of the scene compared to the objects of interest, we can estimate the motion of the camera relative to the background. Afterwards, the estimated camera motion can be compensated for by using the corresponding values in the previous frame for updating background models. After motion compensation, the foreground can be segmented using approaches similar to the methods used for the applications of stationary cameras. Here, we apply an MOG to model the entire foreground using the values that are not absorbed by the background models. The major components of the Gaussian mixture distributions in the background and foreground models are utilized for final binary classification. The details of each step are described in this section.

3.2.1 Global motion estimation

The main purpose behind moving the camera in most applications of video analytics is to focus on the interesting objects and try to keep them in the view field of the camera. In many scenarios, the objects of interest occupy a portion of each video frame, and the remaining majority is considered to be background. Therefore, the majority of point displacements among video frames are caused by the camera motion, which can be estimated by calculating the global motion. For the sake of computational efficiency and accounting for spatial relationships, a similar approach to [166] is applied where the input image is converted to grayscale and divided into a number of grids with equal sizes. The Kanade–Lucas–Tomasi feature tracking approach [137] is applied to the centers of the grid cells from the previous frame. Then a homography matrix is obtained that warps the image pixels at frame t to pixels at frame $t-1$ through a perspective transform. If we denote the intensity of the grayscale image at time t by $I^{(t)}$ and assume consistent intensity between consecutive frames, the corresponding location of each point in the new frame can be used to calculate the global velocity vector as follows:

$$I^{(t)}(x_i + u_i, y_i + v_i) = I^{(t-1)}(x_i, y_i) \tag{3.1}$$

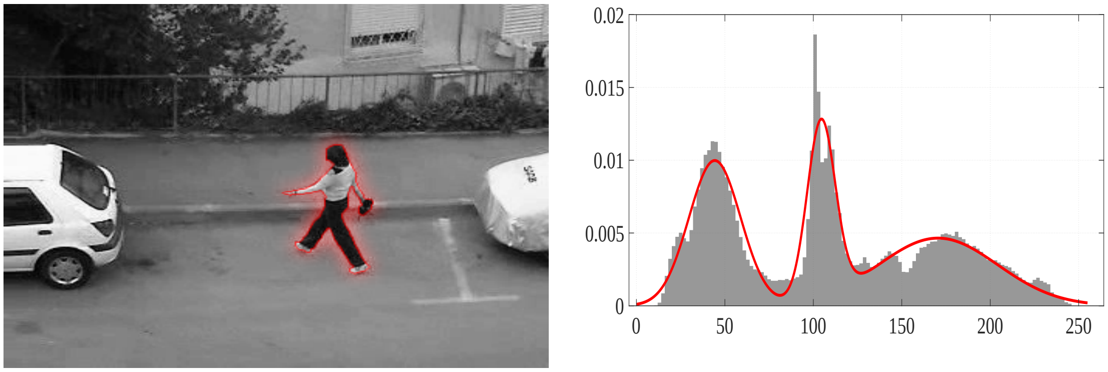
where (u_i, v_i) is the velocity vector of the center point of the i-th block located at (x_i, y_i). Three-dimensional vectors X_i can be constructed as:

$$X_i^{(t-1)} = (x_i, y_i, 1)^T, \quad X_i^{(t)} = (x_i + u_i, y_i + v_i, 1)^T \tag{3.2}$$

and a reverse transformation matrix $H_{t:t-1}$ is obtained that satisfies Equation (3.1) for the largest possible number of samples:

$$\left[X_1^{(t)}, X_2^{(t)}, ...\right] = \mathbf{H}_{t:t-1} \left[X_1^{(t-1)}, X_2^{(t-1)}, ...\right] \tag{3.3}$$

which is solved by applying the RANSAC algorithm [38] in order to remove outliers from further calculations. Also, the center points of the blocks classified as foreground in the previous frame are excluded from this calculation as they do not contribute to the camera motion.



(a) The foreground object

(b) MOG model

Figure 3.2 The foreground is modeled by a mixture of Gaussian distribution.

3.2.2 Background and foreground modeling

Each block of the image is modeled by a mixture of Gaussian distributions and the model is updated at each video frame. In order to update the background models at each frame we have to calculate the corresponding values in the warped background image of the previous frame. The mean and variance of the warped background model are calculated as a weighted sum of the neighboring models, where each weight is proportional to a rectangular area as a bilinear interpolation:

$$\begin{aligned} \tilde{\mu}_i^{(t-1)} &= \sum_{k \in \mathcal{R}_i} \omega_k \mu_k^{(t-1)} \\ \tilde{\sigma}_i^{(t-1)} &= \sum_{k \in \mathcal{R}_i} \omega_k \sigma_k^{(t-1)} \end{aligned} \tag{3.4}$$

where $\mathcal{R}$ is a set of block indices falling in a rectangular region centered at the corresponding point location calculated by the homography matrix in Equation (3.3), ω_k is the weight that indicates the overlapping area between the block i and the corresponding neighbor k, and μ and σ represent the mean and variance of the Gaussian distributions, respectively.

Since the camera might have slight movements in the form of a pan, there can be slight variations in the illumination due to the changes in the angle of view and light direction. Also, even after motion compensation, the pan motion of the camera can cause a part of the background to move out of the scene, which results in a block representing another part. The Gaussian modeling keeps the information from the previous frames and might be slow in catching up with the pace of changing values at the borders of the video frames. In order to make the model parameters adapt to these changes, a global variation factor g is calculated by subtracting the mean intensities in the background model and the current frame:

$$g^{(t)} = \frac{1}{N}\sum_{j=1}^{N} I_j^{(t)} - \frac{1}{B}\sum_{i=1}^{B} \tilde{\mu}_i^{(t-1)} \tag{3.5}$$

with B being the number of blocks and N being the number of pixels. At each frame the parameters of the Gaussian mixture model for each block are updated as follows:

$$\begin{aligned} \mu_k^{(t)} &= \left(n_k^{(t-1)}\left(\tilde{\mu}_k^{(t-1)} + g^{(t)}\right) + M^{(t)}\right)/(n_k^{(t-1)} + 1) \\ \sigma_k^{(t)} &= \left(n_k^{(t-1)}\tilde{\sigma}_k^{(t-1)} + V^{(t-1)}\right)/(n_k^{(t-1)} + 1) \\ n_k^{(t)} &= n_k^{(t-1)} + 1 \\ \alpha_k^{(t)} &= n_k^{(t)}/\sum_{k=1}^{K} n_k^{(t)} \end{aligned} \tag{3.6}$$

where n_k is a counter representing the number of times an input value has been used to update component k, α_k is the weight of the kth component, M and V stand for the mean intensity and the variance of the block, respectively. The component with the largest weight of each Gaussian mixture model is considered to be the background value of the block.

In the case of moving cameras, the objects of interest are usually present in the scene for a longer time as the camera is focused on them. Therefore, it is reasonable to model the values of the foreground objects throughout the video. A similar approach to background modeling is applied to modeling the foreground, except only one mixture of Gaussian distributions is used for the entire foreground pixels. Also, instead of a single component, a number of components from the foreground model that have the largest weights are considered to represent the foreground objects. This is because the foreground objects have multiple parts with different intensity values, and each major component in the foreground

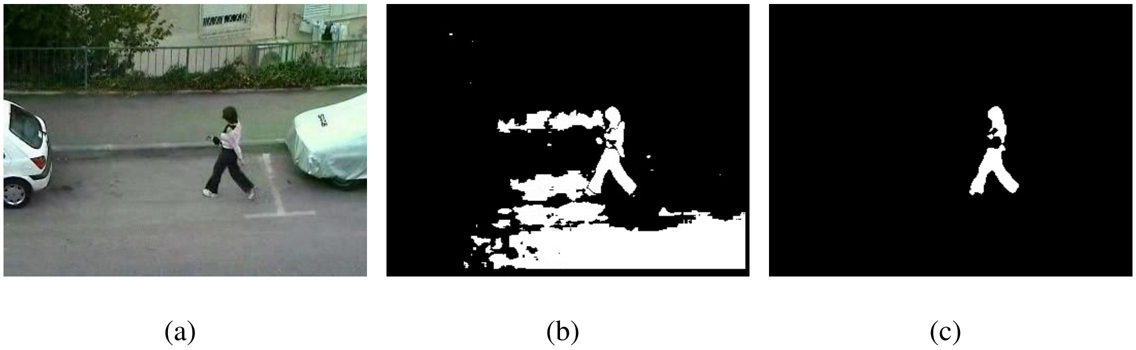


Figure 3.3 Improving the classification results with foreground modeling. (a) Original frame, (b) False positives caused by hysteresis effect in background modeling, (c) False positives are avoided after foreground modeling.

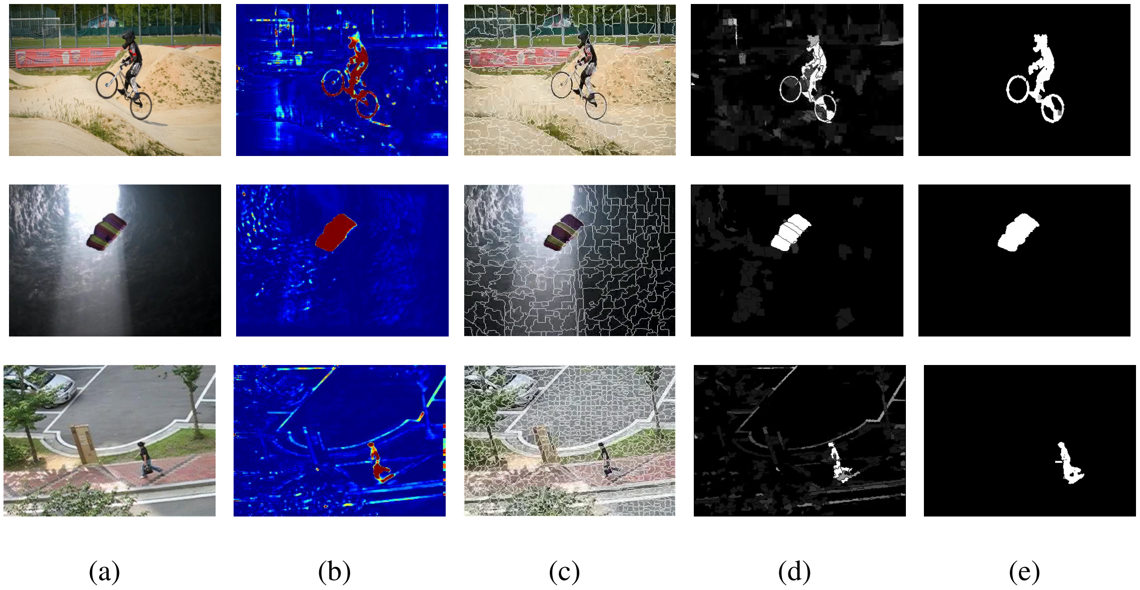


Figure 3.4 The final classification process. (a) Original frame, (b) Heat-map of the foreground probability, (c) Super-Pixels obtained by applying watershed segmentation, (d) Foreground confidence map, (e) Final foreground mask.

model is used to represent one part of the foreground. Figure 3.2 illustrates an example of a foreground object modeled by an MOG with three components.

In addition to the statistical modeling and inspired by the ViBe method [8], we keep a set of sample values as a secondary non-parametric model for each block. This set is initialized by the mean value of the block and its neighboring blocks at the beginning of the first frame. In each of the consecutive frames, one of the values in the set is selected randomly and replaced with the new mean value. We can denote the collection of background sample values for the block i as $\mathcal{S}_i$ as follows:

$$\mathcal{S}_i = \{s_i^1, s_i^2, ..., s_i^K\} \tag{3.7}$$

where s_i^k is the kth sampled mean intensity of block i. The sample-based model is kept and updated mainly to avoid contaminating the foreground model with the background values that do not fit into any of the Gaussian components of the corresponding block model. This problem occurs mostly because of motion compensation errors or new background values being introduced into the scene due to the camera motion. If an input value does not fit into any of the Gaussian components of a background model, the Euclidean distance between the pixel value and each background sample in the set of the corresponding block is calculated. If the number of samples in the set of blocks i that are closer than a distance threshold to the input value is less than a counting threshold, the foreground model is updated by that value. Representing this number of samples by C_i it can be calculated as follows:

$$C_i = \sum_{j=1}^{|\mathcal{S}_i|} \mathbb{1}\left(D(\mathbf{x}, \tilde{s}_j^{(s)}) < \theta_d\right) \tag{3.8}$$

with $\mathbf{x}$ being the input pixel intensity value, D representing the Euclidean distance, θ_d being a predefined threshold, which is set to 20, $\mathbb{1}$ denoting an indicator function, $\tilde{s}_j^{(s)}$ representing the corresponding value of $s_i^{(k)}$ after motion compensation, and $\mathcal{S}_i$ denoting the set of neighboring blocks.

Since the camera is in motion, the parameters in the background models can lag behind the sudden changes caused by motion compensation errors, sudden illumination changes, or new samples appearing at the borders of the frame. Consequently, the distance between the new samples and the mean values may exceed the threshold defined based on the standard deviations, which in turn causes the new samples to falsely be classified as foreground. By keeping a set of values containing a number of recent background samples, we can compensate for the hysteresis effect of Gaussian models representing the older samples. We calculate the Euclidean distances between the new values and the samples in the set and only classify the new values as foreground if they match with less than a few samples in the set. The foreground model is only updated with values that belong to the foreground class with a high certainty, and therefore, the majority of false positive cases are avoided. An example of the classification is illustrated in Figure 3.3. As seen in the Figure 3.3(b), some of the input values do not fit into their corresponding background models due to the camera movements and the motion compensation errors. In Figure 3.3(c) these values are removed from the foreground mask as they do not fit into any of the major components of the foreground model.

3.2.3 Background and foreground classification

For the final classification, at first the foreground likelihood values are calculated for each pixel at an input image as follows:

$$L_{fg}(x,y) = \frac{(I(x,y) - \mu_k)^2}{\sigma_k} \tag{3.9}$$

where $I(x,y)$ and $L_{fg}(x,y)$ are the intensity and foreground likelihood values of the pixel at location (x,y), and μ_k and σ_k are the mean and variance of the corresponding background block, respectively. Afterwards, the watershed segmentation algorithm [100] is applied to each input image in order to extract a set of super-pixels, notated by $\mathbb{P} = \{P_1, P_2, ..., P_k\}$.

For final classification, the mean value of each super-pixel is compared against the major component in the background model of the corresponding block as well as each component in the foreground model. The foreground confidence map $\mathcal{F}$ is obtained by calculating the mean of confidence values in each super-pixel as follows:

$$\mathcal{F}(P_i) = \frac{1}{|P_i|} \sum_{x,y \in P_i} L_{fg}\,(x,y) \tag{3.10}$$

where $|P_i|$ is the number of pixels at super-pixel P_i. Assuming there are M major components in the global foreground model, a background confidence map $\mathcal{B}_m, m \in$

$\{1, ..., M\}$ is similarly obtained based on each component. The Gaussian Naive Bayes (GNB) classifier is applied to each super-pixel in order to calculate the z-score distance between the input value and each class-mean and classify the super-pixel accordingly in order to obtain the final foreground mask $\mathcal{H}$:

$$\mathcal{H}(P_i) = \begin{cases} 1, \text{if } \mathcal{F}(P_i) > \mathcal{B}_m(P_i) \\ 0, \text{otherwise} \end{cases} \tag{3.11}$$

where $\mathcal{B}_m$ is the background confidence map corresponding to the m-th foreground model and $\mathcal{H}(P_i) = 1$ indicates that the super-pixel at location P_i belongs to the moving objects and $\mathcal{H}(P_i) = 0$ means it belongs to the background. The process of segmenting the foreground is detailed in Algorithm 1.

The different stages in the classification process can be seen in Figure 3.4 . From top to bottom, each row in the figure represents a sample video frame from the DAVIS [111], Segment Pool Tracking [81], and SCBU [166] datasets, respectively. The second column represents heatmaps where the pixels with a higher probability of belonging to the foreground are represented by red colors. The third column is the results of the watershed segmentation algorithm applied to each video frame, with the markers chosen uniformly across the image at the same locations as the background block centers. The fourth column illustrates the foreground confidence maps calculated based on Equation (3.10) and the last column is the final results of foreground detection after morphological dilation.

3.3 Experiments

The performance of the proposed method is evaluated using video data collected from the publicly available SCBU dataset [166], which consists of nine video sequences captured by moving cameras. The videos in the dataset impose various challenges in the way of foreground segmentation, such as fast or slow-moving objects, objects of different sizes, illumination changes, and similarities in intensity values between the background and foreground. Figure 3.5 represents the foreground masks detected by various methods. Similar to [165], in addition to background modeling methods [8, 25, 31, 72, 89, 102, 164, 166], the detection results are compared with a number of object-centric methods, such as uNLC [36], which is the unsupervised version of the NLC [36] approach, OSVOS [13] without the fine-tuning step, CIS [161], and BASNet [114]. In terms of time and space complexity, the statistical methods are more efficient as the methods based on deep neural networks require more resources. Therefore, our method is more practical in applications with real-time requirements and edge devices that have lower hardware capacity.

Figure 3.6 represents the foreground detection results in a number of video sequences compared with other background modeling methods. It can be seen that our proposed method is able to detect the foreground in various challenging scenarios. Compared to some of the representative methods, such as MCD [102] and MCD NP [72], our method models the foreground and background separately, which enhances the classification results. One of the limitations in the proposed method is the ability of the foreground model to adapt well to sudden illumination changes caused by the pan movements of the camera. Also, the camouflage problem, where the foreground color values are very similar to those of the corresponding background block, can lead to false negative results (part of the person's head is not detected in Figure 3.5(l)). This problem can be solved by introducing more discriminating features to the statistical modeling process.

The f-score metric is used in order to evaluate the quantitative results:

$$\begin{cases} PRE = T_P/(T_P + F_P) \\ REC = T_P/(T_P + F_N) \\ F_1 = 2 \times (PRE \times REC)/(PRE + REC) \end{cases} \tag{3.12}$$

where T_P, F_P are the number of pixels correctly and incorrectly reported as foreground, and T_N and F_N are the numbers of pixels that are correctly and incorrectly reported as background, respectively. PRE, REC, and F_1 refer to precision, recall, and F1-score, respectively. The F1-scores are listed in Table 3.1 in comparison with other popular methods. The quantitative results demonstrate the robustness of our method in detecting the foreground mask in different videos.

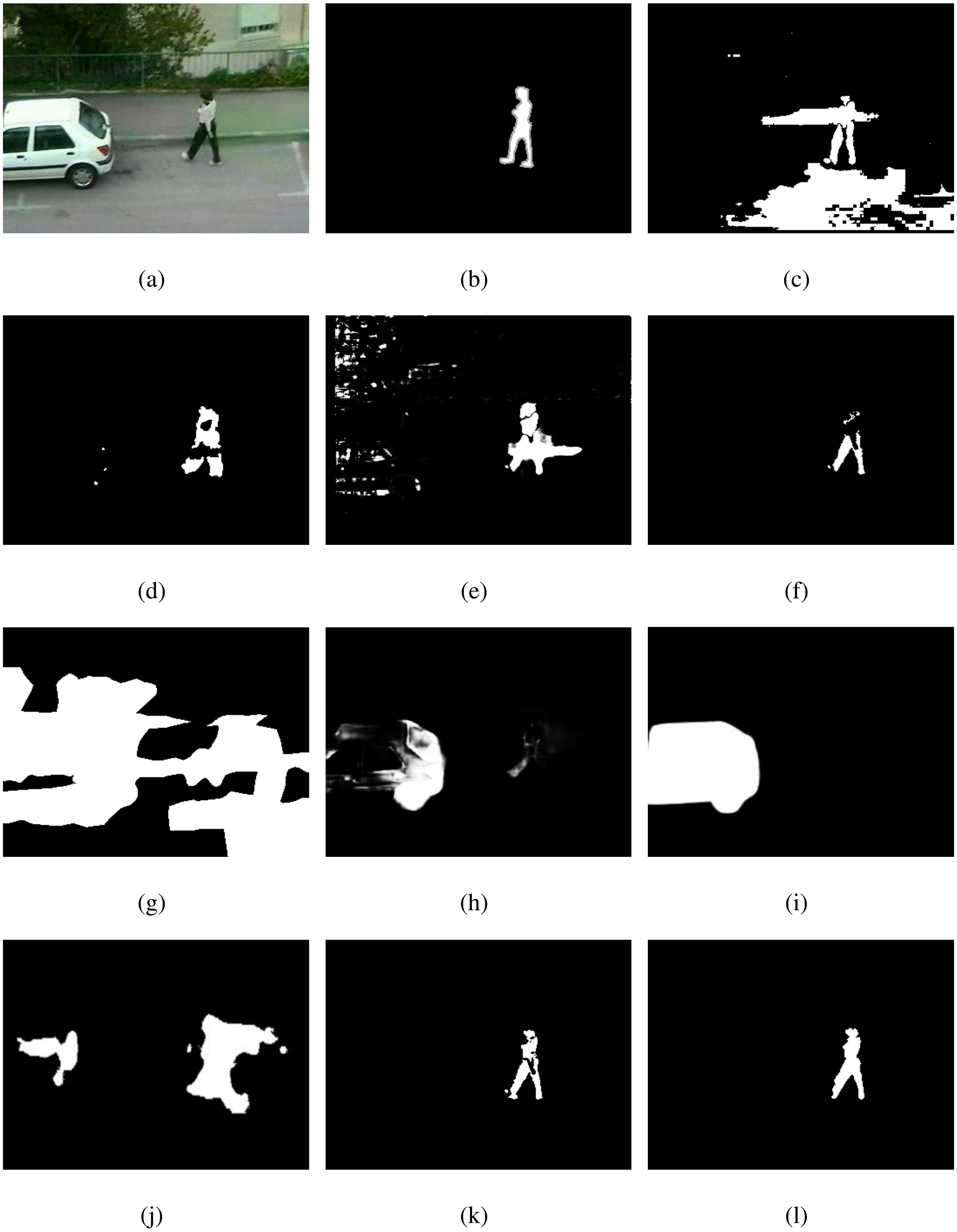
The hardware specification used for the experiments is a 3.4 GHz processor and 16 GB RAM. The average processing speed for video frames of size 320×240 pixels was about ~ 143 frames per second, which is feasible for real-time applications of video analytics. The average running speed of the proposed method is reported in Table 3.2 for each video frame of size 320×240 pixels. The run-time calculations show that the method is feasible to be used as a preprocessing step in real-time traffic video analysis tasks.

3.4 Conclusion

In this study, a new real-time method is proposed for locating the moving objects in videos captured by non-stationary cameras, which poses one of the challenging problems in computer vision. The global motion is estimated and used to compensate for background variations caused by camera movements. Each block is modeled by a mixture of Gaussian distributions, which is updated by the values at the corresponding locations in the warped image after motion compensation. Additionally, the mean values of each block are modeled along with the mean values of its neighboring blocks as a set of samples, which is in turn updated by random selection. The foreground, on the other hand, is modeled by a separate MOG which is updated by values that do not fit into either of the statistical or sample-based background models. For classification, each input value is compared against both the background and foreground models to obtain the definite and candidate foreground locations, respectively. The watershed segmentation algorithm is then applied to detect the final foreground mask. Experimental results demonstrate the feasibility of the proposed method in real-time video analytics systems.

Algorithm 1: Acquiring the foreground mask

```
Input:
        The input video frame in gray-scale $I^{(t)}$
        A set of predefined thresholds
Output:
        The foreground mask $\mathcal{H}$ of the same size as the video frame
initialize $\mathcal{F}$ with $0$;
foreach pixel $p \in I^{(t)}$ do
    if $p$ fits into the MOG model of block $i$ then
        update the $i$th MOG;
    end
    else if $p$ doesn't fit the $i$th sample-based model then
        update the foreground MOG;
    end
end
apply watershed segmentation to obtain $\mathbb{P}$;
$\mathcal{H} = 0$;
foreach super-pixel $P_i \in \mathbb{P}$ do
    calculate $\mathcal{F}(P_i)$;
    foreach component $m$ in foreground model do
        calculate $\mathcal{B}_m(P_i)$;
        if $\mathcal{F}(P_i) > \mathcal{B}_m(P_i)$ then
            $\mathcal{H}(P_i) = 1$;
            break;
        end
    end
end
```



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 3.5 Foreground detection results from some of the popular methods applied on the ”Woman” sequence from the SCBU dataset [166]. (a) Original frame, (b) Ground truth, (c) MCD [102], (d) MCD NP [72], (e) Stochastic approx [89], (f) SC MCD [166], (g) uNLC [36], (h) OSVOS [13], (i) BASNet [114], (j) CIS [161], (k) uMOD [165], (l) Proposed method.

Table 3.1 The F1-Scores of Different Foreground Segmentation Methods

Methods	Walking	Skating	Woman	Ground1	Ground5	Average
ViBe [8]	0.0375	0.2229	0.0375	0.5656	0.1309	0.2107
FIC [31]	0.0613	0.2373	0.0361	0.4543	0.1319	0.1761
BMRI-ViBE [25]	0.0438	0.2402	0.0400	0.4249	0.1377	0.1730
MCD NP [72]	0.4351	0.4164	0.4935	0.2773	0.3540	0.3519
FP Sampling [164]	0.7058	0.8539	0.7268	0.7977	0.8212	0.6646
MCD [102]	0.7349	0.2447	0.3395	0.6573	0.0678	0.4523
SC MCD [166]	0.7496	0.8560	0.6650	0.8965	0.9326	0.8173
Stochastic approx [89]	**0.8335**	0.6543	0.3986	0.2221	0.2181	0.4392
uNLC [36]	0.0158	0.1419	0.0178	0.0570	0.0143	0.0389
OSVOS [13]	0.3397	0.5344	0.0121	0.7697	0.1224	0.4127
CIS [161]	0.0538	0.3036	0.1522	0.1545	0.0184	0.1418
BASNet [114]	0.3433	0.9379	0.0205	0.6039	**0.9829**	0.6188
uMOD [165]	0.7809	0.9600	0.7269	0.9037	0.9793	0.8546
Proposed method	0.8144	**0.9710**	**0.7874**	**0.9112**	0.9686	**0.8725**

Table 3.2 The Average Runtime of Different Foreground Detection Methods for Each Frame

Methods	Run time (ms)	FPS
ViBe [8]	14.6	68.5
MCD [102]	**7.46**	**134**
MCD NP [72]	20.9	47.85
SC MCD [166]	9.56	104.6
uMOD [165]	29.23	34.2
Proposed method	9.4	106.38

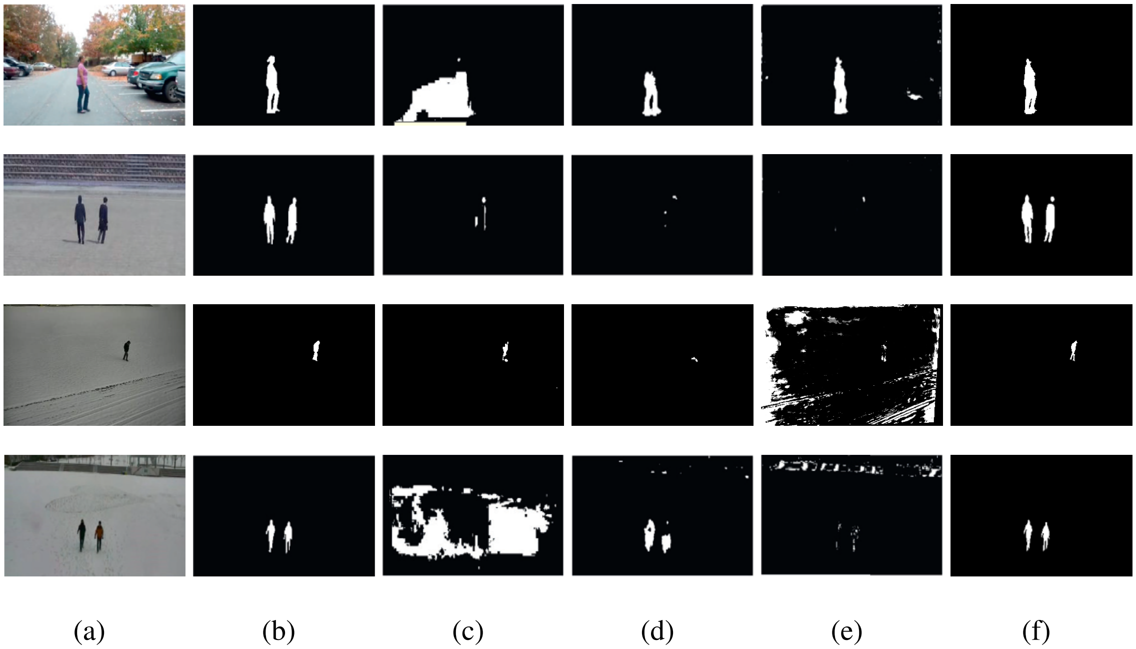


Figure 3.6 Comparison of the qualitative results of background modeling methods. From top to bottom, the rows represent the *Woman2*, *Ground3*, *Ground4*, and *Ground5* sequences. Each subfigure at the first column illustrates one video frame of each sequence with the corresponding ground-truth represented at the second column. The remaining columns represent the classification results of (c) MCD [102], (d) MCD NP [72], (e) Stochastic approx [89], and (f) our proposed methods, respectively.

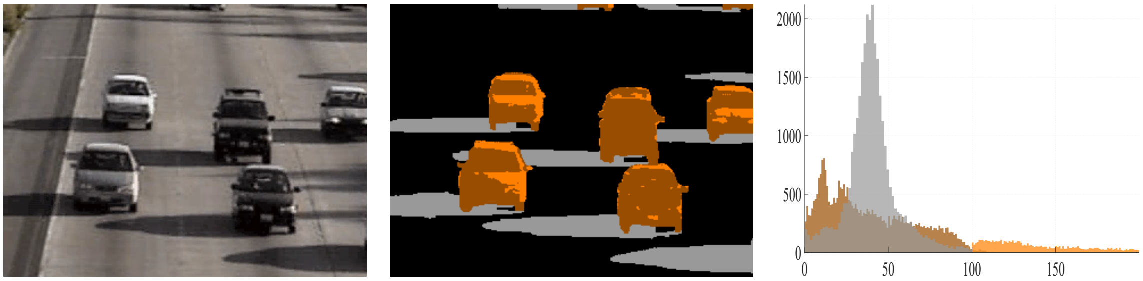
CHAPTER 4

SHADOW SUPPRESSION

4.1 Introduction

Detecting moving objects is a fundamental step in many applications, such as video surveillance, traffic monitoring, content-based video coding, gesture recognition, and human-computer interaction [40]. One of the main challenges in foreground detection is the shadows cast by moving objects in the background, which are often classified as part of the foreground as a result of their similar movement patterns to the moving objects. This misclassification can have severe negative effects on the performance of the further steps in the video analysis systems, such as object classification [37], segmentation [41–43], and object tracking [44, 123]. The task of shadow removal has been addressed in many studies, which have been grouped into seven categories based on the methodologies and exploited features [118], such as color [33] and texture features [120], statistical modeling [58], or a combination of features [48, 124]. Recently, some methods have applied deep convolutional neural networks (DCNNs) for shadow detection [23, 148]. However, these techniques are not suitable for many real-world applications due to the large amount of training data and high demand for computational resources they require. There are a number of other shortcomings in the existing shadow removal methods, such as being limited to specific applications or the requirement for manually specifying sensitive parameters.

In this chapter, a real-time method is proposed to detect and suppress moving shadows with minimal manual involution. First, the global foreground modeling (GFM) method [125] is applied for foreground segmentation due to its efficiency and robustness. Therefore, we employ a region-based classification method, which is capable of dealing with achromaticity and camouflage issues. The watershed segmentation approach [101] is applied in order to extract superpixels. A locally near-invariant illumination feature is



(a) (b) (c)

Figure 4.1 Histogram of RGB norm ratios. (a) Sample video frame [120]. (b) Lighter, darker, and shadowed samples represented by orange, brown, and gray, respectively. (c) Histogram of the RGB norm ratios.

applied to merge correlated superpixels and segment the foreground into a number of regions. These regions are then classified based on the number of candidate shadow samples, foreground-background gradient direction correlation, and the number of external terminal points. In the end, the results of all three steps are integrated for final shadow detection. This integration results in an accurate and robust shadow detection method for real-time video analytics applications.Figure 4.2 shows the system architecture of the proposed shadow detection method.

The remainder of this chapter is organized as follows. In Section 4.2 the major steps of the proposed method, including image segmentation (Section 4.2.1) and region classification (Section 4.2.2) are described in detail. The performance of the proposed method is evaluated on publicly available data in Section 4.3 and Section 4.4 is the conclusion of the chapter.

4.2 A New Cast Shadow Detection Method

In order to subtract the background, the GFM method [125] is applied, which results in a binary motion mask $\mathcal{M}(x, y)$ where $\mathcal{M}(x, y) = 1$ indicates there is significant motion at location (x, y), either caused by an object or moving cast shadow and $\mathcal{M}(x, y) = 0$ means the location (x, y) belongs to the stationary background. The goal here is to classify the foreground pixels into object and shadow classes in order to disregard the moving cast

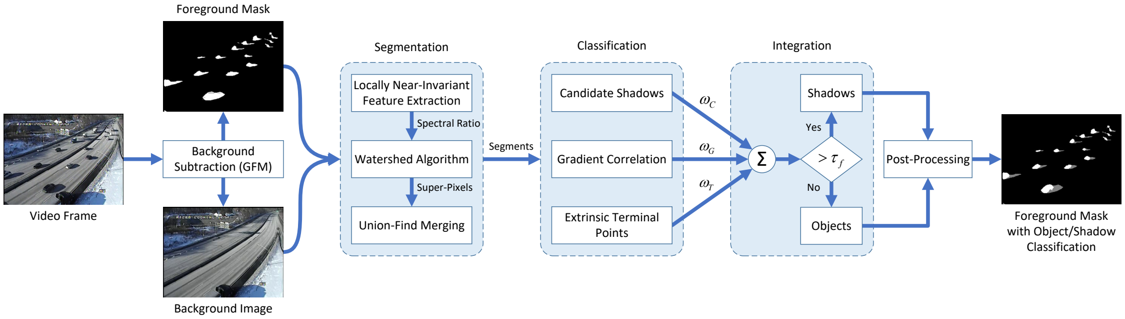


Figure 4.2 The general overview of the system architecture of the proposed shadow detection method

shadows in the further tasks of video analytics. The details of the proposed multi-layer shadow detection method are discussed in this section.

4.2.1 Image segmentation based on locally near-invariant illumination feature

Pixel-wise approaches fail to differentiate between shadows and dark objects that have similar color values (see Figure 4.1) as they are limited only to the variations in the RGB values and do not take the spatial relations between each pixel and its neighborhood into account. Therefore, a combination of pixel-based and region-based techniques can help with locating the dark objects and reducing the misclassification errors. Here, we first apply component analysis [133] in order to partition the binary motion mask $\mathcal{M}(x, y)$ into a set of independent components $\mathbb{R} = \{r_1, r_2, ..., r_k\}$. By assuming that most locations in the scene have rough Lambertian surfaces with negligible specular reflection, there is a single dominant illumination source, there is a specific geometry with constant scene angles, and the camera filters have infinitely narrow bandwidth [33] we can express the camera sensor responses at location (x, y) as follows:

$$C_k(x, y) = q_k \, E(\lambda_k, x, y) S(\lambda_k, x, y) \tag{4.1}$$

where $\lambda_k, k \in \{R, G, B\}$ represents the central frequency of the k-th channel camera filter, q_k, $k \in \{R, G, B\}$ indicates the spectral sensitivities of the three color camera sensors, and

$E(\lambda, x, y)$ and $S(\lambda, x, y)$ are the incident illumination and surface reflectance at location (x, y), respectively [62]. This response can be expressed by the contributions of the direct C_k^d and ambient C_k^a illumination components [97] as follows:

$$C_k = \alpha C_k^d + C_k^a = \alpha q_k E_k^d S_k^d + q_k E_k^a S_k^a, k \in \{R, G, B\} \tag{4.2}$$

where $\alpha \in [0, 1]$ is the attenuation factor that accounts for the unblocked proportion of the direct light, E_k^d, S_k^d, E_k^a, and S_k^a are the incident illumination and surface reflectance of the direct and ambient components, respectively.

With the assumption of $\alpha = 1$ in the background and negligible variations in the ambient illumination, we can define spectral ratio $\overrightarrow{S} = [S_R, S_G, S_B]^T$ as a near-invariant illumination feature:

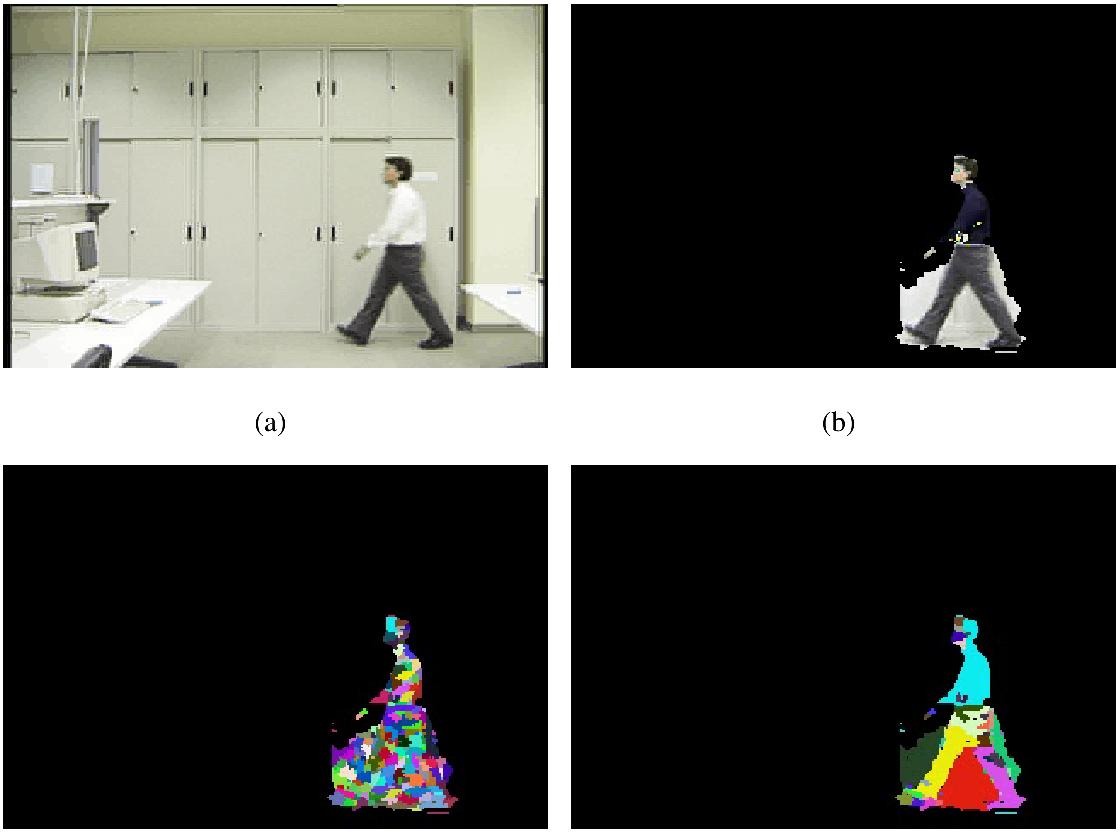
$$S_k = \frac{FG}{BG} = \frac{q_k E_k^d S_k^d}{\alpha q_k E_k^d S_k^d + q_k E_k^a S_k^a} \tag{4.3}$$

where $k \in \{R, G, B\}$ indicates the sensor bands. Since there is little to no direct illumination in the umbra region of the shadow ($\alpha = 0$) and the surface material is the same at location (x, y) in the foreground and background when shadowed ($S_k^d = S_k^a$), the spectral ratio in this region can be indicated as follows:

$$S_k = \frac{E_k^d}{E_k^a} \tag{4.4}$$

which is near-constant among neighboring pixels across the umbra region and the changes are mostly because of the variations in the ambient illumination (Figure 4.3(b)).

We apply the watershed segmentation approach [101] on the spectral ratios of each region in $\mathbb{R}$ to obtain the superpixels. Afterward, correlated superpixels are merged by applying the union-find algorithm [156]. Due to the ratio-invariance property of shadows, two neighboring superpixels are merged if their spectral ratio differences are less than a small threshold across all three color channels. In addition, the edge between two superpixels may have been caused by intersecting shadows, which are difference-invariant [73]. Therefore,



(a)

(b)

(c)

(d)

Figure 4.3 The segmentation process of each frame. (a) Original video frame. (b) Spectral ratio. (c) superpixels. (d) Merged segments based on eq. (4.5).

two neighboring segments are merged if the difference between the foreground values is close to the difference between their background values. Another possible scenario is if the moving shadow is cast over an existing stationary shadow. In this case, the background values are different, but the foreground values are similar and close to the background value of the darker segment.

Two neighboring superpixels/segments s_i and s_j are merged according to three criteria:

$$\begin{cases} {}^{FG_i}\!/_{BG_i} \approx {}^{FG_j}\!/_{BG_j} \\ FG_i - FG_j \approx BG_i - BG_j \\ BG_j \approx FG_i \approx FG_j, \quad BG_i \gg BG_j \end{cases} \tag{4.5}$$

If any of the above conditions hold, the two segments will be merged. Figure 4.3 shows an example of the segmentation and merging process. At this point, each foreground component $r_k \in \mathbb{R}$ is partitioned into a number of segments s_l^k, such that:

$$\bigcup_{l=1}^{n_k} s_l^k = r_k, \quad \bigcap_{l=1}^{n_k} s_l^k = \varnothing, \quad \bigcup_{k=1}^{K} r_k = \mathbb{R} \tag{4.6}$$

where n_k is the total number of segments s_l^k at each region r_k (Figure 4.3(c)). Note that the efficiency of this method is much higher than pixel-wise segmentation methods [3] due to the use of superpixels and applying the union-find algorithm. This way, if two pixels belong to the same superpixel/segment, there is no need to calculate the merging criteria. Otherwise, the dissimilarity measures are calculated in order of priority only for the neighboring pixels of two separate superpixels/segments. If the two superpixels/segments are decided to be merged, all the pixels corresponding to them will be merged at the same time. Figure 4.3 illustrates the steps of the segmentation method in a sample video frame. The white and gray colors represent the 0 and 1 values in the binary masks, respectively.

4.2.2 Segment classification based on various heuristic cues

In order to improve the robustness and accuracy of the object/shadow classification in the foreground, we employ four different heuristic cues simultaneously, including thresholds, gradient correlation, and the number of extrinsic boundary points. The classification results from all three steps are aggregated for final classification. In this section, the steps in the segment classification process are explained in detail.

Extracting candidate shadow pixels Since the HSV color-space separates the chromaticity from the intensity to a good level, it is useful to distinguish the variations in illumination from the changes in material. Figure 4.4 illustrates the potential shadow zone in the RGB color-space which is a portion of the conic region in the RGB space Since shadows have little to no effect on the H(hue) component of the HSV color-space, we choose the S(saturation)

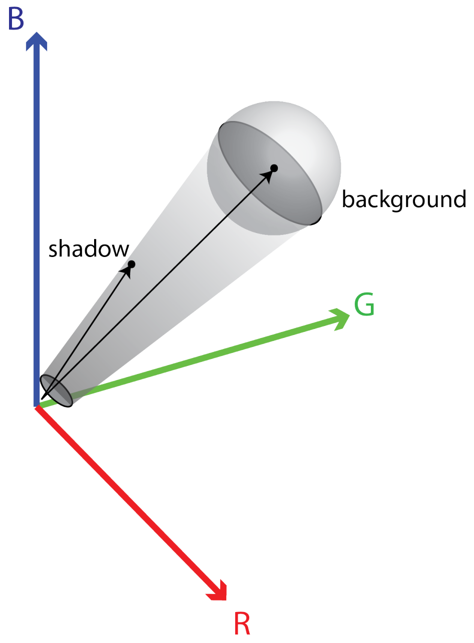


Figure 4.4 The initial shadow candidate detection. Pixels falling in the conic region are considered as potential shadow samples.

and V(value) components to set the criteria. The value ratio can roughly specify the attenuation, which is represented by the vector magnitudes, and the saturation component can determine the apex angle of the cone, which depends on the ambient illumination. By assuming S_f, V_f, S_b, and V_b to be the saturation and value components of the foreground and background, respectively, the chromatic criteria can be formulated as follows:

$$\mathcal{P}(x,y) = \begin{cases} 1, & (\tau_{vl} < {}^{V_f}/_{V_b} < \tau_{vh}) \\ & \wedge(\tau_{sl} < S_f - S_b < \tau_{sh}) \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

where $\mathcal{P}$ is a binary mask where $\mathcal{P}(x,y) = 1$ indicates that pixel at location (x,y) is a potential shadow sample, and $\tau_{vl}, \tau_{vh}, \tau_{sl}$, and τ_{sh} denote the lower and upper thresholds for the value ratio and saturation variation, respectively. All the foreground pixels that satisfy these criteria are considered to be potential shadow candidate samples. Figure 4.5 illustrates an example of potential shadow represented by gray color.

Each segment s_l^k of region r_k is classified as an object or shadow according to its intersection with potential shadow candidates. If most pixels inside a segment are classified as potential shadow candidates, that segment is likely to belong to the shadow class. This can be expressed as follows:

$$\mathcal{C}(s_l^k) = \begin{cases} 1, \text{if } \frac{|\mathcal{P} \cap s_l^k|}{|s_l^k|} > \tau_p \\ 0, \text{otherwise} \end{cases} \tag{4.8}$$

where $\mathcal{C}(s_l^k)$ is a binary mask where $\mathcal{C}(s_l^k) = 1$ if more than τ_p of the pixels in segment s_l^k are classified as potential shadows.

Calculating the gradient direction correlation The amount of gradient information introduced by the objects is generally more than the amount introduced by shadows. The dominant edges are extracted by applying the Canny edge detection method, and the difference in gradient direction between the frame and the background is calculated as follows:

$$\Delta\theta(x, y) = cos^{-1} \frac{\overrightarrow{\nabla} f(x, y) \cdot \overrightarrow{\nabla} b(x, y)}{\left\| \overrightarrow{\nabla} f(x, y) \right\| \left\| \overrightarrow{\nabla} b(x, y) \right\|} \tag{4.9}$$

where $\overrightarrow{\nabla} f(x, y)$ and $\overrightarrow{\nabla} b(x, y)$ are the gradient vectors at location (x, y) in the frame and the background, respectively, and $\Delta\theta(x, y)$ is the angular distance between two vectors. If the gradient direction is highly correlated between the frame and the background in a segment, it has a higher probability of belonging to the shadow class. This criterion is expressed as follows:

$$\mathcal{G}(s_l^k) = \begin{cases} 1, \text{if } \frac{1}{|s_l^k|} \sum_{i=1}^{|s_l^k|} H(\Delta\theta_i - \tau_a) > \tau_e \\ 0, \text{otherwise} \end{cases} \tag{4.10}$$

where $|s_l^k|$ is the number of pixels in the segment s_l^k, $H(.)$ denotes the unit step function which is one if the angular distance is larger than or equal to a threshold τ_a, and $\mathcal{G}(s_l^k)$ is a

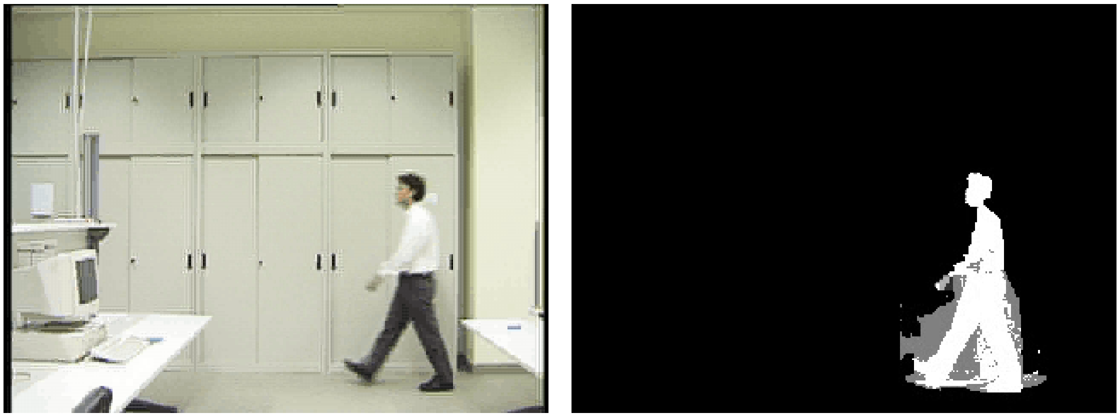


Figure 4.5 Extracting potential shadow candidates. (a) Sample video frame. (b) Potential shadows.

binary mask which is one if a fraction more than τ_e of the pixels in segment s_l^k have similar gradient direction in the frame and the background.

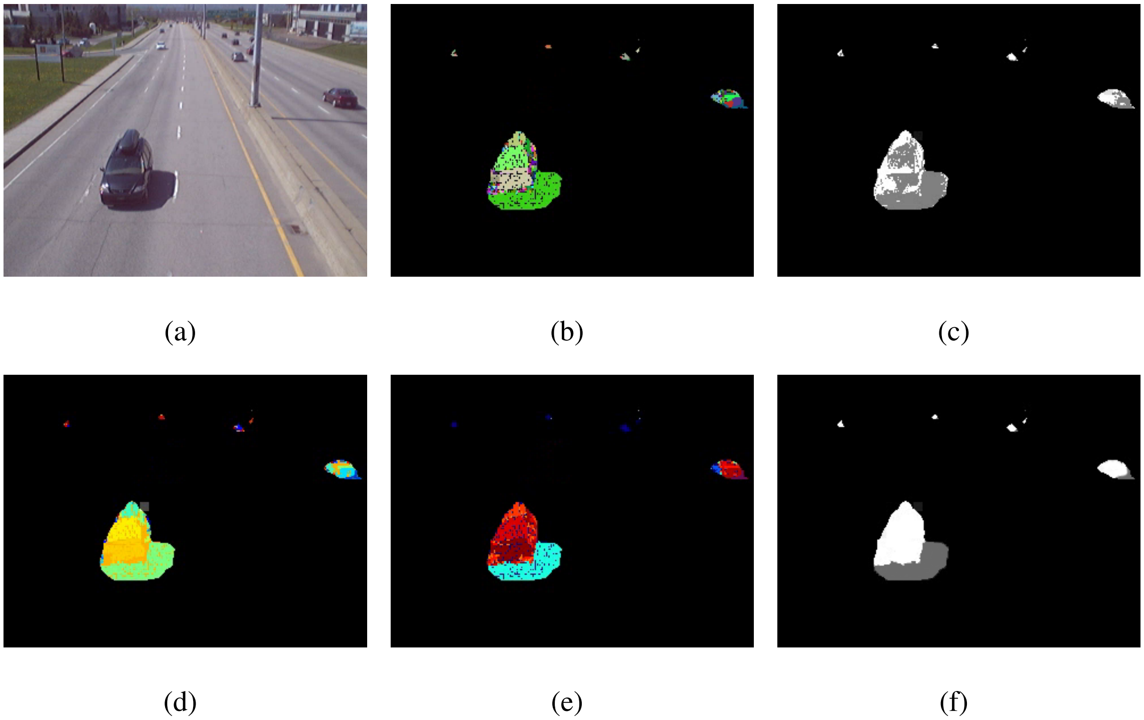
Computing the number of extrinsic terminal points Another observation about shadow samples is their spatial distribution around the objects, which results in shadow segments of each region containing a considerable number of extrinsic terminal points. Such criterion can be expressed in a binary mask $\mathcal{S}_t$ as follows:

$$\mathcal{T}(s_l^k) = \begin{cases} 1, \text{if } \frac{\left|\mathbb{T}(r_k)\cap\mathbb{T}(s_l^k)\right|}{\left|\mathbb{T}(s_l^k)\right|} > \tau_t \\ 0, \text{otherwise} \end{cases} \tag{4.11}$$

where $\mathbb{T}(r_k)$ and $\mathbb{T}(s_l^k)$ are the sets of external boundary points of the foreground component r_k and each of its segments s_l^k, respectively, and $\mathcal{T}(s_l^k)$ is a binary mask which is 1 if a fraction more than τ_t of terminal points are external.

Final shadow detection based on integration of the previous steps For the final object and shadow classification, the results of the previous steps are integrated by calculating the weighted summation as follows:

$$\mathcal{W}(x,y) = w_C\mathcal{C}(x,y) + w_G\mathcal{G}(x,y) + w_T\mathcal{T}(x,y) \tag{4.12}$$



(a) (b) (c)

(d) (e) (f)

Figure 4.6 Classification process of a sample frame from Highway 3 sequence. (a) Original video frame. (b) segmentation. (c) Potential shadows. (e) Heatmap of gradient correlation. (d) Heatmap of external terminal points. (f) Region based classification ($\mathcal{S}$).

where $w_C \in [0, 1]$, $w_G \in [0, 1]$, and $w_T \in [0, 1]$ are the weights indicating the significance of the shadow detection results based on chromatic criteria, gradient correlation, and extrinsic terminal points, respectively. The three weights are normalized and summed up to one:

$$w_C + w_G + w_T = 1 \tag{4.13}$$

We have considered w_G to be twice the value of w_C and w_T. By thresholding the weighted sum values we obtain a binary mask $\mathcal{F}$ which represents the final shadow detection results as follows:

$$\mathcal{F}(x, y) = \begin{cases} 1, \text{if } \mathcal{W}(x, y) > \tau_f \\ 0, \text{otherwise} \end{cases} \tag{4.14}$$

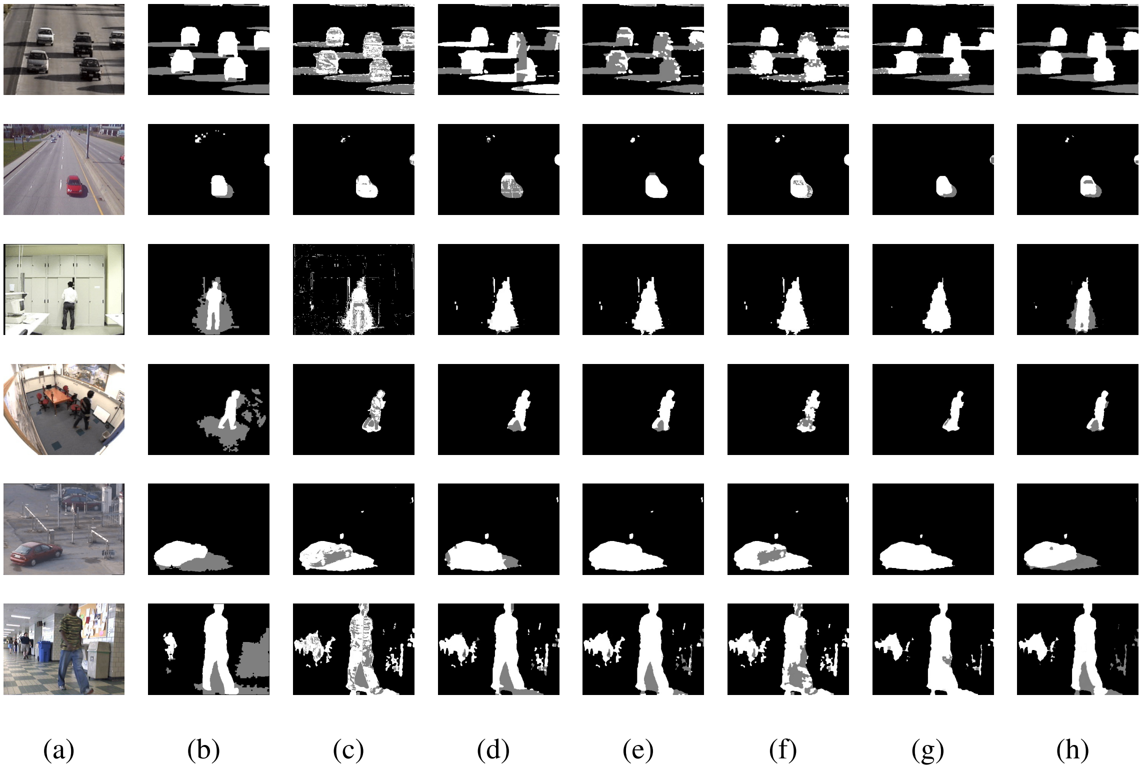


Figure 4.7 The foreground masks and the detected shadows in different methods using various sequences from the ATON dataset [120]. (a) Original video frame. (b) Ground truth. (c), (d), (e), (f), (g), and (h) are the results of Cucchiara et al. [33], Hsieh et al. [56], Sanin et al. [120], Huang and Chen [58], Amato et al. [3], and our proposed method, respectively.

where τ_f is a threshold, $\mathcal{F}(x, y) = 1$ indicates that the pixel at location (x, y) belongs to shadow and $\mathcal{F}(x, y) = 0$ means it belongs to moving objects. Subtracting $\mathcal{F}$ from $\mathcal{M}$ will result in a shadow-free foreground mask. Figure 4.6 shows an example of the described steps in the classification procedure. In the heatmaps, the warmer colors represent the objects and the colder colors represent shadows.

4.3 Experiments

The quantitative and qualitative results of the proposed method are evaluated using publicly available video data [120]. The spatial resolution of each video sequence is 320×240 pixels and each video contains 15 frames per second. The underlying system hardware is

Table 4.1 The Average Shadow Detection Runtime for Each Video Frame in Different Methods

Methods	Runtime (ms)	
	320×240	640×482
Cucchiara et al. [33]	23	141
Zhu et al. [170] (with GPU)	421	1069
Huang and Chen. [58]	16	81
Sanin et al. [120]	61	244
Hsieh et al. [56]	**5**	**16**
leone and Distante. [80]	135	284
Amato et al. [3]	16	102
Proposed method	**5**	**16**

Table 4.2 The Average Runtime of The Main Steps in Shadow Detection After Background Subtraction

Steps	Runtime (ms)	
	320×240	640×482
pre-processing	0.32	0.55
segmentation	2.85	11.44
candidate shadows	0.07	0.28
gradient correlation	1.40	3.36
terminal points	0.27	0.56
post-processing	0.37	0.73
Total	5.29	16.93

a Dell XPS 8900 PC with a 3.4 GHz processor and 16 GB of RAM. The processing time is, on average, 5.48 milliseconds for each frame, which is consistent with the efficiency requirements of real-time applications. Table 4.1 compares the run-time with some of the popular shadow detection methods for video frames of size 320×240 and 640×482 pixels. Table 4.2 contains detailed run-time for each step of the process. The preprocessing step involves removing the fringe of the shadow segments and smoothing each image by Gaussian blurring. The post-processing is a noise correction step that assigns a shadow/object class to each foreground pixel according to the majority of its surrounding pixels.

In Figure 4.7, a sample frame from some videos is illustrated along with the shadow detection results of some of the representative methods. The thresholds τ_p, τ_e, τ_t, and τ_f are all empirically set to 0.5 and show low sensitivity when experimented with various

Table 4.3 The Shadow Detection Results Compared to Other Methods in Terms of F-Measure

	IntelligentRoom	Laboratory	Highway-1	Campus	Highway-3
Cucchiara et al. [33]	78.18	84.33	70.36	53.22	53.40
Hsieh et al. [56]	61.26	56.51	70.55	58.88	54.61
Huang et al. [57]	71.59	54.46	56.79	55.24	48.79
Leone et al. [80]	75.27	84.69	28.69	67.39	10.58
Sanin et al. [120]	88.59	78.05	74.04	66.81	53.56
Wang et al. [143]	**94.63**	**90.30**	84.80	80.42	68.68
Proposed method	92.68	84.22	**88.14**	**89.92**	**84.09**

videos. Three performance measures are calculated for quantitative evaluation of the shadow detection method as follows:

$$\begin{cases} \xi = TP_o/(TP_o + FN_o) \\ \eta = TP_s/(TP_s + FN_s) \\ F_1 = 2 \times (\eta \times \xi)/(\eta + \xi) \end{cases} \tag{4.15}$$

where TP_o and TP_s denote the true positive rates of the object and shadow pixels and FN_o, and FN_s are the false negative rates of the object and shadow pixels, respectively. η, ξ, and F_1 denote the shadow detection rate, shadow discrimination rate, and F-measure, respectively. In Table 4.3 the calculated measures for the performance evaluation are reported along with some of the popular methods [120].

4.4 Conclusion

This chapter presents a new moving cast shadow detection method to separate moving objects from their cast shadows in real-time applications of video analytics. After applying the global foreground modeling (GFM) method for background subtraction, the foreground class contains the moving objects along with their cast shadows. First, a set of chromatic criteria in the HSV color space is applied in order to extract the potential shadow candidates. Then a segmentation technique is used based on the physical properties of the surface reflections to group the correlated pixels in each foreground component and classify the segments according to a set of three criteria. The final decision about shadow and object classification is made through an integration process of the previous steps. The experimental results demonstrate the effectiveness of the proposed method in real-time video analytics applications.

CHAPTER 5

REGION-OF-INTEREST DETECTION

5.1 Introduction

A region of interest (ROI) is a sample within a dataset identified for a particular purpose [12]. In the case of video analysis, a region of interest refers to a subspace of the video frame that is identified as the region of main focus. Selecting one or multiple regions of the video frame to perform video analytic tasks not only reduces the unnecessary and false results, but also decreases the computational complexity due to a lower volume of input data, which means a great deal to real-time applications. One of the main applications of video analysis is in traffic surveillance videos, where the region of interest usually refers to the road area and its proximity. The area of focus in traffic video analysis tasks such as vehicle counting, speed estimation, and detecting traffic incidents such as wrong-way vehicles and vehicle accidents is the road lanes and shoulders. Currently, in most applications, the region of interest is selected manually, which has to be performed for every video and repeated in case of changes in the angle or distance of camera view.

Automatic road recognition has been a popular research topic in applications regarding traffic surveillance videos [121] and in-vehicle perception [17]. Most of the techniques used in these studies are applicable in both areas, with the main motivation of the former being ROI determination and the latter providing useful information for advanced driving assistance systems. In some studies, the local features such as color [87], brightness [147], texture [158], or a combination of them are extracted in order to classify the pixels into road and non-road classes. Some methods tend to rely on the road models in order to match them with low-level features and detect the road region [27]. Several techniques suggest utilizing motion information and temporal features obtained from a sequence of video frames in order to extract the road area [140]. Recently, convolutional deep neural networks have also been applied to segment the road region due to their ability to model non-linear variable relationships [14, 22]. In terms of road detection in traffic video analytic applications, the performance of supervised methods can suffer from a wide range of different illumination and weather conditions, image resolutions, camera viewing angles, and distance from the road surface.

The focus of this study is road recognition and ROI determination in traffic surveillance videos to aid with detection of driving violations, traffic incident recognition, and reduce the computational complexity of urban and highway traffic video analysis tasks. We propose a motion-based statistical method to extract the road region and separate the road map into left and right sides based on the two major moving directions of vehicles in traffic videos. No assumption about the structure of the road is made, and therefore, this method can be used for structured and unstructured road scenarios. The locations of moving vehicles are appropriately assumed to be associated with the roadway region and they are utilized as color samples to estimate the location of road pixels. A novel foreground segmentation technique [125] based on Gaussian mixture models is applied in order to detect the moving vehicles and subtract the stable background. The pixel values of the background image at the corresponding locations of the vehicles are utilized as initial road samples and as seed points by the flood-fill method in an accumulative manner, and several road probability maps are generated. The extracted probability values are then combined in order to estimate a more accurate road region map, which is further refined by using the aggregated foreground mask. The straight and curved road boundaries are estimated by second-degree polynomial curve-fitting to improve the obtained road map from the previous step by removing possible extra pixels that are incorrectly categorized as road pixels by the flood-fill method. The use of color features combined with gradient information and temporal features makes this method robust against illumination changes and severe weather conditions. At the same time, a statistical approach is applied with Lucas-Kanade optical flow and is further refined by a blob-tracking method to separate the two major directions in roads with bidirectional traffic. The detected road regions can further be updated and used as ROI in traffic video surveillance applications.

5.2 A New Automatic Method For Road Region Extraction

Extracting the region of interest is an important preprocessing step in many image and video analytic applications. Currently, the selection of ROI is mostly performed manually by a human agent at the initial stages of preprocessing. Manually determining the region of interest, which in traffic video analysis refers to the road region, is an exhaustive and time-consuming task for human agents. Retrieving the ROI automatically can reduce the need for manual work, and constant updates in the extracted ROI help with adaptation to new scenes when the camera's view changes. We propose a fully automatic method for road recognition that updates the ROI at each frame of the video and therefore can quickly adjust to changes in the camera's view. The proposed method can be performed in real-time and is adaptive to cameraview changes and various illumination scenarios. The only madeassumption is about the location of the vehicles, which are assumed to move mostly along the road region. Our proposed method has three major contributions: (i) The new motion-based statistical method can automatically extract the road region and reduce a great deal of manual work. (ii) The newroad probability estimation method can generate a reliable roadmap from the initial frames of the video without the need to wait for many vehicles to pass along the road region. (iii) The novel ROI determination approach can extract a separate ROI for each side of roads with bidirectional traffic. The ROI determination is fast and robust for real-world application use.

5.2.1 Selection of the initial road samples

In the case of applications with an onboard camera system, initial road samples are usually taken from a triangular area in front of the vehicle. In contrast, in applications with a stationary camera overlooking the roadway, the initial road samples can be extracted based

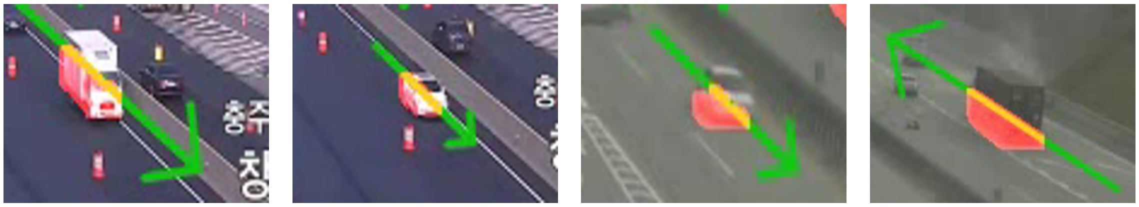


Figure 5.1 Sampling the road pixels from the background image based on the direction of moving vehicles in order to avoid sampling non-road pixels. The red color indicated the location of the sampled road pixels.

on the location of moving vehicles. The further steps for road segmentation based on the initial samples can be commonly used in applications of traffic surveillance and self-driving vehicles. The focus of this study is on automatic ROI determination in traffic surveillance videos. However, our proposed feature extraction and classification approach can work for road segmentation in self-driving vehicles as well.

In order to obtain an estimate of the road region during the initial frames of the video, we first attempt to detect the vehicles and segment them from the still background. The global foreground modeling (GFM) method introduced by Hang and Liu [125] is utilized to detect the location of the moving vehicles and to subtract the stationary background image from the video frames. The GFM foreground segmentation approach was chosen due to its ability to quickly subtract the background in a video captured by a stationary camera. Also, the GFM method is robust in dealing with stopped vehicles, which are continuously detected as foreground and therefore separated from the background image. The road estimation method is applied on the subtracted background with the assumption that most vehicles pass along the roadway. The corresponding locations of the moving and stopped vehicles in the background image are considered to be samples of the road region, which are in turn utilized to estimate the probability of all background pixels. The generated probability maps are further used to classify the pixels into road and non-road in order to segment the road region from other areas and determine the ROI based on the extracted road map.

The selected pixels for road samples should be exclusively from the road region in order to obtain a good estimation of road pixel-values. In many intelligent vehicle systems, such as automatic driving and advanced driver assistance systems, where the field of view is similar to that of the driver, the road region priori is approximated as a triangular region at the mid-bottom of the frame [83]. In the case of traffic surveillance videos, where the cameras are overlooking the road, there can be no initial assumption of the road's location without any observation of the images. On the other hand, in a generally short period of time, vehicles pass along different parts of the road rather than a specific lane. Subsequently, accumulating the motion masks obtained from the foreground segmentation method covers the majority of the road region in a relatively short period of time. Each time a vehicle passes along the road, the pixels of the road map in the corresponding location to its foreground mask are added by a constant positive value. By applying the Otsu's threshold [49], we can get rid of the remaining noise and obtain a binary image representing the estimated active zone of the traffic flow.

Here, a valid assumption is made that most of the pixels in the background image with locations corresponding to those of the vehicles in the foreground mask belong to the roadway region. However, due to the variety of camera view angles, different sizes of vehicles, and occasional movements in the non-road regions, some of the pixels of the foreground mask can belong to the areas outside of the road. In order to discard the faulty outputs of the foreground segmentation method, a tracking approach is utilized to only include the foreground mask of the moving vehicles and discard the pixels that are segmented as foreground due to the possible motion in the areas outside of the roadway. For the sake of simplicity and real-time performance, we apply the blob-tracking method [18] for vehicle tracking. At each frame, the foreground mask of each tracked vehicle is saved separately, and if the life-time and moving length of that track exceeds predefined thresholds, the corresponding pixels of the entire foreground mask of that track in the active traffic region map are added with a positive number. Applying filters to the foreground mask based on track life-time and the moving length of each track ensures that only vehicles passing along the road are considered as part of the active traffic region and noises in the foreground mask are disregarded.

In order to obtain a mask containing pixels that represent road samples Ω_{rsm}, only the foreground mask of vehicles with sizable movement and a long enough tracking lifetime is considered. The moving direction of each vehicle is estimated and updated as follows in each sequence of f frames:

$$\begin{aligned} v_x &= x_{m_2} - x_{m_1} \\ v_y &= y_{m_2} - y_{m_1} \\ d_i &= arctan(v_y, v_x) \\ m_{v_i} &= \sqrt{v_x^2 + v_y^2} \end{aligned} \tag{5.1}$$

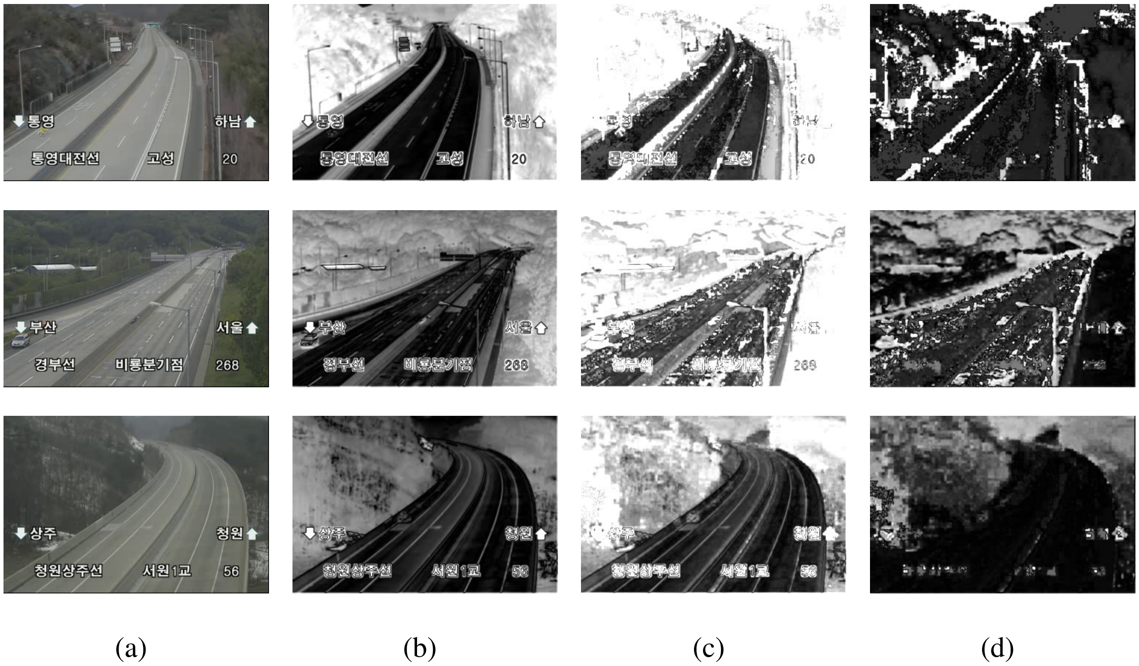
where v_x and v_y are the components of the velocity vector, x_{m_2} and y_{m_2} are the average x and y values of the blob centroid in the most recent $f/2$ frames, x_{m_1} and y_{m_1} are the average x and y values of the blob centroid in the remaining $f/2$ frames, d_i is the estimated direction of the vehicle i, and m_{v_i} is the estimated magnitude of the vehicle i, respectively. The filtered foreground mask of each vehicle is then cropped with regard to its moving direction so that only the part that corresponds to the road region is added to the Ω_{rsm} mask. Figure 5.1 illustrates some examples of the road sampling strategy which helps avoid including non-road regions in the Ω_{rsm} at the boundaries of the roadway. The road samples are accumulated throughout the video, and the Ω_{rsm} mask will cover more parts of the road when more vehicles pass along the roadway.

5.2.2 Road region probability map extraction

Creating a single probability map that represents the roadway region in all cases is rather difficult due to the variety of illumination, texture, color, and other visual conditions. Therefore, generating multiple probability maps and merging them helps obtain a more reliable probability distribution for classifying the pixels into road and non-road regions. In this section, multiple approaches are taken in order to generate a number of probability maps using low-level features, e.g., color, edge, and temporal features. The generated probability maps are further combined together to obtain a binary classification mask, which is in turn refined by the accumulative foreground mask as the number of passing vehicles increases.

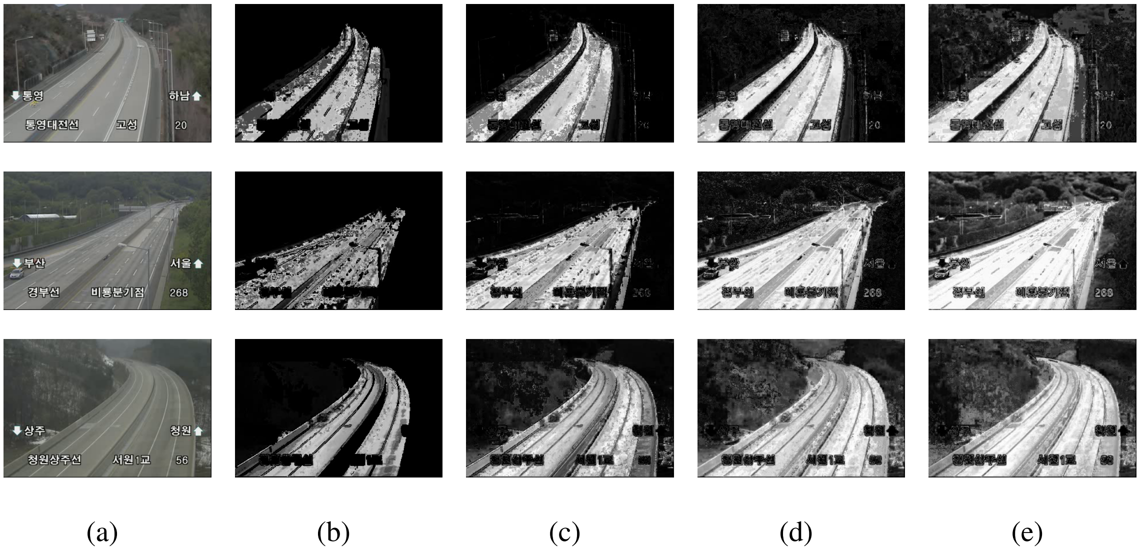
Extraction of probability maps based on difference images One approach to estimating the road probability of the pixels is to compare the pixel's value to the average value of the initially selected road samples in Ω_{rsm}. Similar to the approach used by Wang et al. [146], the gray-scale image G^* of background is first smoothed by applying a Gaussian convolution kernel of size 3×3 to reduce the noise effect. Then the absolute difference between the mean value $G^{\bar{*}}_{rsm}$ of the grayscale image in the location of Ω_{rsm} and each pixel in the smoothed grayscale image is utilized to obtain a gray-scale difference image G. A similar process is carried out on the three channels of the smoothed background image, and the three outputs are added together to obtain another different image C based on the color input. In traffic scenes where the roadway is considerably different in color from the surrounding area, the hue channel of HSV color space can be a distinguishable factor in segmenting the road pixels from the image, especially at the boundaries of the road. The background image is also converted to HSV color space and the hue channel is utilized to acquire a difference image H through a similar process. Figure 5.2 illustrates sample difference images obtained from real traffic video data.

Lower values in the difference images correspond to the parts of the image that are closer to the average value of the road pixels in Ω_{rsm} and have a higher probability of belonging to the road region. Therefore, the probability value of each pixel should be inversely proportional to the corresponding pixel in the difference image.Based on the difference images obtained so far, probability maps can be estimated accordingly based on



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

Figure 5.2 Extracting the auxiliary road region probability maps using difference images. (a) The background image. (b), (c), (c) are the gray-scale, color, and hue difference images, respectively.



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

Figure 5.3 Extracting the auxiliary road region probability maps using difference images. (a) The background image. (b), (c), (d), (e) are the extracted probability maps P_G, P_C, P_H, and P_S, respectively.

which the probability of each pixel is calculated as follows:

$$P'_K(p_i) = \frac{1 - K(p_i)}{max(K(p_i)|p_i \in K)} \tag{5.2}$$

where $i = 1...N$ is the pixel index, $K \in \{G, C, H\}$ refers to each difference image, and $P'_K(p_i)$ is the probability of the pixel p_i belonging to the road region in the difference image K. In order to normalize the brightness and increase the probability contrast of the probability maps, their histograms are normalized to obtain an approximation of the probability density function, and the normalized histograms are equalized as follows:

$$\begin{gathered} H'_{n,P'_K} = \sum_{0 \leq m < n} H_{P'_K}(m) \\ P_K(p_i) = H'_{P'_K}(P'_K(p_i)) \end{gathered} \tag{5.3}$$

where $i = 1...N$ is the pixel index, $K \in \{G, C, H\}$ represents each difference image, $H_{P'_K}$ and $H'_{P'_K}$ are the normalized histogram and the integral histogram of probability map P'_K respectively, and P_K refers to the equalized histogram of each probability map.

The pixels representing the road region in traffic videos usually have a close value in most parts of the roadway contained in the frame, and the road samples represent a high percentage of the road pixels. Therefore, the standard deviation is usually assumed to have a relatively small interval with a high level of confidence. The further the pixel values in G are from the standard deviation of the pixels in the road sample mask Ω_{rsm}, the probability of belonging to the road region should drop. Considering the standard deviation of the road samples, another probability map can be obtained as follows that specifically favors the pixels that are close to the road samples:

$$\begin{gathered} \alpha(p_i) = max(0, sgn(G(p_i) - \sigma_{rsm})) \\ P_S(p_i) = 1 - \alpha(p_i)[\frac{G(p_i)}{k\sigma_{rsm}} + \frac{1}{k^2}], k - 1 \leq \frac{G(p_i)}{\sigma_{rsm}} < k \end{gathered} \tag{5.4}$$

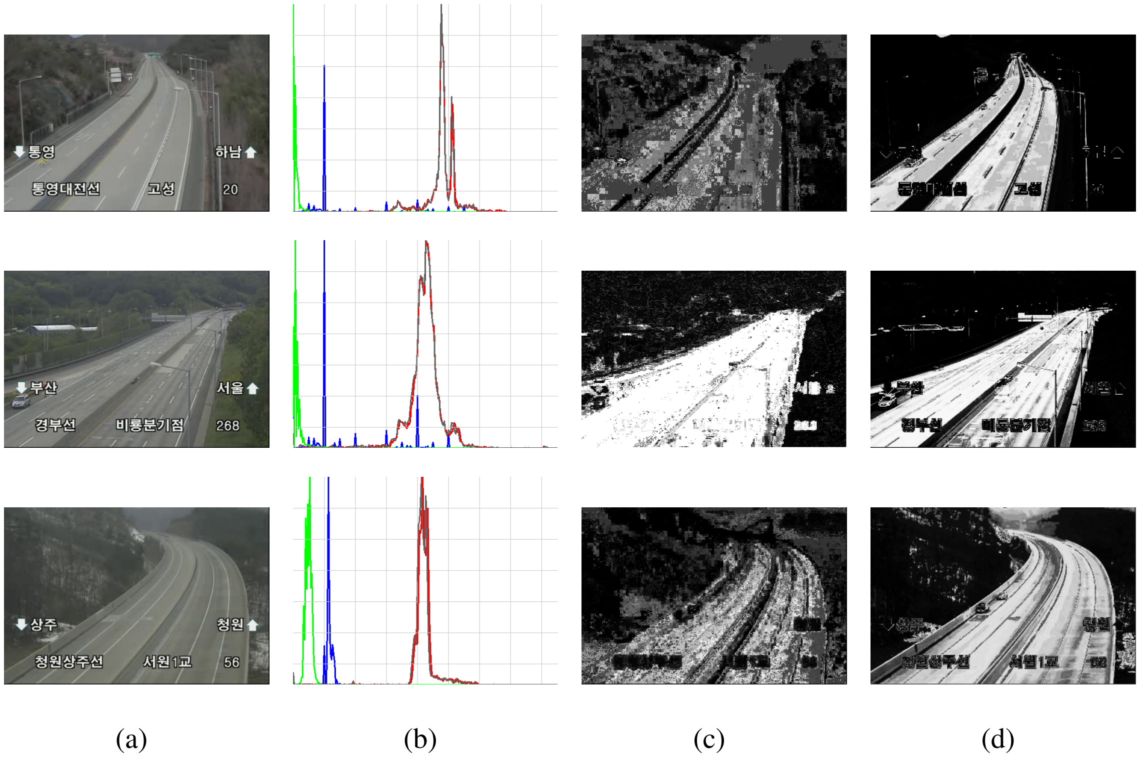
where $p_i \in G$, $i = 1...N$, σ_{rsm} is the standard deviation of the pixel values in Ω_{rsm} mask of G, k is a natural number in $\{k \in \mathbb{N} | 1 < k \leq max(G(p_i) - \sigma_{rsm})\}$, and $P_S(p_i)$ is the resulting probability map. Figure 5.3 represents the extracted probability maps from the difference images.

Extraction of probability maps based on histogram models Another approach to estimating the road region probability of each frame is to utilize histogram models extracted from the road and non-road samples. A similarity measure is used in order to generate probability maps that help classify the road and non-road pixels. The non-road samples are taken from the regions outside of the final estimated road region in the previous frame. The normalized histograms of the blue and green channels of the background image and the gray-scale image G^* are used to estimate probabilities as follows:

$$P_K(p_i) = \frac{N_K^r(K(p_i))}{N_K^r(K(p_i)) + N_K^{nr}(K(p_i))} \tag{5.5}$$

where $i = 1...N$ is the pixel index, $K \in \{Blue, Green, Gray\}$ refers to the blue and green channels of the background image and the gray-scale image G^*, $N_K^r(K(p_i))$ and $N_K^{nr}(K(p_i))$ are the values of the $K(p_i)th$ bin in the histogram models obtained from the road samples in Ω_{rsm} and non-road samples of the previous frame respectively, and $P_K(p_i)$ is the probability of the pixel p_i belonging to the road region in the image K. Since the histogram models of the red channel and gray-scale of background image are close (as seen in Figure 5.4(b)), the red-channel histogram is not considered and two probability maps P_{Ghist} and $P_{GBhist} = P_{Green} + P_{Blue}$ are obtained from the gray-scale image G^* and a combination of green and blue channels of the background image, respectively.

Extraction of probability maps based on edge information In many road detection methods [84, 163] gradient filters are applied in order to differentiate between the road and non-road regions based on the presumed fact that the road region contains considerably less amount of gradient information compared to the surrounding areas. This is usually not the case in traffic surveillance videos, where the objects are further from the camera and the edge density is not much higher in the non-road regions. However, the dominant road



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

Figure 5.4 Extracting the road region probability maps using histogram models. (a) The background image. (b) The histogram plot representing the RGB channels and gray-scale image of the background image. (c), (d) are the extracted probability maps P_{GBhist} and P_{Ghist}, respectively.

boundaries create strong edges, which can be used along with the location of the vehicles to separate the road region from the surroundings. The Canny edge detection method is applied on the gray-scale difference image G with lower and upper thresholds set to $\tau_l = 0.66 \times M$ and $\tau_h = 1.33 \times M$, respectively; where M is the median luminance of G. Figure 5.7(c) represents the edges extracted from the background image. Therefore, since the geometric distortion caused by the perspective view of the camera lens results in the loss of valuable edge information in the regions that are further from the camera, the horizontal line can be estimated and considered as a secondary boundary in addition to the background edges in order to avoid including areas like the sky above the vanishing point inside the road region.

In order to avoid the inclusion of non-road pixels as seed points for flood-fill operation, a single block from the colored difference image C located at one of the corner points of each vehicle's surrounding bounding box is chosen as the road sample. The selected corner is picked according to the moving direction of each vehicle in order to make sure the sample block is certain to belong entirely to the road region. The pixels in the chosen blocks form a flood seed mask Ω_{fsm} which contains the starting nodes for the procedure of flood-fill algorithm. The extracted edges from the gray-scale difference image G along with the horizontal line are used as boundaries for the flood-fill algorithm with a connectivity value of 4, in order to fill the connected components with a constant value in a flood-fill mask image M_F. The maximal lower and upper intensity difference between the currently observed pixel and one of its four nearest neighbors of the same component, or a new seed pixel being added to the component is calculated based on the standard deviation of the colored difference image C as follows:

$$\begin{aligned} m &= \frac{1}{N}\sum_{i=1}^{N} C(p_i) \\ s &= \sqrt{\frac{\sum_{i=1}^{N}(C(p_i)-m)^2}{N}} \\ thr &= max(1, \frac{s}{k}) \end{aligned} \tag{5.6}$$

where m is the mean value of the colored difference image C, N is the total number of pixels in the background image, p_i is the intensity value of the $i-th$ pixel, k is a pre-defined constant, and thr is the maximal lower or upper intensity difference. The maximal lower and upper thresholds are selected based on the general intensity difference among the pixels of the entire background image.

When the dissimilarity among intensity values is relatively large, the connected components in the Flood-Fill method tend to grow slower, and thus a larger value for the maximal thresholds is chosen. On the other hand, in cases where the intensity values are close, e.g., foggy and rainy weather conditions or night time videos, the distinction level between pixels that belong to the road region and pixels that belong to the side of the road is lower. Therefore, in order to avoid connecting the pixels outside of the road area to the

Algorithm 2: Acquiring The Accumulative Foreground Mask

```
   Input:
          The size of each video frame
          The set $T$ of vehicle tracks in the current frame
          The set of blobs for each track $B_t = \{b_1, ...b_n\}$
          A set of predefined thresholds $\mathcal{T} = \{\tau_d, \tau_i, \tau_s\}$
   Output:
          The accumulative foreground mask $\mathcal{F}_{acc}$ of the same size as the video
          frame
1  initialize $\mathcal{F}_{acc}$ with 0;
2  foreach $t \in T$ do
3      if $size(t) < \tau_s$ then
4          continue;
5      end
6      $d = \|\mathbf{t_{cn}} - \mathbf{t_{c1}}\|$;
7      if $d < \tau_d$ then
8          continue;
9      end
10     add track's current blob $b_n$ to track's accumulative mask $\mathcal{F}_t$;
11     if $t_i > \tau_i$ then
12         $\mathcal{F}_{acc}[\mathcal{F}_t] = \mathcal{F}_{acc}[\mathcal{F}_t] + d$;
13     end
14 end
15 $\mathcal{F}_{acc} = \frac{\mathcal{F}_{acc}}{max(\mathcal{F}_{acc})}$;
```

generated components, a smaller value is needed for the maximal thresholds. Another consideration to avoid the inclusion of the pixels outside of the road area as seed points for flood-fill operation, a single seed point is selected for each vehicle based on its moving direction. We consider the moving direction of the vehicle and always select one of the corner points of its surrounding bounding box that is certain to belong to the road area, thus avoiding the selection of non-road pixels as seed points.

After applying the edge detection method, leak segmentation error can still occur due to lack of enough gradient information at the dominant road boundaries, which can be corrected by using the accumulative foreground mask $\mathcal{F}_{acc}$. Algorithm 2 shows the steps of accumulating the foreground masks obtained by the GFM [125] method with false positives and slow-moving object filtered out by applying two thresholds τ_d and τ_s at steps 3–8. The threshold τ_i is used to define how long a track has to be inactive before being removed. The accumulative foreground mask $\mathcal{F}_{acc}$ is added by d in the location of the track only after track t has been removed from the set T (step 11 of Algorithm 2). This way, the tracks with larger movements contribute more to the estimated road region. At the end, $\mathcal{F}_{acc}$ is normalized as it is divided by the maximum value.

The contours of $\mathcal{F}_{acc}$ are smoothed using a Gaussian kernel. The Gaussian coefficients are calculated as follows:

$$\begin{aligned} \sigma' &= \frac{1}{2}\left(c\sigma + 1\right) \\ \mathcal{M} &= 2\left(sgn(\sigma')\lfloor|\sigma'| + 0.5\rfloor\right) - 1 \\ g_i &= \alpha exp\left(\frac{-\left(i - \frac{\mathcal{M}-1}{2}\right)^2}{2\sigma^2}\right) \quad , \sum_{i=0}^{\mathcal{M}-1} g_i = 1 \end{aligned} \tag{5.7}$$

where c is an integer constant, $\mathcal{M} \in \{2n+1 : n \in \mathbb{Z}\}$ is the Gaussian aperture size, σ is the standard deviation, α is the scale factor chosen so that $\sum_{i=0}^{\mathcal{M}-1} g_i = 1$, and g_i is the i-th Gaussian filter coefficient.

The contours are smoothed separately over each X and Y axis:

$$\begin{aligned} C_j^k(n) &= \begin{cases} C_j\left(|C| + n - k\right) & \text{,if } n < k \\ C_j\left(n - k - |C|\right) & \text{,if } n > (k + |C| - 1) \\ C_j\left(n - k\right) & \text{,otherwise} \end{cases} \\ C_j^*(n) &= \sum_{i=0}^{\mathcal{M}-1} C_j^k(n) g_i \quad , k = -\mathcal{L}...\mathcal{L} \end{aligned} \tag{5.8}$$

where $n = 0...\left(|C| - 1\right)$ is the index of each point on the curve,C is the surrounding contour of the accumulative foreground mask,$j \in \{x, y\}$ represents the x or y axis,$\mathcal{L} = \frac{1}{2}\left(\mathcal{M} - 1\right)$,and $C_j^*(n)$ is the position of the n-th point in the smoothed contour.

The sides of the smoothed contours, which correspond to the boundaries of the road, are partitioned into a set of K separate clusters $\mathcal{C} = \{c_k\}_{k=1}^K$ based on their connectivity, which is in turn measured by Euclidean distance. The points of each cluster c_k are resampled by traversing in a pace equal to resample size $m_k = s_k/d$ where s_k is the arc-length of c_k and d is a pre-defined constant.

Then a simple approach is used to estimate the boundaries of the road by fitting a second-degree polynomial curve to each cluster. The l component analysis (PCA) method is applied to each set of re-sampled points in order to calculate the direction of the maximum variation in the set. First a matrix $\mathbf{P}_k \in \mathbb{N}^{m_k \times 2}$ is formed with each row containing the x, y coordinate values of each resampled point from c_k. Then the covariance matrix $\mathbf{S}_k$ is computed as follows:

$$\begin{aligned} \mathbf{u}_k &= \frac{1}{m_k}\sum_{i=1}^{m_k}\mathbf{P}_k \\ \mathbf{S}_k &= \frac{1}{m_k - 1}\sum_{i=1}^{m_k}(\mathbf{P}_k - \mathbf{u}_k)(\mathbf{P}_k - \mathbf{u}_k)^T \end{aligned} \tag{5.9}$$

where $\mathbf{u}_k$ is a row vector that contains the mean $\bar{x}$ and $\bar{y}$ values of each column in $\mathbf{P}_k$. The eigenvalues and eigenvectors of the covariance matrix are calculated as follows:

$$\begin{aligned} \lambda_1^k, \lambda_2^k &= \frac{1}{2}\left(\sigma_{x_k}^2 + \sigma_{y_k}^2 \pm \sqrt{\left(\sigma_{x_k}^2 - \sigma_{y_k}^2\right)^2 + 4\sigma_{x_k y_k}^2}\right) \\ \mathbf{e}_j^k &= \frac{1}{\sqrt{\sigma_{x_k y_k}^2 + \left(\lambda_j - \sigma_{x_k}^2\right)^2}}\begin{bmatrix} \sigma_{x_k y_k}^2 \\ \lambda_j - \sigma_{x_k}^2 \end{bmatrix} \end{aligned} \tag{5.10}$$

where $j \in \{1, 2\}$, $\sigma_{x_k}^2$, $\sigma_{y_k}^2$, and $\sigma_{x_k y_k}^2$ are the variance of x, variance of y, and covariance of xy values in $\mathbf{P}_k$, respectively. λ_j^k and $\mathbf{e}_j^k$ are the eigenvalues and their corresponding eigenvectors of $\mathbf{S}_k$. A matrix $\mathbf{E}_k$ is defined as follows:

$$\mathbf{E}_k = \begin{bmatrix} a_{11}^k & a_{12}^k \\ a_{21}^k & a_{22}^k \end{bmatrix} \tag{5.11}$$

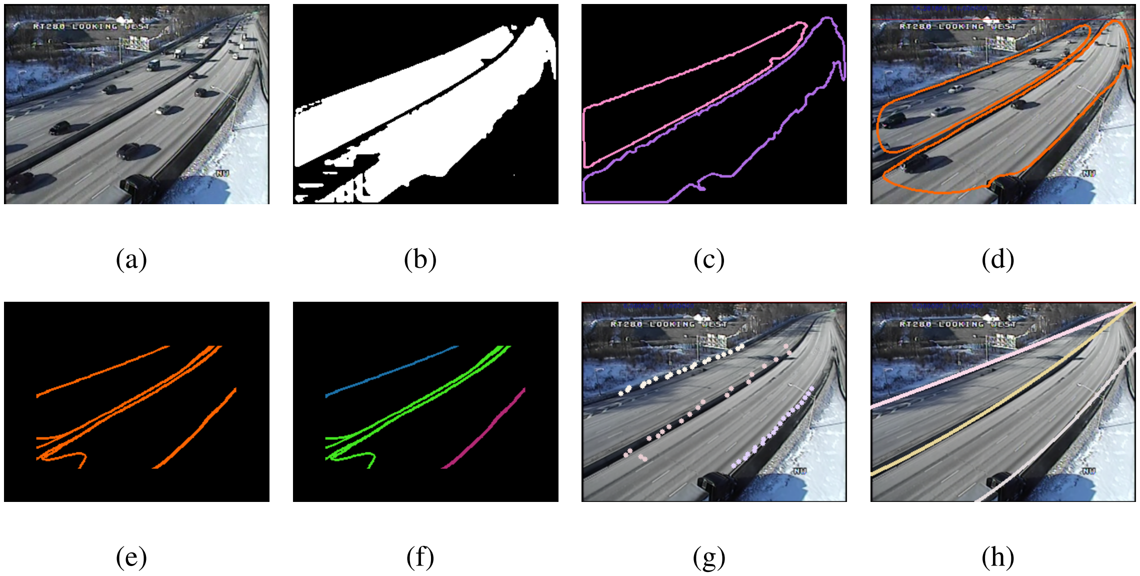

(a) (b) (c) (d)
(e) (f) (g) (h)

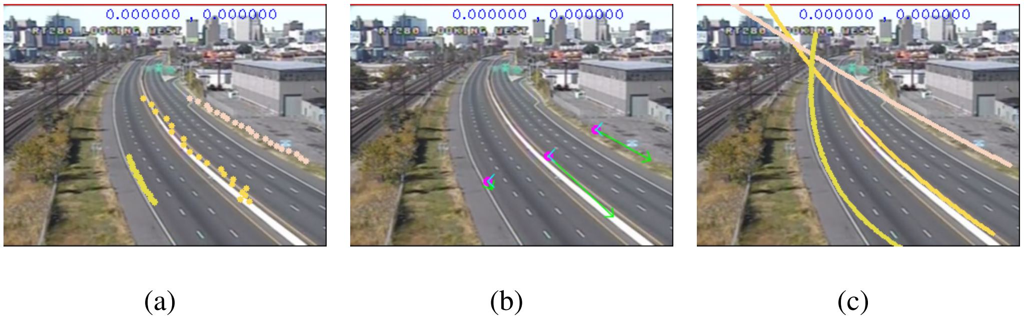
Figure 5.5 The estimation process of the road boundaries. (a) Sample traffic video frame. (b) Accumulative foreground mask after one minute. (c) Contours of the accumulative foreground masks. (d) Smoothed contours. (e) Cropped contours. (f) Clustering. (g) Resampled points. (h) The estimated road boundaries.

where $\mathbf{e}_1^k = \left[a_{11}^k, a_{21}^k\right]^T$ and $\mathbf{e}_2^k = \left[a_{12}^k, a_{22}^k\right]^T$ are the first and second eigenvectors of $\mathbf{P}_k$, respectively. A new axis is generated and the data points from $\mathbf{P}_k$ are rotated as follows:

$$\begin{gathered}
\theta_k = cos^{-1}\left(tr(\mathbf{E}_k)/2\right) \\
\mathbf{R}_k = \begin{bmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{bmatrix} \\
\mathbf{P'}_k^T = \mathbf{R}_k \mathbf{P}_k^T
\end{gathered} \tag{5.12}$$

where θ_k is the direction of maximum dispersion in $\mathbf{P}_k$, $tr(\mathbf{E}_k) = a_{11}^k + a_{22}^k$, $\mathbf{R}_k$ is the rotation matrix, and $\mathbf{P'}_k$ is the matrix containing the rotated points. After second-degree polynomial curve-fitting on each $\mathbf{P'}_k$, the resulting curves are rotated back to the original x and y axis to represent an estimation of the dominant road boundaries. Figure 5.5 and Figure 5.6 represent an example of road boundary estimation.

Figure 5.7 presents examples of the flood-fill algorithm applied on traffic videos in a period of one minute.



(a) (b) (c)

Figure 5.6 Extracting the dominant road boundaries using the PCA method. (a) Resampled points used for curve fitting. (b) The direction of the maximum variation recognized by PCA. (c) The limiting boundaries estimated by curve fitting.

5.2.3 Updating and merging the extracted probability maps

The extracted probability maps are updated in order to take into account the gathered information from all observed frames. As more vehicles pass along different locations of the roadway, the number of pixels in the Ω grows, which makes the probability maps of the latest frames more reliable than the initial values. Also, when a pixel repeatedly appears in the foreground mask of the moving vehicles, it is more likely to belong to the road region. Therefore, all probability maps are updated by applying the temporal fusing algorithm at each frame as follows:

$$
\begin{aligned}
P_K^t(p_i) &= \frac{\sum_{f=1}^{t} w_i^f \times P_K^f(p_i)}{1 + \sum_{f=1}^{t} w_i^f} \\
w_i^f &= \sum_{j=1}^{N} \Omega_M^f(p_j)
\end{aligned}
\tag{5.13}
$$

where $i = 1...N$ is the pixel index, w_i^f is the weight associated with pixel p_i at frame f, $K \in \{G, C, H, S, Ghist, GBhist, F\}$ refers to the source of each probability map, $P_K^f(p_i)$ is the probability value of pixel p_i at frame f, $M \in \{rsm, fsm\}$ is the source of the sample mask containing the initial seed points, $\Omega_M^f(p_i) \in \{0, 1\}$ is the value of p_i in the accumulative road sample mask of frame f, N is the total number of pixels in each frame, and $P_K^t(p_i)$ is the updated probability value of pixel p_i.

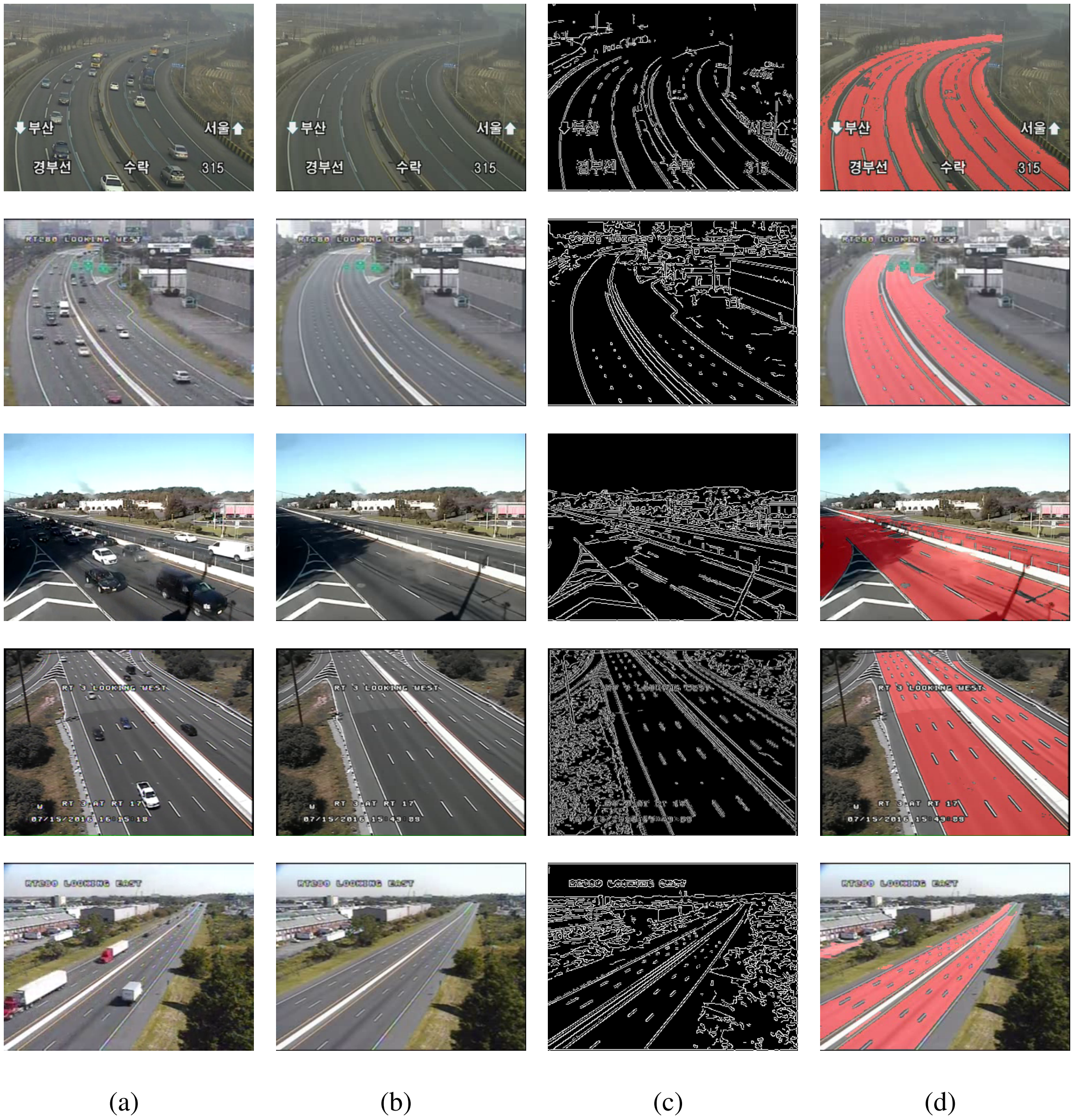


Figure 5.7 Extracting the road region using the cumulative maps of the flood-fill method. (a) Original video frame. (b) The background obtained by the GFM method. (c) The edges of the background image. (d) The retrieved road map.

The updated probability values for each pixel extracted from different sources should be combined with each other in order to obtain a consensus estimation. If we denote the set of all pixels with $\mathcal{N}$ and the set of extracted probability maps with $\mathcal{K}$, the event R_i specifying whether a pixel $i \in \mathcal{N}$ belongs to the road region or not, can be considered as a Bernoulli random variable $Ber(q_i)$ where $q_i \in [0, 1].R_i = 1$ means i belongs to the road region and

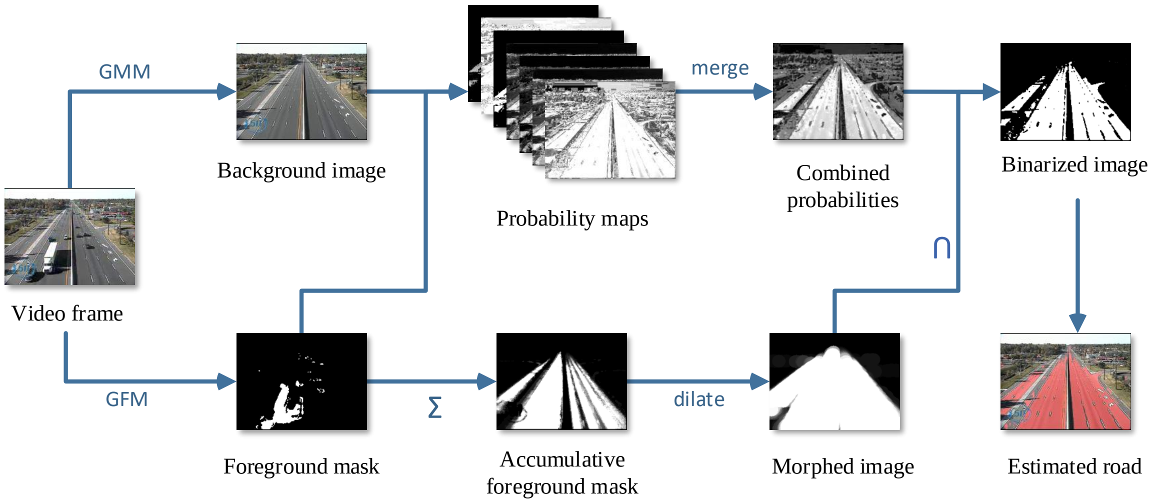


Figure 5.8 The process of merging and refining the probability maps. The extracted probability maps are combined and the Otsu's threshold is applied on the result. The non-road pixels that are misclassified as a part of the road region due to similar color values are later filtered out by intersecting the binary image with the accumulative foreground mask.

$R_i = 0$ means i is a non-road pixel. The set of generated probability maps, $\mathcal{K}$, contains several estimations, each of which is drawn from a different source of information. We denote the probability prediction of source j made on pixel i with $p_{i,j} \in [0, 1]$. To solve a probability aggregation problem, we need to design a function $F : ([0, 1])^{|\mathcal{N}|\times|\mathcal{K}|} \rightarrow [0, 1]^{|\mathcal{N}|}$ that takes the predicted probabilities $\{p_{i,j}\}_{i\in\mathcal{N},j\in\mathcal{K}}$ as input and generates an aggregated probability estimation $\hat{q}_i \in [0, 1]$ for each pixel i.

Some simple approaches to aggregate probability predictions are the arithmetic mean of the probabilities, the median of the probabilities, majority voting, the logarithmic opinion pool, and the Beta-transformed linear opinion pool. Here, we use weighted mean and median in order to solve the aggregation problem by considering the different degrees of reliability among the generated probability maps and also, taking into account that the aggregated estimation should tend towards the majority opinion in extreme cases of probability predictions.The values of each pixel i in the set $\mathcal{K}$are sorted and the resulting ordered list $\mathcal{K}' = \{P'_1, ..., P'_m\}$ is utilized to define the weighted median $p'_{i,k}$ such that:

$$\sum_{j=1}^{k-1} w_j \leq 1/2 \quad and \quad \sum_{j=k}^{|\mathcal{K}'|} w_j \leq 1/2 \tag{5.14}$$

where $j = 1...\mathcal{K}$ is the index of the probability maps and w_j is the weight for each map representing its reliability. Experimental results have shown higher stability of the P_F and P_S probability maps and higher weights are assigned to these source in the aggregation process.

If the values of a pixel in the set of extracted probability maps $\mathcal{K} = \{P_1, ..., P_m\}$ have a large median, it means that the pixel has a high value in most probability maps and, therefore, is most likely inside the road region. On the other hand, low median means most predictions contain a low value for a pixel and it most likely belongs to the non-road area. The aggregated probability values are calculated as follows:

$$\hat{q}_i = \begin{cases} \frac{1}{(m-k+1)} \sum_{j=k}^{m} p'_{i,j} & \text{, if } p'_{i,k} > \theta_1 \\ \frac{1}{k} \sum_{j=1}^{k} p'_{i,j} & \text{, if } p'_{i,k} < (1-\theta_1) \\ \frac{1}{2}(p'_{i,k} + \frac{\sum_{j\in\mathcal{K}} w_j p_{i,j}}{\sum_{j\in\mathcal{K}} w_j}) & \text{, otherwise} \end{cases} \tag{5.15}$$

where $i \in \mathcal{N}$ is a pixel, $p'_{i,j}$ is the probability value of pixel i in the sorted probability set $\mathcal{K}' = \{p'_{i,j}\}_{i\in\mathcal{N}, j\in\mathcal{K}'}$, k is the index of the weighted median value $p'_{i,k}$, θ is a pre-defined threshold close to 1, and $\hat{q}_i$ is the aggregated probability value for pixel p_i. The Otsu's threshold [49] is applied to the resulting map in order to filter out the regions with low probability value.

When the intersection between the binary probability mask and the aggregated foreground mask surpasses a threshold, the cumulative foreground mask has covered most of the road pixels after morphological dilation with a size close to the average size of vehicles. The morphological procedure is performed on $\mathcal{M}_{\mathcal{F}}$ to bridge the gaps and the intersection between its result and $\mathcal{P}^*_{\mathcal{R}}$ is utilized as the final estimated road region as follows:

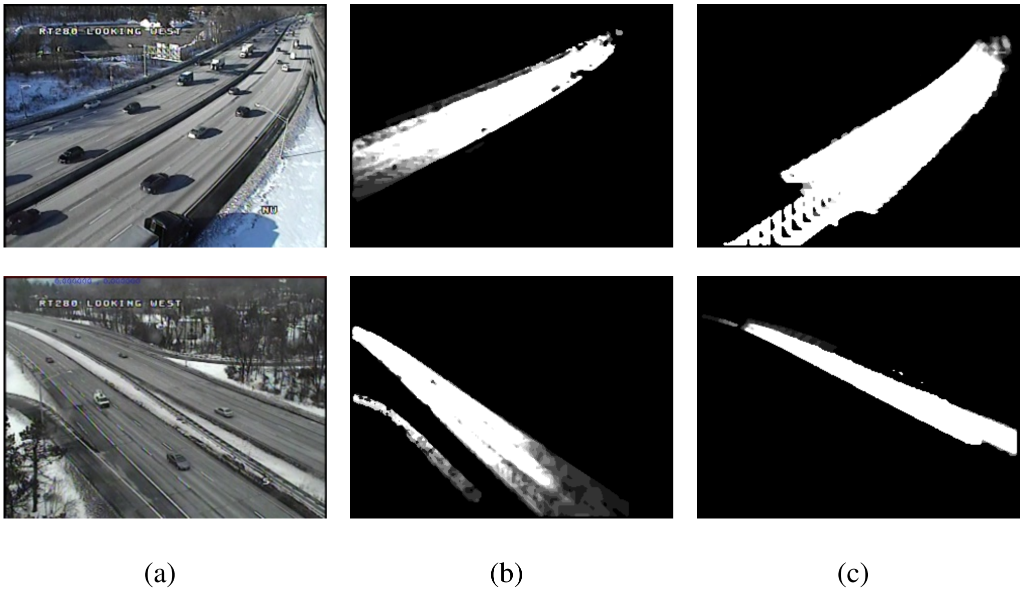
$$\begin{aligned}
\mathcal{M}'_{\mathcal{F}} &= \mathcal{M}_{\mathcal{F}} \oplus B \\
\mathcal{T} &= \frac{|\mathcal{M}'_{\mathcal{F}} \cap \mathcal{P}^*_{\mathcal{R}}|}{|\mathcal{P}^*_{\mathcal{R}}|} \\
\mathcal{M}_{\mathcal{R}} &= \begin{cases} \mathcal{P}^*_{\mathcal{R}} & \text{, if } \mathcal{T} < \theta \\ \mathcal{M}'_{\mathcal{F}} \cap \mathcal{P}^*_{\mathcal{R}} & \text{, otherwise} \end{cases}
\end{aligned} \tag{5.16}$$

where $\mathcal{M}'_{\mathcal{F}} = \{x|[(\hat{B})_x \cap \mathcal{M}_{\mathcal{F}}] \neq \varnothing\}$ is the result of a dilation operation with B as a structuring element,$\mathcal{T}$ is the number of common pixels between the probability mask and the accumulative flood-fill mask,$\theta \in [0, 1]$ is a predefined threshold,and M_R is the final mask representing road pixels.

As illustrated in Figure 5.19, the intersection between the cumulative foreground mask and the binary fused probabilitymask is utilized as the final estimated road region. This way, the possible misclassified non-road regions are removed, and the final road map is refined by the exclusion of the over segmentation and leak segmentation errors.

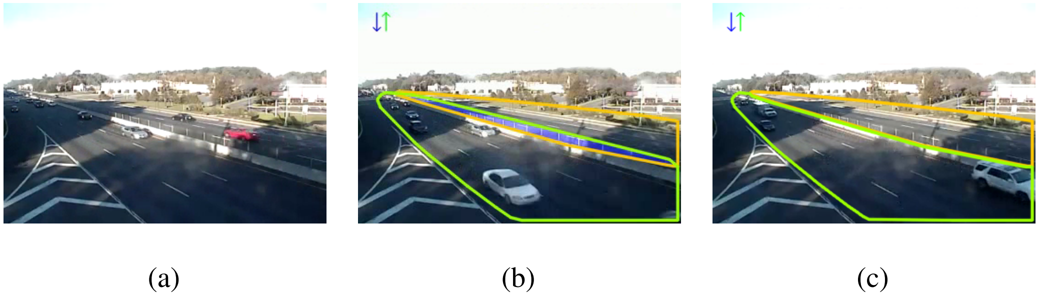
5.2.4 A novel statistical method for separating major traffic directions

Most roads and highways carry traffic in two opposite directions. In the case of most traffic video analytic tasks, a separate ROI is needed for each side of the road. In order to retrieve an ROI for each side of the road, the tracking information obtained from the blob-tracking approach is used to detect the moving direction of each vehicle. The centroid of each track at the starting and ending positions is compared to estimate the direction of its movement. To avoid the effects of noises in the foreground and noisy results of the tracking method, only vehicles with high enough movement size and speed are considered. Each time such a vehicle passes along the road, the pixels with a corresponding location in its foreground mask are added with a positive number in the road map of the correct direction and added with a negative number in the road map in the opposite direction. To avoid having common areas between left and right regions, we try to remove the foreground mask of a vehicle



(a) (b) (c)

Figure 5.9 Separated accumulative foreground masks of the moving vehicles. (a) Original traffic video frame. (b) and (c) are the accumulative foreground masks of the left and right sides, respectively.



(a) (b) (c)

Figure 5.10 Assigning the overlapping area between the maps of the two traffic direction to the correct side. (a) The original traffic video frame. (b) The blue color indicates the overlapping area between two ROIs. (c) The overlapping area is assigned to the correct ROI and removed from the other ROI.

from the opposite side when it is being added to one side, in case it has previously been added to the opposite side by mistake.

For each tracked vehicle that passes along the road, the left and right sides of the road are updated as follows:

$$m = max(0, \alpha \sum_{f=1}^{T} m_v - \beta(m_o + \sum_{f=1}^{T} m_{vo})) \tag{5.17}$$

where m is the traffic region map for one side of the road, m_v is the foreground mask of the vehicle passing along that side at frame f, m_{vo} is the foreground mask of the vehicle passing along the opposite side at frame f, m_o is the traffic region map of the opposite side of the road, T is the current frame, and α and β are predefined coefficients between 0 and 1. In order to speed up the update process of the traffic region maps, α and β should be closer to 1, and in order to reduce the update errors, they should be closer to 0. Each road map is then updated by applying Otsu's threshold:

$$m_f = \begin{cases} 1, & \text{if } m_{acc} \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{5.18}$$

where m_f is the final traffic region binary map for each side, m_acc is the accumulative foreground masks in that side, f is the current frame, τ is the calculated Otsu's threshold, m_f is the foreground mask of frame f, and F is the total number of frames. The Otsu's threshold is applied to remove noises that are mostly caused by occasional noises in the foreground mask. Figure 5.9 shows examples of the separated accumulative foreground masks for the two major directions of the traffic flow.

In order to obtain an ROI for each side of the road that contains the road itself and a good portion of its surroundings, the convex hull of the road map's contour is used for each side. The two convex hulls corresponding to the contours obtained from the road map of each side of the road have proven to be good representations for the ROI, as they involve the entire road and its surroundings while avoiding the regions outside of the road and therefore save the video analytic applications from unnecessary noise and computational overload. However, in videos where the camera angle is from one side of the road, the foreground masks of the vehicles from different sides can overlap each other, which in turn causes an intersecting area between the convex hulls of the two sides in the middle part of the road. The overlapping area should be removed from the ROI of the wrong side to avoid false positive results in further video analysis tasks. In order to decide which side of the

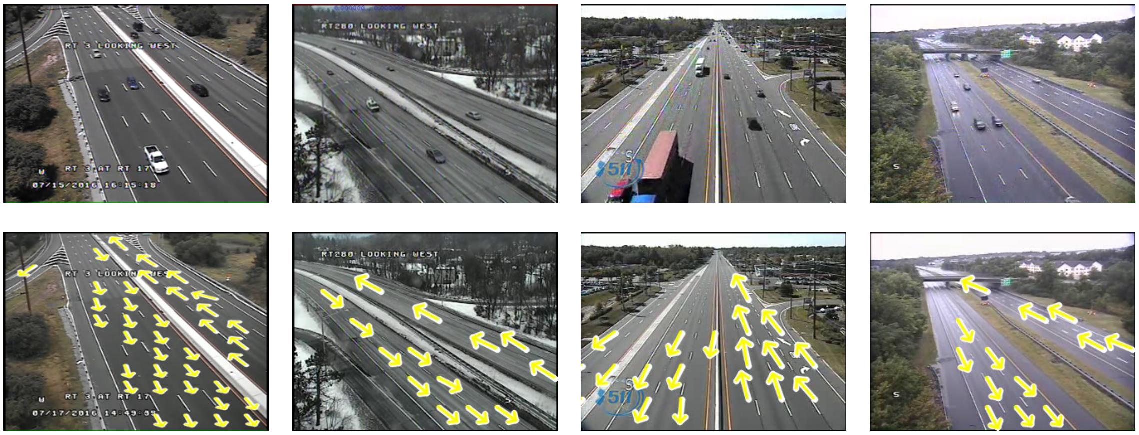
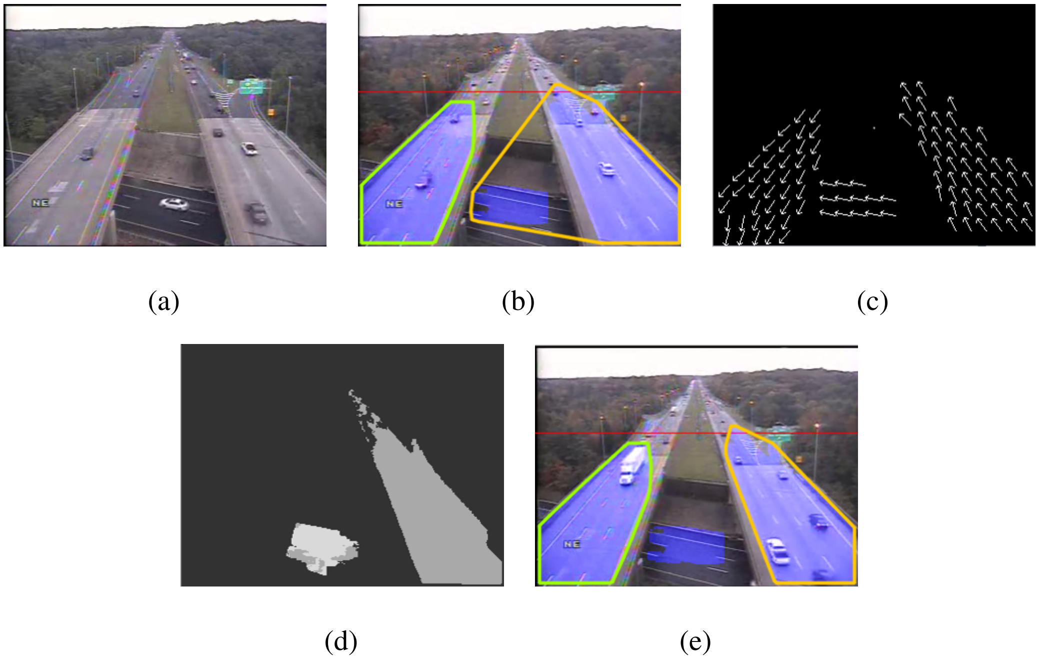


Figure 5.11 Extracting a matrix of motion flow vectors using GMM method with optical flow vectors as input. First row contains sample frames of traffic videos. Second row represents the corresponding flow model matrix obtained from the GMM method.

road the overlapping area belongs to, the intersection between the overlapping area and the convex hull of each side is calculated, and the overlapping area is removed from the ROI of the side with the lower intersection. Figure 5.10 shows the overlapping area removed by our proposed method.

In some videos, the traffic flows in more than two directions, and further steps are required to be taken in order to extract only the regions corresponding to the major directions and exclude others. In this case, using the direction obtained from tracking is not enough to separate the regions with similar directions but different road segments. Here, we have applied a statistical method based on Gaussian Mixture Models (GMM) in order to estimate the general moving velocity of the vehicles at various locations on the road. At each frame, the Lucas-Kanade optical flow method [7] is applied to obtain a matrix of flow vectors in the size of the entire frame. The Lucas-Kanade optical flow method has incorrect outputs, especially in video with low resolution, and the results of a few frames are not reliable for estimating the motion vectors. To overcome this problem, the non-zero magnitude and speed of the optical flow vectors in a sequence of frames are utilized as two-dimensional input vectors by the GMM method in order to estimate the most probable velocity at each pixel.


(a) (b) (c)

(d) (e)

Figure 5.12 Excluding smaller road regions with similar direction to one of the major traffic regions. (a) The original traffic video frame. (b) The road under the bridge is incorrectly grouped with one of the major traffic regions. (c) The flow vectors obtained by the GMM method. (d) Applying K-means clustering method to separate the small region with a similar direction. (e) The small region is excluded from the ROI.

The Gaussian modeling of the optical flow vectors is described as follows:

$$P(\mathbf{x}) = \sum_{k=1}^{K} W_k N(\mathbf{x}|\omega_k) \tag{5.19}$$

$$N(\mathbf{x}|\omega_k) = \frac{\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_k)^t \Sigma_k^{-1}(\mathbf{x}-\mu_k)\right\}}{(2\pi)^{d/2} \mid \Sigma_k \mid^{1/2}} \tag{5.20}$$

$$\sum_{k=1}^{K} W_k = 1 \tag{5.21}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the two-dimensional feature vector containing flow angle and magnitude of each pixel, K is the number of Gaussian distributions in the flow model, W_k is the weight of the k_{th} Gaussian distribution $N(\mathbf{x}|\omega_k).\mu_k$ and Σ_k are the mean vector and the covariance matrix of the k_{th} Gaussian density $N(\mathbf{x}|\omega_k)$. Note that the Gaussian model of each pixel is updated only when the magnitude of the optical flow is greater than zero. The results of the GMM are further refined by removing incorrect estimations based on the general direction of each tracked vehicle. Figure 5.11 shows examples of the optical flow vectors modeled by the GMM method.

After generating the flow matrix, the K-means clustering approach is applied in order to group pixels with vectors of close angles together, thus excluding the regions that are not part of the two major traffic directions. Figure 5.12 shows an example of how the smaller region falsely included in one of the two major traffic regions is separated and removed from the ROI.

5.2.5 Single image based road detection based on illumination invariant image

In this section, we discuss automatic road region extraction in traffic images that aids with ROI determination, which can be useful in the automated detection of obstacles, traffic incidents, and driving violations. We propose an adaptive road recognition method that extracts the road location from single frames. No assumption about the structure of the road is made, and therefore, this method can be used for structured and unstructured road scenarios. A triangular region in front of the vehicle is assumed to belong to the road region and is utilized as the initial road sample. Initially, an illumination-invariant gray-scale image is extracted from the RGB image in order to weaken the effects of shadows that decrease the segmentation performance. Afterward, the boundaries of the road and the horizontal line are estimated. in order to limit the road map from the previous step and avoid possible leak-segmentation errors. Finally, the Chan-Vese segmentation algorithm is applied to the illumination invariant image in order to segment the road region.

Generating the illumination invariant image The shadows cast on the objects in an image captured by a regular camera have negative effects on most computer vision tasks such as segmentation and object detection, especially in outdoor scenes. Therefore, eliminating or weakening the effects of illumination and shadows as a preprocessing step can improve the performance of vision tasks. One of the main methods for weakening the effects of shadows is to derive a one-dimensional illumination invariant image from the three-channel color image based on the relations between the three color values. Assuming a Planckian light source and Lambertian surface for the objects in the natural environment, we can denote the spectral power distribution (SPD) of the light with $E(\lambda, x, y)$ which is incident on a surface with reflectance $S(\lambda, x, y)$. Then the response of the camera sensor is as follows:

$$\rho_k(x, y) = \sigma(x, y) \int E(\lambda, x, y) S(\lambda, x, y) Q_k(\lambda) d\lambda \tag{5.22}$$

Where $k \in 1, 2, 3$, $\sigma(x, y)$ is a constant equal to the dot product of the illumination direction and the surface normal at location (x, y) and $Q_k\left(\lambda\right)$ is the sensitivity of the k-th camera sensor. If we drop the indices for the locations and assume the camera sensors are based on Dirac delta functions $Q_k\left(\lambda_k\right) = q_k \delta(\lambda - \lambda_k)$, we would have:

$$\rho_k = \sigma E\left(\lambda_k\right) S(\lambda_k) q_k \tag{5.23}$$

If the illumination is modeled by Wien's approximation to Planck's law, the SPD can be demonstrated by its color temperature as follows:

$$E\left(\lambda, T\right) = I c_1 \lambda^{-5} e^{-\frac{c_2}{T\lambda}} \tag{5.24}$$

With c_1 and c_2 being constants I being the overall intensity of the light. Therefore, the response of each camera sensor to can be expressed as follows:

$$\rho_k = \sigma\ I c_1 \lambda^{-5} e^{-\frac{c_2}{T\lambda}} S(\lambda_k) q_k \tag{5.25}$$

If we calculate the ratio chromaticities using the color channels, we would have:

$$\chi = \begin{bmatrix} \chi_1 \\ \chi_2 \end{bmatrix} = \begin{bmatrix} R/G \\ B/G \end{bmatrix} = \begin{bmatrix} \left({\lambda_R}^{-5} e^{-\frac{c_2}{T\lambda_R}} S(\lambda_R) q_R\right) / \left({\lambda_G}^{-5} e^{-\frac{c_2}{T\lambda_G}} S(\lambda_G) q_G\right) \\ \left({\lambda_B}^{-5} e^{-\frac{c_2}{T\lambda_B}} S(\lambda_B) q_B\right) / \left({\lambda_G}^{-5} e^{-\frac{c_2}{T\lambda_G}} S(\lambda_G) q_G\right) \end{bmatrix} \tag{5.26}$$

In logarithmic space, we would have:

$$\chi' = \begin{bmatrix} \log \chi_1 \\ \log \chi_2 \end{bmatrix} = \begin{bmatrix} \log \left[\left({\lambda_R}^{-5} S(\lambda_R) q_R\right) / \left({\lambda_G}^{-5} S(\lambda_G) q_G\right) \right] + T^{-1} c_2 \left(\frac{1}{\lambda_G} - \frac{1}{\lambda_R} \right) \\ \log \left[\left({\lambda_R}^{-5} S(\lambda_R) q_R\right) / \left({\lambda_G}^{-5} S(\lambda_G) q_G\right) \right] + T^{-1} c_2 \left(\frac{1}{\lambda_G} - \frac{1}{\lambda_R} \right) \end{bmatrix} = s + eT^{-1} \tag{5.27}$$

Which indicates that by varying the illumination (T) the vector χ' moves along a straight line in the log-chromaticity space for each surface. Therefore, by determining the direction of vector e, we can specify the changes in illumination which is only camera-dependent and by projecting the vector χ' onto the vector $e^{\perp}$ orthogonal to e, a one-dimensional grayscale image is generated as follows:

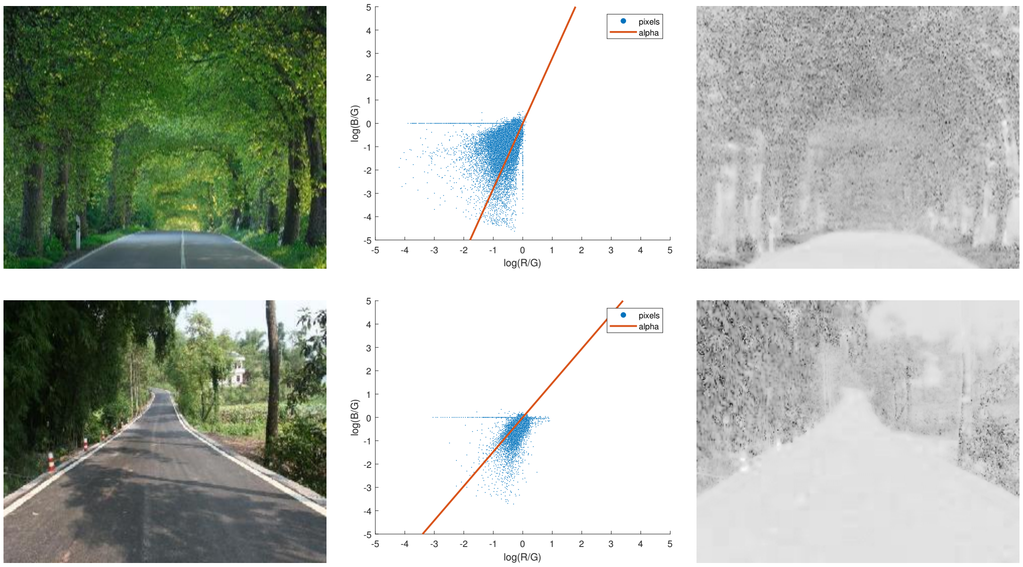
$$G_{inv} = \exp\left({\chi'}^{t} e^{\perp} \right) \tag{5.28}$$

Where the effect of the illumination is weakened.

Here, if we represent the triangular area containing the road samples as Ω, for each road image I the RGB values of pixel $p_i \in I$ where $i \in 1, \ldots, N$ are indicated by $(R\,(p_i)\,, G\,(p_i)\,, B\,(p_i))$ and the corresponding intrinsic image is calculated as follows:

$$\begin{aligned} x_i &= \log\left({}^{R\,(p_i)}/_{G\,(p_i)}\right) - \log\left({}^{R\,(\overline{p_{\Omega}})}/_{G\,(p_{\Omega})}\right) \\ y_i &= \log\left({}^{B\,(p_i)}/_{G\,(p_i)}\right) - \log\left({}^{R\,(\overline{p_{\Omega}})}/_{G\,(p_{\Omega})}\right) \\ G_{inv} &= x_i \cos\alpha + y_i \sin\alpha \end{aligned} \tag{5.29}$$

In order to calculate the angle α, we have used the median of four different values based on moment 1, linear regression, moment 3, and principal component analysis (PCA)



(a) (b) (c)

Figure 5.13 Transformation from RGB to 1D intrinsic image. (a) The original road image. (b) Log-Chromaticity space. (c) The intrinsic image.

to estimate a more accurate and general value as follows:

$$
\begin{gathered}
\alpha_1 = \tan^{-1}\left(sign\left(\sigma_{XY}^2\right)\right)\frac{\sum_i |x_i|}{\sum_i |y_i|} \\
\alpha_2 = \frac{1}{2}\left(\tan^{-1}\left(\frac{\sigma_X^2}{\sigma_{XY}^2}\right) + \tan^{-1}\left(\frac{\sigma_Y^2}{\sigma_{XY}^2}\right)\right) \\
\alpha_3 = \tan^{-1}\left(\frac{\sqrt[3]{\frac{1}{N}\sum_i y_i^3}}{\sqrt[3]{\frac{1}{N}\sum_i x_i^3}}\right) = \tan^{-1}\left(\sqrt[3]{\frac{\sum_i y_i^3}{\sum_i x_i^3}}\right) \\
\alpha_4 = \tan^{-1}\left(\frac{\vec{e_y}}{\vec{e_x}}\right) \\
\alpha_{med} = med\left\{\alpha_i\right\}_{i=1}^{4}
\end{gathered}
\tag{5.30}
$$

where $\vec{e}$ is the first principal component. In Figure 5.13, we can see an example of weakening the shadow effect by projecting the log ratio values onto an orthogonal vector to e.

Road boundaries extraction In order to extract the dominant boundaries of the road, first we need to weaken the shadow effects while preserving the gradient information corresponding to the material changes. Therefore, we cannot use the intrinsic image from the previous step since it also reduces the amount of gradient information at important edges. Here, we have used another shadow feature which is robust to strong shadows in order to reduce the illumination effects while intensifying the edges corresponding to material changes. This feature extraction method has less dependence on the camera settings and relies on the fact that road color values in the RGB space are close to each other. In most cases, the road surface has relatively similar values in red, green, and blue components, whereas the surrounding vegetation has one component considerably higher than the other. This fact can be used as a discriminating feature between shadow edges and the edges corresponding to material changes. By assuming the road to be a homogeneous dielectric surface, the values of pixels of the same material lie on a line passing the origin in the RGB space with a small offset. We can assume a vector for each pixel that belongs to the road surface in the RGB space.

The cause of shadows is the occlusion of sunlight by objects, and in the shadow areas, the road is illuminated by skylight. On the other hand, we know that in the outdoor scenes, the white light emitted from the sun is scattered in all directions by molecules in the air. The Rayleigh scattering effect is higher in the shorter wavelengths, such as blue, which causes the sky to appear bluish. Whereas light in the longer wavelengths, such as red, passes through the atmosphere with less scattering effect. Therefore, we can confidently state that the attenuation is non-proportional due to the ambient illumination, which is blue in this case. As the ambient light can have a Spectral Power Distribution (SPD) different from that of incident light, the decrease in luminance when a surface is under shadow is not proportional among the color channels. Here, we have used the HSV color-space in order to generate a grayscale image where the gradient information is stronger in edges corresponding to material changes in comparison to edges corresponding to illumination changes. The RGB image is converted to HSV color-space where the V component represents the maximum value among the red, green, and blue channels and the S component denotes the saturation

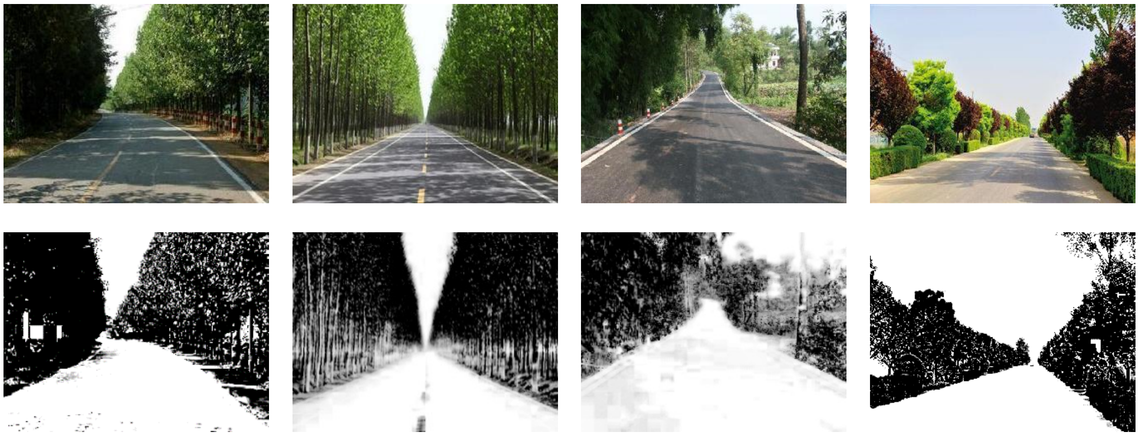


Figure 5.14 Examples of the proposed shadow feature extractor. First row contains some sample road images and the second row represents the corresponding results of the shadow weakening method.

and is calculated as follows:

$$S = \frac{max\left(R, G, B\right) - min\left(R, G, B\right)}{max\left(R, G, B\right)} \tag{5.31}$$

Taking into account that the road surfaces have similar values among the three components, we introduce a feature to weaken the shadow effects while preserving the discriminating properties of material changes as follows:

$$F \triangleq 2 - \frac{V + b}{B + \varepsilon} \tag{5.32}$$

where F represents a feature matrix with the same size as the image, b is a bias that is dependent on the camera sensors and can be estimated by polynomial fitting, and ε is a small positive constant. Some examples of the extracted feature can be seen in Figure 5.14.

After weakening the illumination effects, we can extract the candidate road boundaries by using the global thresholding method and only filtering the bottom section of each image, followed by a morphological operation and connected component analysis to remove the small blobs that are considered to be noise. A middle-to-side operation is performed in order to extract the pixels corresponding to the road boundaries. A bottom-up scan is applied

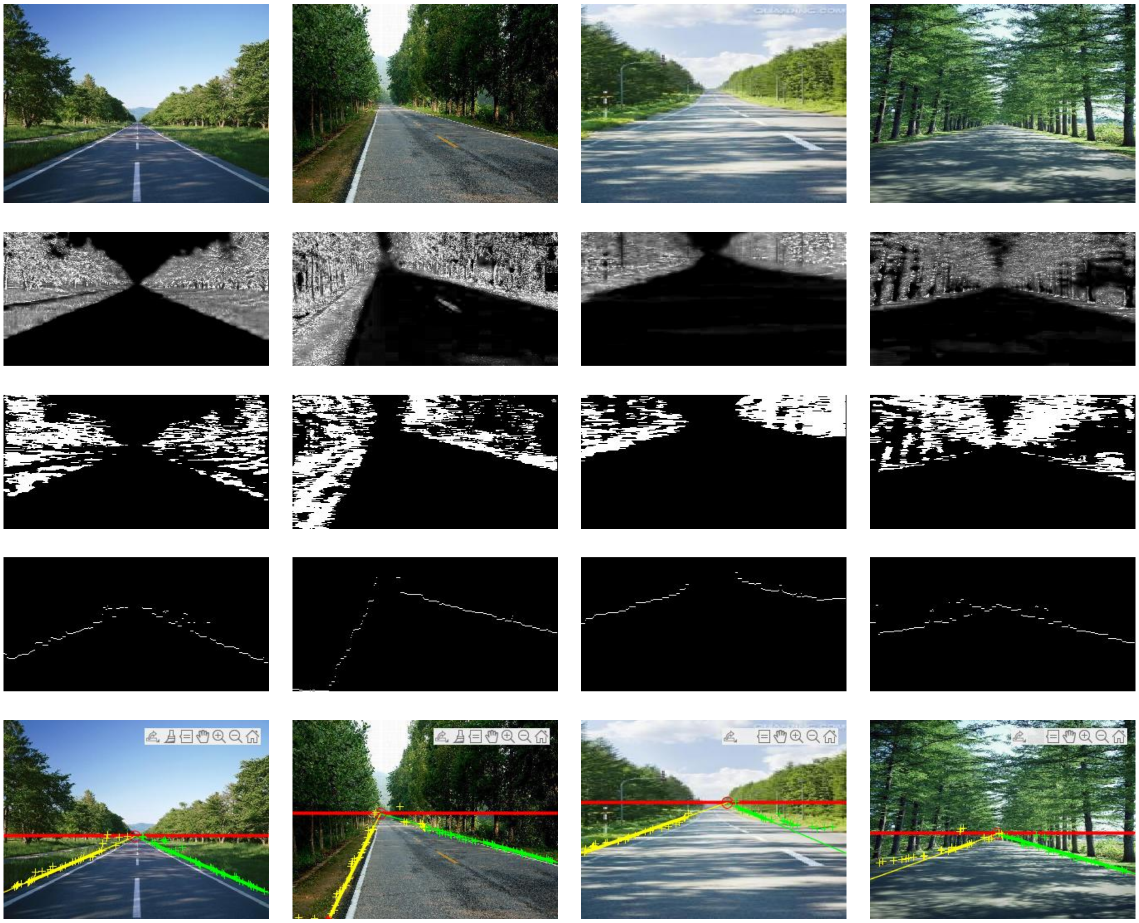


Figure 5.15 An example of road boundary detection. First row contains some sample road images and the second row represents the corresponding results of the boundary detection and horizontal line estimation method indicated by a red line.

to each column, and the first non-zero pixels are marked as boundary candidates. Then a middle-to-left and right-to-left scan is performed on the candidate pixels to extract the left and right boundaries, respectively. Finally, the Hough transformation is applied to fit a straight line on each boundary, and the vanishing point is defined as the intersection of the two lines, which is assumed to be located on the horizontal line. These boundaries are later used in order to limit the segmentation process and reduce the amount of leakage-segmentation errors. An example of road boundary detection is shown in Figure 5.15.

Road region segmentation In order to segment the road region, the Chan-Vese segmentation algorithm is applied. This model for active contours is more robust than traditional segmentation methods such as thresholding or gradient-based methods. The Chan-Vese model is based on the Mumford-Shah function and is mostly used in medical images. In the case of road segmentation, we have already extracted the road boundaries, and a region-growing strategy such as active contour seems to be a choice. The neighboring pixels of a set of initial seed points are examined and decided whether to be added to the region or not in an iterative manner. In the Chan-Vese model, the goal is to minimize the energy defined as the weighted values corresponding to the sum of intensity variations among the pixels inside and outside of the currently segmented region and a term indicating the arc length of the region's boundary. Here, we first apply the segmentation method to the illumination invariant image extracted in the first step. Then the extracted region is filtered by removing the possible pixels segmented as part of the road region as a leak-segmentation error. This error is usually caused by the similarities between the sky and road chromaticity proportions. Some examples of the iterative region-growing method are illustrated in Figure 5.16. The initial seed points are chosen from the triangular area assumed to belong to the road region, and the region is iteratively grown until the boundaries of the road are covered. After this step, the leak-segmentation errors must be removed by using the horizontal line calculated in the previous step.

The similarities between the sky and the road region in terms of chromaticity and color component proportions tend to cause leak-segmentation errors in some images. In order to deal with these types of errors, we can apply to limit boundaries the horizontal line estimated in the previous steps. The resulting mask of the segmentation step is intersected with the resulting binary mask of the road boundary detection step in order to remove areas that are outside of the boundaries (such as the sky) while preserving the exact boundary points of the road instead of the straight lines.

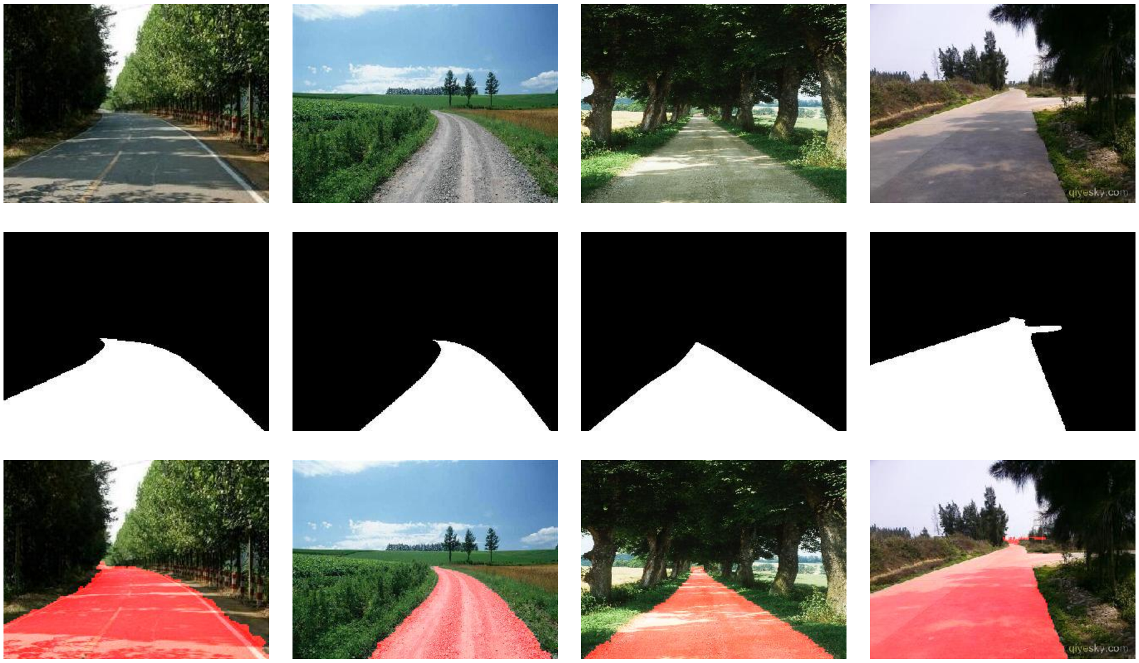
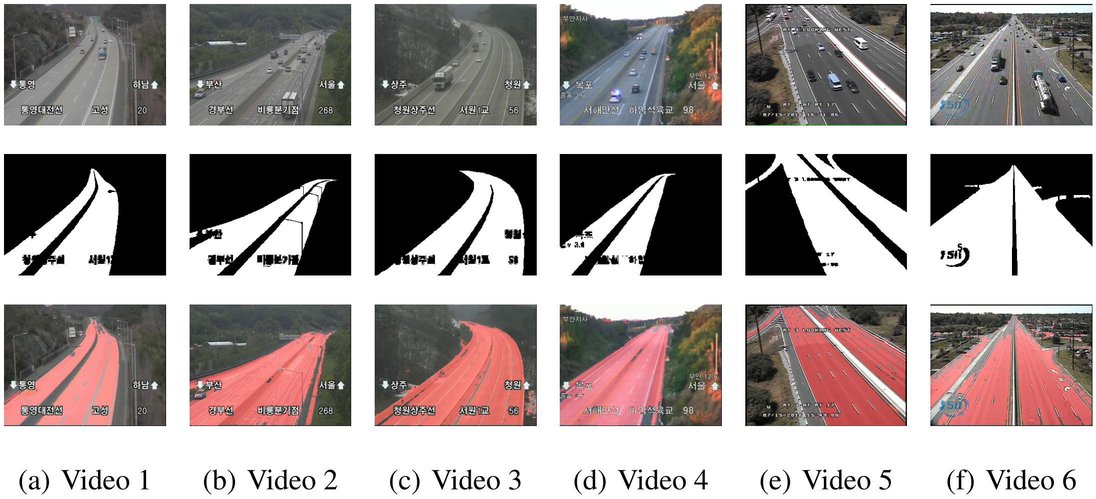


Figure 5.16 A few results of the road segmentation method. The first row contains some sample road images, the second row represents the corresponding ground truth road map, and the third row shows the results of the road segmentation method indicated by a red mask.

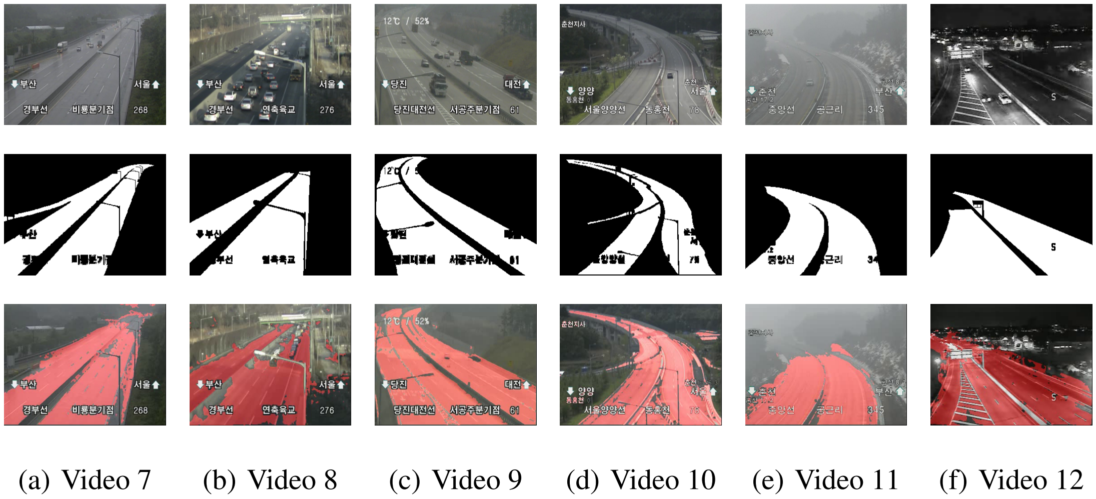
5.3 Experiments

In this section, the performance of the proposed method is evaluated on different videos with various illumination and weather conditions, resolution, and frame-rate values in order to ensure the diversity of the tested data. The used dataset, provided by the New Jersey Department of Transportation (NJDOT), contains 84 real traffic surveillance videos with various illumination conditions, road shapes, resolutions, viewing angles, and frame rates. A sample frame of each video is displayed in the first rows of Figures 5.17 and 5.18. The ground-truth mask representing the road region corresponding to each video is illustrated in the second row of Figures 5.17 and 5.18 and the third rows present the resulting extracted road as a red mask on the background image of each video.



(a) Video 1 (b) Video 2 (c) Video 3 (d) Video 4 (e) Video 5 (f) Video 6

Figure 5.17 Road extraction results in regular traffic videos. The first row displays a sample frame of each video. The second row represents the ground-truth road region masks. The third row illustrates the extracted road region by the proposed method before applying the accumulative foreground mask.



(a) Video 7 (b) Video 8 (c) Video 9 (d) Video 10 (e) Video 11 (f) Video 12

Figure 5.18 Road extraction results in traffic videos with challenging illumination conditions. The first row displays a sample frame of each video. The second row represents the ground-truth road region masks. The third row illustrates the extracted road region by the proposed method before applying the accumulative foreground mask.

5.3.1 Dataset

As the automatic two-direction ROI determination method is a relatively new topic in traffic video processing, there is no publicly available benchmark dataset with ground-truth

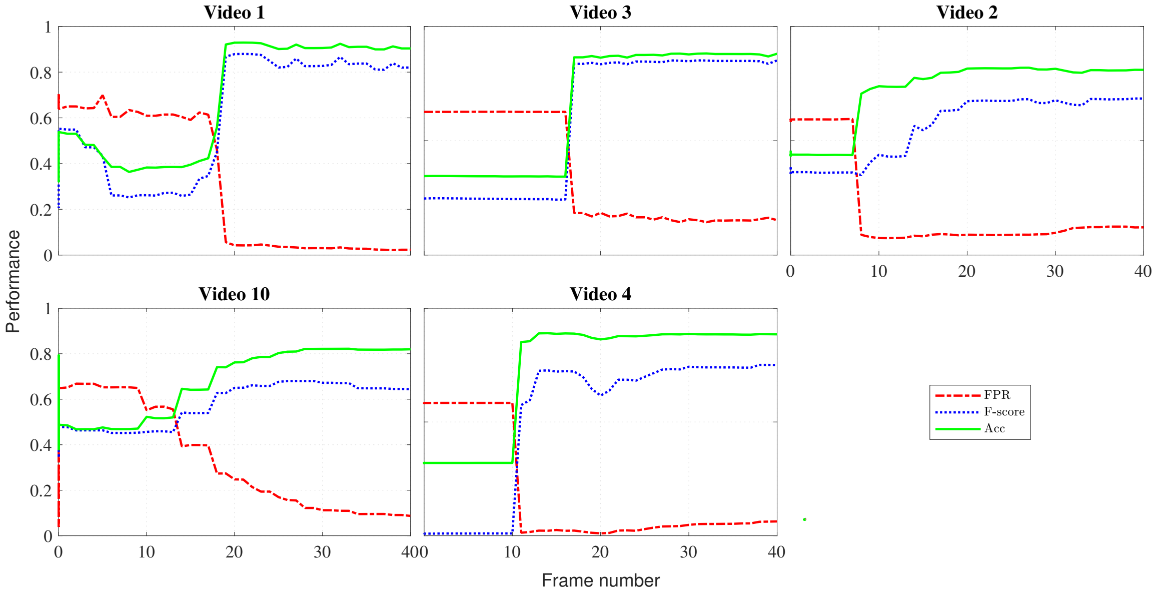
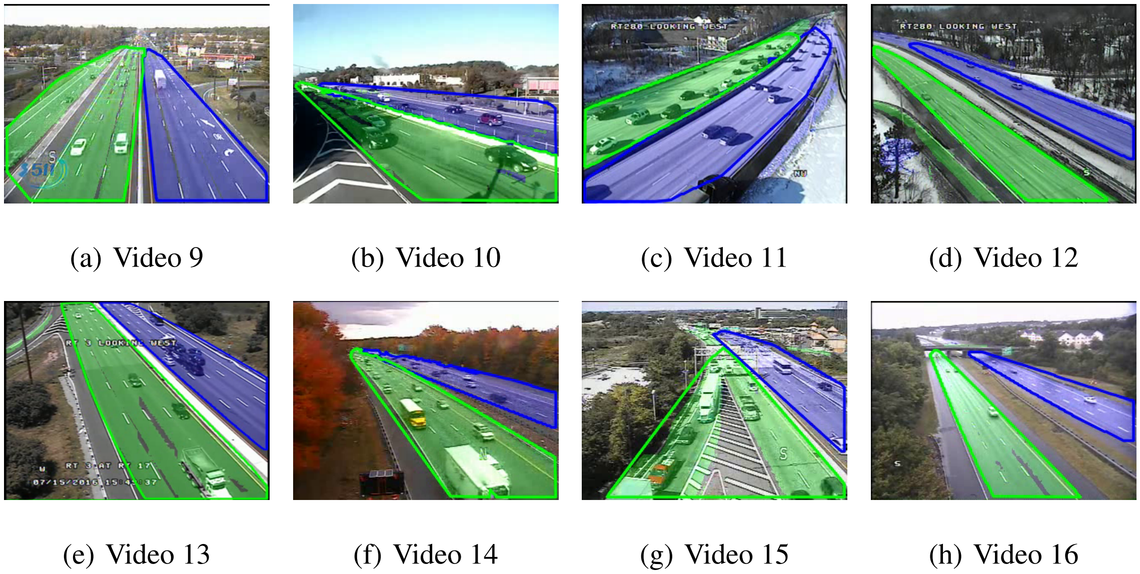


Figure 5.19 The F-measure score, accuracy, and false-positive rate of the proposed road extraction method at different frames, tested on some of the sample traffic videos. The sudden improvement in the performance measures happens when the first vehicle is observed in the video sequence and the initial road samples are obtained based on its location.


(a) Video 9 (b) Video 10 (c) Video 11 (d) Video 12

(e) Video 13 (f) Video 14 (g) Video 15 (h) Video 16

Figure 5.20 Some experimental results of the proposed method on traffic surveillance videos. The blue color and green color indicate the two sides of traffic (ROIs) determined by our proposed method.

Table 5.1 The Quantitative Evaluation of the Proposed Road Detection Method

Video #	**Precision**	**Recall**	**F-Score**
1	0.98	0.96	0.97
2	0.87	0.93	0.90
3	1.0	0.95	0.97
4	0.93	0.94	0.93
5	0.99	0.87	0.92
6	0.99	0.73	0.84
7	0.86	0.98	0.92
8	0.89	0.96	0.92
9	0.97	0.89	0.93
10	0.80	0.92	0.86
11	0.97	0.89	0.93
12	0.99	0.91	0.95
Average	0.94	0.91	0.93

Table 5.2 The Properties of the Traffic Video Sequences Represented in Figure 5.20

Video #	9	10	11	12
Resolution	320×240	352×240	640×482	640×480
FPS	15	15	15	15
Video #	13	14	15	16
Resolution	352×240	320×240	320×240	320×240
FPS	30	15	15	15

data for two-side roadways. We have used real traffic video sequences from the New Jersey Department of Transportation (NJDOT) for evaluation. This dataset contains dozens of diverse traffic surveillance video scenarios, with different illumination circumstances, weather conditions, and spatial resolutions.

5.3.2 Performance analysis

The experiments were carried out using a Dell XPS 8900 PC with a 3.4 GHz processor and 16 GB of RAM. The average speed was ~ 42.22 frames per second for videos of a size 720×480 pixels, which shows the feasibility of the proposed method for real-time applications.

In order to evaluate the quantitative results, several evaluation metrics are utilized as follows:

$$\begin{cases} FPR = F_P/(F_P + T_N) \\ PRE = T_P/(T_P + F_P) \\ REC = T_P/(T_P + F_N) \\ ACC = (T_P + T_N)/(T_P + F_P + T_N + F_N) \\ F_1 = 2 \times (PRE \times REC)/(PRE + REC) \end{cases} \tag{5.33}$$

where T_P, F_P refer to the number of pixels correctly and incorrectly detected as part of the road region, and T_N and F_N are the number of pixels that are correctly and incorrectly detected as part of the non-road region, respectively. FPR, PRE, REC, ACC, and F_1 refer to false positive rate, precision, recall, accuracy, and F-measure respectively. The number of pixels classified as road and non-road are compared with the ground-truth data to calculate each measure. Figure 5.19 demonstrates the accuracy, F1 score, and false-positive rate charts for a number of traffic videos. An instant improvement in the detection results can be seen in the charts shown in Figure 5.19 which corresponds to the frame at which the first vehicle is observed in the video and a number of pixels corresponding to the location of the vehicle can be used as initial road samples.

Table 5.1 shows the quantitative performance of the road extraction method given 12 sample traffic videos. The precision values are higher than the recall values in most cases, which means that the entire roadway region is not always extracted due to under-segmentation. Some examples can be seen in Figures 5.17(e), 5.17(f), 5.18(a) and 5.18(e). This is usually caused by the perspective view and losing the tracking information at the far side of the road. Also, strong cast shadows and congested traffic can result in excluding some road pixels at the initial frames from the road map (e.g., Figure 5.18(b)). In some videos, the recall value is higher than the precision, which means there are more false-positive cases than false-negative ones. This is due to the overestimation or leak segmentation, which is in turn caused by inconspicuous edges and a lack of sufficient gradient information at the road boundaries. Another reason is the illumination effect, which makes the non-road regions such as the sky have similar values to the road pixels. Some examples of this can be seen in Figures 5.17(b), 5.17(d), 5.18(e) and 5.18(f). Here, we do not make any presumptions about the shape of the road in order for the approach to work on unstructured roads. Therefore, segmentation errors cannot be avoided by restrictions based on geometric models.

The performance of the ROI detection method introduced in Section 5.2.4 is evaluated using videos with various view angles and illumination conditions. Table 5.2 shows the video information we have used in our experiment. Figure 5.20 illustrates some examples of the ROIs determined by our proposed method. In each frame, the green and blue colors represent the two traffic regions (ROI) determined by our proposed method, respectively. We can see that the automatically detected regions cover most of the road regions, which can directly be utilized as the ROIs in the applications of traffic surveillance videos.

5.3.3 Discussion

In this chapter, we have not made any assumptions about the shape of the roadway or the viewing angle of the camera. This approach can work on straight, curved, forked, and other road structures. The method is completely automatic and performed in real-time, which makes it applicable in real-world scenarios. However, in some videos with challenging illumination and weather conditions, the initial road region extraction might have leak segmentation errors due to the similarities between the road pixels and the surrounding (e.g. sky). These errors are later dealt with by using the location of moving foreground objects. However, achieving a good ROI determination can take longer.

5.4 Conclusion

Determining the region of interest (ROI) is a fundamental preprocessing step in video analysis applications. In this study, a statistical method is proposed to automatically determine the region of interest corresponding to two major traffic directions in surveillance videos captured from roads with bidirectional traffic. Our proposed method has two contributions. First, the road region is segmented automatically by using color, edge, and temporal features and applying a background subtraction method along with the flood-fill operation. Second, two regions of interest are generated, representing the major traffic directions on roads and highways with bidirectional traffic. As opposed to the supervised learning methods, the proposed method can adapt well to a wide range of videos with different illumination conditions and viewing angles in real-time. The experimental results using real traffic videos provided by NJDOT demonstrate the good performance of the proposed methods.

CHAPTER 6

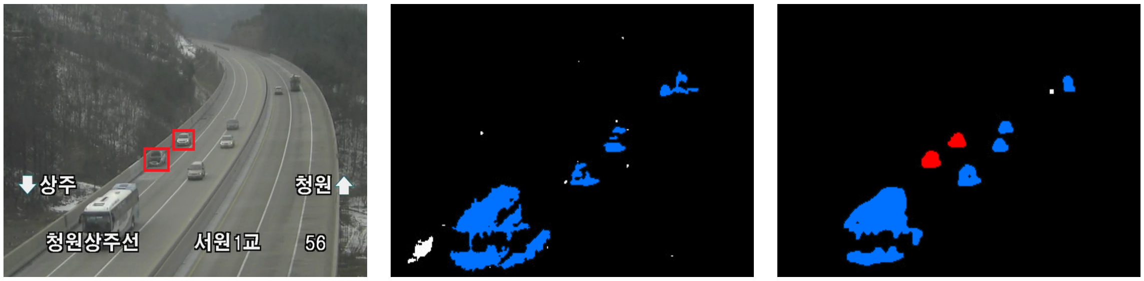
SINGLE-VEHICLE ACCIDENT DETECTION

6.1 Introduction

Vehicle accidents on major roads and highways are one of the main issues in traffic management. It is important to report accidents immediately when they occur so that they can be dealt with without much delay. Automatic detection of traffic accidents helps turn traffic back to normal, and if needed, further medical assistance may be requested in a timely fashion. The term ”accident on the road” may refer to different scenarios, such as rear-end, side-impact, head-on collisions, vehicle rollovers, or single-car accidents. The focus of this study is on single-vehicle accidents when a vehicle strikes a stationary object such as a tree or a barrier on the side of the road. Such incidents are usually caused by the driver losing control of the vehicle and making a sudden turn towards the road-side when there is no turning point.

In order to detect accidents on a highway involving vehicles, the first step is to detect and separate them from the background. Background subtraction methods based on the Gaussian mixture models are statistical techniques that provide a suitable approach to extract the foreground objects with a relatively low time complexity. We apply the Global Foreground Modeling (GFM) method [125] for foreground detection. Note that the GFM method was chosen due to its robustness to noise, efficiency, and ability to keep the temporarily stopped objects in the foreground model. This is helpful in cases where the vehicles involved in an accident stop on the road after the accident.

After the moving objects are detected, they should be tracked as long as they are present in the scene in order to monitor their behavior and classify specific types of motion patterns. We apply the blob tracking method [18] for vehicle tracking. Note that this blob tracking method does not always track the vehicle continuously, but it is chosen for real-time



(a) Original frame (b) MOG foreground (c) GFM foreground

Figure 6.1 The foreground masks extracted using the MOG method and the GFM method, respectively. Note that the GFM method extracts a more accurate foreground mask with both the moving vehicles (blue) and the stopped vehicles (red) clearly detected in the binary mask. In comparison, the MOG method fails to detect the stopped vehicles.

vehicle tracking due to its simplicity and low computational complexity. Note also that in the process of accident detection, the vehicle only needs to be tracked for a short period of time when it is involved in an accident.

The idea of our proposed real-time single-vehicle traffic accident detection framework analyzes the motion of each vehicle and applies heuristics to decide whether the pattern of movement matches those of single-vehicle accidents.First, the boundaries of the active traffic region are automatically detected using the region of interest determination method introduced in the previous chapter. Second, the direction and speed of a vehicle are examined. For a single-vehicle accident to take place, the vehicle should move towards the side of the road at a rather high speed. The tracking information is utilized to estimate the direction for each vehicle, which is detailed in subsection 6.3.1. The average direction of the vehicles is calculated to estimate the correct moving direction at each point in the active traffic region. Finally, after noticing a vehicle making a sudden turn and moving outside of the traffic region, the variations in speed and neighboring foreground pixels are examined to decide whether a single-vehicle crash has happened. Subsection 6.3.2 explains the specific method.

6.2 A Statistical Modeling Method for Detecting Both Foreground Objects and Stopped Moving Objects

Vehicle traffic accidents often involve moving vehicles and stopped moving vehicles, as when a traffic accident occurs, a vehicle is initially moving and then stops. Therefore, traffic accident detection requires a method that is capable of detecting both foreground objects and stopped moving objects. We introduce in this section a statistical modeling approach that applies the Global Foreground Modeling (GFM) method [125], the Mixture of Gaussian (MOG) method [131], and the Bayes Classifier to detect foreground objects.

The GFM method models the foreground objects using a mixture of Gaussian distributions. Taking advantage of the fact that the foreground objects appear at different locations in some continuous frames, the GFM method models all the foreground pixels globally.In addition, the GFM method updates its parameters as the video progresses in order to adapt to different foreground objects.The global foreground model is described as follows:

$$P(\mathbf{x}|M_f) = \sum_{k=1}^{K} W_k N(\mathbf{x}|\omega_k) \tag{6.1}$$

$$N(\mathbf{x}|\omega_k) = \frac{\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_k)^t \Sigma_k^{-1}(\mathbf{x}-\mu_k)\right\}}{(2\pi)^{d/2} \mid \Sigma_k \mid^{1/2}} \tag{6.2}$$

$$\sum_{k=1}^{K} W_k = 1 \tag{6.3}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the feature vector that describes each pixel, M_f means the foreground class, K is the number of Gaussian distributions in the foreground model, W_k is the weight of the k_{th} Gaussian distribution $N(\mathbf{x}|\omega_k)$. μ_k and Σ_k are the mean vector and the covariance matrix of the k_{th} Gaussian density $N(\mathbf{x}|\omega_k)$. Note that every pixel that is classified as foreground is used to update the foreground model $P(\mathbf{x}|M_f)$. The foreground model is called global because it contains all the information about foreground pixels in the frame.

After the global foreground modeling, we also need to estimate a background model. We use the traditional MOG method, which estimates a Gaussian mixture density function for every location in a frame as the background model. The probability density function $P(\mathbf{x}|M_b, L)$ is calculated for location L as described by Stauffer and Grimson [131].

In order to classify a pixel into a foreground class or a background class, we apply the Bayes classifier for the classification.

$$p\left(\mathbf{x}|M_f, L\right) P(M_f, L) > p\left(\mathbf{x}|M_b, L\right) P\left(M_b, L\right) \tag{6.4}$$

For each pixel in a frame, if the inequality 6.4 holds, the pixel is classified as a foreground pixel. Otherwise, it is classified as a background pixel. Note that the conditional probability density functions $p(\mathbf{x}|M_f, L)$ and $p(\mathbf{x}|M_b, L)$ are estimated using the GFM model and the MOG model, respectively.The prior probabilities $P(M_f, L)$ and $P(M_b, L)$ are estimated using the weights of the MOG model [53].

Figure 6.1 shows the foreground masks extracted using the MOG method and the GFM method, respectively. Note that the GFM method extracts a more accurate foreground mask with both the moving vehicles and the stopped vehicles clearly detected in the binary mask. In comparison, the MOG method fails to detect stopped vehicles.

6.3 A Novel Real-Time Traffic Accident Detection Framework

Our proposed real-time single-vehicle traffic accident detection framework consists of three major methods: an automated traffic region detection method, a new traffic direction estimation method; and a traffic accident detection method using first-order logic. These three methods detect the active traffic region, estimate the traffic direction, and detect single-vehicle traffic accidents by applying the assumptions about the abnormality of the movement and specific behaviors of a vehicle that lead to crashing into the traffic barrier.

6.3.1 A new traffic direction estimation method

The first step after segmenting the foreground and tracking the vehicles is to estimate the traffic direction on the road.Since in many traffic videos, the roads are curved, we cannot use a single direction for the entire road segment. Therefore, we divide the road into a number of rectangular areas and estimate one traffic direction for each of these areas based on the average direction and magnitude of the moving vehicles for each area of the road.A number of frames (f) are used to estimate the direction of each vehicle. This is done by finding the mean centroid from the first half and the second half of the f frames to estimate a consistent and smooth direction for the movement of each vehicle. The direction and magnitude of each vehicle are estimated as follows:

$$\begin{aligned} v_x &= x_{m_2} - x_{m_1} \\ v_y &= y_{m_2} - y_{m_1} \\ d_i &= arctan(v_y, v_x) \\ m_{v_i} &= \sqrt{v_x^2 + v_y^2} \end{aligned} \tag{6.5}$$

where v_x and v_y are the components of the velocity vector, x_{m_2} and y_{m_2} are the mean x and y values of the blob centroid in the most recent $f/2$ frames, x_{m_1} and y_{m_1} are the mean x and y values of the blob centroid in the remaining $f/2$ frames, d_i is the estimated direction of the vehicle i, and m_{v_i} is the estimated magnitude of the vehicle i, respectively. Note that we do not consider the slow movements for the direction estimation since, in these cases, the centroids are too close, which can lead to faulty results. Furthermore, the vehicles have to be mostly separated, and in situations when traffic congestion occurs, average directions are not updated. Consequently, only movements with considerable speed and size are considered for estimating the average direction and average speed.

After the calculation of the moving direction of the vehicles, the average direction and magnitude in each part of the active traffic region can be estimated based on equations 6.6

and 6.7 for each frame.

$$avg(c_{k_d}) = (1/F) * \sum_{f=1}^{F} \frac{\sum_{i=1}^{n} d_{i,f}}{N} \tag{6.6}$$

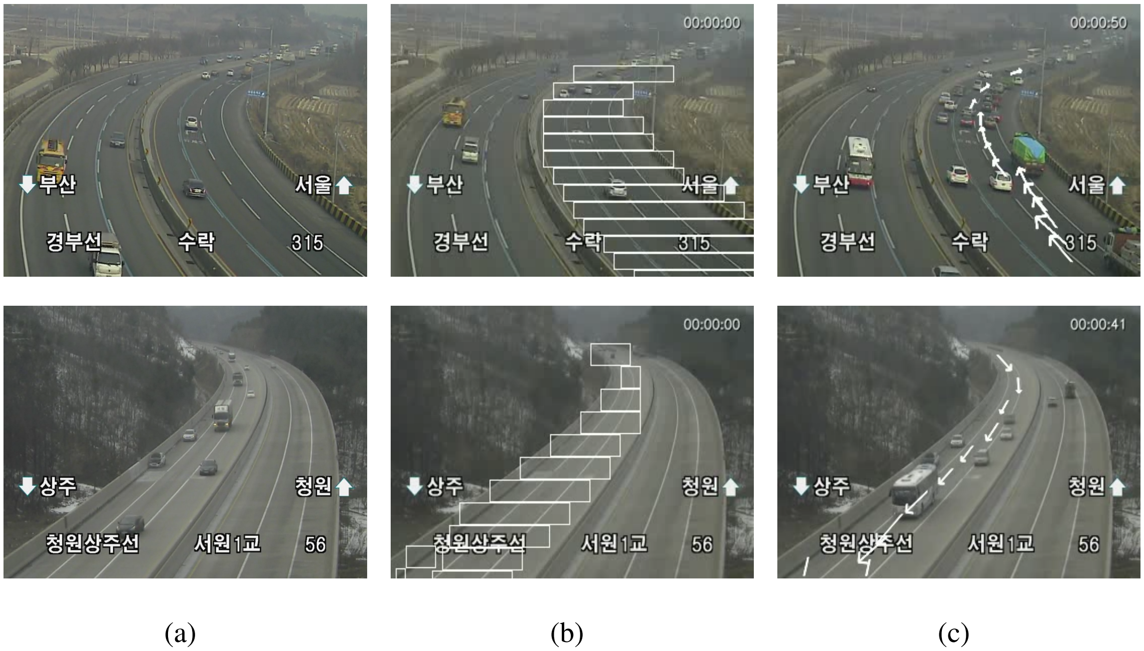
$$avg(c_{k_m}) = (1/F) * \sum_{f=1}^{F} \frac{\sum_{i=1}^{n} m_{i,f}}{N} \tag{6.7}$$

where c_{k_d} and c_{k_m} are the direction and magnitude of $cell_k$ respectively, F is the total number of frames, f is the current frame, n is the number of vehicles at $cell_k$, N is the total number of vehicles passed through $cell_k$, and $d_{i,f}$ and $m_{i,f}$ are the direction and magnitude of vehicle i at frame f respectively, which are calculated based on equation 6.5.

Figure 6.2 shows the estimation of the traffic flow direction at each area of the curved road. The size of these areas can be estimated by considering the size of the road section and the average size of the vehicles. To partition the road, we used the contour derived from the estimated traffic region map.

When a vehicle hits the traffic barrier, it usually starts with an abrupt movement, which is mostly caused by the driver losing control of the vehicle. This rapid movement can be detected by comparing the direction of the moving vehicle with the estimated direction for the area of the road where the vehicle is currently traveling. If the two degrees differ more than a notable value (d) and the magnitude of the movement is also large, it means that the vehicle is making a sudden unexpected move that often can be dangerous. This kind of hasty movement alone does not necessarily result in the vehicle colliding with the traffic barrier or another vehicle.

We should also consider the location of the vehicle after it has made a hasty move. If the vehicle goes beyond the estimated boundary of the road without slowing down its speed, there can be two possibilities. Either the vehicle is making a turn to another road that is not detected in road estimation (because there have not been enough cars making a similar turn), or the driver is making a rapid side move, which can be due to losing control. In the



(a) (b) (c)

Figure 6.2 Estimation of the traffic flow direction at each area of the curved road. (a) The original frame from a traffic video. (b) The automatically partitioned rectangular areas of the curved road. (c) The estimated average direction and magnitude of the moving vehicles for each part of the road.

first case, the vehicle will not crash and will continue its movement, and that road will be added to the estimated road map. However, if the vehicle actually hits an obstacle, it will most probably have a considerable change in its speed, direction, and acceleration. In some scenarios, this type of accident may also lead to vehicle rollovers. After the traffic accident, the vehicle itself and some of the surrounding vehicles usually stop, and traffic congestion occurs. All these cues can help detect a single-vehicle traffic accident. Furthermore, another cue of a single-vehicle collision can be the foreground segmentation mask showing a splash (an unexpected blob detected in the middle of the road) caused by the vehicle hitting the traffic barrier (see Figure 6.3).

6.3.2 A traffic accident detection method using the first-order logic

By considering the occurrence of a sequence of steps, we are able to detect single-vehicle collisions. To keep track of the target vehicle, we can use the stopped vehicle strategy as a

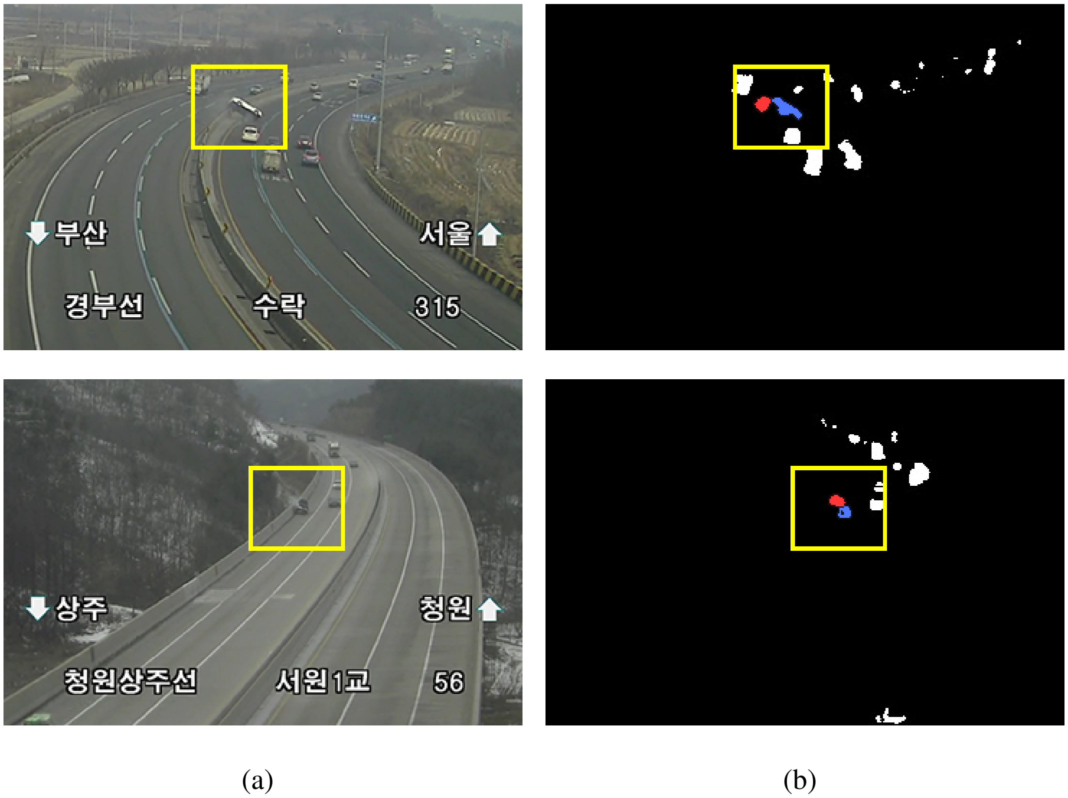


Figure 6.3 Unexpected blob detected in the middle of the foreground mask caused by the vehicle hitting the traffic barrier. The vehicle and the unexpected emerged blob are indicated by blue and red colors respectively. (a) The original frame from traffic video. (b) The foreground mask and the unexpected blob caused by the vehicle crashing the traffic barrier.

factor that makes the assumption more certain. To detect whether the vehicle is stopped, we use the foreground mask from the GFM method, which keeps the corresponding foreground information for temporarily stationary objects. Due to the fact that, in most cases, the vehicle stops after having an accident, and there might be some level of congestion and slow traffic flow. In other words, the probability of an accident having taken place is high if the same vehicle stops after the abnormal movement and if the nearby vehicles also stop or move at a slow speed. We can make an assumption about an accident occurring after having all these incidents happen in close proximity to each other. Here we consider all these factors in order to decide on the possibility of a single-vehicle traffic accident.

The first step of the proposed method is to estimate the location and boundaries of the two directions of the road by thresholding their accumulative foreground masks. As the number of vehicles passing through different parts of the road grows, the probability of that region belonging to the road increases. This step is useful for having an estimate of the correct traffic zone and the boundaries of the road.

The second step is to partition each part of the road into a number of rectangular areas, each of which has an average direction, average speed, and average blob size. The purpose of dividing the road into different areas is to estimate the direction of the traffic flow in each area of the road. Note that while on straight roads the direction of the traffic flow does not change much, on curved roads the direction changes rapidly. Therefore, partitioning the traffic region into smaller areas and assigning a unique average direction and speed to each of them can help improve traffic accident detection accuracy.The rectangular areas on each side of the road are calculated automatically based on the contour of the active traffic region map for that side. Each rectangular area covers the width of the road at the corresponding location and the height of each rectangle is set to be small enough to be reliable even at the curvy parts of the road.

The third step of the proposed method is to detect in real-time single-vehicle traffic accidents. Since crashing the barrier usually starts with an abrupt side-move, the direction of each tracked vehicle (not considering slow vehicles) is compared with the average direction of the corresponding area (part of the road where the vehicle is currently on). If a vehicle makes a rapid side-move, we keep track of that vehicle to see whether it moves out of the road boundaries or whether the abrupt movement ends earlier. In the event that the vehicle moves out of the traffic region, the changes in speed and the neighboring foreground mask are monitored. If the speed decreases suddenly and an unexpected foreground blob appears in the vicinity of the vehicle, it indicates that a crash has happened. Figure 6.4 shows the flowchart of the proposed real-time single-vehicle traffic accident detection framework.

The idea of our proposed real-time single-vehicle traffic accident detection framework may be expressed using the first-order logic knowledge representation language [119]. In

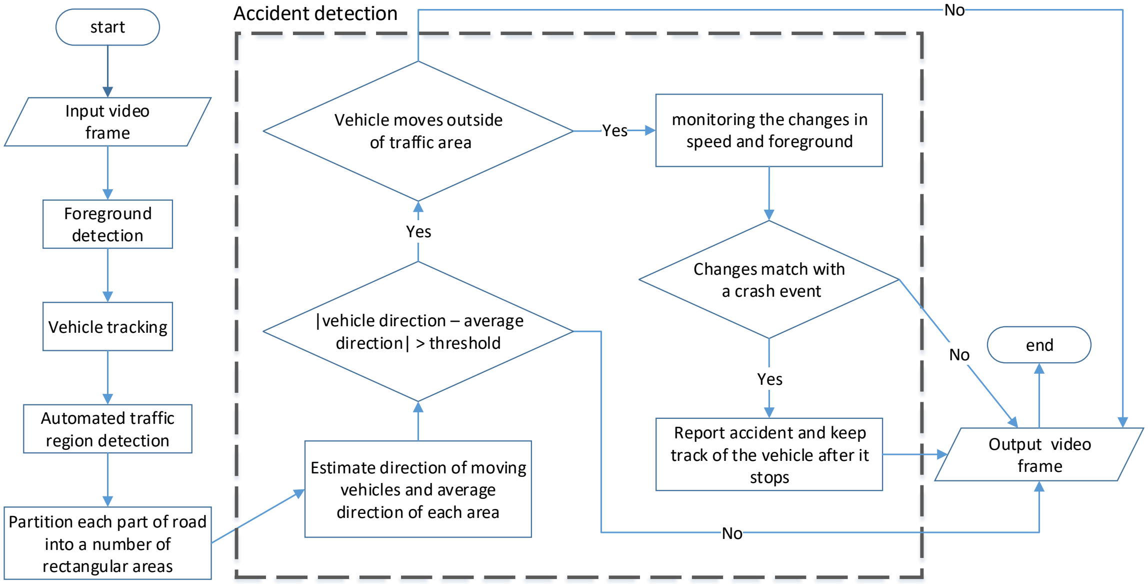

start
Input video frame
Foreground detection
Vehicle tracking
Automated traffic region detection
Partition each part of road into a number of rectangular areas
Accident detection
Estimate direction of moving vehicles and average direction of each area
|vehicle direction – average direction| > threshold
Yes
Vehicle moves outside of traffic area
Yes
monitoring the changes in speed and foreground
Changes match with a crash event
Yes
Report accident and keep track of the vehicle after it stops
No
No
No
end
Output video frame

Figure 6.4 Flowchart of the proposed real time single-vehicle traffic accident detection framework.

particular, the following statements (6.8), (6.9) and (6.10) represent the idea of the traffic accident detection.

$$\begin{aligned} &\forall v Vehicle(v) \land Fast(v) \land Swerve(v) \\ &\land \neg ShortDistance(v) \Rightarrow Rapid(v) \end{aligned} \tag{6.8}$$

where v represents a vehicle, $Vehicle(v)$ means that v is an actual tracked vehicle that is in the current frame, $Fast(v)$ means that the estimated magnitude for v is around the average magnitude of the cell containing its centroid or higher, $Swerve(v)$ indicates that the calculated direction of movement for vehicle v is different from the average direction of the cell containing its centroid by a value more than $45°$. $ShortDistance(v)$ stipulates that the size of movement should not be too small in order to avoid false positives caused by the inaccuracies in the blob detection process. $Rapid(v)$ means that vehicle v has made an abrupt side-move in an unexpected location (see Figure 6.6(b)). These types of movements do not always result in the vehicle crashing into an obstacle on the side of the road. Therefore, in order to draw the conclusion that an accident has happened, more information needs to be considered.

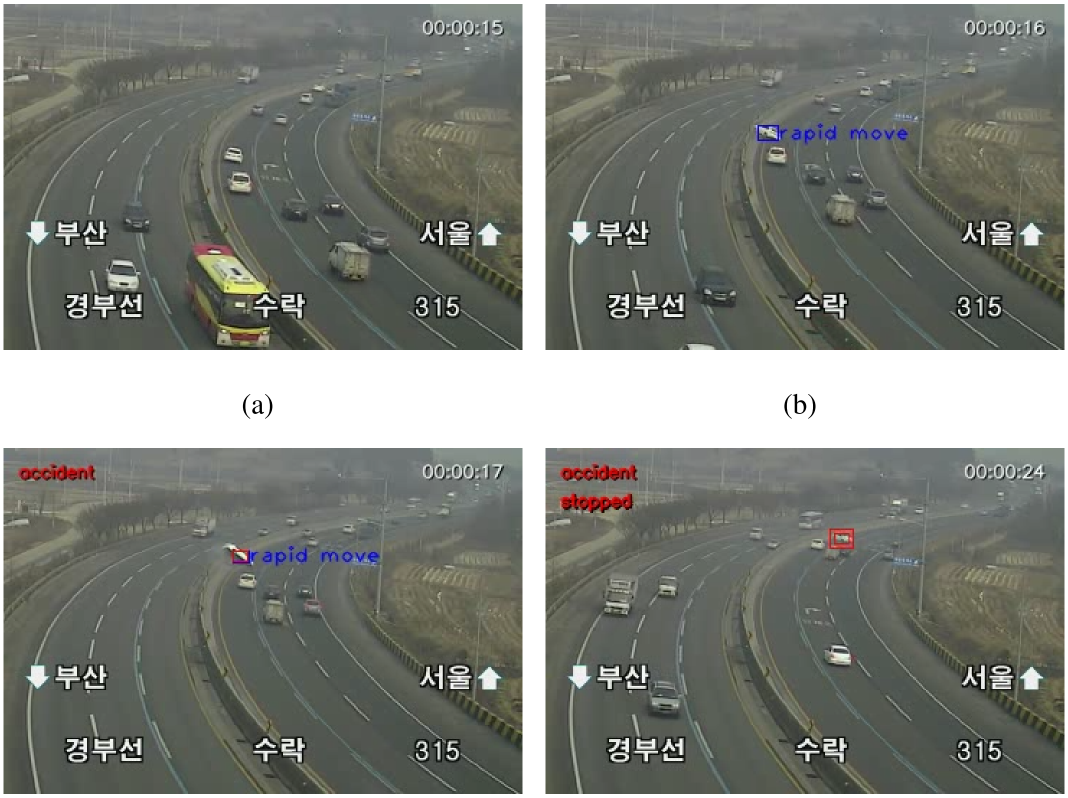
$$\begin{aligned} &\forall v Vehicle(v) \wedge OutOfBoundary(v) \\ &\wedge Splash(v) \Rightarrow Crash(v) \end{aligned} \tag{6.9}$$

where $OutOfBoundary(v)$ is a predicate which indicates that vehicle v has moved outside of the estimated traffic region, $Splash(v)$ means that there is an unexpected blob in the foreground mask in the surrounding block of vehicle v, and $crash(v)$ means that vehicle v has probably collided with some obstacle on the side of the road.

$$\begin{aligned} &\forall v Vehicle(v) \wedge Rapid(v) \wedge Crash(v) \\ &\wedge TimeOf(Rapid(v)) < TimeOf(Crash(v)) \\ &\Rightarrow Accident(v) \end{aligned} \tag{6.10}$$

where $TimeOf()$ is a function which returns the time when its input term has occurred, and $Accident(v)$ indicates that vehicle v has had a single-vehicle accident. Therefore, this statement means a single-vehicle crash happens when a vehicle hits the barrier after moving in that direction at a high speed without slowing down during this abrupt movement.

To prove the rules are complete for FOL, we can use the forward chaining method, which is complete for a Horn knowledge base. The knowledge base is a set of facts representing facts about a particular subject. As for these facts in the case of a single-vehicle road-side accident, we have assumed that if a vehicle makes a sudden turn to the side with a high enough speed and a long enough moving distance, it has made a dangerous move, which we call a rapid move. Also, we assume if a vehicle moves outside of the common traffic region boundaries and at the same time a blob of pixels appears in the foreground mask around the vehicle, a collision with an obstacle might have happened, which we call a crash.



(a)

(b)

(c)

(d)

Figure 6.5 Real time single-vehicle traffic accident detection results using a real traffic video. (a) Vehicles move in the correct traffic direction. (b) A vehicle makes a sudden side move. (c) The vehicle hits the road barrier. (d) The vehicle stops after the accident.

For a single-vehicle road-side accident to occur, the rapid movement should happen before the crash. If we consider V_1 to be a vehicle experiencing both incidents in chronological order, the occurrence of a single-vehicle accident can be concluded. Using the forward chaining method, we can use the known facts to keep proving new information and eventually prove the final clause. Assuming a vehicle V_1 has met all the preconditions of a single-vehicle accident, we can use the known facts to prove the accident has occurred.

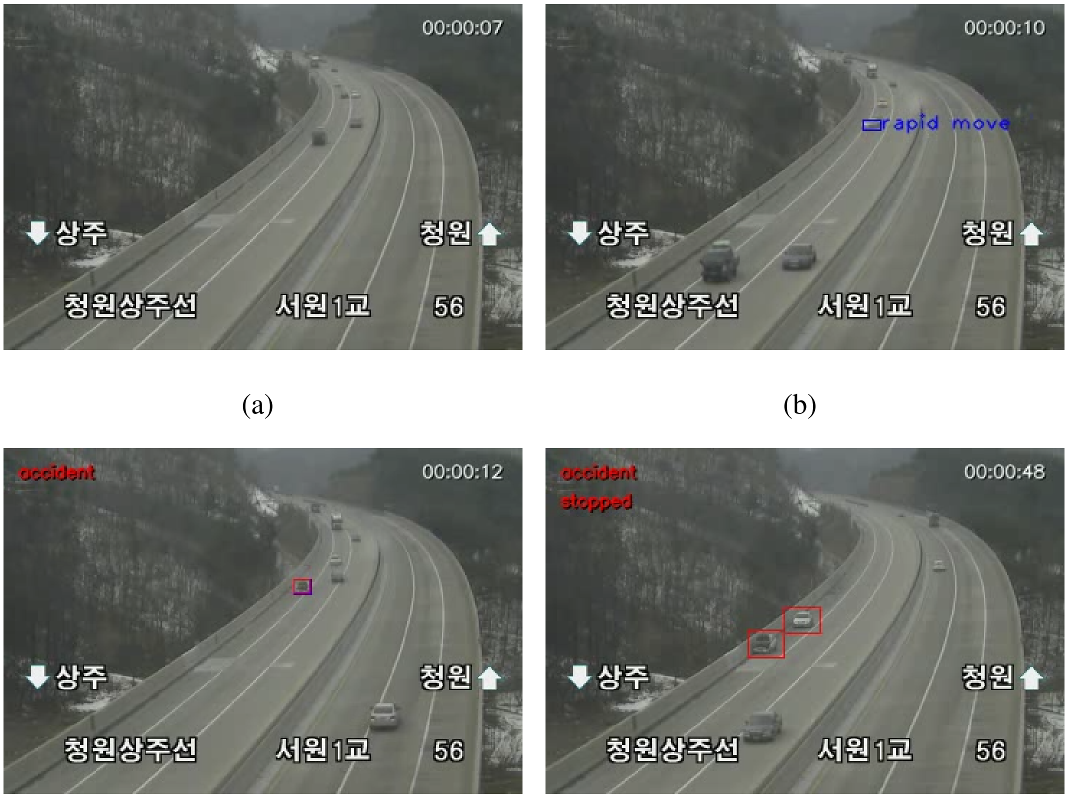
According to the statement 6.8 which is in the form of a Horn clause, the rapid movement of the vehicle V_1 can be proved by considering four facts from the knowledge base to be true for this vehicle. These facts are that V_1 is a vehicle and it has made a large movement at a high speed. In line with statement 6.9 which is also in the form of a

Horn clause, the predefined crash incident can be concluded by considering two more facts from the knowledge-base to be true about vehicle V_1 that are moving outside of the traffic region boundaries and occurrence of an unexpected foreground blob around the vehicle. As stated in 6.10, the resulting clause, which is $Accident(V_1)$ can then be concluded by the conjunction of the previous Horn clauses with another fact from knowledge-based that indicates the right time order.

6.4 Experiments

We apply real traffic videos from the department of transportation to evaluate our proposed method.The spatial resolution of the traffic videos used in our experiments is 720×480 with a frame rate of 30 frames per second. Specifically, first, the motion information from the videos is used to estimate the road boundaries. Second, the tracking and the foreground segmentation results are applied to detect the abnormal motion.And finally, the first-order logic decision-making system is utilized to detect single-vehicle accidents. Traffic accidents are detected in real time in the traffic videos without any false alarms. The experiments are implemented using a Dell XPS 8900 PC with a 3.4 GHz processor and 16 GB of RAM.

Figures 6.5 and 6.6 show the experimental results of real-time single-vehicle traffic accident detection using two real traffic videos from the department of transportation. Considering the limitation of video data for the specific type of single-car traffic accidents, we only apply our method to two video sequences. In particular, Figure 6.5 (a) shows that the vehicles are moving in the right traffic direction in a frame from one traffic video. Figure 6.5 (b) shows that a vehicle makes a sudden side move, which is detected automatically by our proposed method. Figure 6.5 (c) shows that the vehicle hits the road barrier, and our proposed method automatically detects such a single-vehicle traffic accident in real time. Figure 6.5 (d) shows that the vehicle stops after the accident, and our proposed method automatically detects both the traffic accident and the stopped vehicle



(a) (b)

(c) (d)

Figure 6.6 Real time single-vehicle traffic accident detection results using a real traffic video. (a) Vehicles moves in the right traffic direction. (b) A vehicle makes a sudden side move. (c) The vehicle hits the road barrier. (d) The vehicle stops after the accident.

in real time. Figure 6.6 shows the real-time traffic accident results using another real-time traffic video from the department of transportation.

Our proposed method successfully detects the vehicle's sudden move to the side of the road, the traffic accident when the vehicle hits the road barrier, as well as the stopped vehicles in real-time as shown in Figure 6.6 (b), Figure 6.6 (c), and Figure 6.6 (d), respectively. Figure 6.7 shows some other sample experimental results from real traffic videos. The vehicle involved in an accident is indicated by a red bounding box on the right image. Because of using the GFM method for foreground segmentation, we are able to keep track of the crashed vehicle even after it stops. Table 6.1 shows the length (in seconds) of videos; the runtime (in seconds) of our proposed method, the number of frames in each of

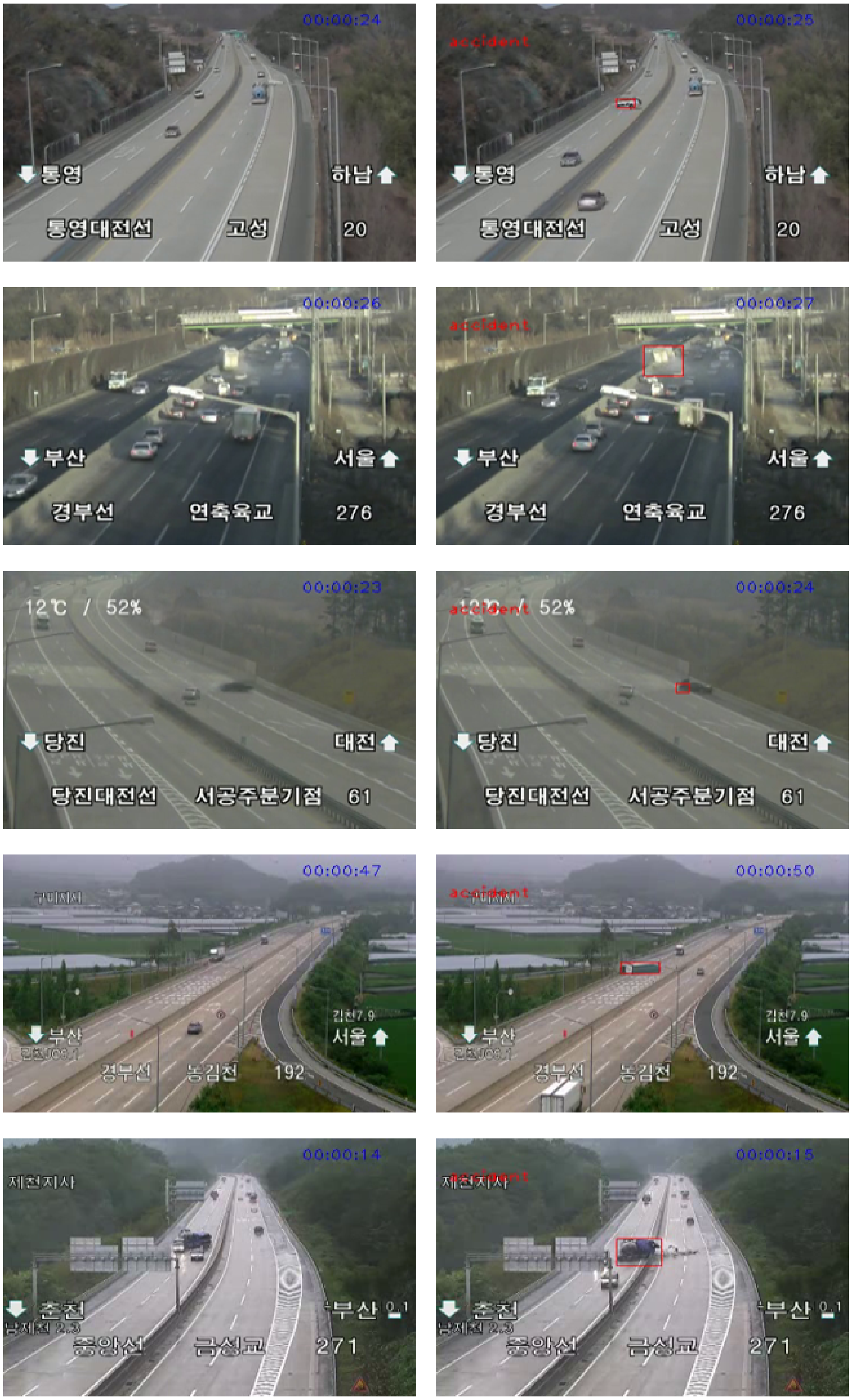
Table 6.1 The Runtime of the Proposed Accident Detection Method

	video 1	video 2	video 3	video 4	video 5	video 6	video 7
Length of video (s)	56	56	60	60	236	178	178
runtime (s)	47.04	45.36	25.20	54.00	198.24	77.43	82.77
Number of frames	1680	1680	900	1800	7080	2670	2670
Run-time per frame (ms)	28	27	28	30	28	29	31

the videos; and the runtime (in milliseconds) for each frame. From the table, we can see that our proposed method runs in real time.

6.5 Conclusion

We have presented in this chapter a novel real-time single-vehicle accident detection method for traffic video analysis. First, we use a statistical foreground modeling method to detect the foreground objects. In order to detect both the moving foreground objects and the temporarily stopped objects, the Global Foreground Modeling (GFM) method is used together with the Mixture of Gaussian (MOG) method. In addition, the Bayes classifier is applied for foreground and background classification. Second, we propose our novel traffic accident detection method. The contributions of our proposed method are three-fold: (i) a new traffic region detection method, (ii) a traffic direction estimation method, and (iii) a single-car run-off-road accident detection method using first-order logic. The traffic region detection method is used to find out the boundaries of the road. By detecting the road boundaries, we are able to detect vehicles that hit or go outside the boundaries. The traffic direction estimation method is able to estimate the correct direction of the moving traffic. A vehicle moving in an abnormal direction may cause a traffic accident. These two methods can provide some clues for detecting a traffic accident. Finally, we use first-order logic to make a final decision based on these clues. We implement our proposed method and evaluate it using real-time traffic video data and achieve good performance in real-time traffic accident detection.



(a) (b)

Figure 6.7 Single-vehicle traffic accident detection results using some traffic videos. (a) A snapshot before the accident. (b) A snapshot after the accident.

CHAPTER 7

ACCIDENT DETECTION AT URBAN INTERSECTIONS

7.1 Introduction

One of the main problems in urban traffic management is the conflicts and accidents that occur at the intersections. Drivers caught in a dilemma zone may decide to accelerate at the time of the phase change from green to yellow, which in turn may induce rear-end and angle crashes. Additionally, despite all the efforts to prevent hazardous driving behaviors, running the red light is still common. Other dangerous behaviors, such as sudden lane changes and unpredictable pedestrian or cyclist movements at the intersection, may also arise due to the nature of traffic control systems or intersection geometry. Timely detection of such trajectory conflicts is necessary for devising countermeasures to mitigate their potential harm.

Currently, most traffic management systems monitor the traffic surveillance cameras by using manual perception of the captured footage. In addition to being a tedious and inefficient task for human operators, manual monitoring may not provide real-time reports of the observed incidents .Since most intersections are equipped with surveillance cameras, automatic detection of traffic accidents based on computer vision technologies will mean a great deal to traffic monitoring systems. Numerous studies have applied computer vision techniques in traffic surveillance systems [37, 41–44, 88, 123] for various tasks. Automatic detection of traffic incidents not only saves a great deal of unnecessary manual labor, but the spontaneous feedback also helps the paramedics and emergency ambulances to dispatch in a timely fashion. The outputs from trajectory conflict analysis offer useful insights into the association between the detected types of conflicts and the number of traffic incidents. An automatic accident detection framework provides useful information for adjusting intersection signal operation and modifying intersection geometry in order to defuse severe traffic crashes.

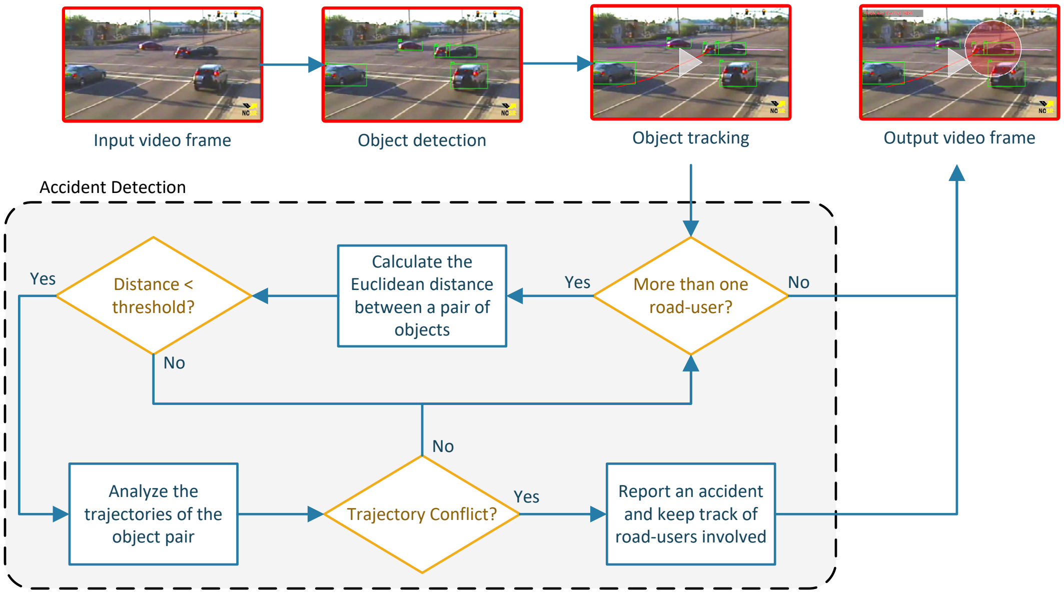


Figure 7.1 The system architecture of our proposed accident detection framework.

The first step in the accident detection framework is detecting objects of interest, such as vehicles, pedestrians, and cyclists.With the recent developments in deep convolutional neural networks (DCNNs), many studies have applied deep learning for the task of object detection in traffic surveillance [167]. Considering the applicability of our method in real-time edge-computing systems, we apply the efficient and accurate YOLOv4 [145] method for object detection. The second step is to track the movements of all interesting objects that are present in the scene to monitor their motion patterns. A new set of dissimilarity measures are designed and used by the Hungarian algorithm [74] for object association coupled with the Kalman filter approach for smoothing the trajectories and predicting missed objects. The third step in the framework involves motion analysis and applying heuristics to detect different types of trajectory conflicts that can lead to accidents. The moving direction and speed of road-user pairs that are close to each other are examined based on their trajectories in order to detect anomalies that could cause them to crash. The variations in acceleration, angle, and velocity are used as factors for detecting the road-user pairs that are involved in a near-accident or accident event. Figure 7.1 illustrates the system architecture of our proposed accident detection framework.

The layout of this chapter is as follows. In Section 7.2, the major steps of the proposed accident detection framework, including object detection (Subsection 7.2.1), object tracking (Subsection 7.2.2), and accident detection (Subsection 7.2.3) are discussed. Subsection 7.3 provides details about the collected dataset and experimental results, and the chapter is concluded in Subsection 7.4.

7.2 Methodology

This section provides details about the three major steps in the proposed accident detection framework. These steps involve detecting interesting road-users by applying the state-of-the-art YOLOv4 [145] method with a pre-trained model based on deep convolutional neural networks, tracking the movements of the detected road-users using the Kalman filter approach, and monitoring their trajectories to analyze their motion behaviors and detect hazardous abnormalities that can lead to mild or severe crashes. The proposed framework is purposely designed with efficient algorithms in order to be applicable in real-time traffic monitoring systems.

7.2.1 Road user detection

As in most image and video analytics systems, the first step is to locate the objects of interest in the scene. Since here we are also interested in the category of the objects, we employ a state-of-the-art object detection method, namely YOLOv4 [145], to locate and classify the road-users in each video frame. The family of YOLO-based deep learning methods demonstrates the best compromise between efficiency and performance among object detectors.

The first version of the You Only Look Once (YOLO) deep learning method was introduced in 2015 [115]. The main idea of this method is to divide the input image into an

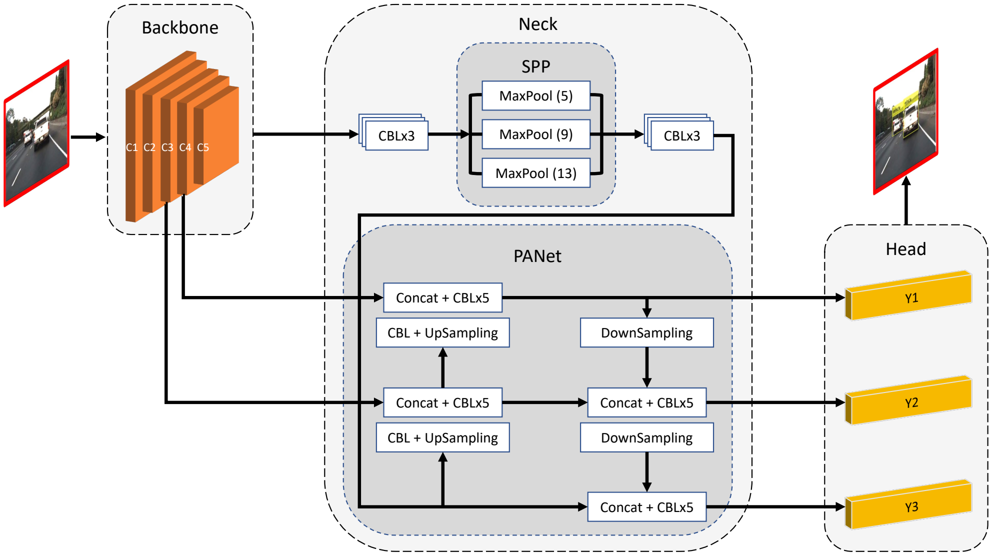


Figure 7.2 Architecture of the YOLOv4 model with three major component.

$S \times S$ grid where each grid cell is either considered as background or used for detecting an object. A predefined number (B) of bounding boxes and their corresponding confidence scores are generated for each cell. The intersection over union (IOU) of the ground truth and the predicted boxes is multiplied by the probability of each object to compute the confidence scores. Furthermore, the non-maximum suppression is applied to remove the repetitive bounding boxes. In later versions of YOLO multiple modifications have been made in order to improve the detection performance while decreasing the computational complexity of the method. Although there are online implementations, the latest official version of the YOLO family is YOLOv4 [145], which improves upon the performance of the previous methods in terms of speed and mean average precision (mAP). As illustrated in Figure 7.2, the architecture of this version of YOLO is constructed with a CSPDarknet53 model as the backbone network for feature extraction, followed by a neck and a head part. The neck refers to the path aggregation network (PANet) and spatial attention module, and the head is the dense prediction block used for bounding box localization and classification. This architecture is further enhanced by additional techniques referred to as bag of freebies and bag of specials.

Here, we have applied the YOLOv4 [145] model pre-trained on the MS COCO dataset [86] for the task of object detection. Although the model is pre-trained on a dataset with different visual characteristics in terms of object sizes and viewing angles, YOLOv4 proved to generalize well to images with overhead perspective. We are interested in trajectory conflicts among the most common road-users at regular urban intersections, namely, vehicles, pedestrians, and cyclists. Due to the hesitant nature of the decision-makers at intersections, trajectory conflicts can be the cause of mild-to-severe crashes.

7.2.2 Road user tracking

Multiple object tracking (MOT) has been intensively studied over the past decades [90] due to its importance in video analytics applications. Here we employ a simple but effective tracking strategy similar to that of the Simple Online and Realtime Tracking (SORT) approach [10]. The Hungarian algorithm [74] is used to associate the detected bounding boxes from frame to frame. Additionally, the Kalman filter approach is used as the estimation model to predict the future locations of each detected object based on their current location for better association, smoothing trajectories, and predicting missed tracks.

The inter-frame displacement of each detected object is estimated by a linear velocity model. The state of each target in the Kalman filter tracking approach is presented as follows:

$$o_i^t = [x_i, y_i, s_i, r_i, \dot{x_i}, \dot{y_i}, \dot{s_i}] \tag{7.1}$$

where x_i and y_i represent the horizontal and vertical locations of the bounding box center, s_i, and r_i represent the bounding box scale and aspect ratio, and $\dot{x_i}, \dot{y_i}, \dot{s_i}$ are the velocities in each parameter x_i, y_i, s_i of object o_i at frame t, respectively. The velocity components are updated when a detection is associated with a target. Otherwise, in the case of no association, the state is predicted based on the linear velocity model.

Considering two adjacent video frames t and $t+1$, we will have two sets of objects detected at each frame as follows:

$$\begin{aligned} O^t &= \{o_1^t, o_2^t, \dots, o_n^t\} \\ O^{t+1} &= \{o_1^{t+1}, o_2^{t+1}, \dots, o_m^{t+1}\} \end{aligned} \tag{7.2}$$

Every object o_i in set O^t is paired with an object o_j in set O^{t+1} that can minimize the cost function $C(o_i, o_j)$. The index $i \in [N] = 1, 2, \dots, N$ denotes the objects detected at the previous frame and the index $j \in [M] = 1, 2, \dots, M$ represents the new objects detected at the current frame.

In order to efficiently solve the data association problem despite challenging scenarios such as occlusion, false positive or false negative results from object detection, overlapping objects, and shape changes, we designed a dissimilarity cost function that employs a number of heuristic cues, including appearance, size, intersection over union (IOU), and position. The appearance distance is calculated based on the histogram correlation between an object o_i and a detection o_j as follows:

$$C_{i,j}^A = 1 - \frac{\sum_b \left(H_b(o_i) - \bar{H}(o_i)\right)\left(H_b(o_j) - \bar{H}(o_j)\right)}{\sqrt{\sum_b \left(H_b(o_i) - \bar{H}(o_i)\right)^2 \sum_b \left(H_b(o_j) - \bar{H}(o_j)\right)^2}} \tag{7.3}$$

where $C_{i,j}^A$ is a value between 0 and 1, b is the bin index, H_b is the histogram of an object in the RGB color-space, and $\bar{H}$ is computed as follows:

$$\bar{H}(o_k) = \frac{1}{B} \sum_b H_b(o_k) \tag{7.4}$$

in which B is the total number of bins in the histogram of an object o_k.

The size dissimilarity is calculated based on the width and height information of the objects:

$$C_{i,j}^S = \frac{1}{2}\left(\frac{|h_i - h_j|}{h_i + h_j} + \frac{|w_i - w_j|}{w_i + w_j}\right) \tag{7.5}$$

where w and h denote the width and height of the object's bounding box, respectively. The more different the bounding boxes of the object o_i and detection o_j are in size, the more $C_S^{i,j}$ approaches one. The position dissimilarity is computed in a similar way:

$$C_{i,j}^P = \frac{1}{2}\left(\frac{|x_i - x_j|}{x_i + x_j} + \frac{|y_i - y_j|}{y_i + y_j}\right) \tag{7.6}$$

where the value of $C_{i,j}^P$ is between 0 and 1, approaching more towards 1 when the object o_i and detection o_j are further. In addition to the mentioned dissimilarity measures, we also use the IOU value to calculate the Jaccard distance as follows:

$$C_{i,j}^K = 1 - \frac{Box(o_i) \cap Box(o_j)}{Box(o_i) \cup Box(o_j)} \tag{7.7}$$

where $Box(o_k)$ denotes the set of pixels contained in the bounding box of object k.

The overall dissimilarity value is calculated as a weighted sum of the four measures:

$$C_{i,j} = w_a C_{i,j}^A + w_s C_{i,j}^S + w_p C_{i,j}^P + w_a C_{i,j}^A + w_k C_{i,j}^K \tag{7.8}$$

in which w_a, w_s, w_p, and w_k define the contribution of each dissimilarity value in the total cost function. The total cost function is used by the Hungarian algorithm [74] to assign the detected objects at the current frame to the existing tracks. If the dissimilarity between a matched detection and track is above a certain threshold (τ_d), the detected object is initiated as a new track.

7.2.3 Accident detection

In this section, details about the heuristics used to detect conflicts between a pair of road users are presented. Conflicts among road-users do not always end in crashes. However, near-accident situations are also of importance to traffic management systems as they can indicate flaws associated with the signal control system and/or intersection geometry. Logging and analyzing trajectory conflicts, including severe crashes, mild accidents, and near-accident situations, will help decision-makers improve the safety of urban intersections. The most

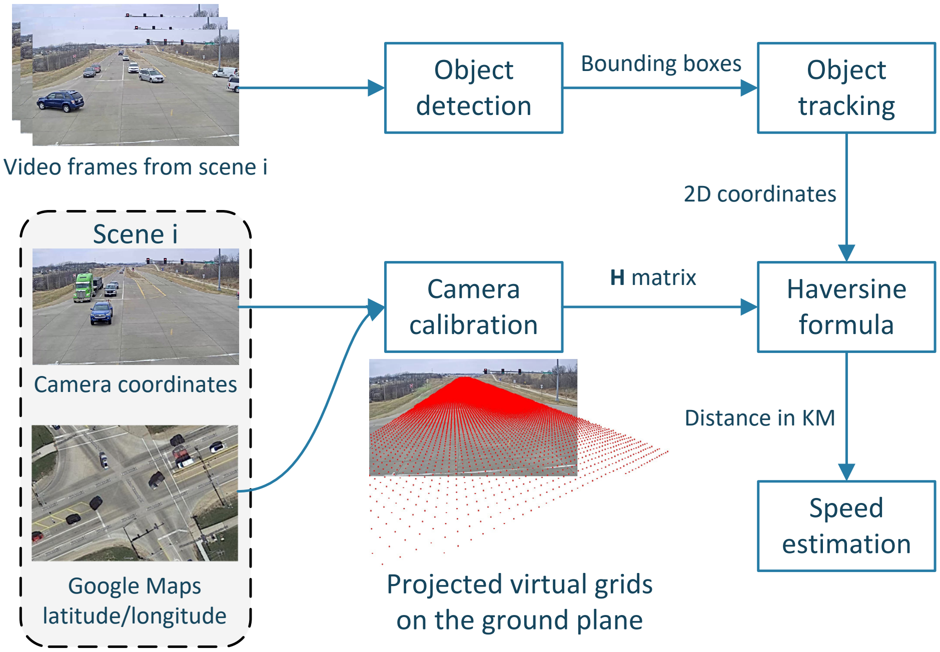


Figure 7.3 The workflow of the speed estimation method demonstrated on a scene from the NVIDIA AI City Challenge 2022 dataset [105].

common road-users involved in conflicts at intersections are vehicles, pedestrians, and cyclists. Therefore, for this study, we focus on the motion patterns of these three major road-users to detect the time and location of trajectory conflicts.

The Euclidean distances among all object pairs are calculated in order to identify the objects that are closer than a threshold to each other. These object pairs can potentially engage in a conflict, and they are, accordingly, chosen for further analysis. The recent motion patterns of each pair of close objects are examined in terms of speed and moving direction.

As there may be imperfections in the previous steps, especially in the object detection step, analyzing only two successive frames may lead to inaccurate results. Therefore, a predefined number f of consecutive video frames is used to estimate the speed of each road-user individually. The average bounding box centers associated to each track at the first half and second half of the f frames are computed. The two averaged points, p and q are transformed to the real-world coordinates using the inverse of the homography matrix $\mathbf{H}^{-1}$, which is calculated during camera calibration [135] by selecting a number of points on the frame and their corresponding locations on the Google Maps. The distance in kilometers can then be calculated by applying the haversine formula [39] as follows:

$$\begin{gathered} h = \sin^2\left(\frac{\phi_q - \phi_p}{2}\right) + \cos\phi_p \cdot \cos\phi_q \cdot \sin^2\left(\frac{\lambda_q - \lambda_p}{2}\right) \\ d_h(p, q) = 2r \arcsin\left(\sqrt{h}\right) \end{gathered} \tag{7.9}$$

where ϕ_p and ϕ_q are the latitudes, λ_p and λ_q are the longitudes of the first and second averaged points p and q, respectively, h is the haversine of the central angle between the two points, $r \approx 6371$ kilometers is the radius of earth, and $d_h(p, q)$ is the distance between the points p and q in real-world plane in kilometers. The speed s of the tracked vehicle can then be estimated as follows:

$$S = \frac{d_h(p, q) \times 3600 \times fps}{f} \tag{7.10}$$

where fps denotes the frames read per second and S is the estimated vehicle speed in kilometers per hour. Note that if the locations of the bounding box centers among the f frames do not have a sizable change (more than a threshold), the object is considered to be slow-moving or stalled and is not involved in the speed calculations.

Another factor to account for in the detection of accidents and near-accidents is the angle of collision. Traffic accidents include different scenarios, such as rear-end, side-impact, single-car, vehicle rollovers, or head-on collisions, each of which contains specific characteristics and motion patterns. When it comes to an intersection, most accidents occur due to reckless driving, running red lights, or risky decisions to pass the intersection when the vehicles are caught in the dilemma zone. These hazardous behaviors may induce angle or rear-end collisions. Accordingly, our focus is on the side-impact collisions at the intersection area where two or more road users collide at a considerable angle. The bounding box centers of each road user are extracted at two points: (i) when they are first observed and (ii) at the time of conflict with another road user. Then, the approaching angle of a pair

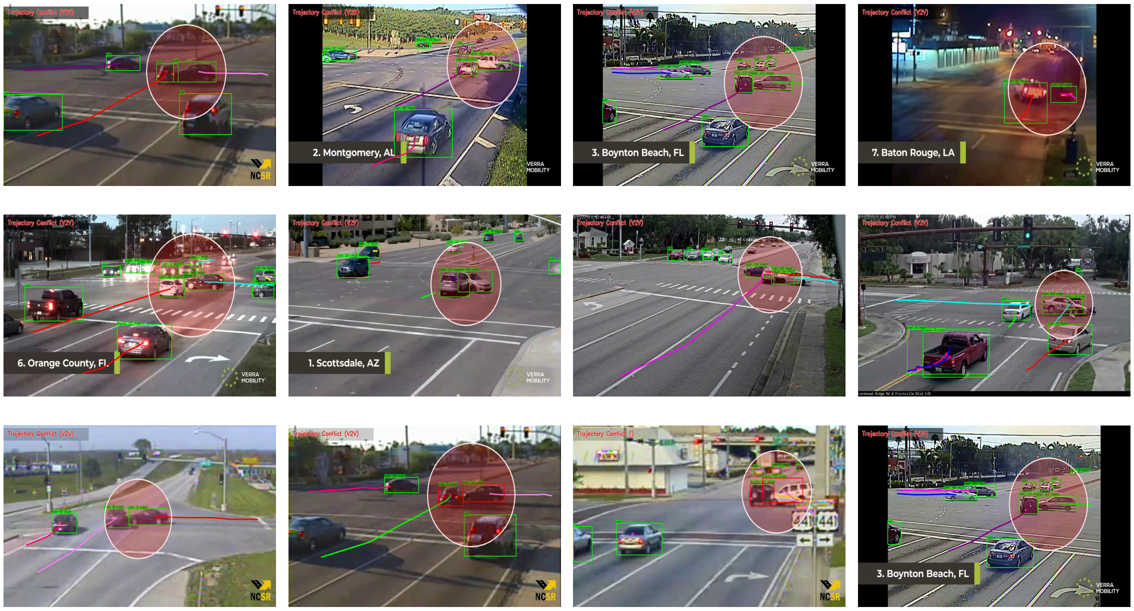


Figure 7.4 Vehicle-to-Vehicle (V2V) traffic accidents at intersections detected by our proposed framework. The red circles indicate the location of the incidents.

of road users a and b is calculated as follows:

$$\begin{aligned} m_a &= \frac{(y_a^t - y_a^{t'})}{(x_a^t - x_a^{t'})} \\ m_b &= \frac{(y_b^t - y_b^{t''})}{(x_b^t - x_b^{t''})} \\ \theta &= arctan\left(\frac{m_a - m_b}{1 + m_a m_b}\right) \end{aligned} \tag{7.11}$$

where θ denotes the estimated approaching angle, m_a and m_b are the the general moving slopes of the road-users a and b with respect to the origin of the video frame, x_a^t, y_a^t, x_b^t, y_b^t represent the center coordinates of the road-users a and b at the current frame, $x_a^{t'}$ and $y_a^{t'}$ are the center coordinates of object a when first observed, $x_b^{t''}$ and $y_b^{t''}$ are the center coordinates of object b when first observed, respectively.

If the bounding boxes of the object pair overlap each other or are closer than a threshold, the two objects are considered to be close. The trajectories of each pair of close road-users are analyzed with the purpose of detecting possible anomalies that can lead to accidents. The approaching angle of each pair of close objects is calculated and examined to see whether it is higher than the pre-defined threshold. The variations in the calculated magnitudes of the velocity vectors of each approaching pair of objects that have met the distance and angle conditions are analyzed to check for signs that indicate anomalies in the speed and acceleration. If the pair of approaching road-users move at a substantial speed towards the point of trajectory intersection during the previous f frames and the speed of one or both shows a sudden drop at the most recent frames, a trajectory conflict is reported. Trajectory conflicts involve near-accident and accident occurrences and include three types, namely, vehicle-to-vehicle (V2V), vehicle-to-pedestrian (V2P), and vehicle-to-bicycle (V2B).

7.3 Experiments

Due to the lack of a publicly available benchmark for traffic accidents at urban intersections, we collected 29 short videos from YouTube that contain 24 vehicle-to-vehicle (V2V), 2 vehicle-to-bicycle (V2B), and 3 vehicle-to-pedestrian (V2P) trajectory conflict cases. The dataset includes day-time and night-time videos of various challenging weather and illumination conditions. Each video clip includes a few seconds before and after a trajectory conflict. The spatial resolution of the videos used in our experiments is 1280×720 pixels with a frame-rate of 30 frames per second. We used a desktop with a 3.4 GHz processor, 16 GB of RAM, and an Nvidia GTX-745 GPU, to implement our proposed method. The average processing speed is 35 frames per second (fps), which is feasible for real-time applications.

The results are evaluated by calculating detection and false alarm rates as metrics:

$$\begin{aligned} DR &= \frac{\text{detected conflict cases}}{\text{total number of conflicts}} \\ FAR &= \frac{\text{number of false alarms}}{\text{total number of conflicts}} \end{aligned} \tag{7.12}$$

The proposed framework achieved a Detection Rate of 93.10% and a False Alarm Rate of 6.89%. The performance is compared to other representative methods in Table 7.1. The

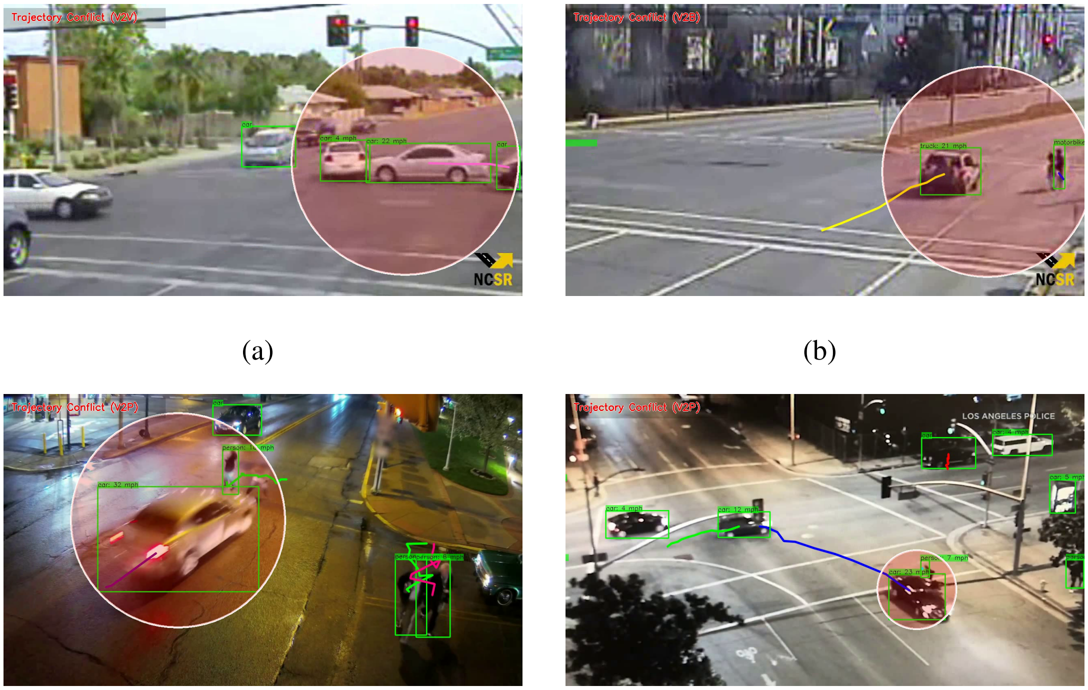
Table 7.1 Performance Comparison With Other Accident Detection Methods

Methods	Num. of videos	DR %	FAR %
Ki et al. [69]	1	63	6
Singh et al. [127]	7	77.5	22.5
Ijjina et al. [63]	45	71	**0.53**
Wang et al. [144]	–	92.5	7.5
Pawar et al. [110]	7	79	20.5
Proposed method	29	**93.1**	6.89

object detection and object tracking modules are implemented asynchronously to speed up the calculations.The trajectory conflicts are detected and reported in real-time with only 2 instances of false alarms, which is an acceptable rate considering the imperfections in the detection and tracking results. Our framework is able to report the occurrence of trajectory conflicts along with the types of the road-users involved immediately. Additionally, it keeps track of the location of the involved road-users after the conflict has happened. Figure 7.4 shows sample accident detection results by our framework given videos containing vehicle-to-vehicle (V2V) side-impact collisions. Furthermore, Figure 7.5 contains samples of other types of incidents detected by our framework, including near-accidents, vehicle-to-bicycle (V2B), and vehicle-to-pedestrian (V2P) conflicts.

7.4 Conclusion

In this chapter, a new framework is presented for automatic detection of accidents and near-accidents at traffic intersections. The framework integrates three major modules, including object detection based on the YOLOv4 method; a tracking method based on the Kalman filter and Hungarian algorithm with a new cost function; and an accident detection module to analyze the extracted trajectories for anomaly detection. The state-of-the-art YOLOv4 object detection method is applied due to its high performance and efficiency to locate and classify different road-users that are most commonly seen at urban intersections. The robust tracking method accounts for challenging situations, such as occlusion, overlapping objects, and shape changes in tracking the objects of interest and recording their trajectories. The



(a) (b)

(c) (d)

Figure 7.5 Different types of conflicts detected at the intersections. (a) Vehicle to Vehicle (V2V) near accident, (b) Vehicle to Bicycle (V2B) near accident, (c) and (d) Vehicle to Pedestrian (V2P) accident.

trajectories are further analyzed to monitor the motion patterns of the detected road-users in terms of location, speed, and moving direction. Different heuristic cues are considered in the motion analysis in order to detect anomalies that can lead to traffic accidents. A dataset of various traffic videos containing accident or near-accident scenarios has been collected to test the performance of the proposed framework against real videos. Experimental evaluations demonstrate the feasibility of our method in real-time applications of traffic management.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusions

This dissertation presents a fully automatic, real-time framework for single-vehicle and intersection accident detection in traffic video. Specifically, the moving objects are detected using a statistical method based on a mixture of Gaussians in order to locate the moving vehicles. Since shadows cast by moving objects cause issues for the following steps, in Chapter 4 a shadow detection and removal method is introduced. In this method, the potential shadow candidates are first extracted based on physics laws for reflection models. The attenuation in different channels of the RGB color space is examined in the case of pixels that are classified as foreground using two reference points from the background and foreground models. Various features are integrated to construct a feature vector for each pixel in the foreground class, and multivariate Gaussian mixture models are applied in order to classify the extracted features into objects and shadows. The classification results are further enhanced by applying the k-means clustering algorithm and separating the classes based on the location of the pixels. After removing the cast shadows from the foreground, the remaining pixels are considered to be objects.An efficient blob-tracking method is applied to the resulting foreground mask in order to track multiple objects simultaneously at a low computational cost.

In Chapter 5, the tracked foreground objects and the subtracted background are used to estimate the region of interest, which helps reduce the computational load as well as faulty results in the successive video analysis operations. On the other hand, the segmented road region is used to estimate the location of each vehicle relative to the boundaries of the road, which in turn helps with the detection of single-vehicle accidents. For the task of road segmentation, initial road samples are chosen from the corresponding locations of the moving objects in the background image. A feature vector is established for each pixel that consists of features from the grayscale, RGB, and HSV color-spaces, and a probability map is generated according to the standardized Euclidean distance between the feature vectors. After applying a binary threshold on the resulting probability map, an initial road mask is obtained, which is further refined by the integration of the pixels located by the Flood-fill algorithm. In the case of roads with bi-directional traffic, the extracted road is divided into separate regions based on estimating the traffic direction and modeling the results of the Lucas-Kanade optical flow algorithm in a mixture of Gaussians.

In Chapter 6, the sequential steps of the single-vehicle accident detection framework are explained in detail. The average direction and magnitude of the velocity vectors are calculated at each location and considered to be the correct moving pattern.At the same time, the motion pattern of each tracked vehicle is monitored separately in order to detect rapid changes that can lead to run-off-road collisions. In the case of a sudden change in direction, the variations in velocity of the corresponding vehicles are examined along with the changes in the neighboring foreground pixels in order to report single-vehicle accidents.The experimental results using public datasets and the videos provided by NJDOT demonstrate the practicability of our method in real-world applications.Intersection accidents are discussed in Chapter 7, where the road users are detected using the YOLOv4 method and further tracked by applying a number of heuristics in the association process. Later, the trajectories are analyzed in order to detect trajectory conflicts and different types of crashes at the intersections.

8.2 Challenges

On the whole, an intelligent traffic monitoring system should be concomitantly accurate, responsive, and generalizable. The Advanced Traffic Management Systems (ATMSs) are envisaged to enhance mobility, improve safety, increase transport efficiency, reduce environmental costs, and increase economic productivity in land transportation. The real-time data from multiple sensors, including monocular cameras, flowing into the Transportation Management Centers (TMCs) should be handled and processed using advanced intelligent systems with the goal of producing useful information and taking appropriate actions. The intelligent algorithms designed to process the visual data are responsible for providing the traffic monitoring systems with useful information, such as speed, volume, and different classes of road-users.Additionally, they are used for generating alerts for various events, including stopped vehicles, wrong-way vehicles, slow speed, congestion, trajectory conflicts, and accidents.The essential need for developing accurate, robust, and efficient algorithms for locating the objects of interest in traffic surveillance videos opens new horizons and prospects for future research studies.The main challenges and future research scope of intelligent traffic video analytics systems are briefly discussed in this section.

8.2.1 Performance and reliability

Without the ability to perform at a reasonable level of expectations, intelligent traffic video analytics systems will not be able to rely on automatic processing of the visual data and achieve sustainable development. A traffic operation center may house a large number of video feeds concurrently, and an unerring intelligent video analytics system can help lighten the workload of the human operators to a great extent. On the other hand, a faulty system that fails to generate reliable information and triggers too many false alarms will be liable and therefore futile. The algorithms designed to process the visual data should, first and foremost, perform satisfactorily enough to be trusted with the significant traffic data processing.

The performance of intelligent visual traffic monitoring systems revolves around their ability to locate the objects of interest, as it is the foundation of all the other major modules, such as object tracking, classification, and event detection.In spite of recent advancements in object detection techniques, most of the existing traffic surveillance systems use outdated methods. Despite all the shortcomings, motion segmentation techniques are still prevalently used due to old infrastructures, computational limits, lower costs, and a lack of sufficient training data for deep learning models. On the other hand, deep learning techniques are primarily designed to work with high-resolution and high-quality videos. Yet real-world traffic surveillance footage consists mostly of low-resolution videos with low frame rates. Designing effective algorithms that can perform well despite all these limitations is a challenging problem, which requires substantial effort by researchers in the field.

8.2.2 Versatility and flexibility

Traffic surveillance videos are continuously captured during night and day from different locations, and they can vary in illumination conditions, resolution, viewing angle, viewing distance, frame-rate, and weather conditions. Many of the Closed Circuit Television (CCTV) cameras provide the operators with the functionality of adjusting the Pan, Tilt, and Zoom (PTZ) movements to survey an area of interest.These variations introduce numerous possibilities in the visual characteristics of the traffic videos that only make it more challenging to locate the objects. In order for a video analytics system to robustly locate the objects of interest in various situations, it should be generalizable and adaptable to all sorts of changes in the visual properties. Otherwise, an algorithm that is limited to specific situations and cannot be deployed in real-world systems, due to significant performance degradation in adverse weather conditions, illumination changes, or different perspective views.In spite of the efforts to integrate other sensors, such as thermal cameras [52, 104] and LiDAR [155], they are not commonly used in real-world systems due to the additional costs. Developing algorithms to increase the generalization ability of object detection methods in monocular traffic surveillance can make worthwhile contributions to intelligent traffic monitoring systems.

8.2.3 Efficiency and responsiveness

Processing time is one of the main factors in intelligent traffic video analytics systems. A traffic operation center may house a large number of video feeds concurrently, and there is limited processing capability of the servers. On the other hand, in addition to locating the objects, there are multiple other tasks to be undertaken in order to analyze the continuous streams of video frames and produce useful information. Not to mention, many traffic incidents should be reported promptly with little to no tolerance for delay.

An intelligent traffic video analytics system is expected to operate responsively and process the video frames in real-time.The quality and the resolution of the video streams, frame rate, the computational capacity of the underlying platforms, and compliance with specified cost-efficiency policies are among the key points of consideration when defining the limits for the complexity of the designed algorithms.The object detection methods should be configured in a way that complies with the real-time requirements to be applicable in real-world systems.There have been numerous studies attempting to increase the efficiency of object detection if traffic surveillance applications on edge computing platforms [85, 88, 142].However, enabling the object detection algorithms, especially those that are designed based on deep learning, to achieve the desired real-time characteristics is still an open challenge that requires substantial effort.

8.3 Future Directions

With regard to future research directions, we plan to focus on improving the core steps of the video analysis applications, such as foreground segmentation and shadow detection, as well as reducing the computational complexity of the algorithms. We also want to work on improving the performance and generalizability of accident detection methods in order to detect other types of traffic incidents with few to no false alarms.

First, we are looking forward to improving the performance of the fundamental video analytics tasks such as foreground segmentation and shadow removal. The use of machine learning methods for learning the parameters of Gaussian mixture models in an online manner instead of manual initialization is one of the possible ways to achieve a more adaptive and automatic approach. Also, the effects of considering the relations among each pixel and its neighboring pixels, gradient features, and fuzzy Gaussian mixture models on the results of the foreground segmentation can be studied in order to further improve the results of the object detection step. In terms of detecting and removing shadows, we plan to examine the state stability of Gaussian distributions in the foreground model in order to see whether we can enhance the results of the Gaussian mixture model used in shadow detection.This is based on the observation that each pixel is generally more affected by recurrent shadow values than various objects.Also, the features extracted based on the laws of physics to extract the initial shadow samples are going to be one of the main points of focus in our future studies.

Second, computational complexity reduction is another direction for future research. The possibility of applying the analysis algorithms in a more selective manner and on a smaller number of pixels in each video frame can reduce the need for computational resources and also leave more room to include additional processing steps. Obtaining a more accurate and adaptive region of interest can specifically increase the speed and decrease the memory usage by a considerable amount.

Third, we are planning to include the detection of other types of traffic accidents, such as head-on collisions, rear-end collisions, and rollovers, in videos captured from highways and urban areas. Detecting different vehicle accidents in various visual conditions without false alarms or misdetections is a difficult task which requires extracting high-level features.Most current methods for accident detection tend to model the motion patterns statistically, which are regarded as normal movements, and abnormal motions are reported as anomalies. This can refer to collisions or stopped, wrong-way driving, and slow-speed vehicles. However, detecting the exact incident that is considered an anomaly is usually not straightforward. Therefore, another direction for future research can be the traffic incident type determination.

REFERENCES

[1] Parvin Ahmadi, Mahmoud Tabandeh, and Iman Gholampour. Abnormal event detection and localisation in traffic videos based on group sparse topical coding. *IET Image Processing*, 10(3):235–246, 2016.

[2] Emilio J Almazan, Yiming Qian, and James H Elder. Road segmentation for classification of road weather conditions. In *European Conference on Computer Vision (ECCV)*, October 8-10, 2016, Amsterdam, The Netherlands.

[3] Ariel Amato, Mikhail G Mozerov, Andrew D Bagdanov, and Jordi Gonzalez. Accurate moving cast shadow suppression based on local color constancy detection. *IEEE Transactions on Image Processing*, 20(10):2954–2966, 2011.

[4] Vicente Enrique Machaca Arceda and Elian Laura Riveros. Fast car crash detection in video. In *XLIV Latin American Computer Conference (CLEI)*, October 1-5, 2018, São Paulo, Brazil.

[5] Serge Ayer and Harpreet S Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *IEEE International Conference on Computer Vision (ICCV)*, June 20-23, 1995, Cambridge, Massachusetts, USA.

[6] Chris H Bahnsen and Thomas B Moeslund. Rain removal in traffic surveillance: Does it matter? *IEEE Transactions on Intelligent Transportation Systems*, 20(8):2802–2819, 2018.

[7] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[8] Olivier Barnich and Marc Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, 2010.

[9] Elizaveta Batanina, Imad Eddine, Ibrahim Bekkouch Ibrahim Bekkouch, Adil Khan, Asad Masood Khattak, and Mikhail Bortnikov. Domain adaptation for car accident detection in videos. In *International Conference on Image Processing Theory, Tools and Applications (IPTA)*, November 6-9, 2019, Istanbul, Turkey.

[10] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *IEEE international conference on image processing (ICIP)*, September 25-28, 2016, Phoenix, AZ, USA.

[11] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 13-19, 2020, Seattle, WA, USA.

[12] Ron Brinkmann. *The Art and Science of Digital Compositing: Techniques for Visual Effects, Animation and Motion Graphics*. Burlington, MA, USA: Morgan Kaufmann, 2008.

[13] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 21-26, 2017, Honolulu, HI, USA.

[14] Luca Caltagirone, Mauro Bellone, Lennart Svensson, and Mattias Wahde. Lidar–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111:125–131, 2019.

[15] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.

[16] Yi-Tung Chan. Comprehensive comparative evaluation of background subtraction algorithms in open sea environments. *Computer Vision and Image Understanding*, 202:103101, 2021.

[17] Chin-Kai Chang, Jiaping Zhao, and Laurent Itti. Deepvp: Deep learning for vanishing point detection on 1 million street view images. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 21-25, 2018, Brisbane, Australia.

[18] Fu Chang, Chun-Jen Chen, and Chi-Jen Lu. A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93(2):206–220, 2004.

[19] Marie-Neige Chapel and Thierry Bouwmans. Moving objects detection with a moving camera: A comprehensive review. *Computer Science Review*, 38:100310, 2020.

[20] Chia-Chih Chen and Jake K Aggarwal. Human shadow removal with unknown light source. In *International Conference on Pattern Recognition (ICPR)*, August 23-26, 2010, Istanbul, Turkey.

[21] Yanfeng Chen and Qingxiang Wu. Moving vehicle detection based on optical flow estimation of edge. In *International Conference on Natural Computation (ICNC)*, August 15-17, 2015, Zhangjiajie, China.

[22] Zhe Chen, Jing Zhang, and Dacheng Tao. Progressive lidar adaptation for road detection. *IEEE Journal of Automatica Sinica*, 6(3):693–702, 2019.

[23] Zhihao Chen, Liang Wan, Lei Zhu, Jia Shen, Huazhu Fu, Wennan Liu, and Jing Qin. Triple-cooperative video shadow detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 19-25, 2021, Virtual.

[24] Zhihao Chen, Lei Zhu, Liang Wan, Song Wang, Wei Feng, and Pheng-Ann Heng. A multi-task mean teacher for semi-supervised shadow detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 13-19, 2020, Seattle, WA, USA.

[25] Fan-Chieh Cheng, Bo-Hao Chen, and Shih-Chia Huang. A background model re-initialization method based on sudden luminance change detection. *Engineering Applications of Artificial Intelligence*, 38:138–146, 2015.

[26] Gong Cheng, Yiming Qian, and James H. Elder. Fusing geometry and appearance for road segmentation. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, October 22-29, 2017, Venice, Italy.

[27] Gong Cheng, Yue Wang, Yiming Qian, and James H Elder. Geometry-guided adaptation for road segmentation. In *Conference on Computer and Robot Vision (CRV)*, May 13-15, 2020, Ottawa, ON, Canada.

[28] Haris Cheong, Sripad Krishna Devalla, Tan Hung Pham, Liang Zhang, Tin Aung Tun, Xiaofei Wang, Shamira Perera, Leopold Schmetterer, Tin Aung, Craig Boote, et al. Deshadowgan: A deep learning approach to remove shadows from optical coherence tomography images. *Translational Vision Science and Technology*, 9(2):23–23, 2020.

[29] Ameni Chetouane, Sabra Mabrouk, Imen Jemili, and Mohamed Mosbah. Vision-based vehicle detection for road traffic congestion classification. *Concurrency and Computation: Practice and Experience*, 34(7), 2022.

[30] Sen-Ching S Cheung and Chandrika Kamath. Robust background subtraction with foreground validation for urban traffic video. *EURASIP Journal on Advances in Signal Processing*, 2005(14):1–11, 2005.

[31] 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*, 116(2):179–193, 2012.

[32] Goktug T. Cinar and José C. Príncipe. Adaptive background estimation using an information theoretic cost for hidden state estimation. In *International Joint Conference on Neural Networks (IJCNN)*, July 31 - August 5, 2011, San Jose, California, USA.

[33] Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, 2003.

[34] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014.

[35] Yuanquiang (Evan) Dong, T. X. Han, and Guilherme N. DeSouza. Illumination invariant foreground detection using multi-subspace learning. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 14(1):31–41, 2010.

[36] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *British Machine Vision Conference (BMVC)*, September 1-5, 2014, Nottingham, UK.

[37] Mohammad O Faruque, Hadi Ghahremannezhad, and Chengjun Liu. Vehicle classification in video using deep learning. In *Machine Learning and Data Mining in Pattern Recognition (MLDM)*, July 13-28, 2020, New York, USA.

[38] 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*, 24(6):381–395, 1981.

[39] Kenneth Gade. A non-singular horizontal position representation. *The Journal of Navigation*, 63(3):395–417, 2010.

[40] Belmar Garcia-Garcia, Thierry Bouwmans, and Alberto Jorge Rosales Silva. Background subtraction in real applications: Challenges, current models and future directions. *Computer Science Review*, 35:100204, 2020.

[41] Hadi Ghahremannezhad, Hang Shi, and Chenajun Liu. Robust road region extraction in video under various illumination and weather conditions. In *IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, December 9-11, 2020, Virtual Event, Italy.

[42] Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. Automatic road detection in traffic videos. In *IEEE International Confenrence on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, December 17-19, 2020, Exeter, United Kingdom.

[43] Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. A new adaptive bidirectional region-of-interest detection method for intelligent traffic video analysis. In *International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, December 9-13, 2020, Laguna Hills, CA, USA.

[44] Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. A real time accident detection framework for traffic video analysis. In *Machine Learning and Data Mining in Pattern Recognition (MLDM)*, July 18-23, 2020, New York, USA.

[45] Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. Illumination-aware image segmentation for real-time moving cast shadow suppression. In *IEEE International Conference on Imaging Systems and Techniques (IST)*, June 21-23, 2022, Kaohsiung, Taiwan.

[46] Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. Real-time accident detection in traffic surveillance using deep learning. In *IEEE International Conference on Imaging Systems and Techniques (IST)*, June 21-23, 2022, Kaohsiung, Taiwan.

[47] Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. Real-time hysteresis foreground detection in video captured by moving cameras. In *IEEE International Conference on Imaging Systems and Techniques (IST)*, June 21-23, 2022, Kaohsiung, Taiwan.

[48] Hadi Ghahremannezhad, Hang Shi, and Chengjun Liu. A new online approach for moving cast shadow suppression in traffic videos. In *IEEE International Intelligent Transportation Systems Conference (ITSC)*, September 19-22, 2021, Indianapolis, IN, USA.

[49] Ta Yang Goh, Shafriza Nisha Basah, Haniza Yazid, Muhammad Juhairi Aziz Safar, and Fathinul Syahir Ahmad Saad. Performance analysis of image thresholding: Otsu technique. *Measurement*, 114:298–307, 2018.

[50] Vitor Gomes, Pablo Barcellos, and Jacob Scharcanski. Stochastic shadow detection using a hypergraph partitioning approach. *Pattern Recognition*, 63:30–44, 2017.

[51] Ruiqi Guo, Qieyun Dai, and Derek Hoiem. Paired regions for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2956–2967, 2013.

[52] Yajing Han and Dean Hu. Multispectral fusion approach for traffic target detection in bad weather. *Algorithms*, 13(11):271, 2020.

[53] Eric Hayman and Jan-Olof Eklundh. Statistical background subtraction for a mobile observer. In *IEEE International Conference on Computer Vision (ICCV)*, October 14-17, 2003, Nice, France.

[54] Mohamed A Helala, Ken Q Pu, and Faisal Z Qureshi. Road boundary detection in challenging scenarios. In *IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, September 18-21, 2012, Beijing, China.

[55] Martin Hofmann, Philipp Tiefenbacher, and Gerhard Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, June 16-21, 2012, Providence, RI, USA.

[56] Jun-Wei Hsieh, Wen-Fong Hu, Chia-Jung Chang, and Yung-Sheng Chen. Shadow elimination for effective moving object detection by gaussian shadow modeling. *Image and Vision Computing*, 21(6):505–516, 2003.

[57] Jia-Bin Huang and Chu-Song Chen. A physical approach to moving cast shadow detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 19-24, 2009, Taipei, Taiwan.

[58] Jia-Bin Huang and Chu-Song Chen. Moving cast shadow detection using physics-based features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 20-25, 2009, Miami, Florida, USA.

[59] Xiaohui Huang, Pan He, Anand Rangarajan, and Sanjay Ranka. Intelligent intersection: Two-stream convolutional networks for real-time near-accident detection in traffic video. *ACM Transactions on Spatial Algorithms and Systems*, 6(2):1–28, 2020.

[60] Agha Asim Husain, Tanmoy Maity, and Ravindra Kumar Yadav. Vehicle detection in intelligent transport system under a hazy environment: A survey. *IET Image Processing*, 14(1):1–10, 2019.

[61] Juan Antonio Guerrero Ibáñez, Sherali Zeadally, and Juan Contreras-Castillo. Sensor technologies for intelligent transportation systems. *Sensors*, 18(4):1212, 2018.

[62] Manuel José Ibarra-Arenado, Tardi Tjahjadi, and Juan Pérez-Oria. Shadow detection in still road images using chrominance properties of shadows and spectral power distribution of the illumination. *Sensors*, 20(4):1012, 2020.

[63] Earnest Paul Ijjina, Dhananjai Chand, Savyasachi Gupta, and K Goutham. Computer vision-based accident detection in traffic surveillance. In *International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, July 6-8, 2019, Kanpur, India.

[64] Md Milon Islam, Md Repon Islam, and Md Saiful Islam. An efficient human computer interaction through hand gesture using deep convolutional neural network. *SN Computer Science*, 1(4):1–9, 2020.

[65] Wenyang Ji, Lingjun Tang, Dedi Li, Wenming Yang, and Qingmin Liao. Video-based construction vehicles detection and its application in intelligent monitoring system. *CAAI Transactions on Intelligence Technology*, 1(2):162–172, 2016.

[66] Moonyong Jin, Kiseon Jeong, Sook Yoon, and Dong Sun Park. Real-time pedestrian detection based on GMM and HOG cascade. In *International Conference on Machine Vision (ICMV)*, April 16-17, 2013, London, United Kingdom.

[67] Felipe Jorquera, Sergio Hernández, and Diego Vergara. Probability hypothesis density filter using determinantal point processes for multi object tracking. *Computer Vision and Image Understanding*, 183:33–41, 2019.

[68] Muhammad Junaid, Mubeen Ghafoor, Ali Hassan, Shehzad Khalid, Syed Ali Tariq, Ghufran Ahmed, and Tehseen Zia. Multi-feature view-based shallow convolutional neural network for road segmentation. *IEEE Access*, 8:36612–36623, 2020.

[69] Yong-Kul Ki and Dong-Young Lee. A traffic accident recording and reporting model at intersections. *IEEE Transactions on Intelligent Transportation Systems*, 8(2):188–194, 2007.

[70] Jong Bae Kim and Hang Joon Kim. Efficient region-based motion segmentation for a video monitoring system. *Pattern Recognition Letters*, 24(1-3):113–128, 2003.

[71] Kyungnam Kim, Thanarat H Chalidabhongse, David Harwood, and Larry Davis. Background modeling and subtraction by codebook construction. In *International Conference on Image Processing (ICIP)*, October 24-27, 2004, Singapore.

[72] 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*, 24(5):1015–1028, 2013.

[73] Frederick AA Kingdom. Perceiving light versus material. *Vision Research*, 48(20):2090–2105, 2008.

[74] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[75] Jinhui Lan, Jian Li, Guangda Hu, Bin Ran, and Ling Wang. Vehicle speed measurement based on gray constraint optical flow algorithm. *Optik*, 125(1):289–295, 2014.

[76] Hieu Le and Dimitris Samaras. From shadow segmentation to shadow removal. In *European Conference on Computer Vision (ECCV)*, August 23-28, 2020, Glasgow, UK.

[77] Hieu Le, Tomas F Yago Vicente, Vu Nguyen, Minh Hoai, and Dimitris Samaras. A+ d net: Training a shadow detector with adversarial shadow attenuation. In *European Conference on Computer Vision (ECCV)*, September 8-14, 2018, Munich, Germany.

[78] Jong Taek Lee, Kil-Taek Lim, and Yunsu Chung. Moving shadow detection from background image and deep learning. In *Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, November 23-27, 2015, Auckland, New Zealand.

[79] Woochul Lee and Bin Ran. Bidirectional roadway detection for traffic surveillance using online cctv videos. In *IEEE Intelligent Transportation Systems Conference (ITSC)*, September 17-20, 2006, Toronto, Ontario, Canada.

[80] Alessandro Leone and Cosimo Distante. Shadow detection for moving objects based on texture analysis. *Pattern Recognition*, 40(4):1222–1233, 2007.

[81] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video segmentation by tracking many figure-ground segments. In *IEEE International Conference on Computer Vision (ICCV)*, December 1-8, 2013, Sydney, Australia.

[82] Haoran Li, Yaran Chen, Qichao Zhang, and Dongbin Zhao. Bifnet: Bidirectional fusion network for road segmentation. *IEEE Transactions on Cybernetics*, 52(9):8617–8628, 2022.

[83] Yong Li, Weili Ding, XuGuang Zhang, and Zhaojie Ju. Road detection algorithm for autonomous navigation systems based on dark channel prior and vanishing point in complex road scenes. *Robotics and Autonomous Systems*, 85:1–11, 2016.

[84] Yong Li, Guofeng Tong, Anan Sun, and Weili Ding. Road extraction algorithm based on intrinsic image and vanishing point for unstructured road image. *Robotics and Autonomous Systems*, 109:86–96, 2018.

[85] Siyuan Liang, Hao Wu, Li Zhen, Qiaozhi Hua, Sahil Garg, Georges Kaddoum, Mohammad Mehedi Hassan, and Keping Yu. Edge yolo: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[86] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, September 6-12, 2014, Zurich, Switzerland.

[87] Dongfang Liu, Yaqin Wang, Tian Chen, and Eric T Matson. Accurate lane detection for self-driving cars: An approach based on color filter adjustment and k-means clustering filter. *International Journal of Semantic Computing*, 14(01):153–168, 2020.

[88] Guanxiong Liu, Hang Shi, Abbas Kiani, Abdallah Khreishah, Joyoung Lee, Nirwan Ansari, Chengjun Liu, and Mustafa Mohammad Yousef. Smart traffic monitoring system using computer vision and edge computing. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):12027–12038, 2022.

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

[90] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, 293:103448, 2021.

[91] Boutheina Maaloul, Abdelmalik Taleb-Ahmed, Smail Niar, Naim Harb, and Carlos Valderrama. Adaptive video-based algorithm for accident detection on highways. In *IEEE International Symposium on Industrial Embedded Systems (SIES)*, June 14-16, 2017, Toulouse, France.

[92] Lucia Maddalena and Alfredo Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168–1177, 2008.

[93] Nicholas A Mandellos, Iphigenia Keramitsoglou, and Chris T Kiranoudis. A background subtraction algorithm for detecting and tracking vehicles. *Expert Systems With Applications*, 38(3):1619–1631, 2011.

[94] Nicolas Martel-Brisson and Andre Zaccarin. Learning and removing cast shadows through a multidistribution approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1133–1146, 2007.

[95] Nicolas Martel-Brisson and André Zaccarin. Kernel-based learning of cast shadows from a physical model of light sources and surfaces for low-level segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 24-26, 2008, Anchorage, Alaska, USA.

[96] Ester Martinez-Martin and Angel P Del Pobil. Robot vision for manipulation: A trip to real-world applications. *IEEE Access*, 9:3471–3481, 2020.

[97] Bruce A Maxwell, Richard M Friedhoff, and Casey A Smith. A bi-illuminant dichromatic reflection model for understanding images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 24-26, 2008, Anchorage, Alaska, USA.

[98] Stephen J McKenna, Sumer Jabri, Zoran Duric, Azriel Rosenfeld, and Harry Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, 2000.

[99] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI Conference on Artificial Intelligence*, February 2-7, 2018, New Orleans, Louisiana, USA.

[100] Fernand Meyer. Topographic distance and watershed lines. *Signal Processing*, 38(1):113–125, 1994.

[101] Fernand Meyer. Color image segmentation. In *International Conference on Image Processing (ICIP)*, April 7-9, 1992, Maastricht, Netherlands.

[102] Kwang Moo Yi, Kimin Yun, Soo Wan Kim, Hyung Jin Chang, and Jin Young Choi. Detection of moving objects with non-stationary cameras in 5.8 ms: Bringing motion detection to your mobile device. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, June 23-28, 2013, Portland, OR, USA.

[103] Sohail Nadimi and Bir Bhanu. Physical models for moving shadow and object detection in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1079–1087, 2004.

[104] Yunyoung Nam and Yun-Cheol Nam. Vehicle classification based on images from visible light and thermal cameras. *EURASIP Journal on Image and Video Processing*, 2018(1):1–9, 2018.

[105] Milind Naphade, Shuo Wang, David C Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Yue Yao, Liang Zheng, Pranamesh Chakraborty, Christian E Lopez, et al. The 5th ai city challenge. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, June 19-25, 2021, Virtual.

[106] Wafa Nebili, Brahim Farou, and Hamid Seridi. Background subtraction using artificial immune recognition system and single gaussian. *Multimedia Tools and Applications*, 79(35-36):26099–26121, 2020.

[107] SeungJong Noh and Moongu Jeon. A new framework for background subtraction using multiple cues. In *Asian Conference on Computer Vision (ACCV)*, November 5-9, 2012, Daejeon, Korea.

[108] Adi Nurhadiyatna, Wisnu Jatmiko, Benny Hardjono, Ari Wibisono, Ibnu Sina, and Petrus Mursanto. Background subtraction using gaussian mixture model enhanced by hole filling algorithm. In *2013 IEEE international conference on systems, man, and cybernetics*, October 13-16, 2013, Manchester, United Kingdom.

[109] Deepak Kumar Panda and Sukadev Meher. A new wronskian change detection model based codebook background subtraction for visual surveillance applications. *Journal of Visual Communication and Image Representation*, 56:52–72, 2018.

[110] Karishma Pawar and Vahida Attar. Deep learning based detection and localization of road accidents from traffic surveillance videos. *ICT Express*, 8(3):379–387, 2022.

[111] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 27-30, 2016, Las Vegas, NV, USA.

[112] Hung Ngoc Phan, Long Hoang Pham, Nhat Minh Chung, and Synh Viet-Uyen Ha. Improved shadow removal algorithm for vehicle classification in traffic surveillance system. In *International Conference on Computing and Communication Technologies (RIVF)*, October 14-15, 2020, Ho Chi Minh City, Vietnam.

[113] Andrea Prati, Ivana Mikic, Mohan M Trivedi, and Rita Cucchiara. Detecting moving shadows: Algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):918–923, 2003.

[114] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 16-20, 2019, Long Beach, CA, USA.

[115] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE conference on computer vision and pattern recognition (CVPR)*, June 27-30, 2016, Las Vegas, NV, USA.

[116] Jianqiang Ren. Detecting and positioning of traffic incidents via video-based analysis of traffic states in a road segment. *IET Intelligent Transport Systems*, 10:428–437, 2016.

[117] Mosin Russell, Ju Jia Zou, and Gu Fang. Real-time vehicle shadow detection. *Electronics Letters*, 51(16):1253–1255, 2015.

[118] Mosin Russell, Ju Jia Zou, and Gu Fang. An evaluation of moving shadow detection techniques. *Computational Visual Media*, 2(3):195–217, 2016.

[119] Stuart J Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. London, United Kingdom: Pearson, 2016.

[120] Andres Sanin, Conrad Sanderson, and Brian C Lovell. Shadow detection: A survey and comparative evaluation of recent methods. *Pattern Recognition*, 45(4):1684–1695, 2012.

[121] Marcelo Santos, Marcelo Linder, Leizer Schnitman, Urbano Nunes, and Luciano Oliveira. Learning to segment roads for traffic analysis in urban images. In *IEEE Intelligent Vehicles Symposium (IV)*, June 23-26, 2013, Gold Coast City, Australia.

[122] Ajmal Shahbaz, Joko Hariyono, and Kang-Hyun Jo. Evaluation of background subtraction algorithms for video surveillance. In *Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, January 28-30, 2015, Mokpo, South Korea.

[123] Hang Shi, Hadi Ghahremannezhad, and Chengjun Liu. Anomalous driving detection for traffic surveillance video analysis. In *IEEE International Conference on Imaging Systems and Techniques (IST)*, August 24-26, 2021, Kaohsiung, Taiwan.

[124] Hang Shi and Chengjun Liu. A new cast shadow detection method for traffic surveillance video analysis using color and statistical modeling. *Image and Vision Computing*, 94:103863, 2020.

[125] Hang Shi and Chengjun Liu. A new foreground segmentation method for video analysis in different color spaces. In *International Conference on Pattern Recognition (ICPR)*, August 20-24, 2018, Beijing, China.

[126] Hang Shi and Chengjun Liu. Moving cast shadow detection in video based on new chromatic criteria and statistical modeling. In *IEEE International Conference On Machine Learning And Applications (ICMLA)*, December 16-19, 2019, Boca Raton, FL, USA.

[127] Dinesh Singh and Chalavadi Krishna Mohan. Deep spatio-temporal representation for detection of road accidents using stacked autoencoder. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):879–887, 2018.

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

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

[130] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin. A self-adjusting approach to change detection based on background word consensus. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, January 5-9, 2015, Waikoloa, HI, USA.

[131] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 23-25, 1999, Ft. Collins, CO, USA.

[132] Maryam Sultana, Arif Mahmood, and Soon Ki Jung. Unsupervised moving object segmentation using background subtraction and optimal adversarial noise sample search. *Pattern Recognition*, 129:108719, 2022.

[133] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[134] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 21-26, 2017, Honolulu, HI, USA.

[135] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, June 18-22, 2018, Salt Lake City, UT, USA.

[136] Sinnu Susan Thomas, Sumana Gupta, and Venkatesh K Subramanian. Event detection on roads using perceptual video summarization. *IEEE Transactions on Intelligent Transportation Systems*, 19(9):2944–2954, 2017.

[137] Carlo Tomasi and Takeo Kanade. Detection and tracking of point. *International Journal of Computer Vision*, 9:137–154, 1991.

[138] Guofeng Tong, Yong Li, Anan Sun, and Yuebin Wang. Shadow effect weakening based on intrinsic image extraction with effective projection of logarithmic domain for road scene. *Signal, Image and Video Processing*, pages 1–9, 2019.

[139] Daniel Toth, Ingo Stuke, Andreas Wagner, and Til Aach. Detection of moving shadows using mean shift clustering and a significance test. In *International Conference on Pattern Recognition (ICPR)*, August 23-26, 2004, Cambridge, UK.

[140] Li-Wu Tsai, Yee-Choy Chean, Chien-Peng Ho, Hui-Zhen Gu, and Suh-Yin Lee. Multi-lane detection and road traffic congestion classification for intelligent transportation system. *Energy Procedia*, 13:3174–3182, 2011.

[141] Tomas F Yago Vicente, Minh Hoai, and Dimitris Samaras. Leave-one-out kernel optimization for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):682–695, 2017.

[142] Shaohua Wan, Songtao Ding, and Chen Chen. Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles. *Pattern Recognition*, 121:108146, 2022.

[143] Bingshu Wang, Yong Zhao, and CL Philip Chen. Moving cast shadows segmentation using illumination invariant feature. *IEEE Transactions on Multimedia*, 22(9):2221–2233, 2019.

[144] Chen Wang, Yulu Dai, Wei Zhou, and Yifei Geng. A vision-based video crash detection framework for mixed traffic flow environment considering low-visibility condition. *Journal of Advanced Transportation*, 2020.

[145] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 19-25, 2021, Virtual.

[146] Ende Wang, Yong Li, Anan Sun, Huashuai Gao, Jingchao Yang, and Zheng Fang. Road detection based on illuminant invariance and quadratic estimation. *Optik*, 185:672–684, 2019.

[147] Jun Wang, Tao Mei, Bin Kong, and Hu Wei. An approach of lane detection based on inverse perspective mapping. In *International IEEE Conference on Intelligent Transportation Systems (ITSC)*, October 8-11, 2014, Qingdao, China.

[148] Tianyu Wang, Xiaowei Hu, Chi-Wing Fu, and Pheng-Ann Heng. Single-stage instance shadow detection with bidirectional relation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 19-25, 2021, Virtual.

[149] Yang Wang. Real-time moving vehicle detection with cast shadow removal in video based on conditional random field. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(3):437–441, 2009.

[150] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. Cdnet 2014: an expanded change detection benchmark dataset. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, June 23-28, 2014, Columbus, OH, USA.

[151] Ben G Weinstein. A computer vision for animal ecology. *Journal of Animal Ecology*, 87(3):533–545, 2018.

[152] 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*, 19(7):780–785, 1997.

[153] Jie Wu, Xionghui Wang, Xuefeng Xiao, and Yitong Wang. Box-level tube tracking and refinement for vehicles anomaly detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 19-25, 2021, Virtual.

[154] Siyu Xia, Jian Xiong, Ying Liu, and Gang Li. Vision-based traffic accident detection using matrix approximation. In *Asian Control Conference (ASCC)*, May 31-June 3, 2015, Kota Kinabalu, Malaysia.

[155] Yingji Xia, Zhe Sun, Andre Tok, and Stephen Ritchie. A dense background representation method for traffic surveillance based on roadside lidar. *Optics and Lasers in Engineering*, 152:106982, 2022.

[156] Deliang Xiang, Wei Wang, Tao Tang, Dongdong Guan, Sinong Quan, Tao Liu, and Yi Su. Adaptive statistical superpixel merging with edge penalty for polsar image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 58(4):2412–2429, 2019.

[157] Yan Xu, Xi Ouyang, Yu Cheng, Shining Yu, Lin Xiong, Choon-Ching Ng, Sugiri Pranata, Shengmei Shen, and Junliang Xing. Dual-mode vehicle motion pattern learning for high performance road traffic anomaly detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, June 18-22, 2018, Salt Lake City, UT, USA.

[158] Guoan Yang, Yuhao Wang, Junjie Yang, and Zhengzhi Lu. Fast and robust vanishing point detection using contourlet texture detector for unstructured road. *IEEE Access*, 7:139358–139367, 2019.

[159] Honghong Yang and Shiru Qu. Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition. *IET Intelligent Transport Systems*, 12(1):75–85, 2018.

[160] M-T Yang, K-H Lo, C-C Chiang, and W-K Tai. Moving cast shadow detection by exploiting multiple cues. *IET Image Processing*, 2(2):95–104, 2008.

[161] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 16-20, 2019, Long Beach, CA, USA.

[162] Mehran Yazdi and Thierry Bouwmans. New trends on moving object detection in video images captured by a moving camera: A survey. *Computer Science Review*, 28:157–177, 2018.

[163] Yuan Yuan, Zhiyu Jiang, and Qi Wang. Video-based road detection via online structural learning. *Neurocomputing*, 168:336–347, 2015.

[164] Kimin Yun and Jin Young Choi. Robust and fast moving object detection in a non-stationary camera via foreground probability based sampling. In *IEEE International Conference on Image Processing (ICIP)*, September 27-30, 2015, Quebec City, QC, Canada.

[165] Kimin Yun, Hyungil Kim, Kangmin Bae, and Jongyoul Park. Unsupervised moving object detection through background models for ptz camera. In *International Conference on Pattern Recognition (ICPR)*, January 10-15, 2021, Milan, Italy.

[166] Kimin Yun, Jongin Lim, and Jin Young Choi. Scene conditional background update for moving object detection in a moving camera. *Pattern Recognition Letters*, 88:57–63, 2017.

[167] Xingchen Zhang, Yuxiang Feng, Panagiotis Angeloudis, and Yiannis Demiris. Monocular visual traffic surveillance: A review. *IEEE Trans. Intell. Transp. Syst.*, 23(9):14148–14165, 2022.

[168] Yindan Zhang, Gang Chen, Jelena Vukomanovic, Kunwar K Singh, Yong Liu, Samuel Holden, and Ross K Meentemeyer. Recurrent shadow attention model (rsam) for shadow removal in high-resolution urban land-cover mapping. *Remote Sensing of Environment*, 247:111945, 2020.

[169] Yuxiang Zhao, Wenhao Wu, Yue He, Yingying Li, Xiao Tan, and Shifeng Chen. Good practices and a strong baseline for traffic anomaly detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 19-25, 2021, Virtual.

[170] Lei Zhu, Zijun Deng, Xiaowei Hu, Chi-Wing Fu, Xuemiao Xu, Jing Qin, and Pheng-Ann Heng. Bidirectional feature pyramid network with recurrent attention residual modules for shadow detection. In *European Conference on Computer Vision (ECCV)*, September 8-14, 2018, Munich, Germany.

[171] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *International Conference on Pattern Recognition (ICPR)*, August 23-26, 2004, Cambridge, UK.

[172] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *International Conference on Pattern Recognition (ICPR)*, August 23-26, 2004, Cambridge, UK.

[173] Zoran Zivkovic and Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2006.

[174] Hue Zu, Yaohua Xie, Lu Ma, and Jiansheng Fu. Vision-based real-time traffic accident detection. In *World Congress on Intelligent Control and Automation (WCICA)*, June 29- July 4, 2014, Shenyang, China.