# NOTE TO USERS

Université d'Ottawa • University of Ottawa

# Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Yi LIU

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M. Sc. (Systems Science)

GRADE - DEGREE

Systems Science

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Movement Detection in Outdoor Scenes for Traffic Monitoring

P. Payeur

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

B. Cameron

CO-DIRECTEUR DE LA THÈSE - THESIS CO-SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

A. Adler

E. Dubois

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

# MOVEMENT DETECTION IN OUTDOOR SCENES FOR TRAFFIC MONITORING

Yi Liu

A Thesis submitted to the Faculty of Graduate and Postdoctoral Studies in partial fulfillment of the requirements for the degree of Master of Science, System Science

System Science
School of Management
University of Ottawa
Ottawa, Ontario, Canada

*Your file Votre référence*
*ISBN: 0-494-01533-0*
*Our file Notre référence*
*ISBN: 0-494-01533-0*

*To my parents and brother*

# Contents

# List of Figures

# Abstract

One important application of image processing and computer vision is traffic monitoring and control. In this thesis, a system for detection of moving vehicles approaching an intersection from a sequence of color images acquired by a stationary camera in the context of traffic light control systems is presented. As the system is dedicated to outdoor applications, efficient and robust vehicle detection under various weather and illumination conditions is examined. To deal with these ever changing conditions, vehicle detection relies on motion segmentation and on an original algorithm using color mapping to achieve feature space segmentation. Experimental results on real outdoor sequences of images demonstrate the system's robustness under various environmental conditions.

# Acknowledgements

First, I would like to thank my supervisor, Dr. Pierre Payeur with my heart for his valuable advice and guidance throughout my graduate study.

I would like to thank defence committee members for being the readers of this thesis and for their insightful comments and suggestions.

I would like to express my sincere gratitude to VIVA laboratory members for all their friendships and great help during my school life; when I encountered difficulties, they were always there for me.

It has been an honor for me be a member of the VIVA laboratory at the University of Ottawa, which has provided me a great opportunity for research and development as well as personal growth in terms of patience, friendships and support.

This thesis is dedicated to my dear parents, who make my life full of love and give me courage to overcome any difficulties.

# Chapter 1

# Introduction

In the automobile age, traffic congestion on our major roads has become a fact of life. Over the last decade, there has been considerable research work performed on traffic monitoring systems, which aim at optimizing traffic to maximize the efficiency of existing roads. Applications of computer vision in intelligent transportation systems(ITS) are now gaining in popularity. Vision-based systems for ITS are adopted for improving the external environment understanding using either static cameras mounted in fixed positions or moving cameras. By analyzing a variety of real-time traffic data collected from different kinds of sensors, it becomes possible to monitor, understand and eventually control activities in large, complex and spatially distributed areas.

## 1.1 Traffic Monitoring and Sensors

Traffic monitoring involves the collection of data describing the characteristics of vehicles and their movement on road networks. Vehicle counts, vehicle speed, vehicle path and vehicle density are all examples of useful data. Currently the main types of detectors or sensors used in road-traffic monitoring can be classified in two categories: intrusive and non-intrusive [30].

Intrusive detectors such as loop detectors are installed under the pavement surface of a roadway and detect the presence of vehicles as they pass over the sensor zone.

Loop detectors consist of several wires buried in the roadway, and they are connected to an electronic circuit which can detect a car passing over it.

Non-intrusive detectors are not embedded in the pavement and are usually located overhead or on the side of a roadway. Passive or active infrared, radar, microwave and video camera all belong to the non-intrusive technology group. Their main advantage is that there is no need to dig in the road to install them. For example, active infrared devices detect the presence of vehicles by emitting low-energy laser beams at the road surface and measuring the time for the reflected signal to return to the device. The presence of a vehicle is measured by the corresponding reduction in time for the signal return.

Among the great number of technologies that are available, loop detectors are the well-known mature technology which are widely used in traffic light systems. Although the inductive loop detectors have the advantage of being very accurate when operating correctly, they have several disadvantages over non-intrusive sensors. As their installation disrupts traffic, they cannot be constructed during high traffic day time. Moreover, they are hard to maintain, once installed they may not be easily moved. Also such an installation may weaken the pavement. A critical weakness of loop detectors is their limitation to fixed detection zones, since they are only capable of detecting vehicles directly overhead.

## 1.2 Traffic Monitoring and Computer Vision

Computer vision is a branch of artificial intelligence and image processing. It is concerned with computer processing of images from the real world. Typically image processing consists of two levels. At the low level, image enhancement is performed by noise filtering, contrast adjustment, etc. At the higher level, pattern recognition and image understanding are applied to recognize features present in the images. Over the last decades, computer vision has been widely applied to traffic monitoring and control areas. A typical vision system for road traffic monitoring can be depicted as in figure

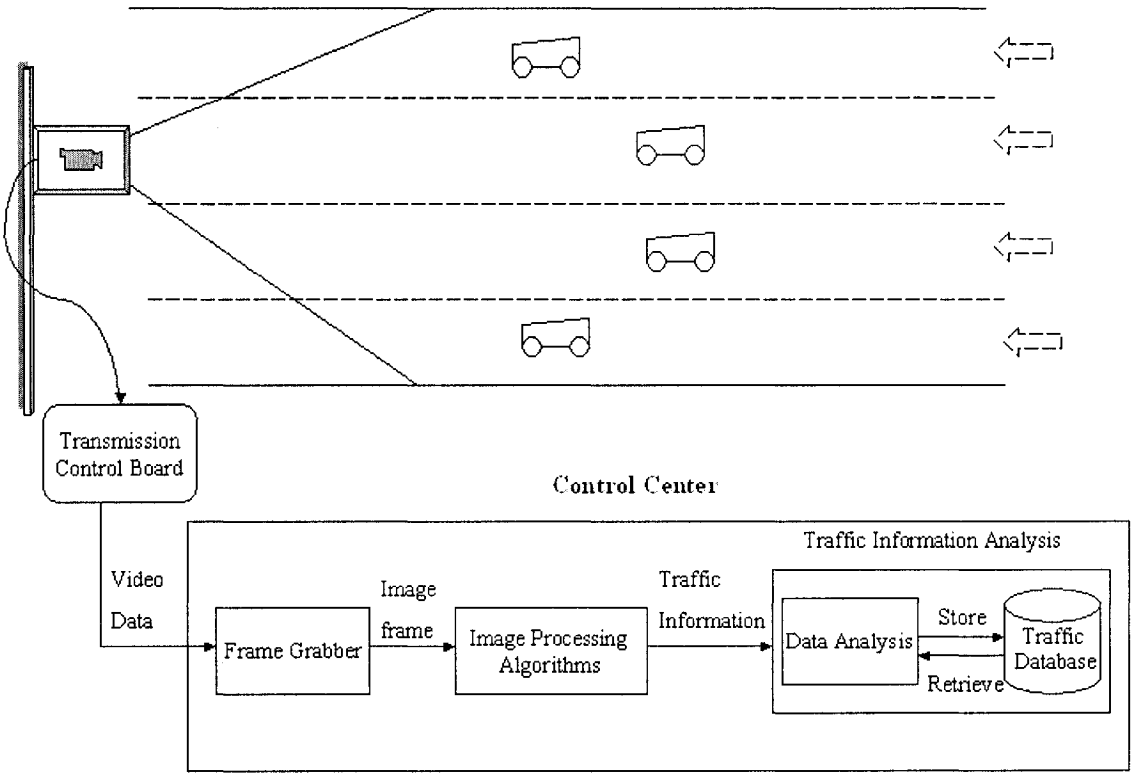



Figure 1.1: A typical vision system for road-traffic monitoring.

1.1. The video camera placed on the pole takes a continuous real-time video signal. It is transmitted to a control center via a cabled or a wireless network. In the control center, images are digitized by a frame grabber and each image frame is processed to perform vehicle detection, tracking, classification or identification to provide desired traffic information. Furthermore, a traffic information analysis module deals with these information and store them in the traffic database for backup or later retrieval usage.

Computer vision based monitoring systems allows to overcome many of the disadvantages associated with loop detectors. First, installing loop detectors on moderate or high volume roadways results in traffic disruption. Temporary closure of traffic lanes is needed to ensure the safety of installation personnel. Video cameras can be installed on any road or highway with minimal disruption. Second, a single camera is able to monitor more than one traffic lane as well as several hundred meters of road, whereas the area monitored by loop detectors is very limited. Third, compared with

loop detectors, vision-based systems may extract a much richer variety of information such as vehicle shape, vehicle path and color. A vision system could theoretically have the same capability of observation as a human observer but without the detrimental effects of tiredness and boredom. But in spite of their numerous advantages, video detectors still have shortcomings that need to be overcome. Among other things, the contrast and clarity of images tend to be degraded by inclement weather. For example, mist or fog lower the visibility of moving objects in the distance. Moreover, shadows, reflections from wet pavement and day-night transitions can result in missed or false detections. In addition, large vehicles can occlude smaller vehicles, which leads to vehicles' miscounting and misclassification.

One of the main unsolved issues for existing vision based vehicle detection systems is their lack of robustness to various weather conditions. In this thesis, we address this problem and introduce promising solutions for moving vehicles detection under different weather and luminance conditions.

## 1.3 Objectives

Applications of computer vision in traffic monitoring are usually composed of moving object detection, tracking and matching components. However, achieving reliable and effective moving object detection in natural environments is a critical issue that influences the performance of the entire system. Here, a typical problem arises from the complexity of changes occurring in the real-world which result from variations in environmental conditions. Lighting conditions such as the time of the day, the sun position, shadows and weather conditions, all have a direct impact on the way useful information can be extracted from the images. How to detect moving vehicles efficiently and robustly in various natural environments remains a critical issue. In this thesis, the main objective is to achieve robust detection of moving vehicles under different weather and luminance conditions in outdoor scenes. The end result of this effort is a set of algorithms that are expected to enhance the efficiency of real-time

vision-based motion detection systems.

## 1.4 Thesis Overview

In the thesis, we propose an adaptive and alternative system for detecting moving vehicles in sequences of color images acquired by a stationary camera. The proposed approach is developed in the context of a traffic light control system. The processing scheme is a combination of motion detection with a color based switching algorithm that relies on features detection and matching. Experimental results obtained on real outdoor scenes demonstrate the system's robustness under many difficult situations, such as a snowy night and rainy weather. This thesis describes how computer vision techniques have been used to achieve robust moving vehicles detection in natural outdoor environments. The following chapters are structured as follows:

- Chapter 2 presents a literature review of general moving objects detection as well as an overview of specific algorithms or techniques proposed for outdoor applications.
- Chapter 3 details the proposed system structure for moving vehicles detection in changing environmental conditions.
- Chapter 4 presents experimental results and an evaluation of the system performance. A discussion of the main constraints and possible applications of this system is also provided.
- Chapter 5 concludes on the work that has been performed, summarizes the contributions and examines possible refinements and extensions.

# Chapter 2

# Literature Review

Vehicle detection systems usually perform three key operations: image acquisition, image segmentation and vehicle identification. The image acquisition task consists of collecting digital images of the scene of interest. The goal of image segmentation is to isolate meaningful objects from the digital images. Even though segmentation is a critical step in any image processing operation, it still remains a difficult task to perform. The vehicle identification procedure classifies and labels a moving object as a vehicle if it meets specific requirements on the size and characteristics. This component relies heavily on the quality of the image segmentation process.

A large body of literature exists about moving object classification, which is usually composed of two steps: motion detection and object recognition. The following sections examine each of them and summarize the current state of the technology.

## 2.1 Motion Detection

The process of motion detection refers to the procedure of labelling pixels which are associated with different coherently moving objects or regions. It is an extremely important step of image processing for moving object detection. Its accuracy influences the eventual success rate of other computerized analysis procedures. In the literature,

many standard algorithms have been proposed for detecting moving points, with various performance levels and computation complexity. These can be classified in two categories: non-model based methods and model-based methods.

### 2.1.1 Non-Model Based Methods

Different non-model based techniques have been applied in traffic monitoring systems. Their aim is to extract the moving regions from the images. After efficiently exploiting suitable characteristics of these regions, one can perform vehicle classification, vehicle count or tracking as desired. Among all the various methods, frame differentiation, feature-based motion detection and optical flow segmentation are the most frequently used.

#### 2.1.1.1 Frame differentiation

As there are usually some differences between two frames that are taken at different times or that have different background references, frame differentiation is able to reveal regions of motion by subtracting the two frames. In the literature, many different approaches have been proposed that include temporal differentiation [1, 2], double frame difference [3], and background suppression [4].

Temporal differentiation refers to operate a pixel-wise difference between two images taken at different times. The resulting difference image is then thresholded at a certain value to highlight large variations in pixel intensity over that period.

Double frame differentiation requires that three consecutive frames $f1$, $f2$ and $f3$ are selected from the image sequence for computation. The procedure performs two pixel-wise differences respectively between $f1$ and $f2$ and between $f2$ and $f3$. The final difference image results from a logical AND operation on these two temporal differences. Using this technique, the edges belonging to the moving object in the middle frame are highlighted.

Background subtraction refers to the comparison of each pixel of the current image with a set of pixels in the neighborhood of the corresponding pixel in the background

image. This comparison usually involves subtracting the intensity value of a pixel in the current image and the intensity value of the corresponding pixel in the background image to determine if the pixel in the current image represents a part of a foreground moving object. The ideal background image is merely an image of the scene without moving objects (here vehicles). If there is no vehicles in the current frame, then no difference exists between the current image and the background image. On the contrary, if the current image does contain vehicles, the difference image turns out to be distinct.

The temporal differentiation technique is rarely used when monitoring traffic since it has the tendency to create several disconnected blobs for a single moving vehicle. On the other hand, the background subtraction method is the most popular for traffic applications since it tends to preserve the integrity of the resulting blob. For this reason, this approach has been examined in more details. In order to be adaptive with the real changes in the environment, the background suppression approach requires a computationally expensive background update procedure. The development of a background estimation technique requires that several factors are taken into consideration. The main ones are the changes in lighting conditions, the changes in the content of the background, the occurrence of shadows and small movements experienced by the camera. Many research works proposed different ways to perform the background update procedure. Here is a summary of these approaches.

Oha *et al.* [4] perform a background update by using statistical functions on a representation of the most recent sampled frames. They first define mathematical models representing the relationship between the illumination intensity, a reflection index of objects and a pixel value. They also mathematically define an assumption about the illumination which requires that the distribution of the illumination intensity in a small region does not change. Then it formalizes the background subtraction problem as a statistical test based on the models and assumption. Apart from its mathematical complexity, this approach requires images of a large dimension to correctly estimate the background model.

Ridder *et al.* [5] apply Kalman filter theory to the background estimation problem. It presents an algorithm where the process of gray value variations in a background image sequence are described as signal parameters (for each pixel). The system state is represented by the gray value difference in addition to the first derivative. The Kalman filter makes their system more robust to lighting changes in the scene. While this method does have a pixel-wise automatic threshold, it performs a slow adaption on the background sequence and does not properly handle bimodal backgrounds.

Cucchiara *et al.* [6] define a specific approach based on background subtraction with a statistical and knowledge-based background update. It combines both a statistical update and a selective update that computes a new background value only if a point is not marked as a motion point. The advantage of this approach is that it can reduce the number of false detections. This problem arises whenever moving objects are stopped for a long time and become part of the background. When these objects start to move again, a false detection occurs in the area where they were stopped.

Lai *et al.* [7] apply running mode and running average algorithms, which estimate for each pixel in the image the gray scale intensity of the background from a sequence of input images. The running mode technique calculates the most frequently occurring intensity values among a group of images. It is reputed to be quite accurate but its computing speed is very slow and therefore cannot be considered for real-time applications. On the other hand, by taking the average of intensity values on a set of subsequent images, the running average approach is faster but its estimation accuracy is lower.

Kamijo *et al.* [8] compose background images from image sequences captured during a certain period of time. It builds histograms of intensities at each pixel along an image sequence, and then determines the intensity that has a maximum frequency in the histograms and uses this value as a background intensity at each pixel. This background composition can detect slower variations in the image sequence and finds interesting application for detecting vehicles on parking lots.

Grimson *et al.* [9] model each pixel in the background image as a mixture of Gaussian distributions to cope with small and frequent luminance variations, typical for instance of tree leaves swaying under windy conditions. Based on the persistence and the variance of each of the Gaussians of the mixture, it determines which Gaussian may best represent background colors. Pixel values that do not fit with the background distributions are considered foreground until there is a Gaussian that includes them with sufficient and consistent evidence. Unfortunately, this approach cannot be used to reconstruct the background image when the illumination level changes unevenly over the image's pixels.

Tsuchikawa *et al.* [15] reconstruct a background image by using statistical characteristic analysis of temporal changes of a target pixel. Each target pixel is judged based on stored images collected over the last few seconds. Different modes are considered: slow illumination change, rapid illumination change and moving object passing by. The background image is reconstructed based on each mode. The advantage of this approach is that it can adapt the background under drastic illumination changes. However, as it has been successfully tested only in indoors environments, its capability to deal with the complexity of outdoor scenes is not demonstrated.

Each of these methods is developed to dynamically update the reference background frame so that it adapts to changes in the scene illumination. Many of the previously mentioned background models are based on statistical modelling of pixel intensity. The model adaptation integrated in these techniques makes it possible to adapt to gradual changes in illumination. However, apart for their heavy computational workload, sudden changes in illumination still represent a challenge for such models.

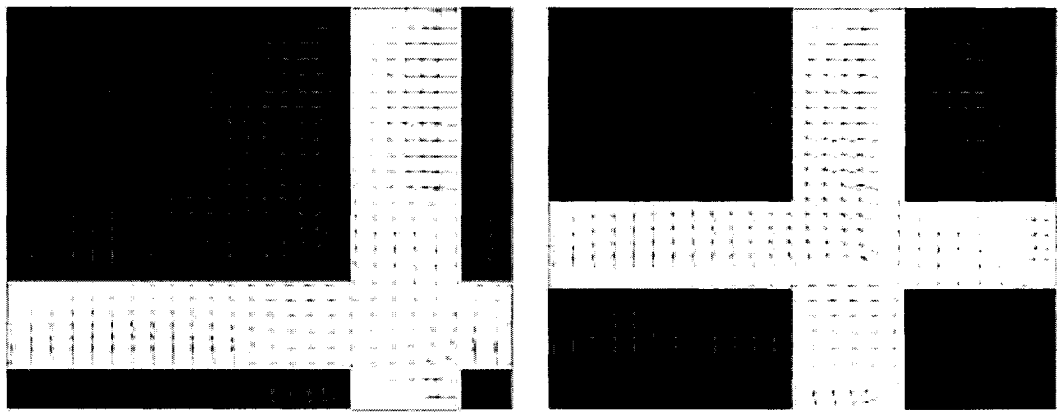
Other techniques use statistical functions computed on a sequence of previous background representations and on the current frames. These techniques also offer good performance while requiring a limited computational workload compared with statistical models.

#### 2.1.1.2 Feature-based motion detection

An alternative approach to locate objects of interest abandons the idea of detecting objects as a whole and rather concentrates on sub-features such as distinguishable points or lines on the object [10]. Feature-based techniques start with a feature extraction step. From these corresponding features, three-dimensional motion parameters are derivable. The advantage of such an approach is that even in the presence of partial occlusion, some of the features of the moving object remain visible. On the other hand, it introduces a new problem of grouping the features that belong to the same object. Huang *et al.* [29] present a review of these algorithms and their performance for determining the three-dimensional motion and structure of rigid objects when their corresponding features are known at different times or are viewed by different cameras.

#### 2.1.1.3 Optical flow segmentation

To detect objects motion, optical flow techniques can also be utilized. The optical flow represents the distribution of velocity with respect to an object over the points in an image. Optical flow is determined by computing the motion vector of each pixel in an image. As an example, the motion fields generated by the displacement of a white cross are shown in figure 2.1. Detecting these fields helps in separating the changes introduced due to the cross movements. As seen in figure 2.1, most of the motion vectors between $(a)$ and $(b)$ are due to the displacement of the cross in a single direction. Some disparity in the direction of some of the motion vectors results from noise. Watanabe *et al.* [11] present a method which can effectively recognize moving objects by analyzing optical flow information acquired from dynamic images. This method can work under various disturbances by accumulating information over space and time, but it appears not to be suitable in complex and unstructured environments such as outdoor scenes. Moreover the computation of the optical flow is generally very time consuming.



(a) (b)

Figure 2.1: Optical flow example.

### 2.1.2 Model-Based Methods

The basic idea of model-based methods is to classify each pixel using a model of how that pixel looks when it is part of different classes.

Friedman *et al.* [12] propose a probabilistic approach applied to vehicles detection called Expectation Maximization (EM) for a mixture Gaussian classification model for each pixel. Their method attempts to explicitly classify the pixel values into three separate, predetermined distributions corresponding to the road color, the shadow color and other colors corresponding to vehicles. This makes their work somewhat restricted to such scenario, although the method can probably be applied to other ones.

Fuh *et al.* [26] present a model for detecting the displacement field in spatio-temporal images that allows for affine shape deformations of corresponding spatial regions and for affine transformations of the image intensity range. This model includes the block matching method, which uses a full search method to exhaustively evaluate all possible candidate pixels over a predetermined search window in order to find the optimum. In addition, a least-square algorithm is used to find the model parameters.

In the model-based approach, each model is instantiated when its parameters are

defined within a domain of possible values. Usually this kind of approach may reach high accuracy at the cost of high computation. For this reason, it can hardly be used in real-time applications.

## 2.2 Object Recognition

The issues of object recognition and matching are other important areas of research in computer vision. Recognition and matching are usually performed by comparing some compact description of an object derived from sensed images with its model, or with another object. There are three main methods found in the literature to perform this task [28]. These are template matching, feature-based recognition and model-based object recognition.

### 2.2.1 Template matching

Template matching is a technique used to isolate certain features in an image. These features can be single pixels, lines, edges or complete objects. In general, the approach consists of convolving an image with a template window to produce a correlation estimate between the image and that template. The result is then an image with high values where there is a strong correlation (that is, where the template matches a part of the image) and low values elsewhere. Template matching is often used with some kind of matching criterion, usually under the form of a threshold that is applied to the image resulting from the convolution between an image and a template. This threshold ensures that features are detected only if they result in a large response in the convolved image. Apart from requiring much computing power, this method is not well suited to the detection of asymmetric or complex objects.

### 2.2.2 Feature-based recognition

These techniques extract significant features (points, edges, clusters, structures, etc) from the input images, which represent the main characteristics of the objects. These

features are then compared to a database of feature descriptors for all possible input images. The best matching descriptor leads to recognition. Color space plays an important role in feature extraction. Numerous techniques use color as a cue to extract information from images by a specific color representation, and then an appropriate method analyzes the extracted color information to achieve the best possible result. The usage of color information and the combination of feature-based and correlation-based matching leads to robust and efficient algorithms.

### 2.2.3 Model-based object recognition

The model-based object recognition methods are more advanced as they strive to gain an understanding of the image. To achieve such an understanding, models are used to represent the available knowledge on the appearance of objects. There are two main approaches found in the literature to realize model-based object recognition. These are either defined in a parameter space or in a correspondence space.

Searching for objects in parameter space involves the use of a parameterized model of the object. This typically takes the form of 3D CAD or wire frame model. The model parameters specify the pose of the object in the three dimensional world. The presence of suitable image features in the vicinity of the projected model features can be used to produce a score. For example, if a projected model line happens to fall exactly over a straight line in the image then a high score would be returned. The model parameters which give rise to the best score represent the object pose which best fits the evidence in the image. As the score is a function of the model parameters, a $n$-dimensional space must be searched to find the maximum, where $n$ is the number of parameters. Unfortunately, for realistic problems the search space will contain many local optima and its optimization becomes extremely difficult. Moreover the location of the global optimum can never be guaranteed.

The alternative to searching pose space is to perform the search in a correspondence space. If the correspondence between image features and model features is known, a score function may be derived which reflects how close a projected model

feature is to its corresponding image feature. The problem of this approach is how to find the correct correspondence. An exhaustive search may be performed for small problems by exploring what is known as an Interpretation Tree. However, the size of the tree is exponential in the number of images and model features, therefore real problems must make use of methods to substantially prune the interpretation tree.

### 2.2.4 Review of object recognition application

This section reviews some of the proposed systems which adopt any of the three methods described above.

Dubuisson *et al.* [3] present a method which integrates a motion segmentation technique using image subtraction and a color segmentation technique based on the split-and-merge algorithm. Split and merge is an image segmentation procedure that initially subdivides an image into a set of arbitrary, disjoint regions and then merges and/or splits these regions in an attempt to reach the conditions that ensure that the final regions are homogeneous. It is a feature-based matching method. The split and merge algorithm extracts the contours of an object, which contains relevant information about its shape. The resulting contours can then be used to classify the objects in different generic categories.

Parodi *et al.* [10] propose a method for the interpretation of traffic scenes based on the detection and recognition of objects. It adopts a feature-based, top-down reconstruction and recognition of the 3D scene and of the main objects therein contained. Each of the different classes of objects that they expect to find in a typical scene is identified according to some selected features and only later their shape is reconstructed more thoroughly. This prevents the combinatorial explosion of the recognition of objects of parameterized shape which is typical of general model-based recognition schemes.

Dufaut *et al.* [13] propose to use pattern recognition associated to some elements of graph theory to recognize vehicles at night. The approach computes an area factor, a distance factor and a spacing factor from the object area and perimeter. Decision

functions are used to arrange these data in categories so that it distinguishes highlight bulbs from other shapes. Although it is effective in matching vehicles at night, the algorithm is complex and relies on perspective observation.

Gupte *et al.* [14] propose a system which first operates motion segmentation by background differentiation. The result of the segmentation step is a collection of connected regions. Then it recovers important vehicle parameters such as length, width and height. A vehicle identification stage groups regions together to form vehicles. Vehicles are modelled as rectangular patches with certain dynamic behavior whose size depend on the dimensions of its constituent regions.

Each of these methods is developed to recognize a specific object. Some of these methods rely on motion of certain body parts; others use the shape of the object. The criteria of choosing a suitable method for a specific application mostly depends on its accuracy and computation requirement. However, this brings important constraints on the diversity of objects that can be monitored.

## 2.3 Specific Algorithms Applied on Natural Environments

It is interesting to observe that in the literature, most part of the research work performed in the context of traffic monitoring has been targeted at motion detection under fair weather situations and clear lighting only. A limited number of projects have focused on very specific weather and luminance related issues such as the followings:

Under snowfall conditions, Hase *et al.* [16] propose an algorithm for noise reduction in an image by applying a median filter on the intensity of the successive images. The proposed approach succeeds to reduce the noise from the images for relatively low intensity snowfalls, but is limited to this specific weather condition.

Under heavy rainy conditions, Kyo *et al.* [17] propose to first divide the input image into several search windows. Then the approach applies a Robinson filter, that

is a template-based edge filter, in each window to extract edges of vehicles under low contrast conditions. The horizontal and vertical edge segments are detected by thresholding the Robinson edge filtering result and assigning a unique label to each connected component. This approach succeeds to detect and track various vehicles in both fair and bad weather conditions especially on heavy rainy scenes, but it has limitations under night conditions.

For daytime and nighttime, when illumination conditions differ, Cucchiara *et al.* [18] propose an approach for detecting vehicles in urban traffic scenes by using different detection modes according to lighting conditions (day/night) in order to classify vehicles by types. In these image processing modules, visual data is extracted from the scene by a spatio-temporal analysis during daytime and by a morphological analysis of headlights at night. At daytime, the visual data refers to blocks of pixels in motion corresponding to the identifiable vehicles in the scene. At night, the visual data refers to vehicle headlight pairs, which are the most salient features of vehicles under low illumination. This system generally succeeds to detect vehicles during the day but does not deal with other aspects such as the effect of shadows. Also it tends to lead to a misclassification of vehicles under noisy conditions such as snowfall or highly reflective wet surfaces.

Despite the relatively large number of image processing algorithms that have been published and that are dedicated to traffic and outdoor scene monitoring, no general methods have yet been found to efficiently deal with the wide diversity of images characteristics that are encountered in real outdoor applications. Most existing vision based vehicle detection systems still suffer from a lack of robustness in dealing with these various conditions [19, 20].

## 2.4 Existing Commercial Systems

A number of video detection products have been developed and used in real traffic world, such as the AUTOSCOPE$^{TM}$2003, the TrafficVision and the Vantage$^{TM}$

video detection systems. These products use different fore-end hardware devices to capture the real traffic data. For example, AUTOSCOPE$^{TM}$2003 uses zoom-lens color camera, TrafficVision makes use of video camera and Closed Circuit Television, and Vantage$^{TM}$ utilizes wireless camera to collect images from the roadway. But in spite of these differences, the main configuration of these products is quite similar. All of them configure a video image processor to perform all the image processing algorithms.

The product features of most of these companies claim that they can realize good and accurate detection of vehicles in outdoors lighting and weather conditions. Unfortunately, detailed technical descriptions of these systems and evidence of their robustness are extremely difficult to find. Therefore, it is hard to evaluate the actual performance of these products. Moreover, it seems that these systems have been developed under relatively good weather conditions. Whether they could operate under the extreme Canadian climate, like heavy snow, is not clearly demonstrated. According to the literature, no system has been found that can deal with such diversified outdoor conditions.

# Chapter 3

# Proposed System Structure

A close analysis of images collected on real traffic scenes under various environmental conditions such as sun, clouds, rain, snow and night lighting, allowed to put in evidence important factors that influence the detection of moving objects. Especially the perception contrast appeared to be critical as it directly influences the area of objects that is visible. For example, under sunny conditions, the visual perception contrast is high and the complete surface of moving vehicles can be extracted. However, at night or under heavy snow or rain conditions, the perception contrast is lower and only vehicles' headlights appear as clear features in the image.

Under low contrast conditions, traffic images are generally characterized by the following elements:

- Bright blobs with a regular shape representing the headlights of the vehicles.
- Blobs with an irregular shape resulting from the reflection on the ground of the illumination originating from the headlights of the vehicles.

Therefore, vehicle detection at night and under other low contrasting situations must rely on the extraction of some representative blobs among all blobs that are present in the scene, as well as on the matching between these representative blobs and the moving objects in the scene. Following the Canadian motor vehicle safety standards (CMVSS) 108 issued in 1993, all new manufactured vehicles are required to be

equipped with two or more headlights that must be equally distributed on each side of the front side of the vehicles. Under this assumption, we may consider headlights as a reliable feature for detecting vehicles under low contrast conditions.

Since the relevant features under various environmental conditions significantly differ, designing a single algorithm to cover all cases does not seem suitable. Meanwhile, it has been observed that all weather and lighting conditions can be grouped in two categories under a proper mapping of color information. The proposed approach then relies on two slightly different algorithms that work in parallel to extract different features from moving regions and on a switching criterion to automatically select the most appropriate mode given the perception contrast achieved in the current set of images. Figure 3.1 shows the structure of the entire proposed system.

While setting up the system, one or two suitably sized windows are defined on the area where the lanes of the road are located in the image. Assuming a fixed viewpoint for the camera, these windows can be kept in a constant position. Defining such windows of interest on the image does not only emphasize the key observation area but also significantly reduces computation time, which is critical for real-time applications such as traffic monitoring. The search for vehicles is then performed only in the regions delimited by the windows of interest.

The first stage of the system operates motion detection and segmentation on the images to detect regions of interest inside the search window. Motion detection components consist of background differentiation, ghost detection, segment filtering and adaptive background update modules. The second stage consists of a dual processing scheme that refines the detected motion parameters following different strategies that depend on the perceptual contrast level observed in the current set of images. A switching algorithm is implemented to automatically select the best mode of operation.

The analysis of color images revealed that the Hue-Saturation-Value (HSV) color

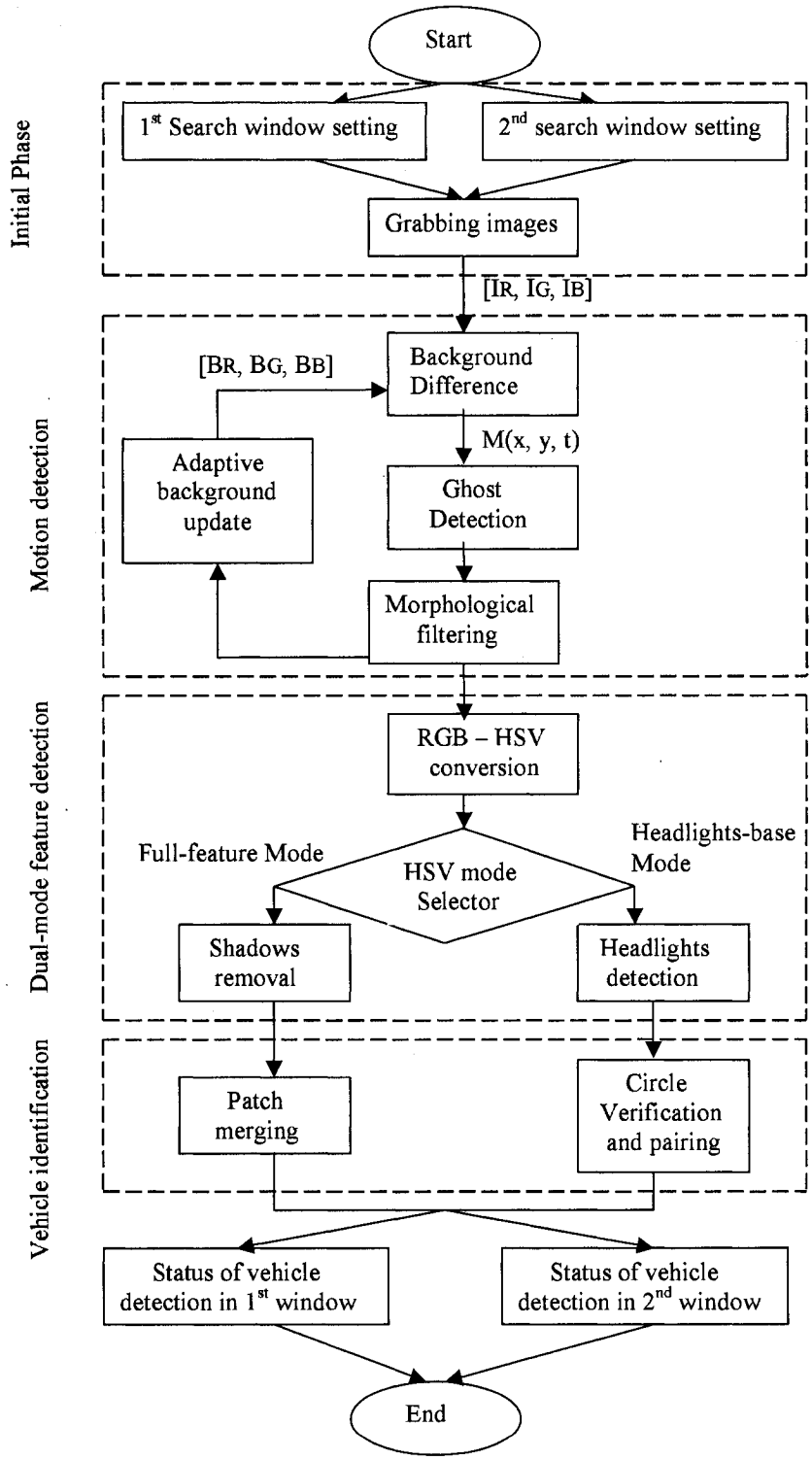



Figure 3.1: System Structure.

space can advantageously be used to distinguish between different weather and illumination conditions. Moveover, under highly contrasting conditions, shadows associated with moving objects must be analyzed separately as they tend to be merged with foreground objects. In nighttime conditions and low perceptual contrast, foreground objects must be associated with vehicles' headlights. Different verification methods are then used to detect and match the relevant features while the system is able to automatically adapt to the current environmental conditions. Each module in the system is described in details in the following sections. Selected images extracted from the set of sequences that will be described in chapter 4 are presented to illustrate the performance of each module.

## 3.1 Initial Phase

In our experimental setup, the system works on sequences of images previously recorded by a fixed video camera installed above the road to observe a proper area of the traffic scene. In the initial phase, some preprocessing is applied on the sequence to:

- define the detection windows on the areas of interest in the scene;
- compute the geometrical properties of the windows;
- convert the original analog signal to digital images on which image processing will be applied.

In order to speed up computation and to meet the requirement for real-time operation, some data reduction is performed spatially and temporally. Spatial data reduction involves processing only small portions of each image that correspond to regions of interest. In a typical traffic scene, much of the image is of little interest as it contains buildings, vegetation or sidewalks. These areas are unlikely to contain a vehicle. It is therefore useless to spend processing time on them. As shown in figure 3.2, two windows are set along the road, each having an area that corresponds to

the space we are interested in for traffic flow observation. Temporal data reduction is achieved by only processing every $n^{th}$ frame in the traffic sequence. As the frame grabbing speed is usually faster than the processing speed, this allows to circumvent the cases where motion difference between two successive frames is very small and could hardly be detected. After setting the interest windows, we compute the window positions in the image and obtain the windows' widths and heights. These parameters remain constant as we assume a fixed viewpoint for the camera.

Finally, images are digitized with a Matrox frame grabber board. Since all the data the system will process are based on the digital signal format, it is necessary to convert the original analog images into digital images.

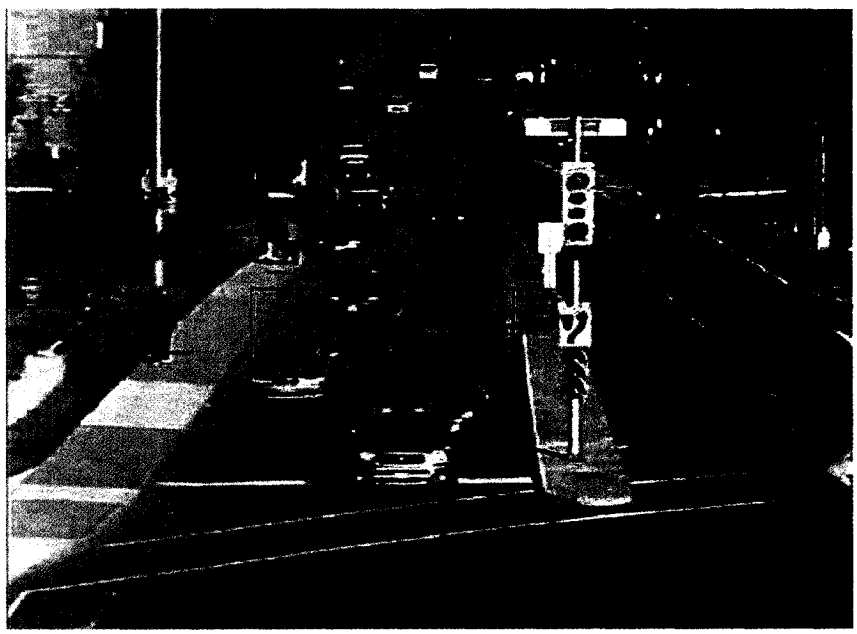


Figure 3.2: Definition of windows of interest.

## 3.2 Motion Detection

One of the crucial elements of a vehicle detection system is the motion analysis component which segments moving objects from an image. As discussed previously, motion segmentation can be based on various approaches like feature detection, frame differentiation, optical flow extraction, etc. Each of these methods has its benefits and its drawbacks. The goal of this thesis being to achieve high robustness to environmental factors (weather and lighting) with high performance and a low cost of implementation, the adaptive background suppression technique has been selected. It is important to observe that in the proposed approach, no initial background image is required in opposition to most techniques found in the literature that use background suppression and require a vehicle free background image to initialize the procedure.

The color images that are provided by the camera are encoded following the RGB mapping scheme with each component (red-green-blue) represented separately for each pixel. A binary mask of the image of differences between a background representation and the current frame is initially created in order to isolate pixels that are changing over the time. Ghost effects that result from particular displacements of vehicles between successive frames are then removed to improve the estimation of the moving objects location. Finally the background representation is updated as part of the motion detection module.

### 3.2.1 Background Differentiation

For the purpose of detecting motion, foreground segmentation on the luminance of each pixel is used. Our experiments demonstrated that the foreground pixels corresponding to moving objects can be segmented with a reasonable accuracy by using the luminance. The background image that is considered is the set of pixels located inside of a search window and that might contain every type of objects that can be found in the camera field of view (road, static objects, and moving, stopped or parked vehicles). The foreground image is defined as a pixel map that contains only

the detected moving vehicles.

The principle of image binarization using a brightness criterion is extended to color image segmentation as color differences appear to be less sensitive to illumination variations and inter-reflections than gray scale differences. Figure 3.4 shows how a difference algorithm applied on color images improves the detection of moving pixels in comparison with the application on gray scale images. In figure 3.3, when a car approaches the window of interest during the night, a large portion of foreground pixels result from the reflection of lights on the ground. As a consequence, the number of white pixels in figures 3.4($a1$) to 3.4($d1$) is obviously larger than in the corresponding pixels maps of figures 3.4($a2$) to 3.4($d2$) where the segmentation is performed on color images. Therefore, color image differentiation is preferred to gray scale image differentiation in this work as it increases the accuracy of movement detection.

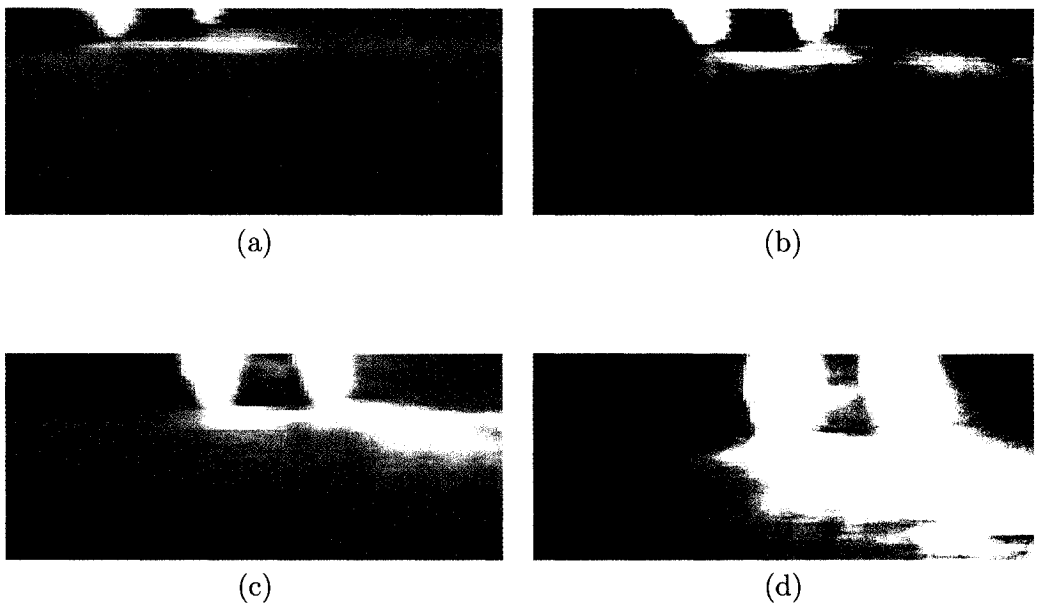



Figure 3.3: Results of differentiation on gray and color images taken at night. (a)-(d) original sequence of images at night.

The proposed algorithm for motion segmentation operates as follows: the three

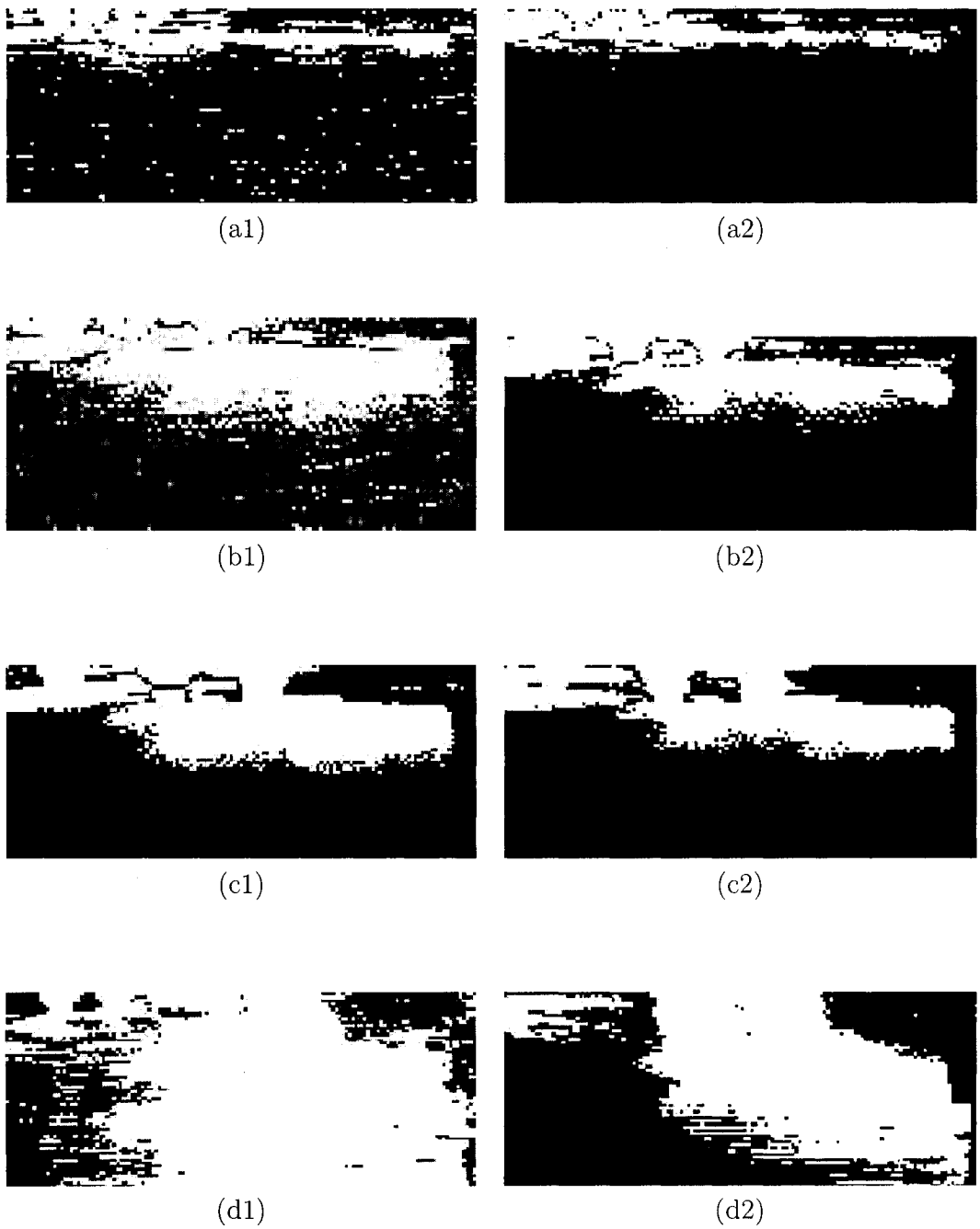



Figure 3.4: Results of differentiation on gray and color images taken at night. (a1)-(d1) results of differentiation on gray scale images; (a2)-(d2) results of differentiation on color images.

color components of each pixel in the current window image, $[I_R, I_G, I_B]$, are extracted at a time $t$ and compared with the previous background image components for the corresponding pixel $[B_R, B_G, B_B]$ estimated at time $t-1$ to compute a spatio-temporal difference image, $D$, defined as:

$$\mathbf{D}(x,y,t) = \begin{array}{ll} |I_R(x,y,t) - B_R(x,y,t-1)| & + \\ |I_G(x,y,t) - B_G(x,y,t-1)| & + \\ |I_B(x,y,t) - B_B(x,y,t-1)| & \end{array} \tag{3.1}$$

This difference image is then thresholded at a given level, $t_D$, in order to extract only the pixels where a significant change in all color components has occurred. A binary mask image, $M$, is then defined as a result of the threshold operation as follows:

$$\mathbf{M}(x,y,t) = \begin{cases} 1 & \text{if } D(x,y,t) > t_D \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

To achieve robustness in real-world outdoor conditions, the threshold value is dynamically estimated from the histogram of $D(x,y,t)$. As a major portion of the images consists of the background representing static objects, a large number of pixels usually get low difference values and are mapped around the peak of the histogram, as shown in figure 3.5($a$). However, under some particular environmental conditions like when clouds suddenly cover the sun or when vehicles enter or leave the search window during the night, abrupt illumination changes occur over a wider part of the search window. This makes the histogram to be spread on a wider range of values and the peak to be shifted to the right of the histogram, as shown in figure 3.5($b$). In this latter case, it becomes difficult to select a proper threshold point as the histogram is more uniform. Our experimentation revealed that background differentiation is not efficient in such a case. Therefore, we rather consider the time difference, $F(x,y,t)$, between the current pixels color components $[I_R(x,y,t),\ I_G(x,y,t),\ I_B(x,y,t)]$ and those of the previous frame $[I_R(x,y,t-1),\ I_G(x,y,t-1),\ I_B(x,y,t-1)]$, when background differentiation is not effective. The time difference image, $F(x,y,t)$, is then defined

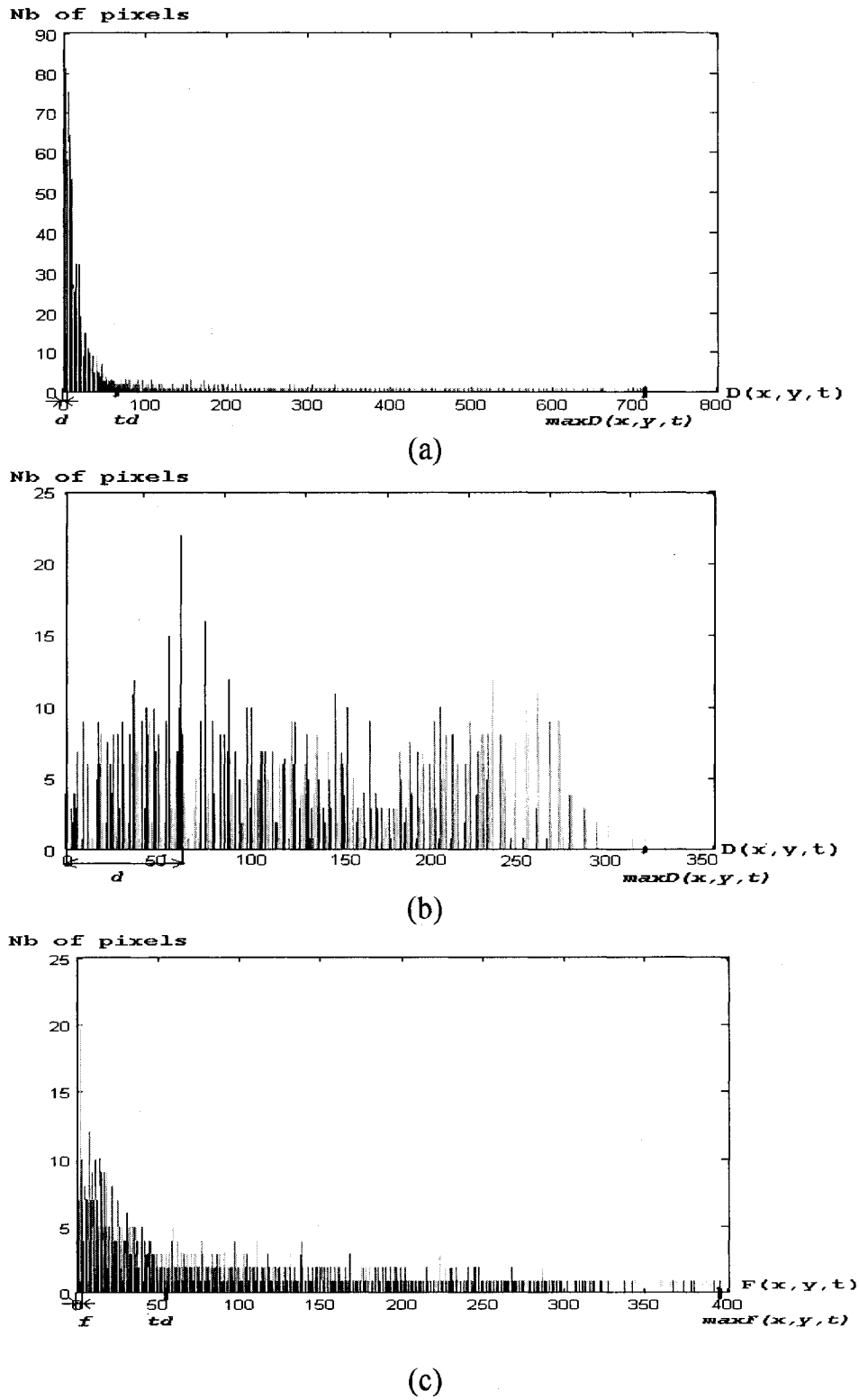



Figure 3.5: Histograms of differentiation under different illumination variation conditions. (a) background differentiation histogram under slow illumination changes; (b) background differentiation histogram under fast illumination changes; (c) time differentiation histogram under fast illumination changes.

as follows:

$$\mathbf{F}(x,y,t) = \begin{array}{l} |I_R(x,y,t) - I_R(x,y,t-1)| \quad + \\ |I_G(x,y,t) - I_G(x,y,t-1)| \quad + \\ |I_B(x,y,t) - I_B(x,y,t-1)| \end{array} \tag{3.3}$$

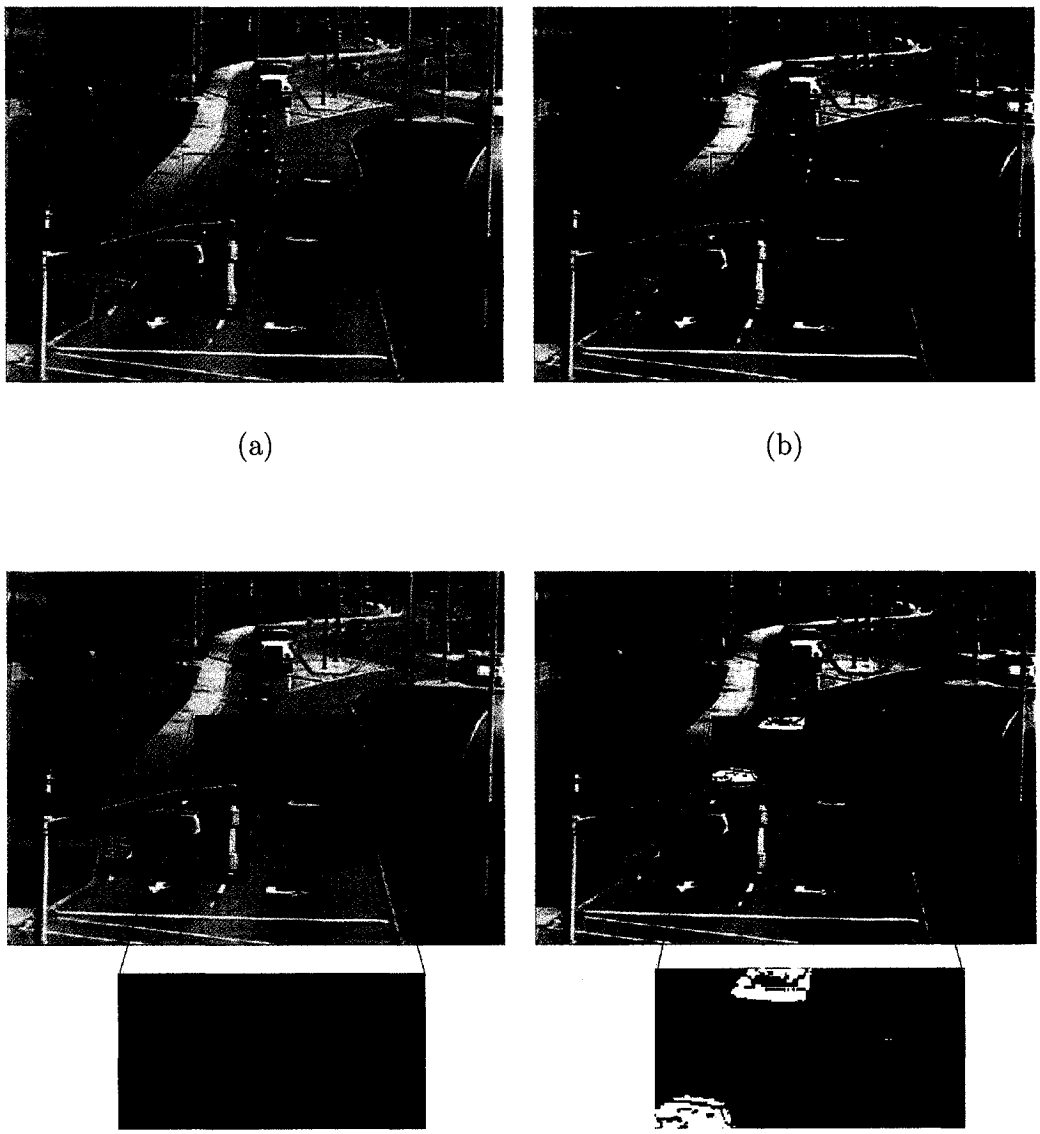
The resulting histogram of time differentiation is shown in figure 3.5($c$). The adaptive threshold selection method consists in finding a dip point at the right side of the peak of the histogram that has a value significantly lower than the peak value itself [14]. We propose the following algorithm for the selection of $t_D$.

$$\mathbf{t_D} = \begin{cases} d * s & \text{if } \frac{max(D(x,y,t))}{d} > 10 \\ f * s & \text{otherwise} \end{cases} \tag{3.4}$$

where $d$ and $f$ both represent the distances between the peak position and the origin of the histogram in the histograms of background differentiation or time differentiation respectively. $maxD(x,y,t)$ is the maximum difference value extracted from the background differentiation. The system automatically applies the proper differentiation algorithm according to the ratio of $maxD(x,y,t)$ to $d$, that is to use background differentiation when slow illumination changes occur or to apply time differentiation when sudden and widely spread illumination changes occur. The threshold, $t_D$, is calculated by multiplying the distance $d$ or $f$ by a parameter $s$. This parameter has been experimentally set to 7 to achieve the best results.

Figure 3.6 and 3.7 show two examples of abrupt changes in the global lighting of the scene. The first case occurs at the transition between sunny and cloudy. The illumination suddenly changes as the sun is overclouded or disclosed.

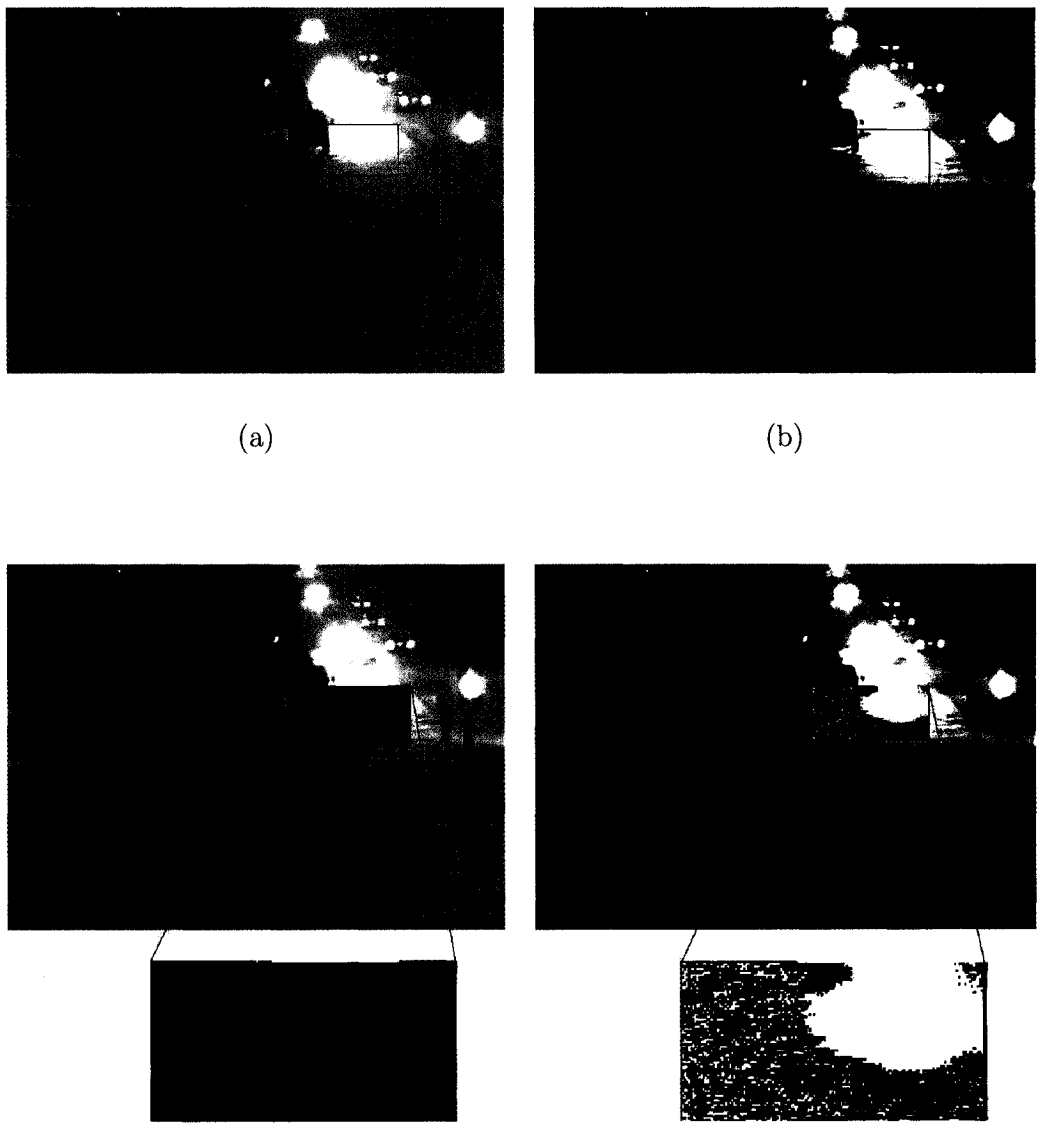
Our experiments demonstrated that if we use a background differentiation procedure at such a special time, no moving points can be detected, as shown in figure 3.6($c$). But if a time differentiation is performed, then moving points are successfully extracted, as shown in figure 3.6($d$). The second example happens at nighttime. When a car approaches the observation window, the illumination changes significantly over a large part of that window because of the strong reflection of the headlights on



(a) (b)

(c) (d)

Figure 3.6: Results of background differentiation and time differentiation under abrupt illumination changes during a sunny day. (a)-(b) successive original images; (c) results of background differentiation only; (d) results of time differentiation only.



(a) (b)

(c) (d)

Figure 3.7: Results of background differentiation and time differentiation under abrupt illumination changes at snowy night. (a)-(b) successive original images; (c) results of background differentiation only; (d) results of time differentiation only.

the ground. Comparing figures 3.7($c$) and ($d$), we observe that the detection relying on time differentiation significantly improves the segmentation results. The noise around the detected object area that can be seen in figure 3.7($d$) is due to strong reflections on the ground, since time differentiation is sensitive to illumination variations. However, as it does not make a local and compact cluster of pixels like the detected vehicle area, this noise can easily be removed by filtering of the binary mask.

### 3.2.2 Ghost Removal

Given the fact that the background image is progressively updated with some of the current frame information, a static object initially represented in the background representation that starts to move (e.g. a stopped car) requires a short period of time before it fades out from the background image. Therefore it influences the movement detection over this period of time as shown in figure 3.8($b$) where pixels located just above the car, shown in red colors, are still detected as moving points. These erroneous regions of the binary mask image that result from the temporary memory of the background image are called "ghosts" as they correspond to virtual objects that temporarily appear in the background model. These ghosts must be eliminated to preserve the coherence of the binary mask that must map only objects that are really moving.

(a) original image. (b) mask with ghost pixels. (c) mask after ghost removal.

Figure 3.8: Effect of ghost pixels on binary mask.

A correction is applied by differentiating the current and the previous frames over all moving pixels (identified in the binary mask). It is observed that ghost pixels have

a small difference between two successive frames since their motion is only virtual and results from erroneous background pixel values. Therefore a proper correction consists in eliminating these apparently moving pixels that have a small inter-frame difference from the binary mask. Figure 3.8($c$) illustrates such a corrected binary mask. The large patch in the middle of the mask image represents only the moving car without the ghost area. Other small patches in the binary image are caused by significant illumination changes, which will be removed by the following morphological operator.

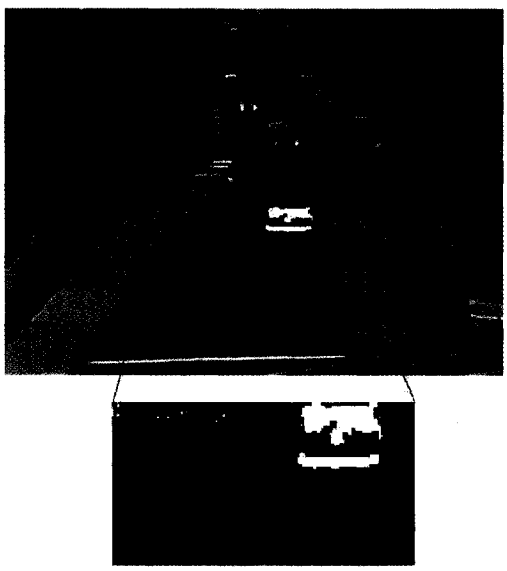
### 3.2.3 Morphological Operator

Once the relative correct motion mask has been formed, it must be further refined to get rid of the noise in the image or to fill holes in the object. These holes or noise mainly result from previous computations or slightly unexpected camera movements. The goal of the morphological operator is to generate integrated objects. Here, the Close morphological operator is applied on the binary mask, $M(x, y, t)$. The Close operator is implemented by a dilation operator followed by an erosion operator. Intuitively, the dilation operator increases the size of the objects in a binary image so that narrow gaps close up (eg. objects fuse together). The erosion operator does the opposite of the dilation operator. The purpose of erosion is to get rid of unwanted noise. Therefore the Close operator eliminates some defaults in the shape of the objects and corrects some holes due to segmentations errors. Furthermore, in order to speed up the computations, small connected components in the mask image, whose size does not pass over a threshold, are removed. Figure 3.9 presents the mask images before and after morphological operator application. The separated components which belong to the same vehicle get connected together as shown in image ($c$).

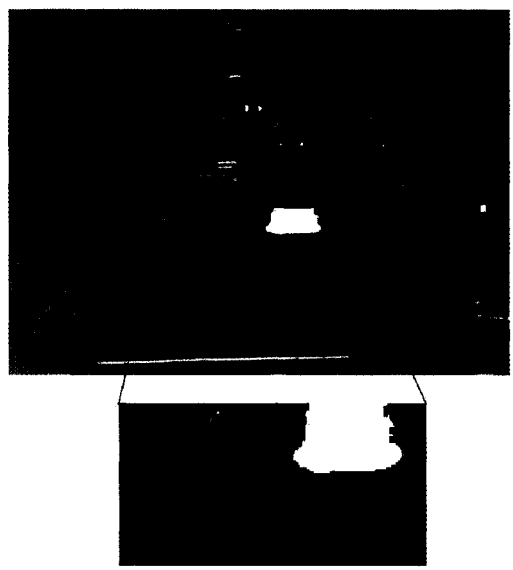
### 3.2.4 Adaptive Background Representation

In general, motion tends to generate rapid changes in the content of images while illumination and weather conditions tend to vary in a smoother way. However, in order to keep a track of the impact of the environmental factors on the overall image

(a) original image.



(b) motion mask.



(c) mask image after morphological operator.

Figure 3.9: Morphological operator results in a cloudy weather conditions scene.

over the time, an adaptive background image needs to be created and frequently updated.

Background changes are mainly due to two factors: lighting condition variations, and the status of vehicles that changes from stopped to moving or vice versa. Therefore, an adaptive background image is required to allow reliable segmentation between fixed and moving components.

In order to compute the current background representation, a temporary background, $TB(x, y, t)$, is computed with the help of the current binary mask image, $M(x, y, t)$. In this mask image, all pixels that correspond to foreground objects have a value equal to one, while all other pixels are set to zero. $TB(x, y, t)$ is obtained by combining the values of the previous background, $B(x, y, t-1)$, for areas corresponding to moving objects, with the values of the current image, $I(x, y, t)$, for fixed areas. The temporary background is then computed as follows:

$$TB(x, y, t) = \overline{M(x, y, t)} \wedge I(x, y, t) + M(x, y, t) \wedge B(x, y, t-1) \tag{3.5}$$

where $\wedge$ represents the logic AND operator.

Finally, the updated background image, $B(x, y, t)$, associated with the current frame is computed as a weighted average of the temporary background, $TB(x, y, t)$, and the previous background, $B(x, y, t-1)$, as follows:

$$B(x, y, t) = (1 - \alpha) * B(x, y, t-1) + \alpha * TB(x, y, t) \tag{3.6}$$

where the parameter $\alpha$ allows to control the background update rate. Its value is estimated adaptively in order to increase robustness when dealing with large variations in illumination or motion. As discussed previously, a peak in the histogram of the difference image mainly results from illumination changes. When the illumination variation is low, the peak tends to be located in the leftmost part of the histogram. For larger variations in the illumination, the peak is shifted towards the right.

These observations allowed to achieve a function for the estimation of the $\alpha$ parameter that relies on two measurements extracted from the histogram of the difference image, $D(x, y, t)$ or $F(x, y, t)$. Depending on what kind of differentiation is operated,

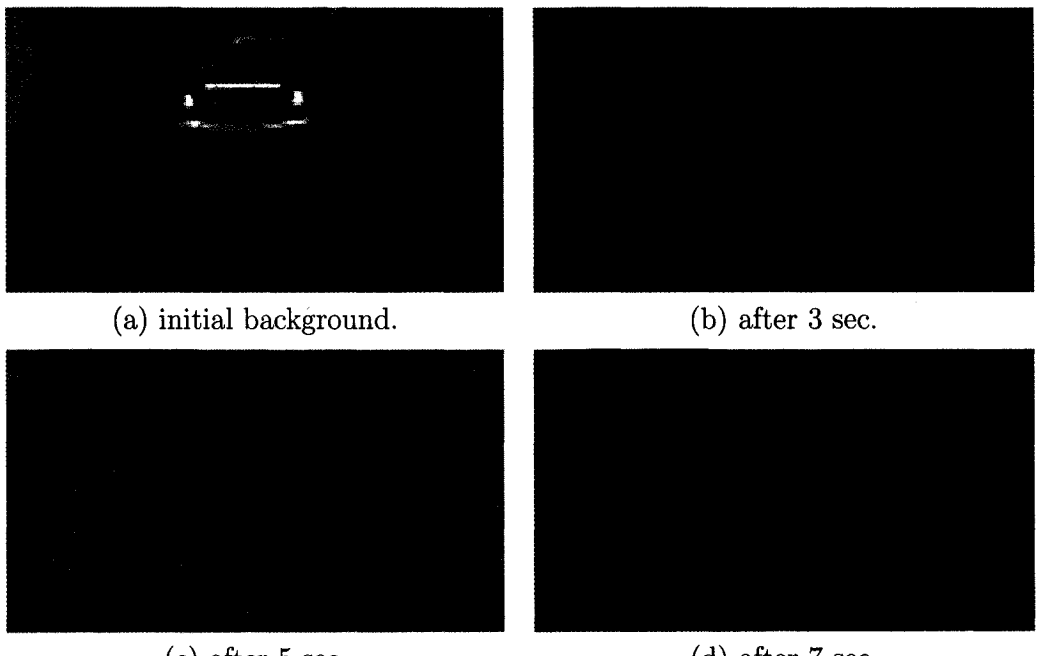
the first measurement, $d$ or $f$, corresponds to the distance between the peak and the origin of the corresponding histogram. The second measurement, $maxD(x, y, t)$, or $maxF(x, y, t)$, is the maximum difference value. The parameter $\alpha$ is then estimated as follows:

$$\alpha = min(0.1 + \frac{d}{max(D(x, y, t))}, 0.5) \tag{3.7}$$

or

$$\alpha = min(0.1 + \frac{f}{max(F(x, y, t))}, 0.5)$$

That is, the $\alpha$ value varies with the ratio of the peak distance on the histogram to the maximum difference value with a maximum of 0.5. When $\alpha$ reaches larger values,



(a) initial background. (b) after 3 sec.

(c) after 5 sec. (d) after 7 sec.

Figure 3.10: Evolution of the adaptive background representation.

the background update rate is increased as it corresponds to faster variations in the content of the scene. In practice, this results in an increased background transition

rate when illumination changes are important as the background representation must then map faster variations in the static components of the scene. Figure 3.10 shows an example of a background image updated with the proposed approach over a period of 7 seconds. Here $\alpha$ has been set to a value of 0.148 as illumination is quite constant under clear weather conditions.

## 3.3 Dual-Mode Feature Mapping

In order to monitor the vehicle activities occurring in the observation area, we have to be able to recognize the vehicle that passes in front of the video. After the previous steps of the process, we obtain motion regions of the image, which may or may not contain vehicles. A further analysis on the motion mask would help us to determine whether they are vehicles or not. As none of the existing methods can be applied alone to detect the vehicles under various weather and illumination conditions, we concluded that it is better to adopt different algorithms under different conditions. In our system, refinement of the detected moving objects is performed through a dual mode processing scheme that maps vehicles as rectangular patches that surround the complete structure of the vehicle under clear and highly contrasting daytime conditions, which is defined as the full features (FF) mode. Under night or lighter contrasting conditions such as heavy rain, snow or fog, vehicles are mapped as groups of circular regions corresponding to their headlights, which is defined as the headlight-based (HB) detection mode. An automatic switching approach has been developed that relies on RGB color images converted to the Hue-Saturation-Value (HSV) color mapping.

### 3.3.1 Mode Selection Module

In the previous motion detection phase, we benefit much from the RGB color representation provided by video camera. However RGB representation loses its superiority in the refinement phase. Based on the requirements of the task to perform, the required

color space must have the following characteristics:

1. It should have distinct characteristics under various weather conditions.
2. It must allow the detection of headlights and shadows.

There are many existing color spaces and each color space is suitable for a certain application area. Some color space likes RGB, CMY, CMYK, YIQ are specific to different hardware devices, like printers, color monitors, TV broadcast. Other color space like the HSV color space is frequently used for color manipulation. So here, the HSV representation appears to be the most relevant, as it is proven to be closer to human color perception than RGB. The HSV color model stands for Hue, Saturation and Value (luminance), while RGB is a model that decomposes color into channels of red, green and blue intensity. Both of them are introduced in Appendix A.1 and A.2 respectively. Because most color cameras provide a RGB signal, we must first convert pixels from the RGB color scheme to their equivalent HSV representation. The conversion formulas that have been used are the ones adopted from the literature [27] and are detailed in Appendix A.3.

Since we apply a dual mode processing, it is critical to determine how to efficiently switch between the FF and HB modes. By observation, we found that images' contrast varies under different weather and illumination conditions. Usually, at clear daytime, the contrast is so high that the vehicle's shape can be easily detected, while at nighttime or under bad weather conditions, the contrast of images gets lower so that the vehicles' headlights become the prominent feature to be detected. Although the contrast of an image can be perceived directly by human eyes, it is complex in the computer's perspective. In addition, the computation speed is also an important factor we need to consider. Fortunately, two apparent phenomena are found in these traffic images. One is the variations of the window color, the other is the variations of the ratio of vehicles' headlights. Each of them has been analyzed under different outdoor conditions and leaded to the following observations.

Under a HSV mapping, it has been observed that color histograms of images collected under different environmental conditions clearly differ as shown in figure

3.11. The hue has the greatest discrimination power among the other coordinates. Also, to extract the color information of the observation window, a two dimensional histogram of Hue and Saturation is used.

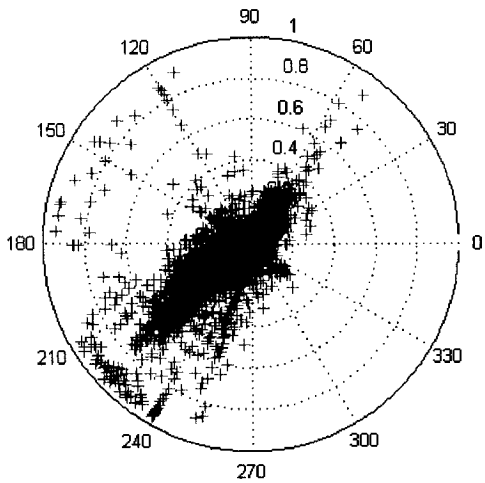
The distribution of clusters of points in the HS histogram tends to move toward different areas of the polar representation with changes in weather and illumination. On the other hand, HS histograms of different color vehicles collected under similar conditions remain rather similar one to each other, provided that the vehicles do not cover the totality of the window of interest.

As the hue is the actual color of the object, there are three main factors that influence the hue of a colored objects:

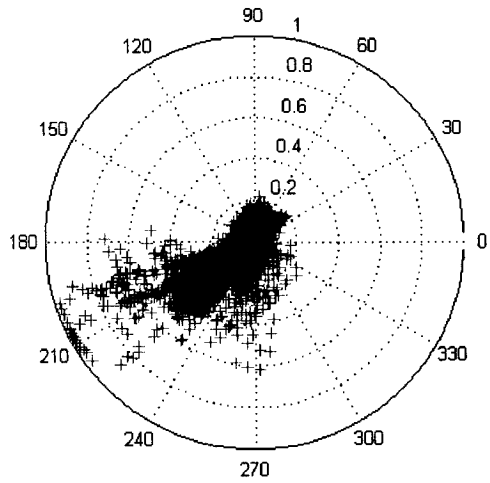
- the local colour of the object;
- the light source which illuminates it;
- any color reflecting on surfaces near the object.

The local color is the colour of an object that is not modified by light, reflection, weather or distance. Local colour is sometimes referred to as true colour. The light source that illuminates an object affects the local colour of that object. These factors explain the phenomenon that appears on the Hue and Saturation histogram of figure 3.11 which makes the cluster distribution move in different areas with changing weather and illumination.

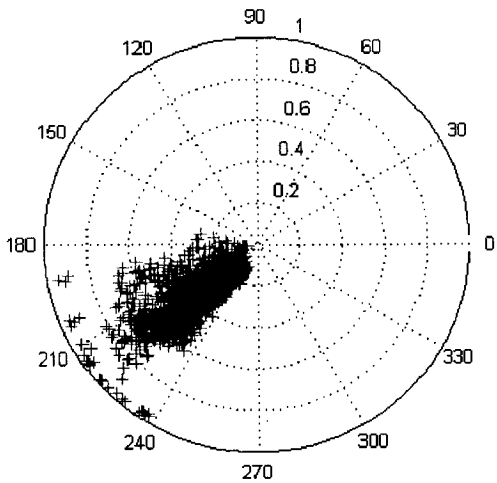
In the HSV color space, the hue is given by the angle with red located at 0°, yellow at 60°, green at 120°, cyan at 180°, blue at 240°, and magenta at 300°. Since in the observation window, the ground usually covers the main part of the window, the ground color determines the main color of the window area. Further, it appears that the ground color is usually strongly affected by the illumination and weather conditions. Under high contrast conditions such as sunny or cloudy days, the color of the ground is close to its original color, which is between cyan and blue, so accordingly its Hue is between 180° and 270°. As a result under sunny conditions, as shown in figure 3.11($a$), the majority of points on the HS histogram are located in that



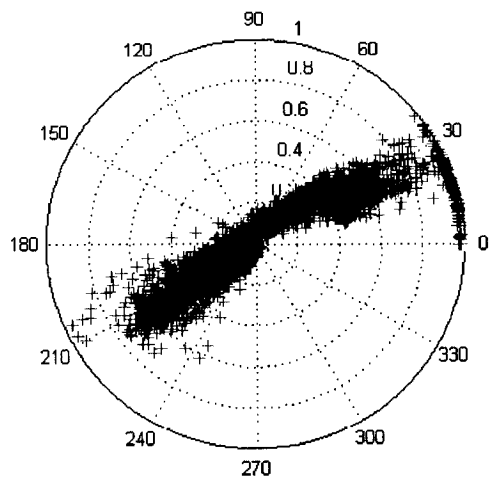

(a) sunny day.



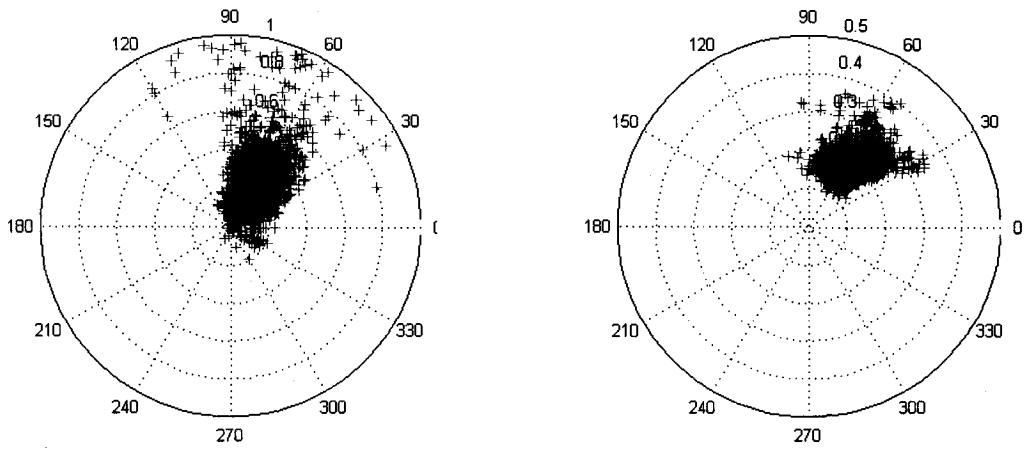

(b) cloudy day.




(c) snowy day




(d) rainy day.




(e) clear night. (f) snowy night.

Figure 3.11: HS polar histograms of images under different environmental and contrast conditions.

specific range of values. Points mapped at other angles mainly correspond to the vehicles' color. Under low contrast conditions such as during a clear night or snowy night, the color of the ground is changed by environmental factors like headlight illumination, snow or water reflection and road lamps. The illumination from the headlights and road lamps is the main factor that determines the ground color. Under these situations, the color of the ground surface tends to be located between orange and yellow, thus its Hue is between 0° and 90°. When snow is covering the ground, some white color is added on the ground. This results in lower saturation. Water on the ground has the tendency of adding dark to the shade of the ground, which also slightly affects the hue.

These changes in the window color with different conditions is exploited to determine the best detection mode under different environmental conditions. The experimental histogram illustrated in figure 3.11, shows that in the HS plane the first quarter between 0° and 90° and the third quarter between 180° and 270° are two important areas. During a sunny, cloudy and snowy day, as shown in figures 3.11($a$), ($b$) and ($c$), the majority of points on the HS polar histogram are located in the third

quarter. Since the vehicle's shape is the prominent feature under these conditions, the FF mode is the most suitable choice for the detection system. Conversely, on a clear night or snowy night, as shown in figure 3.11($e$) and ($f$), most of the points are located in the first quarter. Since the headlights are the obvious feature to detect, the HB detection mode is the best choice under these kinds of situations. By comparing these two areas of the polar histogram, we can estimate what kind of feature is dominant under a specific weather and illumination condition. Here, we count the number of pixels that are mapped to the first quarter of the HS plane as $NU$, and the number of pixels that are mapped to the third quarter of the HS plane as $NL$. $NU$ and $NL$ are computed by an examination of the Hue angular values, respectively from 0° to 90° and from 180° to 270°, for each pixel in the search window to determine its location in the HS polar plot. However, this comparison is not suitable for rainy weather, as shown in figure 3.11($d$), as the points are equally distributed in the first and third quarters. This results from the similarity between the luminosity of the natural lights and that of the headlights or reflections. As a result, under rainy conditions as well as at dusk or at dawn there is a period of gray time when either of the detection modes can be successfully used. It is not easy to determine the best detection mode in these circumstances. In order to allow a proper selection of the detection mode, a supplementary criterion is developed.

As we mentioned previously, the luminosity variations of vehicles' headlights is another key factor. Usually when the contrast in an image is low, the luminosity of vehicles' headlights is strong such that the headlights are larger in appearance. On the contrary, when an image's contrast is high, headlights' illumination becomes faint or can even be totally covered by bright daylight so that the headlights are usually too small to be observed. These observations lead to the development of a function for the estimation of the ratio of vehicles' headlights in the observation windows based on two measurements. The first measurement, $N$, corresponds to the number of pixels representing headlights in the search window. The second measurement, $W$, represents the total number of pixels in the search window. The estimation of the

ratio, L, is then computed as follows:

$$L = \frac{N}{W} \tag{3.8}$$

That is, the $L$ value varies with the ratio of the number of headlights pixels to the total pixel number in the observation window. When $L$ reaches larger values, vehicles' headlights look bigger and brighter, and then the headlight features are visible.

By combining these two important factors, the distribution of pixels on the HS histogram and the ratio, $L$, of headlights in the window, we can select the most suitable detection mode at a given time. After analyzing the real natural conditions, we divide all these different kinds of weather and illumination conditions into four cases. The first case happens mainly on clear days. With strong daylight and weak ratio of headlights, the images' contrast is high so that we choose the full feature detection mode at that time. The second situation happens at night. With strong headlight ratio and weak natural light, the headlight feature is prominent so that we select the headlight-based detection mode. The third and the fourth situations may happen at dusk, at dawn or under rainy weather. Since the illumination is between bright and dark, the ground color may show either natural asphalt color or the color of reflected light. Thus, the detection mode is rather determined by the ratio, $L$, of the headlights. In the third case, even though the daylight is strong enough for the vehicle's shape to be detected, as long as the ratio of the headlights reaches a certain level, the system prefers to choose the headlight-based detection mode. On the other hand, in the fourth case, although the daylight is not very bright, since the features of headlights are not easy to detect, the system rather selects the full feature detection mode. These four cases generalize all the situations that happen in natural outdoor conditions, and they are summarized as follows:

Case 1: If $NU < NL$ and $L < l$, Then mode = FF;

Case 2: If $NU \geq NL$ and $L \geq l$, Then mode = HB;

Case 3: If $NU < NL$ and $L \geq l$, Then mode = HB;

Case 4: If $NU \geq NL$ and $L < l$, Then mode = FF;

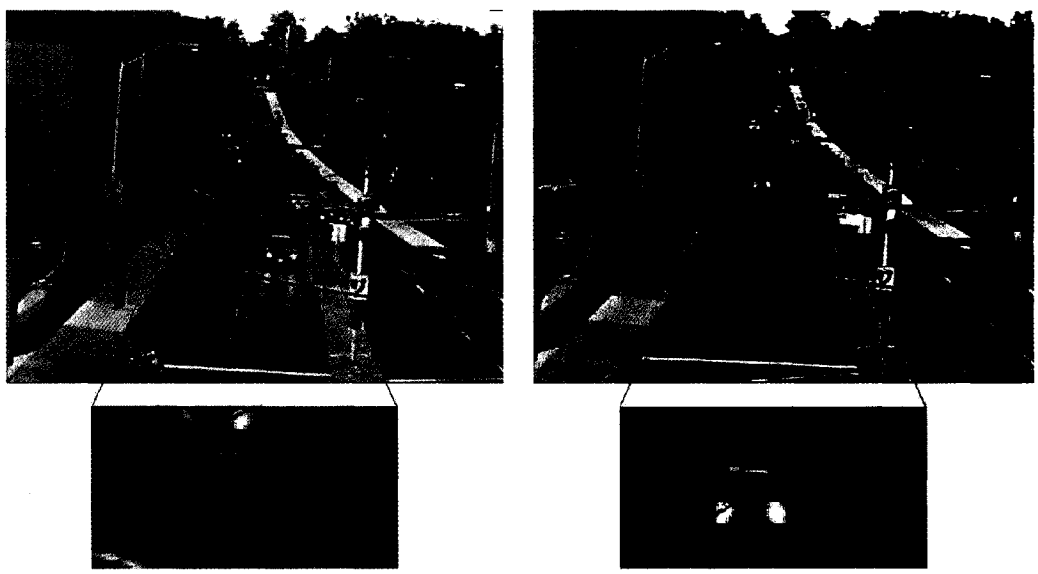
A proper $L$ threshold level, $l$, has been experimentally estimated to be 0.02, that is 2% of the pixels being highlighted. This value tends to slightly emphasize the application of the headlight-based detection mode as it provides a clearer representation of vehicles, reduces the impact of occlusion problems between vehicles and is less sensitive to shadow effects. Figure 3.12 shows a sequence of detection results obtained by applying the proposed dual-mode switching algorithm at dusk. In these images, the regions are indicated for detected cars by green bounding boxes. The whole sequence was taken between 20:30-21:10 p.m on July 29, 2003. During the sunset, the illumination becomes dimmer and the luminosity of vehicles' headlights gets stronger, accordingly the detection mode switches from FF mode, as shown in figures 3.12($a$) and ($b$), to HB mode, as shown in figures 3.12($c$) and ($d$). Figure 3.13 shows another example of the mode switching in which a car is passing through the window. Headlight-based detection mode is active in figures 3.13($a$), ($b$) and ($c$). As soon as the car's headlights leave the window, since the $L$ is below the threshold level, $l$, the algorithm immediately switches to the full feature detection mode as shown in image ($d$). This allows to preserve the detection of the car even though its headlights are no longer inside the search window.

### 3.3.2 Shadow Removal

Although background subtraction can provide effective means of locating and tracking moving vehicles, moving shadows do cause serious problems since they differ from the background image and are therefore identified as parts of the moving objects.

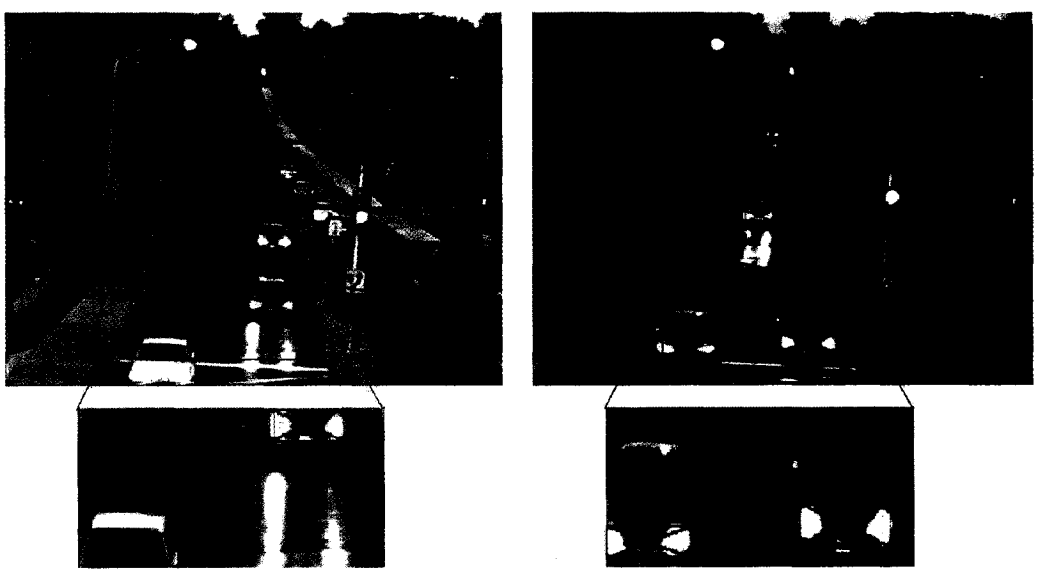
Shadows of foreground elements tend to generate changes on the adaptive background representation. Without special treatment these variations in image content tend to be considered as being part of the foreground components in the difference image resulting from color segmentation. Such a behavior is not suitable as shadows should not be detected as moving objects.

Many algorithms that detect shadows take into account the location of the light



(a) mode=FF, at 20:31 p.m.

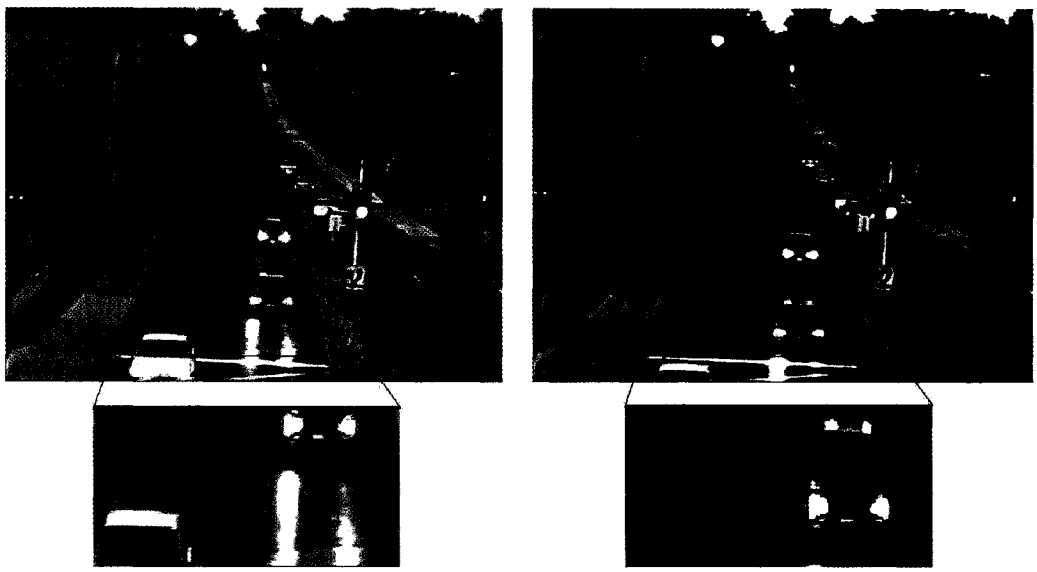
(b) mode=FF, at 20:52 p.m.



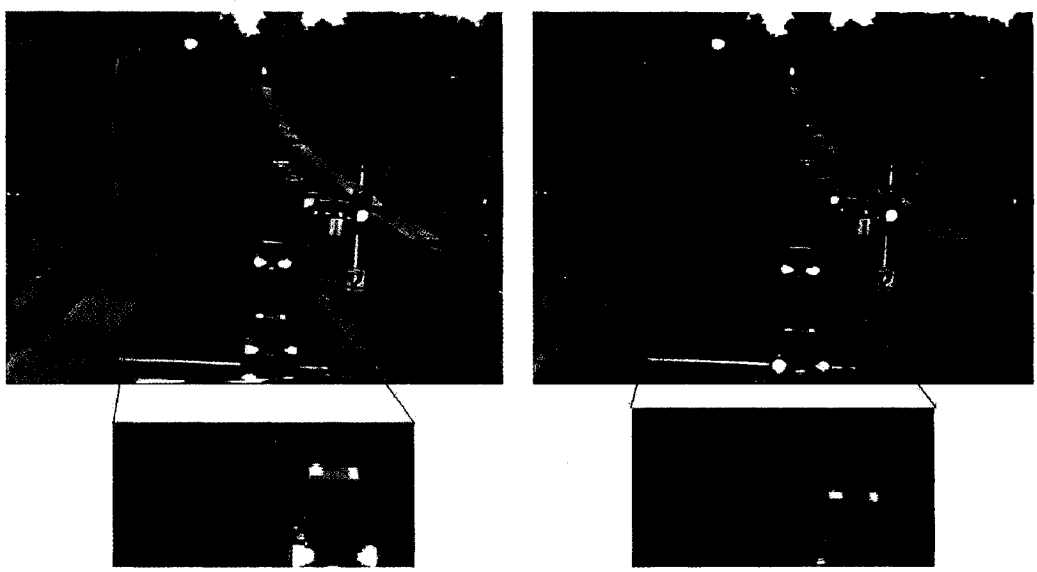
(c) mode=HB, at 21:04 p.m.

(d) mode=HB, at 21:10 p.m.

Figure 3.12: Results of vehicle detection using the dual-mode switching algorithm at dusk.



(a) mode=HB. (b) mode=HB.



(c) mode=HB. (d) mode=FF.

Figure 3.13: Dual-mode switching during a car detection.

source, the geometry of the scene and the models of moving objects [24].

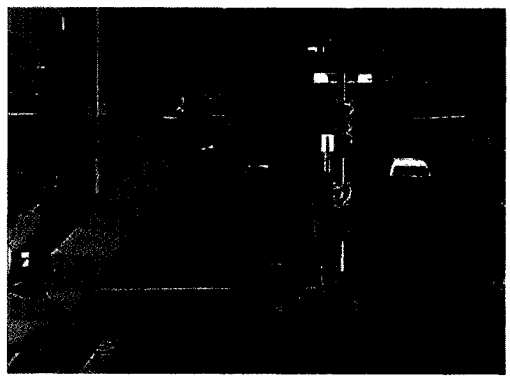
Fortunately, shadows can be discriminated from objects by means of the HSV color space mapping. We use an approach inspired by [25], in which each point of a shadow mask, $SM(x,y,t)$, must satisfy three conditions described as follows:

$$\mathbf{SM}(x,y,t) = \begin{cases} 1 & \text{if } (\alpha < \frac{V_I(x,y,t)}{V_B(x,y,t)} < \beta) \wedge \\ & ((H_I(x,y,t) - H_B(x,y,t)) < \varepsilon_H) \wedge \\ & ((S_I(x,y,t) - S_B(x,y,t)) < \varepsilon_S) \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

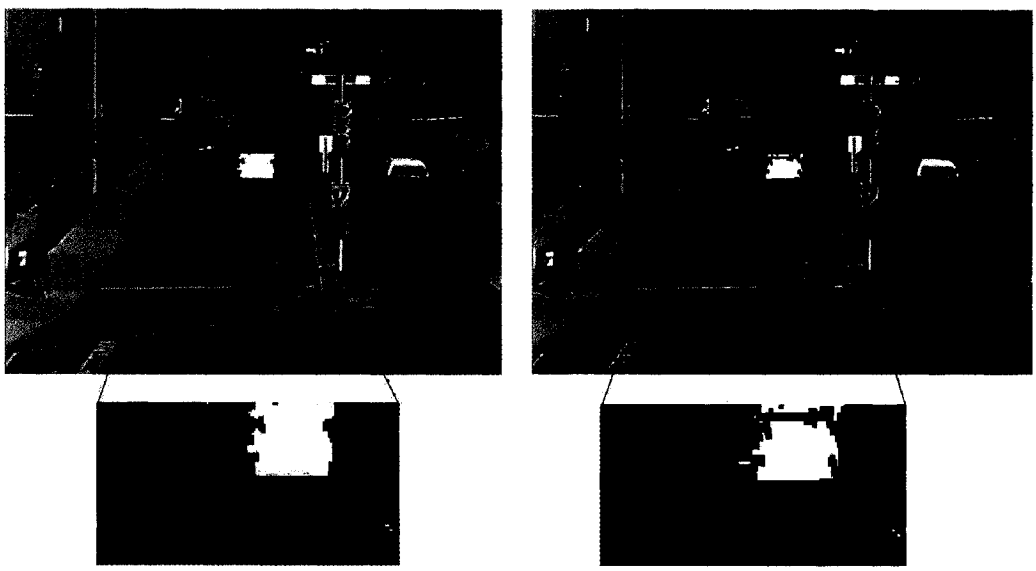
where $H_I$, $S_I$ and $V_I$ respectively represent the Hue, Saturation and Value values of a pixel in the current image. Similarly, $H_B$, $S_B$ and $V_B$ represent the Hue, Saturation and Value values of a pixel in the background image. $0 < (\alpha, \beta) < 1$ as shadowed areas have lower luminance than the background, and $(\varepsilon_H, \varepsilon_S)$ are kept very small under the assumption that the chrominance of shadowed and non-shadowed areas is similar. In order to speed up the computation, we only examine the moving points detected in the binary mask, $M(x,y,t)$, for shadow detection. Furthermore, in order to remove the shadow from objects, we combine the original mask image, $M(x,y,t)$, with the complement of the shadow mask, $SM(x,y,t)$, to obtain the object mask, $OM(x,y,t)$, as follows:

$$OM(x,y,t) = \overline{SM(x,y,t)} \wedge M(x,y,t) \tag{3.10}$$

Figure 3.14 presents some shadow removal results. The original image $(a)$ is taken under sunny conditions with moving shadows cast by the vehicles. After motion detection, the resulting mask image $(b)$ contains many shadow pixels shown in red. Applying the shadow removal function, we can successfully remove the moving shadows around the vehicle and get a new object mask, as shown in image $(c)$. Here the parameters have been set to $\alpha$=0.05, $\beta$=0.95, $\varepsilon_H$=0.3 and $\varepsilon_S$=0.3. One slight limitation of this approach that has been observed is that sometimes it tends to treat the windows of a vehicle as shadows and erroneously removes them from the mask image as shown in figure 3.14$(c)$. This phenomenon is due to the fact that the windows



a) original image.



b) initial motion mask.

c) mask image after shadow removal.

Figure 3.14: Performances of the shadow removal procedure applied on a binary mask under bright sunny conditions.

have similar characteristics as shadows in the HSV representation. But this does not significantly degrade the vehicle detection as windows are contained within the area covered by the vehicle.

### 3.3.3 Headlights Detection Algorithm

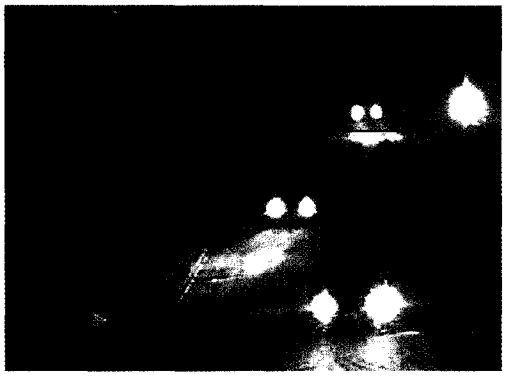
At night or under low illumination conditions, vehicles' headlights are the prominent features that can be observed and extracted from the motion mask to help in determining the moving vehicles.

Highlights correspond to those points that have very high luminosity in an image. The light of headlights is a kind of highlight which is useful for the vehicle detection. However, some other highlights, such as strong reflection of headlights on the ground could prevent us from properly detecting the vehicles. Highlights detection could be achieved by using Saturation and Value values based on the observation that when compared with other pixels the highlight pixels always have very low saturation with very high intensity. Therefore, we might consider moving pixels as highlights if and only if their Saturation values are less than a threshold $s$, and their Value values are larger than a threshold $v$. Following this idea, a highlight pixel satisfies the following equation:

$$\mathbf{HM}(x,y,t) = \begin{cases} 1 & \text{if } (S_I(x,y,t) \leqslant s) \wedge (V_I(x,y,t) \geqslant v) \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

where $S_I$ represents the Saturation value of a pixel in the current image taken at time $t$ and $V_I$ corresponds to the Intensity value of that pixel.

From our observations, we found that the reflections on the ground surface tend to cause some variations in the saturation and the intensity. In most cases, the purity of the color of reflections is higher than that of headlights, whereas the intensity of reflections is lower than that of headlights. Therefore a careful refinement of $s$ and $v$ values could separate the color clusters of headlights from most of the small reflections on the ground. In order to examine the influence of $s$ and $v$ in the performance of the detection, we compared the results of headlight detection between five sets of $(s, v)$



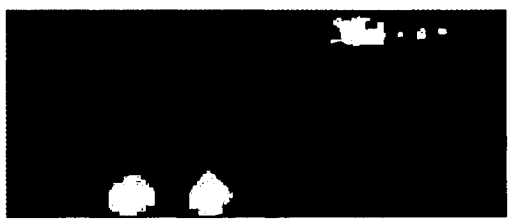
(a)original image.

(b) motion segmentation result.

(c) 0 <s$\leqslant$ 0.01, v $\geqslant$ 0.995.

(d) 0 < s $\leqslant$ 0.02, v $\geqslant$ 0.995.

(e) 0 <s $\leqslant$ 0.1, v $\geqslant$ 0.995.



(f) 0 <s $\leqslant$ 0.2, v $\geqslant$ 0.995.

(g) 0 <s $\leqslant$ 0.2, v $\geqslant$ 0.90.

Figure 3.15: Results of headlights detection with different setting of $s$ and $v$ values.

values, as showed in figure 3.15. With slightly increasing $s$ values and decreasing $v$ values, the detected size of headlights and reflection parts begins to enlarge. Figure 3.15($c$) shows the slimmest detection area while ($g$) shows the largest one. There is no significant differences between ($d$), ($e$) and ($f$). Since the setting of ($d$) shows the least influence of reflection lights, we selected these values to achieve the best detection performance. Resulting from these experiments, we set $s$ equal to 0.02 and $v$ equal to 0.995 respectively.

To obtain the object mask, $OM(x, y, t)$, we combine the original motion mask, $M(x, y, t)$, with the headlights mask, $HM(x, y, t)$, as follows:

$$OM(x, y, t) = HM(x, y, t) \wedge M(x, y, t) \tag{3.12}$$

Further differentiation and refinement of headlights from large ground surface reflection interference is also performed done by circular shape verification as discussed in the following section.

## 3.4 Vehicle Identification

Once the corresponding features under different detection modes have been extracted, they must be analyzed to determine if they represent vehicles. Vehicle identification is the last step of the detection procedure. It consists in identifying vehicles by matching simple geometrical patterns with the detected moving regions in order to better define their respective area. Under daytime conditions with relatively clear weather (sun, cloud or even drizzle), vehicles are segmented as large rectangular patches corresponding to the whole vehicle shape. At night or under noisy situations (heavy rain, snow), vehicles are rather segmented as pairs of circles corresponding to their headlights as these features remain the most visible.

### 3.4.1 Full Features Detection

Under the full feature detection mode, after the shadow removal, the object mask possibly turns out to be fragmented. In order to identify the vehicle object clearly, a merge procedure is performed at this point. First, to speed up the merging process, all segments whose areas are less than a preset small value are removed. Afterwards the total number of fragments left is counted. Each fragment in the mask is represented by a rectangle, which is set to surround the fragment. Usually the fragment is irregular, therefore we compute the distance between the leftmost and the rightmost point of the fragment as the rectangle's width, and compute the distance between the upper and the lower point of the fragment as the rectangle's height. As shown in figure 3.16,

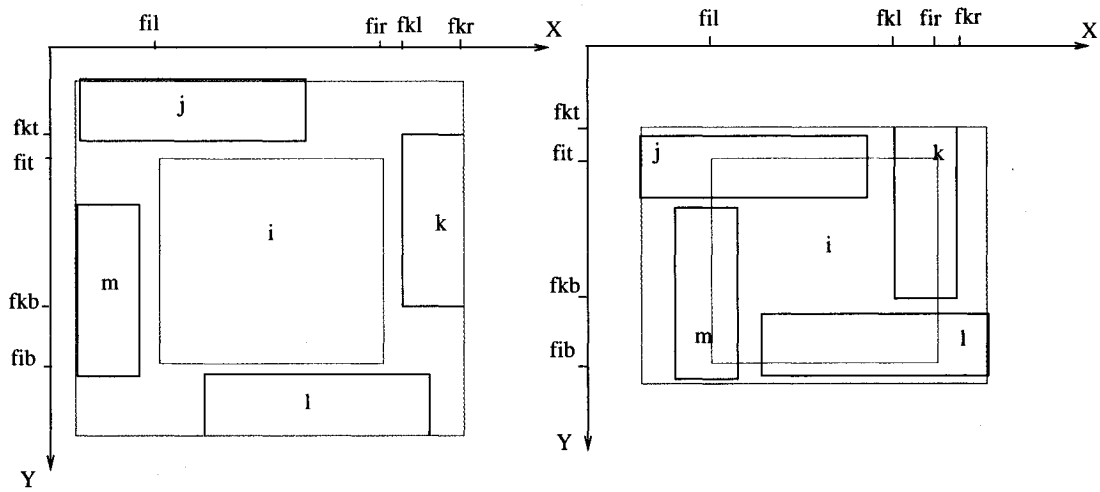



Figure 3.16: Rectangle merging function under FF mode .

all the $i$, $j$, $k$, $l$, $m$ rectangles present fragments. We define four variables $f_{ir}$, $f_{il}$, $f_{it}$, $f_{ib}$ for each fragment, which respectively represent the $i^{th}$ fragment's right, left, top and bottom side position in the window sized image coordinate. By scanning on all the fragments, the pairs whose distance falls below a threshold are merged. For example, if $|f_{kl} - f_{ir}| < \mu$ and $|f_{it} - f_{kt}| < \nu$ $(0 < \mu < \nu)$, then segment $k$ could be merged with the segment $i$. Here $\mu$ and $\nu$ are small values which are no more

than 10 pixels length. In the same way, segment $i$ can be merged with three other segments $j$, $l$ and $m$ and generates a new larger rectangle. If there are some other segments in that window, the merging algorithm would recursively combine them with the new large rectangle until no segments are left. The final obtained rectangle represents a moving vehicle. In addition, we can easily estimate the vehicle's position by computing the rectangle's position in the window. Figure 3.17 shows the good performance of this rectangle merging function applied on an object mask image. As shown in figure 3.17($c$), the vehicle is segmented in two components. By using rectangle merging function, the system integrates these two components and regards them as a complete one, as shown in figure 3.17($d$).

### 3.4.2 Headlights Based Detection

Under low illumination and bad weather conditions, vehicles' headlights are the features to be matched. In order to speed up the matching, we apply some preprocessing. If the detected circular area is smaller than a preset value, the object is removed. Similarly if the circular area is too large, the object is removed, as this usually corresponds to the reflections from a strong illumination by the headlights of the moving objects on the ground surface.

The prior knowledge that we have about the headlights is:

- a regular shape of headlights (as seen in a far picture) in general is circular.

  The area, $A$, and the perimeter, $P$, are two parameters which describe the size of a circle. In order to compare objects which are observed from different distances, it is important to use shape parameters that do not depend on the scaling factor of the object on the image plane. The circularity, $c$, is independent of the object's size [28]. It can be computed as follows:

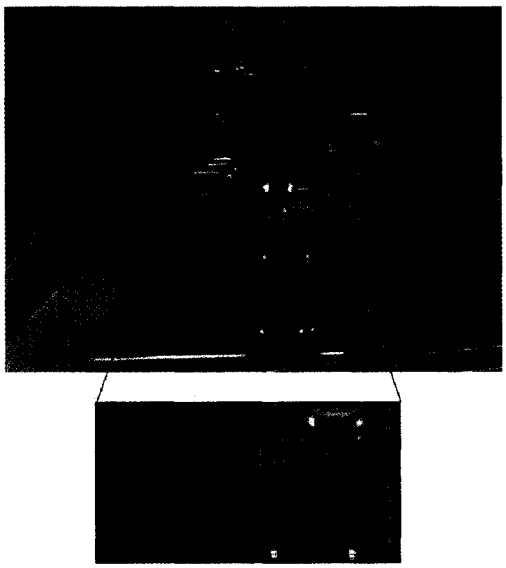
$$c = \frac{P^2}{A} \tag{3.13}$$

  Typically a value of $c$ that is around $4\pi \approx 12$ indicates a circular shape.

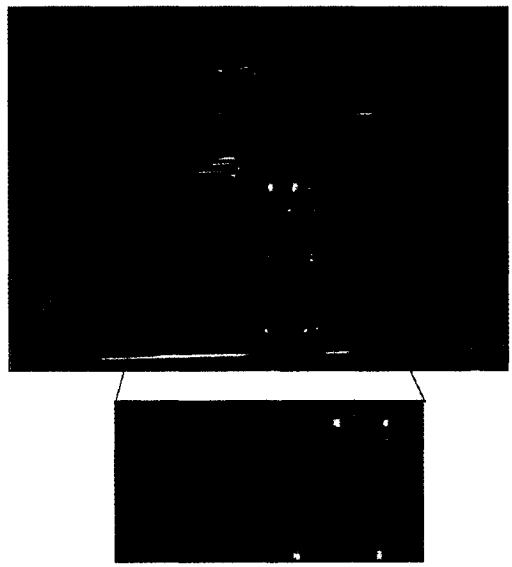
a) original image.



b) object mask.



c) matching result without merging function.



d) matching result with merging function.

Figure 3.17: Performance of rectangle merging function applied on an object mask image under cloudy conditions.

- the two headlights of a same vehicle should be close to each other (with respect to the size of the camera field of view).

  As we assume that each vehicle has a pair of headlights, which are installed at the same height on the vehicle with a certain distance between them, which limits the search region for the matched headlights. The inter-circles' distances on both the X and the Y axes are considered.

  On the X axis, as shown in figure 3.18, the Euclidean distance, $d$, between the two circular objects should be in a certain range. It satisfies the following condition:

$$d_{min} < d = |X_i - X_j| < d_{max} \tag{3.14}$$

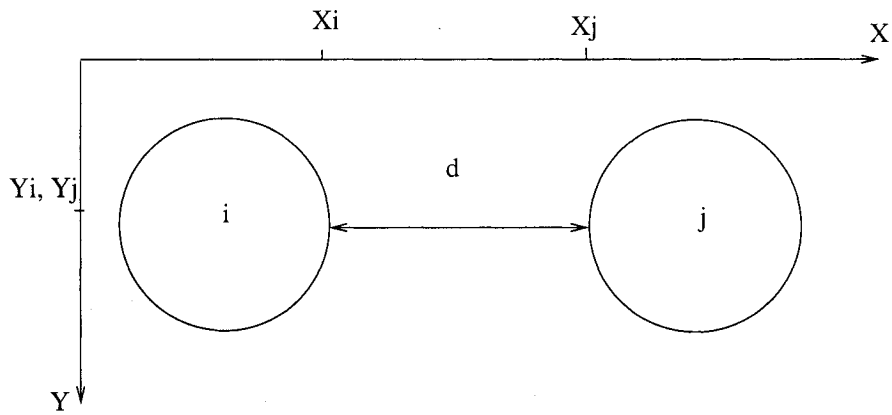



Figure 3.18: Circle pairing function under HB mode.

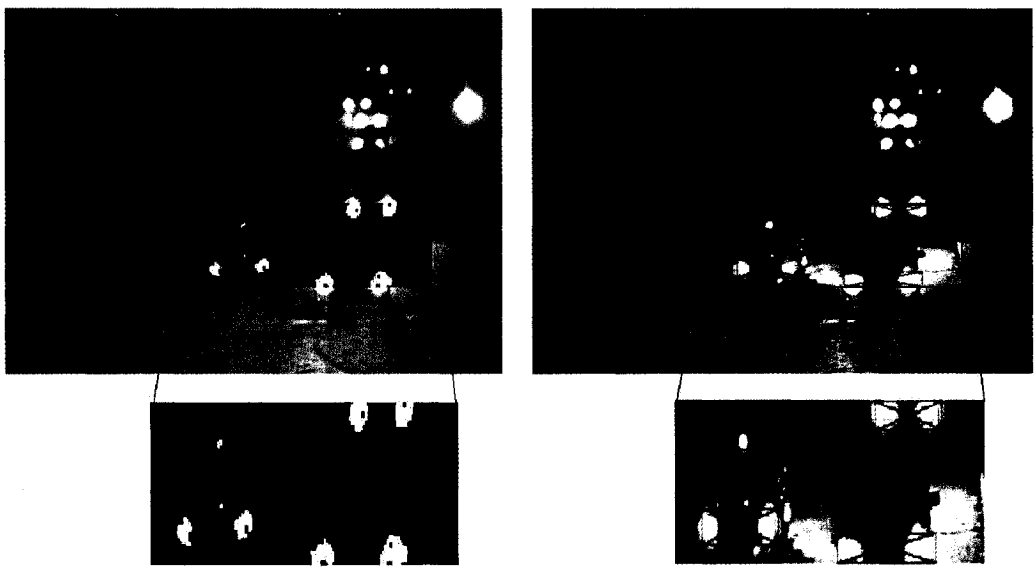
where $X_i$ and $X_j$ are the abscissa values in an image framework, representing the rightmost point of the object $i$ and the leftmost point of the object $j$ on the image plane. The $d_{min}$ is a predefined small value and is set to be 10 pixels in the experiments. The $d_{max}$ is equal to 40% of the window width. The image distance, $d$, between two headlights is variable with the window position along the road. If the window is set too far, this image distance becomes short and

can reach zero at the end of the scene. With such a setup, we obtain large bright irregular blobs. Correctly setting the search window during the system initialization phase could prevent this problem.

On the Y axis, as shown in figure 3.18, the two circular objects should be at the same or similar height, which means the Euclidean distance between $Y_i$ and $Y_j$ should be small. Thus it satisfies the following condition:

$$|Y_i - Y_j| < y \tag{3.15}$$

where $Y_i$ and $Y_j$ are the image ordinate values of objects $i$ and $j$ center points. Here $y$ is a predetermined small value and is set to 5 pixels in the experiments.



(a) object mask. (b) matching result.

Figure 3.19: Performance of circle pairing applied on an object mask image under snowy night conditions.

Figure 3.19 shows some circle pairing results under snowy night conditions. By efficient pairing the headlights, three vehicles are detected, as shown in figure 3.19($b$).

Observe that the taxi top light is not detected as it cannot be paired with another source of light in its proximity.

# Chapter 4

# Experiments and Discussion

## 4.1 Experimental Setup

This work was developed with actual traffic scenes captured by means of a Sony Digital8 handy video camera DCR-TRV525 using the NTSC standard. Sony Digital8 handy cameras use a standard 8mm videotape and apply 5:1 compression. Depending on the image contents, the encoder adaptively decides whether to compress picture fields separately or combine two fields into a single compression block. Thus, Digital8 coding can be thought of as something half-way between Motion JPEG and MPEG. The camera was operated on a normal mode with any special feature activated under all weather and lighting conditions. For the reproduction of these images in the laboratory, a video recorder SONY VHS SLV-R1000 was used. The video signal was digitized using a Matrox Orion AGP frame grabber board programmed with the Matrox Imaging Library (MIL) to generate a series of composite color images. The digitized images are stored in RGB format with a resolution of 640 * 470 pixels with 256-color display resolution.

Three different scenes, all in Ottawa, Canada, have been used to collect image sequences and to test the proposed algorithm. The first one is at the intersection of Mann Ave and King Edward Street (site1). The second one is at the intersection of King Edward Street South and Templeton Street (site2). The third one is at the

intersection of King Edward Street North and Mann Ave (site3). These intersections were chosen because we had an easy access to good filming locations in the proximity of our research laboratory. The traffic scenes were recorded from points located on a highpass on Nicholas Street which is about 5 meters above Mann Ave and King Edward Street. The camera was mounted on a fixed tripod.

In our system, images are collected by a fixed video camera manually installed. In order to catch the headlights of coming vehicles, the height, angle and focal length of the camera are very important. To improve the system performance and to ensure the stability of the application, the field of view of the video camera is to be wide enough to cover all coming vehicles in different lanes from one direction. The goal is to use only one camera per direction of traffic in this application for traffic lights control.

## 4.2 Tests and Analysis of Results

### 4.2.1 Tests with Scenarios

We set up two test scenarios for our experiments. In the first scenario, the initial background is free of moving objects. It has two kinds of background representations. One has only the street in the background, the other may contain static objects such as traffic light poles and parked cars that will not move during the test period. This is the simplest scenario since the initial background representation is perfect, there is no need to obtain a genuine background. Thus the background differentiation provides clear segmentation results of motion detection right from the beginning.

The second scenario is designed to have moving objects in the initial background representation. Since the initial background is not a pure one, the motion could not be easily detected by background differentiation at the beginning. After a few seconds, by calculation of the weighted average of the current image and the previous background image, the current background evolves to a representative one.

Both scenarios have been tested under sunny, cloudy, clear night and snowy night

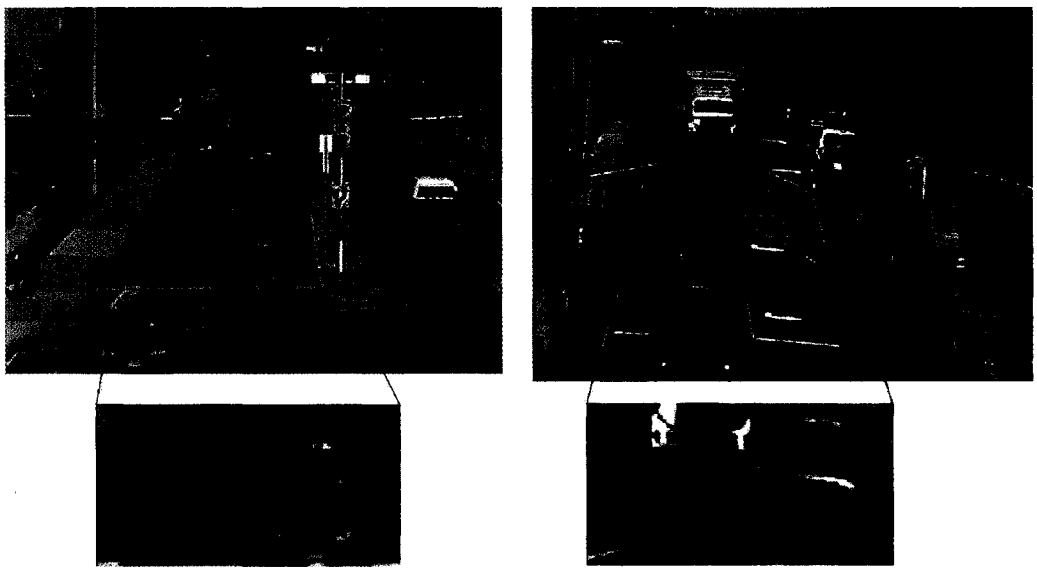
conditions respectively. Thus there are eight test cases that have been investigated. All these eight sequences of test results are presented in Appendix B.1 to B.4. By efficiently using the background updating algorithm, detections are successful in all situations. One situation presents a stopped car contained in the background image that starts to move. The other situation presents a moving car that stops in the sequence. Moreover, the "ghost" phenomenon, as we illustrated in Section 3.2.2, always happens when the moving objects temporarily included in the background start to move. The generic examples presented in Appendix B also demonstrate the excellent ghost removal performance of the proposed system.

The examples presented in the following sections of this chapter aim at demonstrating the performance of all functionalities included in the proposed detection scheme. For compactness a limited set of examples showing more specific cases has been selected, while Appendices B, C and D propose more extensive sets of results. In those examples, the successful detections are marked with green bounding boxes.

### 4.2.2 Tests under Sunny Conditions

Under sunny conditions, the overall shape of the vehicle can be easily observed and detected. The image contrast is very high, and the system automatically selects the full feature detection mode. Shadows can reveal to be useful for the detection. When they appear under or to the side of a vehicle on bright days, as shown in figure 4.1($a$). Because dark shadows underneath a moving vehicle increase the contrast between the vehicle and the road surface, they make the vehicle easier to detect. On the other hand, they generate phenomena that interfere in the validation of vehicle detection as discussed in section 3.3.2. Our experimentation demonstrated that long shadows that extend across multiple lanes of traffic (e.g. beside the white truck in figure 4.1($b$)) from a moving vehicle are difficult to handle. Both of figures 4.1($a$) and 4.1($b$) illustrate the performance of the proposed shadow removal procedure on small and large shadow areas.

In addition, figure B.1, figure B.2 in Appendix B.1 and figure C.1 in Appendix



(a) short shadow. (b) long shadow.

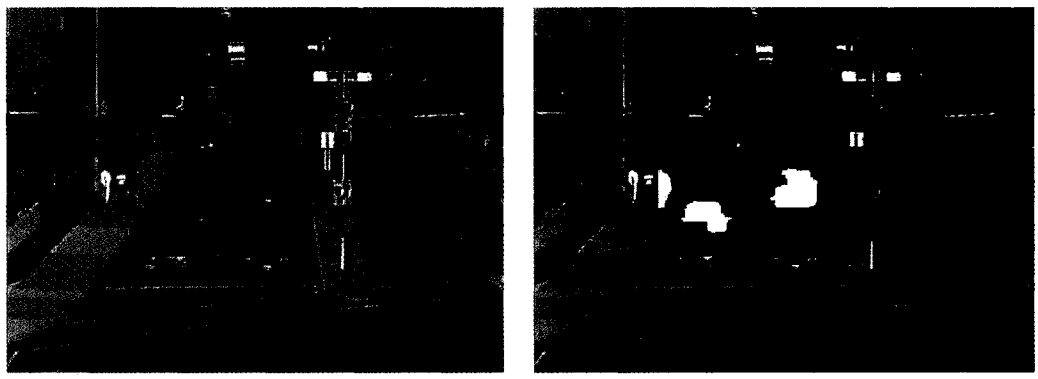
Figure 4.1: Shadows at sunny day.

C.1 present some more examples of vehicle detection under sunny conditions. As the experimental results show, the proposed approach successfully separates the shadows from the vehicles body.

Moreover, our system can also detect pedestrians on the sidewalk, as shown in figure 4.2, or in-line skaters, as shown in figure 4.3, under sunny conditions. In figures 4.3($a$) and ($b$), the moving shadow cast by the boy is successfully excluded from the detection bounding box.

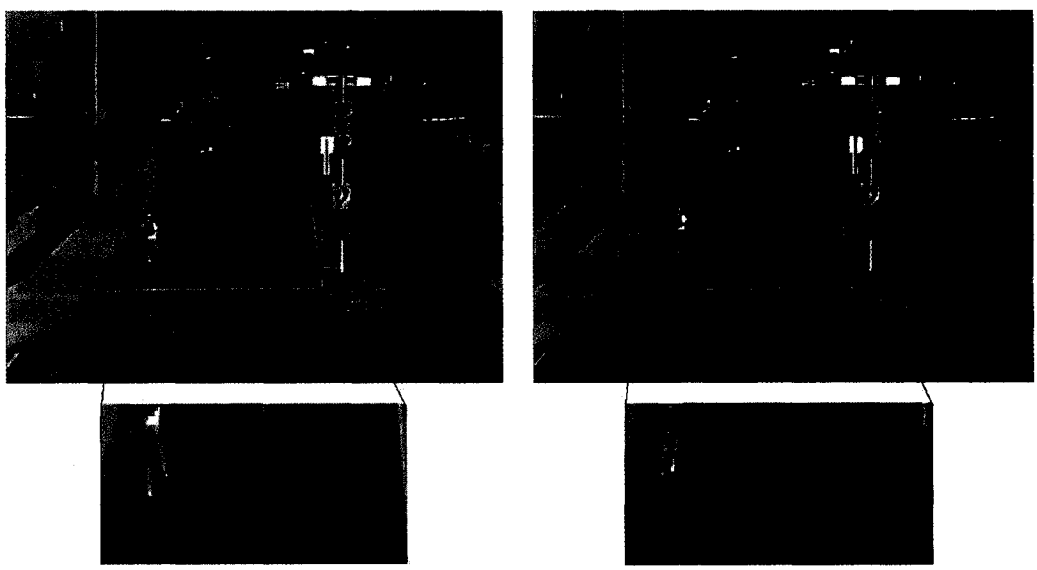
### 4.2.3 Tests under Cloudy Conditions

Under cloudy conditions, vehicles are detected by their full shape. Since the illumination on cloudy days is not as strong as on sunny days, shadow is not a critical feature. The results of the sequences presented in figure B.3, figure B.4 in Appendix B.2 and



(a) original image. (b) object mask image.

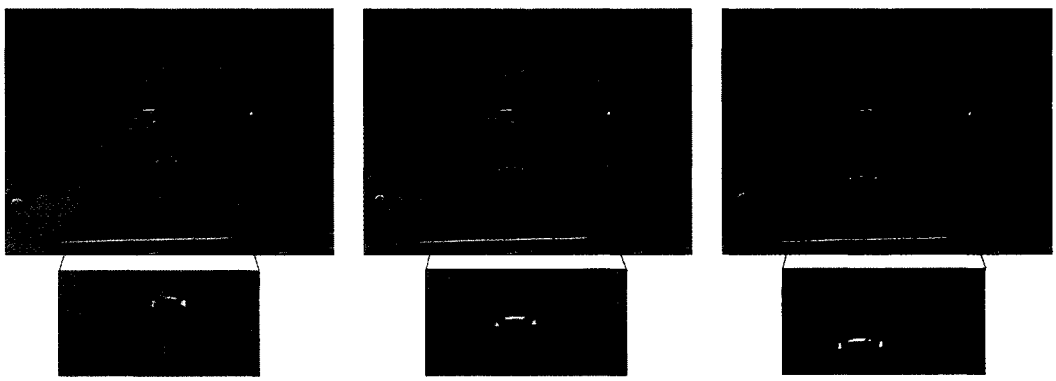
Figure 4.2: Detection of vehicles and pedestrian during a sunny day.



(a) (b)

Figure 4.3: Detection of a skater during a sunny day.

figure C.3 in Appendix C.2 show supplementary examples of successful detection under cloudy conditions. In those sequences, most of the detected vehicles keep driving in the same lane. However our system is not constrained by any specific lanes. We can detect any vehicle that drives along the road, turns left or right, or even makes a U-turn. Figure 4.4 shows an interesting example, in which the moving car makes a U-turn and crosses two lanes to the walk side.



(a) (b) (c)

Figure 4.4: Detection of a vehicle changing lane on a cloudy day.

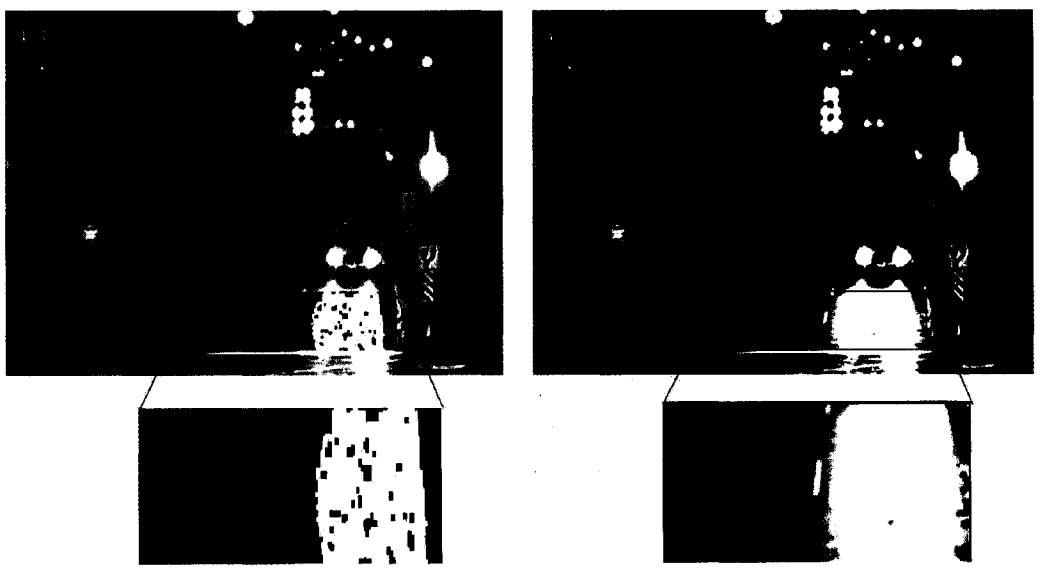
In addition, our system can also detect other moving objects on the road, such as the bike in figure 4.5. Both the vehicle's head and bike's shape are segmented, as shown in the object mask figure 4.5($b$). This shows that the system could be easily extended to detect other moving objects under clear weather conditions.

### 4.2.4 Tests at Night

When the system operates at night, vehicles' headlights are the prominent features that can be easily observed and detected. However, strong ground reflections introduce supplementary difficulties in traffic detection and highly affect the detection validation. When headlights project far ahead in front of the vehicle, they create an

(a) original image. (b) object mask.

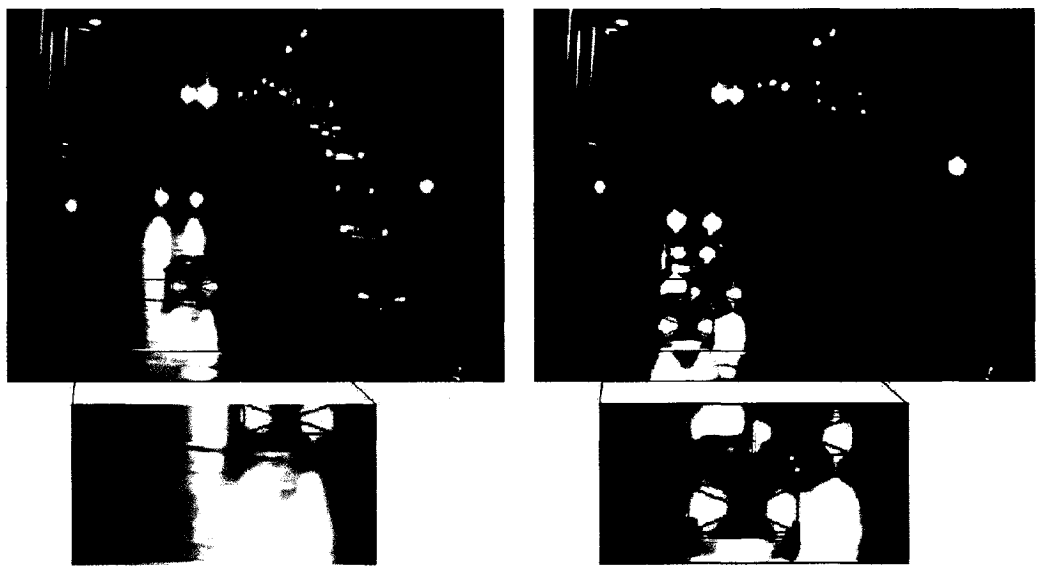
Figure 4.5: Detection of vehicles and bikes on a cloudy day.



(a) object mask. (b) vehicle verification result.

Figure 4.6: Example of long ground reflections at night.

illusion of either two vehicles or one very long vehicle as shown in figure 4.6*(a)*. To compensate for this, a different image processing algorithm has been developed for nighttime detection, which consists of feature extraction, shape matching and pattern correspondence with the help of the HSV color space. After processing, large reflections on the road can be eliminated and are no longer treated as a vehicle, as shown in figure 4.6*(b)*.



(a) (b)

Figure 4.7: Typical examples of vehicle detection at night showing the effects of ground reflection, partial beam interference and headlights glaring.

Figure 4.7 shows two typical examples of vehicle detection at night. In figure 4.7($a$), one vehicle arrives in the front window with intense headlights reflection ahead. Meanwhile it is followed by another car. The light beam from the car behind is in partial occlusion with the front car. Figure 4.7($b$) illustrates a more complex situation in which four cars are coming in close succession. Their headlights and ground reflection are mixed up, which misleads the detection at night. The test results demonstrate that our algorithm can successfully filter out most of the reflection under

low illumination conditions.

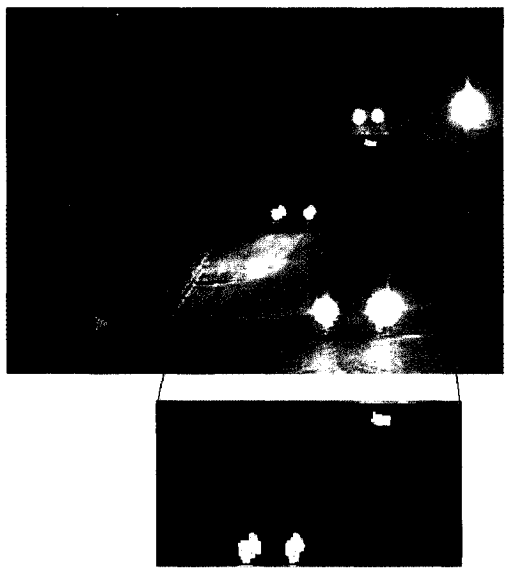
In addition, figure B.5, figure B.6 in Appendix B.3, and figure C.3 in Appendix C.3 illustrate some more examples of vehicle detection under clear night conditions. The experimental results show that the system can successfully separate the headlights from small and large ground reflection lights, and then detect the arrival of moving vehicles.

### 4.2.5 Tests at Night under Snowy Conditions

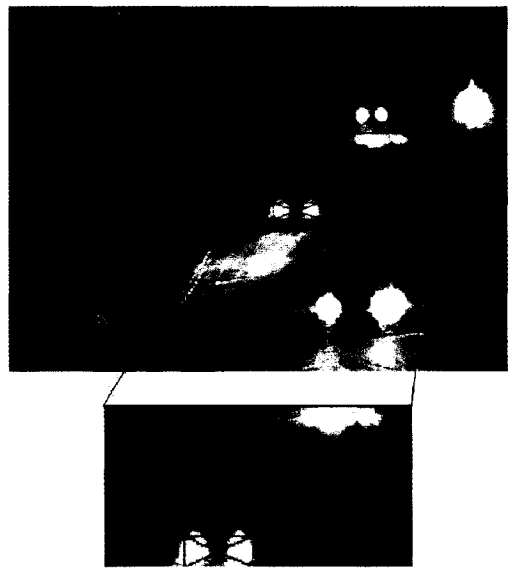
A snowy night is a typical bad weather. Under these conditions, all the objects on the street, such as ground, trees or vehicles are covered with snow. The strong reflection characteristic of snow makes headlights and their reflections on the ground confusing. It is then hard to observe the headlights clearly even by human eyes. Moreover, heavy snow adds noise to the images, which also affects the validation of detection. By applying the headlight-based detection mode, the proposed algorithm does not need to filter out the noise as in [16].

Figure 4.8 presents two examples showing vehicle segmentation and verification results under snowy night conditions. In each example, small reflections on the ground are detected as shown in figures 4.8($a1$) and 4.8($a2$). With the vehicle verification function, they are removed successfully as shown in figures 4.8($b1$) and 4.8($b2$).

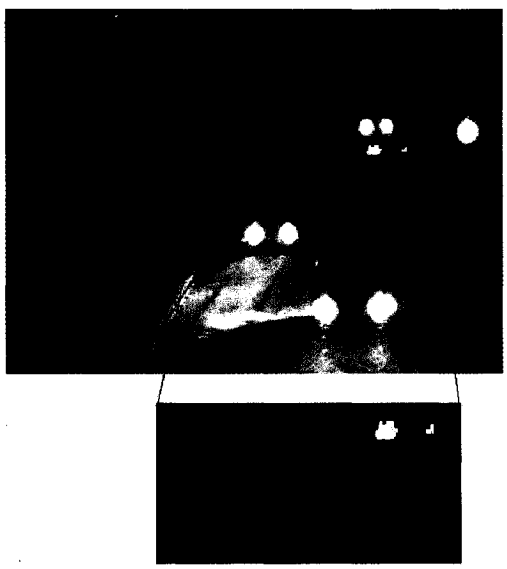
Figure 4.9 shows another complex situation in which there are two or more cars approaching the front window. More specifically, one of these cars stops before the intersection and the other is passing the window to make a right turn. Different status happen in the same window at the same time. The proposed approach only detects the moving vehicles as shown in figures 4.9($a$) and 4.9($b$), without detection of the stopped cars. In addition, figure B.7, figure B.8 in Appendix B.4 and figure C.4 in Appendix C.4 show some more examples of vehicle detection under snowy night conditions. These successful results demonstrate that the system is efficient and robust enough to deal with complex situations under difficult weather conditions.



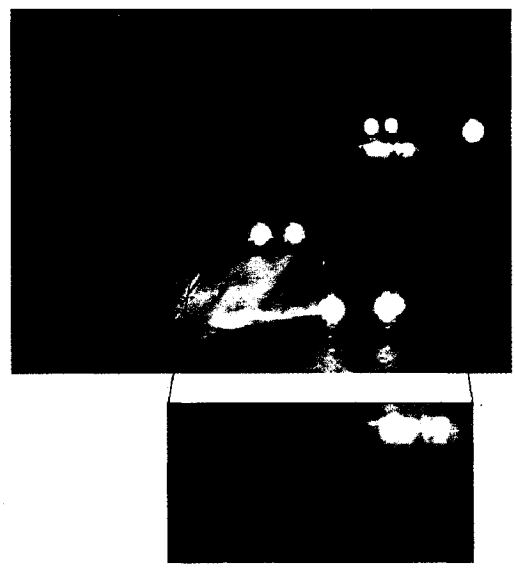
(a1) object mask.



(b1) vehicle verification result.

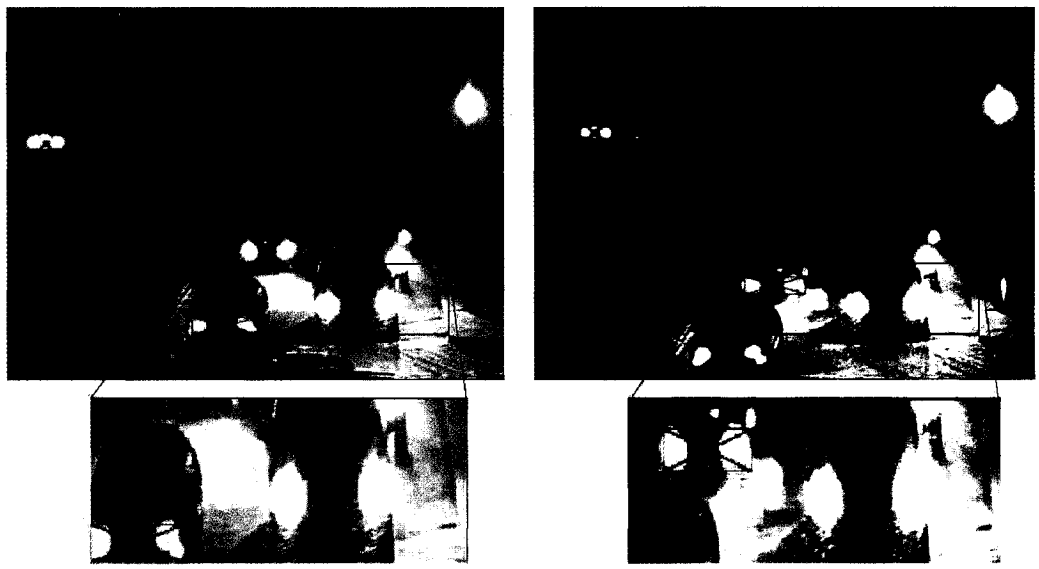


(a2) object mask.



(b2) vehicle verification result.

Figure 4.8: Results of vehicle segmentation and verification during a snowy night.



(a) (b)

Figure 4.9: Complex example of vehicle detection during a snowy night.

### 4.2.6 Supplementary Tests

Our approach has been evaluated on a large number of sequences of real traffic scenes under various weather and illumination conditions for detection of vehicles in one or multiple windows. To demonstrate the efficiency of the proposed vehicle verification algorithm, numerous tests have been performed under sunny, cloudy, clear night and snowy night conditions and extensive sets of results are presented in Appendix B, C and D.

More specifically, Appendix D contains the results of tests that have been conducted on sequences taken at site2 and site3 as described in section 4.1. Without any change on the parameters of our program, the quality of these detection results demonstrates the robustness and adaptability of the proposed system to various configurations of the environment.

| Step | Time | Average Time |
|---|---|---|
| Motion Detection | 0.12-0.19(s) | 0.14s |
| Feature Extraction | 0.01-0.02(s) | 0.01s |
| Vehicle Verification | 0.01-0.21(s) | 0.06s |

Table 4.1: Running time for each step.

## 4.3 Performance Evaluation

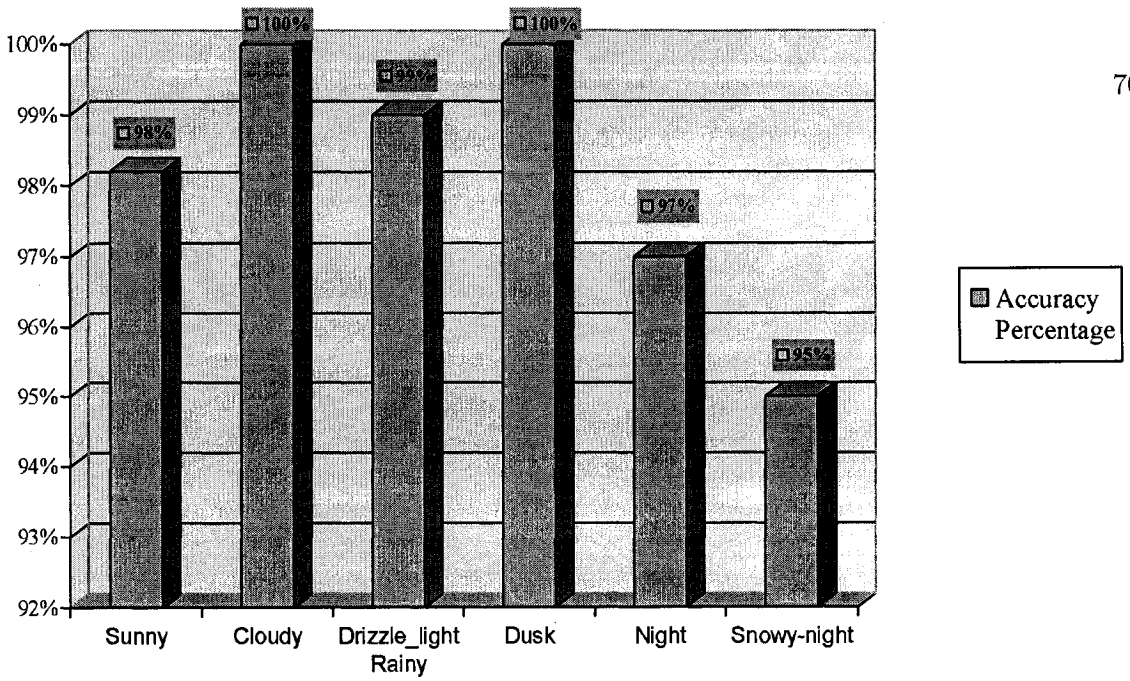
The approach has been encoded using C++ language and the Intel's Computer Vision library. A grabbing rate of 5 frames per second is achieved without code optimization on a Pentium II 350 MHz processor. The average processing time of the detection algorithm for each frame is around 0.2s excluding grabbing time. The running time for each step is listed in Table 4.1. Among them, the fastest step is feature extraction which only takes 0.01s on average. As a consequence of pixel based computation on the window area, motion detection is the slowest step with an average of 0.14s. The most variable step is vehicle verification, which varies from 0.01s to 0.21s depending on how many vehicles are in the observation window to be verified.

The detection accuracy has been evaluated by calculating the detected vehicle rate as follows:

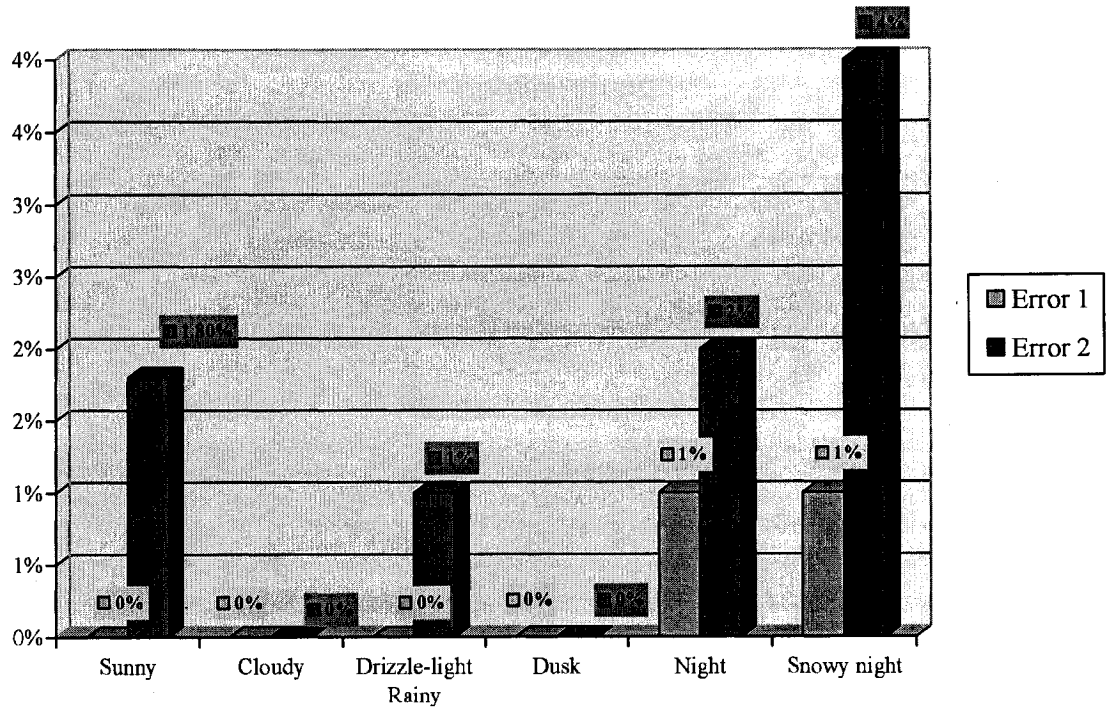
$$rate = A/B * 100\% \tag{4.1}$$

where A is the number of vehicles successfully detected in the observation window by the system over a given period of time; B is the number of vehicles manually observed in the same observation window.

The system has been tested under six different kinds of illumination and weather conditions: sunny, cloudy, drizzle-light rainy, dusk, night and snowy night. Among these real traffic sequences, the one taken at dusk is the longest and contains 18,400 images. While the other sequences have around 2,400 images. Figure 4.10($a$) shows the comparison results of the detection accuracy under these six conditions. The accuracy rate reaches 100% during cloudy and dusk periods, and 95% at snowy night.




(a)




(b)

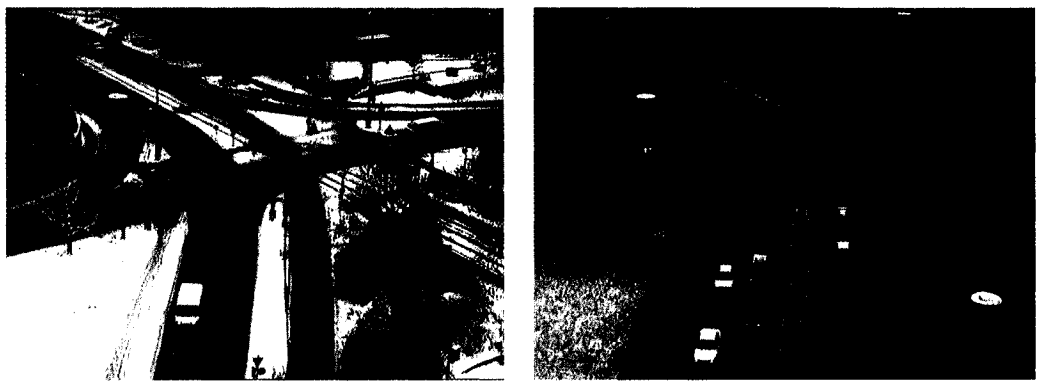
Figure 4.10: Performance comparison under different weather and illumination conditions. (a) Accuracy rate comparison; (b) Error rate comparison.

As expected, the detection accuracy under high illumination and clear weather conditions is higher than under low illumination and bad weather conditions.

Further analysis of the error detections is shown in Figure 4.10($b$). We category all the detection errors into two types. The false detections which mistreat non-vehicles as vehicles are classified as Error1. Error2 indicates the missed detections of vehicles. When the efficient shadow and ghost removal functions that are proposed and activated for the FF detection mode, there are no Error1 observed under sunny, cloudy, drizzle-light rainy and dusk conditions. Whereas, under the HB detection mode, the vehicle verification module could misinterpret some reflections on the pavement portion which are then classified as vehicles. In spite of that, the rate of Error1 is still limited to 1% under both night and snowy night conditions. By comparing of the rates of the Error1 and Error2, we find that the Error2 is the main type of errors, as it is higher than Error1 over all the situations. This is explained by the fact that Error2 detected under sunny and drizzle-light rainy conditions are caused by the imperfection of the segmentation of background or time differentiation. Since our segmentation approach is intensity based, vehicles whose intensity is similar to the road surface are sometimes missed or detected as fragments that are too small to be reliably separated from noise. On the other hand, at night or snowy night, headlights are usually mingled with strong reflection on the ground or lights from other vehicles, which significantly interfere with the detection process. Depending on the influence of the interference, the rate of Error2 might reach 2% at night and 4% at snowy night. In addition, when multiple vehicles occlude each other, they happen to be detected as a single vehicle. But as the goal of the system is to detect moving vehicles approaching the intersection, and not to count or even to recognize them, merge vehicles detection does not degrade performance. Overall, the results of operation of the algorithm compared with manual observations of images demonstrate that the proposed system is robust enough to achieve reliable detection of vehicles under a wide variety of illumination and weather conditions.

## 4.4 Limitations of the System

During the development and the test phase, some limitations have been observed. The main limitation of the current system corresponds to the situations in which it is impossible to detect vehicle headlights when the headlight-based detection mode is chosen. For instance, when the camera's position is too far away from the observation area or positioned at a high elevation with respect to the detected vehicles, an accurate detection can not be achieved. Figure 4.11 shows examples of images from a sequence taken from a camera installed on a tall building in Karlsruhe in Germany. Figure 4.11($a$) is taken at a clear winter day, while figure 4.11($b$) is taken at a snowy day. The distance between the road and the camera being large, the vehicles' headlights can not be clearly detected in these images. This limitation could be solved by placing some constraints on the camera height and angle.



(a) (b)

Source: Institut für Algorrithmen und Kognitive Systeme, Universität Karlsruhe.

Figure 4.11: Limitations due to camera placement.

Another limitation is observed in sequences in which the intensity of moving objects is quite similar to that of the background (e.g, the black car in figure 4.11($a$)). Under such conditions, the validation of the detection is degraded, when the full-feature mode is used. This comes from the fact that the segmentation is not really

effective or only small differences can be extracted by the background differentiation technique.

Another constraint is illustrated in figure 4.12, when a vehicle is moving very slowly. As the absolute moving distance is pretty small between successive images, the system can hardly detect the movement of the vehicle, as shown in figures 4.12($d$) to 4.12($h$) where each frame is separated by about 0.2 seconds. Here since the vehicle has already been detected in figure 4.12($a$), ($b$) and ($c$), its slow speed over the last few frames does not affect the window detection result. However if there is a car approaching the intersection with such a slow speed, the system can not detect its arrival at the intersection. This situation usually happens near the intersection when the traffic light turns on red, the coming vehicles gradually slow down and finally stop behind the waiting queue. However, this problem can be overcome by predicting the arrival of cars by means of a detection of their traversal of a second window set far from the intersection, as shown in some of the previous examples. It is assumed that there is no exit along the road between these two windows and that the waiting queue is seldom extended up to the second window. Since the car doesn't intend to stop close to the second window, its speed should be fast enough for the system to detect it. The estimated slowest speed of the car the system can detect is around 10 km/hour.

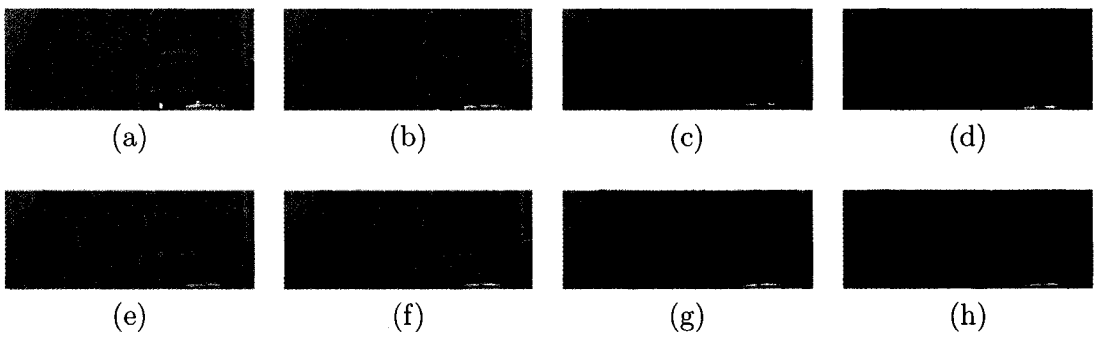



Figure 4.12: Slow speed limitation.

Because the system attempts to mark the arrival when a vehicle crosses a given

window on the roadway, the processing rate is another inherent problem that introduces errors which are independent of this algorithm. It takes 0.4s on average to perform the frame grabbing and to detect moving vehicles. This implies a possible interval of 0.4 sec between two successive vehicle detection. As the window size can be adjusted during the initial phase, the corresponding length and width of the mapped area on the road can be estimated. Therefore, the maximal speed that a vehicle can have without taking the risk of being skipped by the detection system can be computed by dividing the physical length of the detection area by the total average processing time. For instance, if the detection area has 11 meters long, the system could detect the vehicle at a maximum of 100 km/hour speed. Any vehicle driving faster than 100 km/hour could possibly be omitted. Defining a proper window size given the speed limitation imposed in the area where the system is installed is a simple way to ensure proper detection performance. Also, reducing grabbing and detection time would be easy to achieve by using a faster processor and code optimization.

## 4.5 System Applications

In urban area, important improvements on traffic flow can be achieved by an optimal programming of traffic lights according to the current traffic flow rather than time of the day. Real-time measurement of traffic flow at intersections is very important for the analysis of intersection traffic conditions and to adjust traffic lights. At critical intersections, optimizing traffic light cycles is especially effective for the reduction of traffic jams and waiting time. It might also have significant environmental impacts by reducing pollution.

The proposed vehicle detection algorithm has been developed in the context of traffic lights control systems, as shown in the diagram of figure 4.13. Two windows are set and make use of images collected from a fixed camera mounted at the top of a pole and aligned to cover multiple lanes. The first window is located close to the intersection to feedback the current traffic volume at the intersection. The second

window of interest is set to correspond to an area located at a distance from the intersection. This second window allows the implementation of an anticipative controller for the traffic lights. The status of each window is then taken into account to optimize the lights cycle. Usually four cameras should be needed at main intersections.

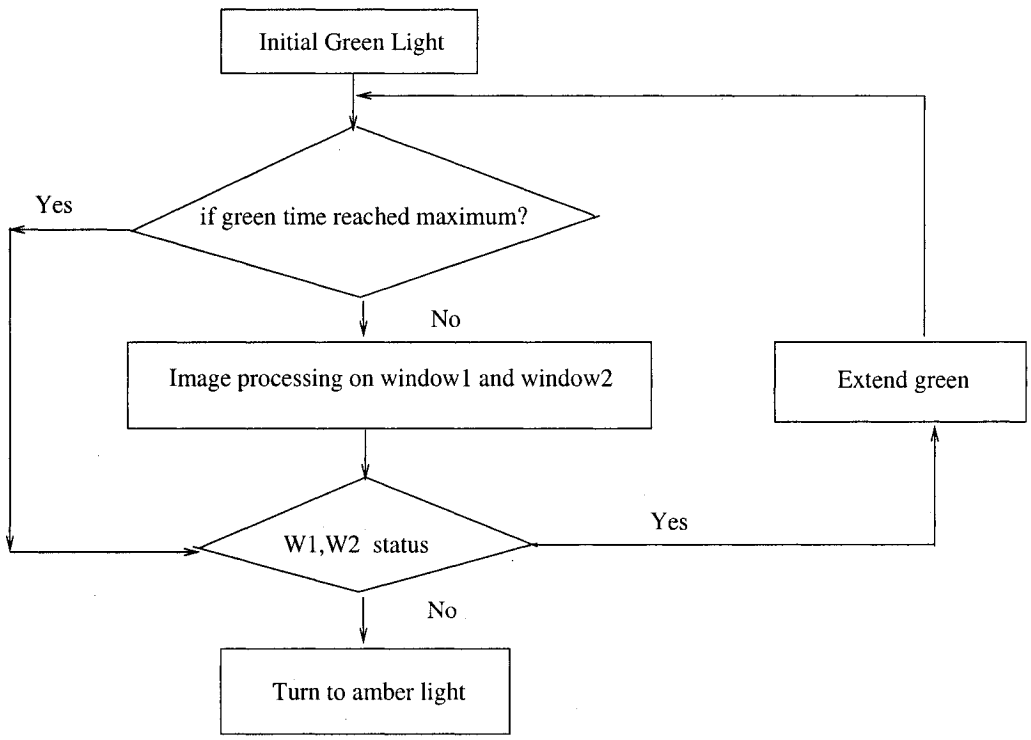



Figure 4.13: Traffic light control application of a vehicle detection system based on two detection windows.

Another possible application of this system is to measure vehicles' speed. When an initial movement is detected in the remote window, the time is recorded and compared with the time of the initial movement in the near window. Knowing the distance between windows and estimating the time it takes the vehicle to cross this distance allows to estimate the speed of the vehicle. The speed of a vehicle and the period of time spent by the vehicle to pass through one detector zone can also be used to estimate the length of the vehicle. This would find interesting applications

on streets with a single lane and no exit along the road between these two detection windows.

In addition, this system could be applied to detect specific traffic activities by setting observation windows on specific lanes. For instance, observation of the volume of vehicles that turn right, turn left or go straight, could help in optimizing the use of existing roadways.

# Chapter 5

# Conclusion and Future Work

## 5.1 Summary of the Thesis

This thesis presents an approach for the detection of moving vehicles based on color image processing under various uncontrolled weather and lighting conditions.

Some sensors currently used for traffic monitoring, including magnetic loops, infrared and videos, are introduced in Chapter 1. Their advantages and disadvantages are discussed and a classical vision system structure for road-traffic monitoring is introduced.

General image processing algorithms applied to moving object detection are analyzed in Chapter 2, along with some more specific methods and systems dedicated to traffic monitoring. Their respective strengths and weaknesses are discussed. It is observed that in spite of a significant amount of literature on the problem of video traffic monitoring, no efficient and feasible methods have yet been found to deal with the complex problem of various weather conditions. This still remains unsolved issues in the field of vision based detection systems in outdoor scenes.

In order to override these limitations, a new vehicle detection system is proposed in Chapter 3. It takes advantage of motion characteristics, of the HSV color mapping and of feature characteristics of vehicles to provide a robust method for detecting moving vehicles in natural environments. The system is auto-adaptive as it automatically

selects the most salient features to detect under the given environmental conditions. The procedure to extract objects' characteristics reveals to be straightforward as the goal is not to identify and classify vehicles but rather to detect their arrival within a given area delimited by a predefined window of interest.

The proposed algorithm provided excellent experimental results on sequences of color images gathered in outdoor scenes with standard video cameras, as demonstrated in Chapter 4. The system has been tested on 10 real traffic sequences containing a total of 40,000 images for which the arrival of vehicles has been manually counted to provide a ground truth reference for these tests. Overall, by averaging the detection accuracy under high and low illumination conditions, the total detection accuracy reaches 97.6%. The approach demonstrated excellent robustness to lighting and weather changes, succeeding to detect vehicles under the worst snowy night conditions. Shadow detection and ghost removal modules also showed very good performance. The average detection time of 0.2s appears to be sufficient for operation in usual urban areas.

## 5.2 Thesis Contributions

The proposed system has numerous new features and advantages over existing approaches found in the literature.

First, the new algorithm for motion detection automatically selects the most suitable differentiation method and thresholding value according to the illumination variation rate. It applies background differentiation under slow illumination changes and uses successive frames differentiation under sudden illumination changes. This enables the system to operate under a larger range of lighting conditions without modifications to the tuning.

Second, the background image is adaptive to the real environment. It does not rely on any pure initial background image for initialization and its updating speed is auto-adjustable according to the real world changing pace. As a result, a faster

update of the background representation is achieved when there is a rapid change in the brightness of the roadway surface, especially under rainy weather or snowy weather, and when there is an increase in traffic flow.

Third, taking advantage of the HSV color mapping scheme, the system automatically selects the most salient features under given environmental conditions. This feature enhances the system's robustness for detecting the arrival of vehicles in daylight, twilight or nighttime conditions as well as under different kinds of weather conditions.

Finally, the addition of shadow removal functionalities under clear weather conditions and of direct feature extraction and matching operation under bad visibility significantly contributes to decrease the error detection and highly increases the system robustness.

This thesis brings a new insight in the technology of moving vehicle detection in outdoor environments by bringing together existing approaches that have been developed to deal with specific problems and bridging the gap between them. The result is a robust, accurate and efficient detection system. The work presented in this thesis already led to two conference papers [31, 32] and one journal publication [33].

## 5.3 Future Work

Traffic monitoring by image processing is a very interesting topic, that responds to practical requirements. But as these systems are designed to operate in outdoor environments, they are submitted to severe constraints. Some further developments might improve the performance of the proposed system.

First, experimentations should be conducted to measure the performance of the system when it runs in the field for a long period of time without interruption. The experiments presented in this thesis demonstrate excellent performance and robustness over short continuous video sequences that usually cover about 10 minutes each, but a long time test should be conducted to ensure stability.

Second, as we have mentioned in the discussion of the system's limitations, a successful detection appears to be more difficult to achieve when the color or intensity of the moving objects is similar to the background. Although this case only happens at day time, it is the main reason for the few failures to detect moving vehicles. We could consider to take advantage of the recent development of real-time depth computation from stereo camera. Because the shapes of the objects in the scene are not affected by intensity and color, a method combining depth and color is a possible way to solve this problem.

In addition, we would benefit from speeding up computation to increase the frame processing rate by code optimization. Using the latest processor technology would also significantly improve the frame rate.

Finally, our system is designed specifically for vehicles which have two headlights. Extending the system to deal with different types of objects, such as bicycles, motorcycles and pedestrians would also be an interesting direction as it has already been demonstrated that the current system is able to detect such objects under clear day time conditions.

# Appendix A

# Color Models

## A.1 RGB Color Model

As shown in figure A.1, the RGB color space can be represented by a cubic model with its vertices assigned to the different colors. It is composed of the primary colors Red, Green and Blue whose values are between 0 and 255. The diagonal from black to white represents the gray-scale. The RGB color model is also called the "additive primaries" since the colors are added together to produce the desired color. For example adding pure red and green produces yellow, adding red and blue makes magenta, blue and green combine to make cyan, and all three together, when mixed at full intensity, create white.

## A.2 HSV Color Model

The Hue/Saturation/Value model was proposed by A.R.Smith in 1978. It is based on the concept of intuitive color characteristics such as tint, purity and intensity. As shown in figure A.2, the coordinate system is a hexa-cone structure, and the colors are defined inside the top hexagon plane. The Hue is the actual color, or the angle of the point in the hexagon plane. The hue value, $H$, goes from 0° to 360°. The Saturation is the "purity" of the color. The saturation value, $S$, ranges from 0 to

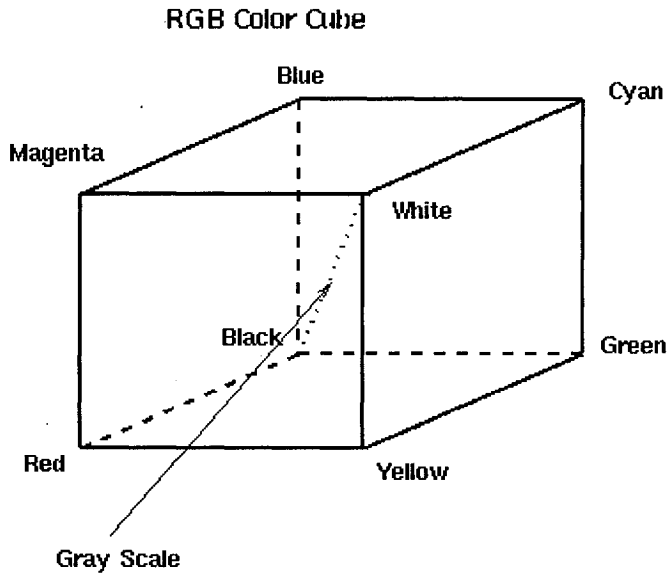



Figure A.1: RGB color model.

1. The Value is the intensity, or brightness, and is the height of the point along the pyramid. The brightness, $V$, also ranges from 0 to 1, where 0 corresponds to black.

## A.3 RGB-HSV Model Conversion

Because the relationship between the standard RGB mapped color representation and the HSV color representation is nonlinear, there is no transformation matrix for the RGB/HSV conversion, the following algorithm is used:

$V = max(r, g, b)$;

$S = (max(r, g, b) - min(r, g, b))/max(r, g, b)$;

$d = max(r, g, b) - min(r, g, b)$;

If $(r == V)$, then $h = (g - b)/d$;

If $(g == V)$, then $h = 2 + (g - r)/d$;

If $(b == V)$, then $h = 4 + (r - g)/d$;

If $(r = g = b = 0)$, then $H = -1$;

$H = h * 60$;
If $h < 0$ then $H = h + 360$;

The resulting $S$ and $V$ values are between 0 and 1, while the $H$ value is between 0° and 360°. In the case where R = G = B = 0, $H$ is undefined, hence it is assumed to be -1.

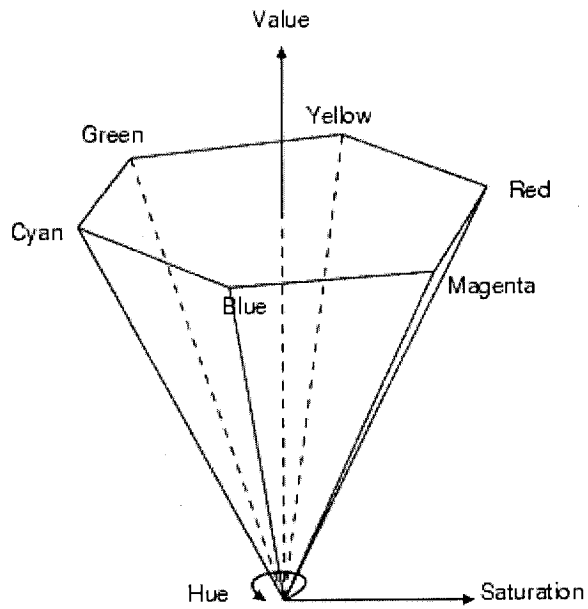



Figure A.2: HSV color model.

# Appendix B

# Sequences Showing Detection Results with Two Test Scenarios under Different Weather Conditions

## B.1 Test Sequence Results under Sunny Conditions

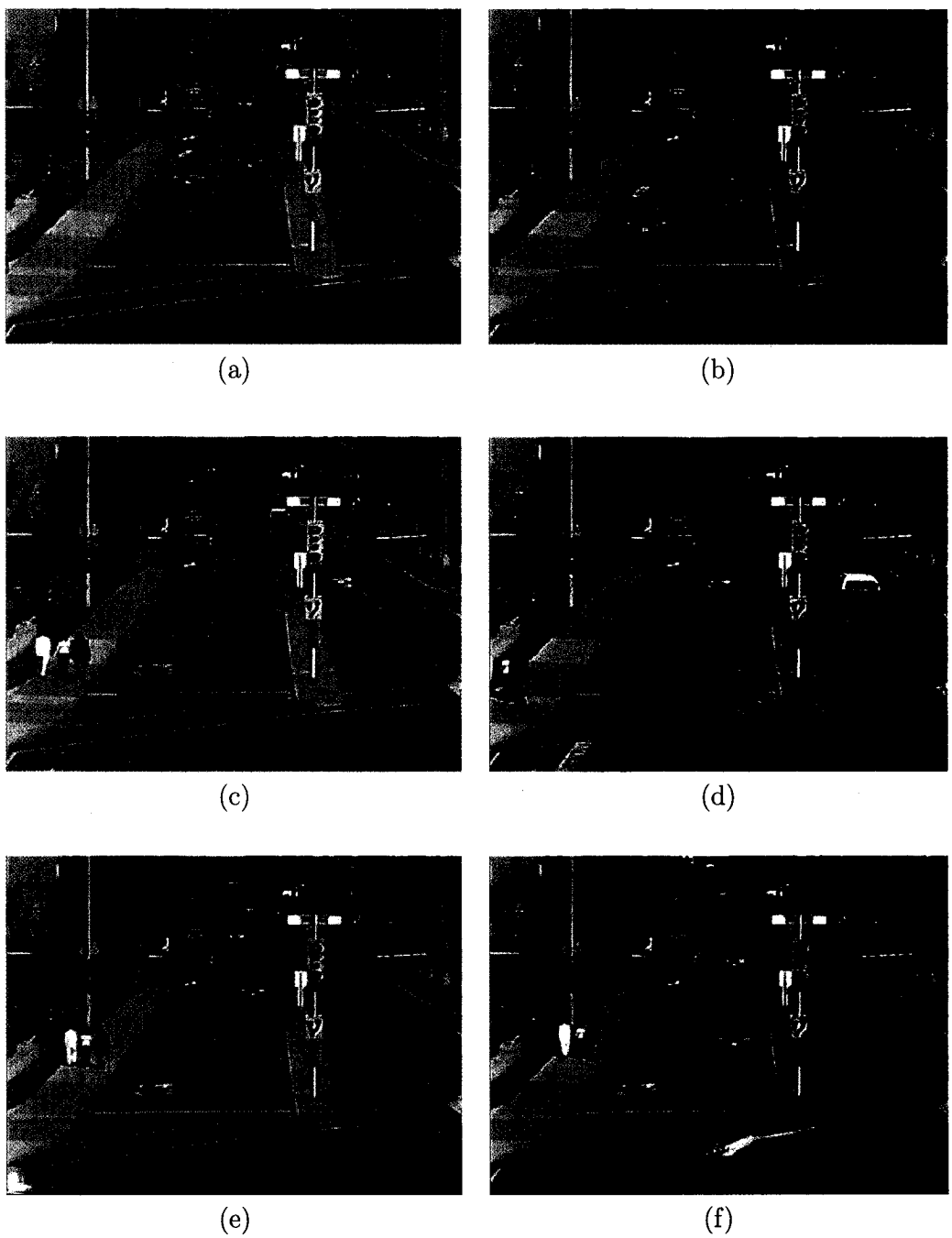
Figure B.1 shows a set of results based on a sequence of images taken at the intersection of Mann Ave and King Edward Street during a sunny day. Under bright illumination conditions, vehicles' shape is the prominent feature that can be easily detected.

In the first scenario, the initial image contains no moving objects in both observation windows. Some cars parked along the sidewalk in the back window (farther from the intersection) of figure B.1($a$), could be regarded as components of the background image since these cars never move during the test. Under bright sunny conditions, the vehicle's full shape can be efficiently detected as highlighted by green bounding

boxes in figures B.1($b$) and B.1($c$). When the traffic light turns on red, the coming vehicle slowly approaches to the intersection and then stops. Since the system merges the stopped car into the background representation at that time, the stopped vehicle is not detected anymore. This procedure is shown in figures B.1($e$) to B.1($f$).

In the second scenario, as shown in figure B.2($a$), the initial situation is that stopped cars are waiting at the intersection. When the traffic light turns on green, the waiting vehicles start to move, and then they are immediately detected by the system, as shown in figures B.2($c$) to B.2($f$). As we illustrated in Section 3.2.2, the "ghost" phenomena appears when a static object initially represented in the background representation starts to move. At the beginning of this test, we have two waiting vehicles which occupy the front window of the initial background image. This situation brings much "ghost" interference for a short period of time until these two vehicles fade out from the background image. From these images we can see that the green boxes only bound vehicles' real size excluding ghost pixels and shadow pixels.

These instantaneous and accurate vehicle detections demonstrate that the background representation is able to update itself according to the fast changes in the real traffic situations. Also the proposed system shows good performance on the shadow removal and ghost removal.

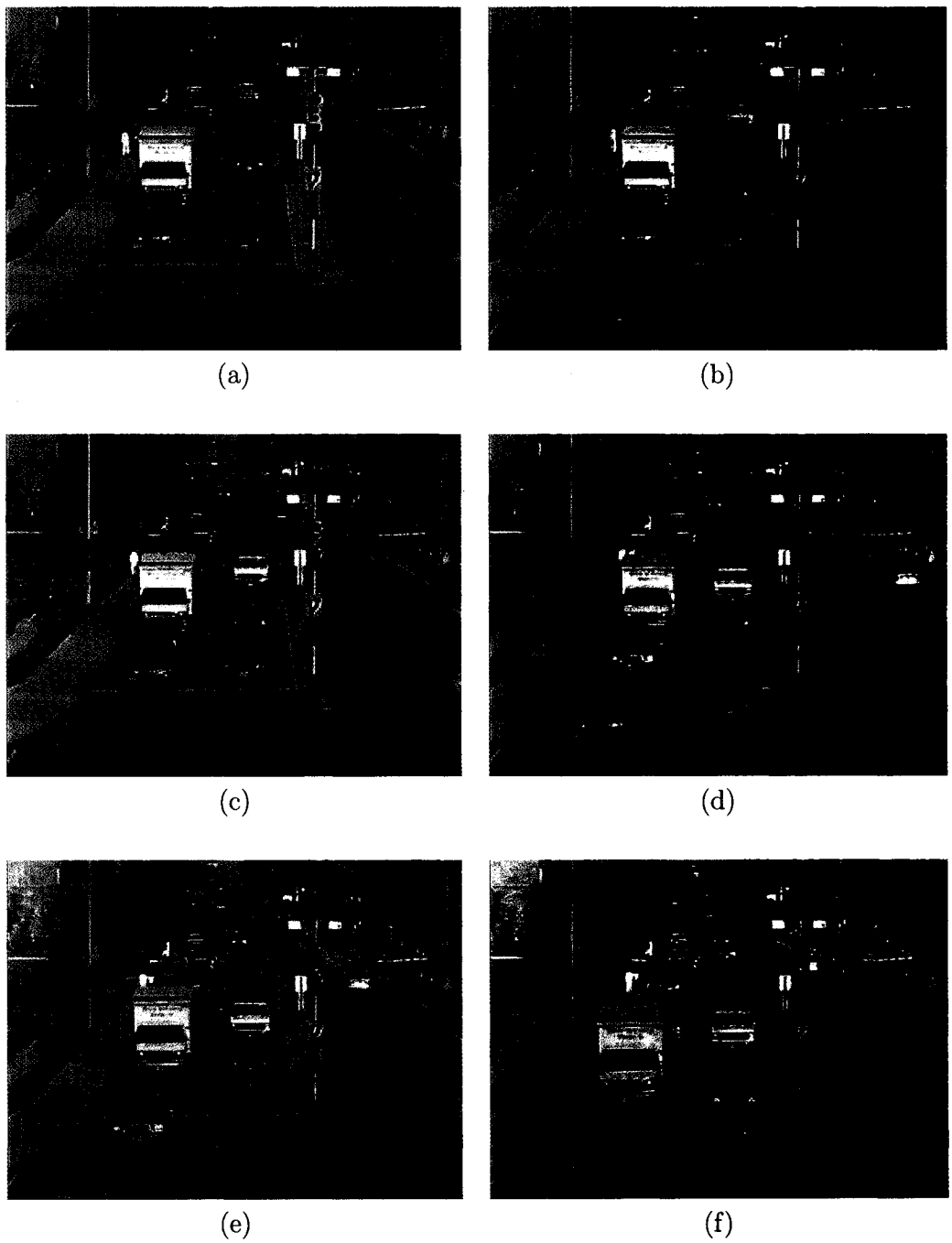


(a) (b)

(c) (d)

(e) (f)

Figure B.1: Test sequence results with the first scenario under sunny conditions.



(a) (b)

(c) (d)

(e) (f)

Figure B.2: Test sequence results with the second scenario under sunny conditions.

## B.2 Test Sequence Results under Cloudy Conditions

Under cloudy weather conditions, vehicles can be detected by their full size.

In the first scenario, figure B.3 illustrates a situation where the initial figure B.3($a$) taken by the video camera does not contain moving objects in both observation windows. As shown in figures B.3($b$) to B.3($d$), the system can successfully detect all the vehicles passing through these two windows. As soon as the vehicle stops in the front window, it starts to be merged into the background image and is no longer detected, as shown in figures B.3($e$) and B.3($f$).

Figure B.4 illustrates the test results based on the second scenario under cloudy weather conditions. Two vehicles are stopped in the front window as shown in figure B.4($a$) such that they are integrated in the initial background representation. When the stopped vehicles start to move, they are immediately detected as shown in figures B.4($b$) to B.4($e$).

These test cases demonstrate the validation and efficiency of the proposed approach under relatively weak illumination conditions as compared with sunny conditions.

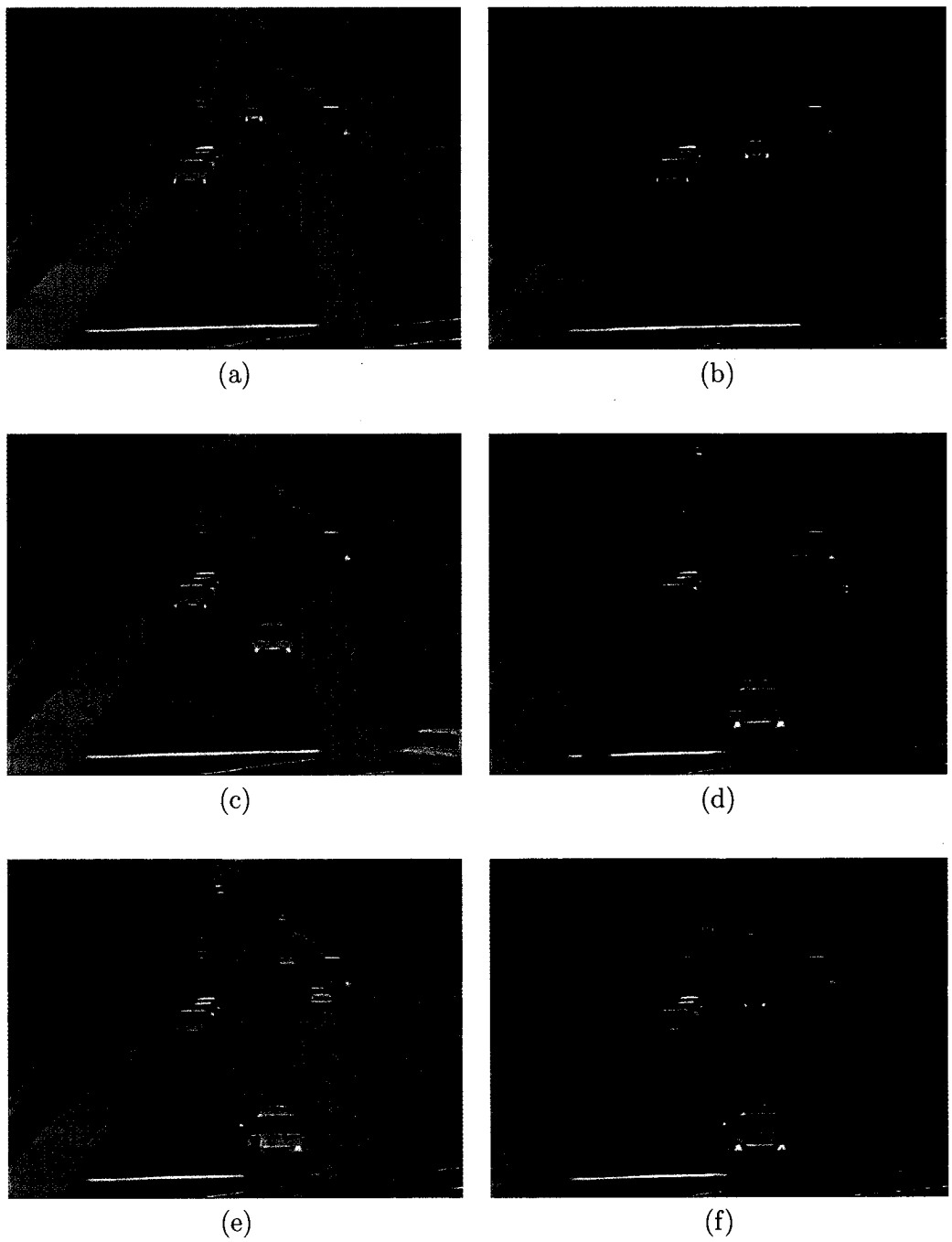
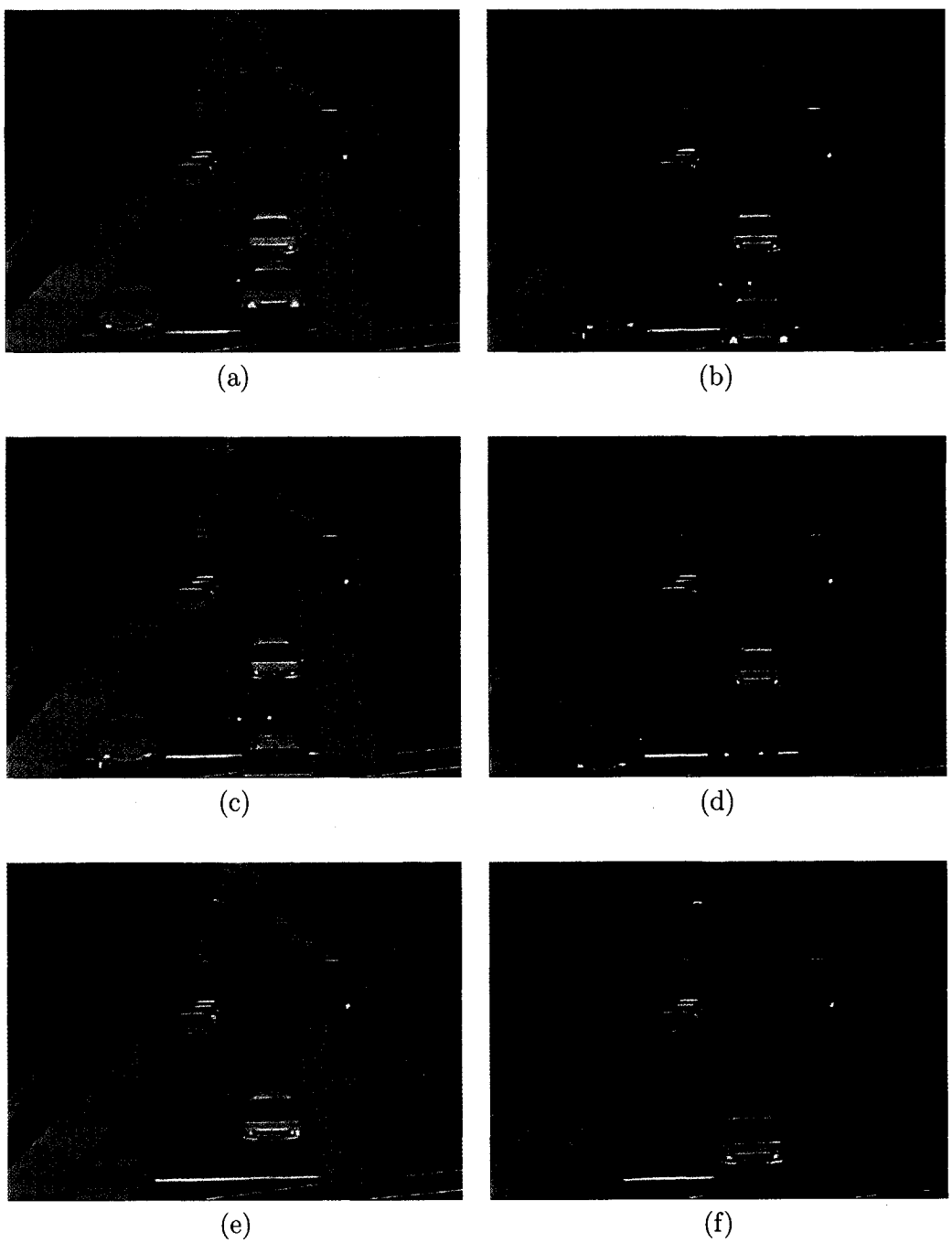


Figure B.3: Test sequence results with the first scenario under cloudy conditions.



(a) (b)

(c) (d)

(e) (f)

Figure B.4: Test sequence results with the second scenario under cloudy conditions.

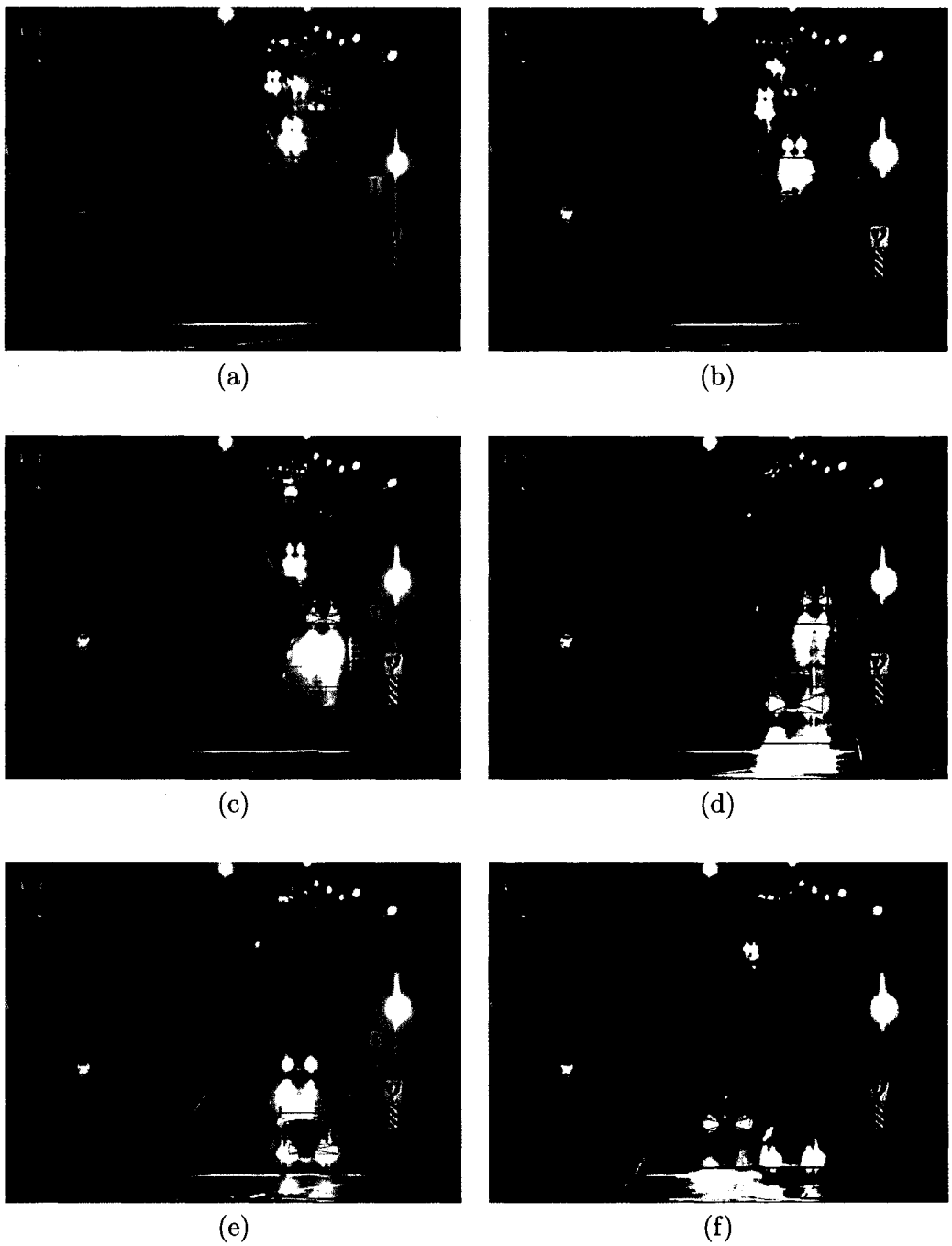
## B.3 Test Sequence Results at Night

When the system operates at night, vehicles' headlights are the prominent feature which can be easily detected.

In the first scenario, with an initial background image exempt of vehicles, as shown in figure B.5($a$), the vehicles' headlights are efficiently detected as shown in figures B.5($b$) to B.5($e$). Later on when the car stops with strong headlights flashing, as shown in the front window of figure B.5($f$), the proposed system is able to merge the static vehicle into background image and do not treat it as a moving object. These detection results demonstrate the robustness of the proposed system that it can efficiently distinguish between the vehicles' headlights and the reflections from ground surface, as well as between static objects and moving objects under low illumination conditions.

In the second scenario, there are two vehicles waiting in the front window of figure B.6($a$). When both of them start to move, they are detected by the system, as shown in figures B.6($c$) to B.6($f$).

These two cases demonstrate that the background updating algorithm can also work very well under low illumination conditions. They also show that the headlights are reliable features to represent vehicles at night.

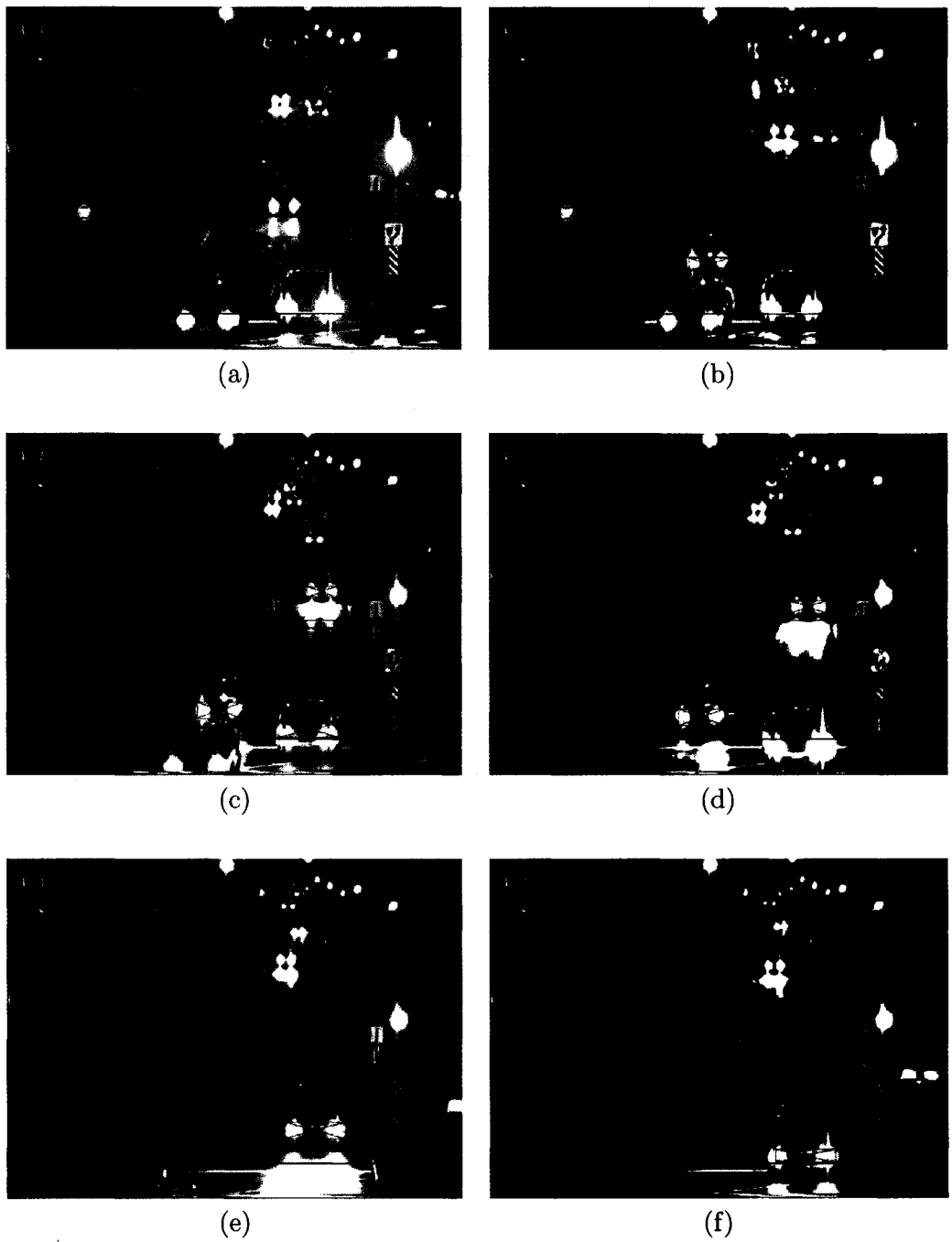


(a) (b)

(c) (d)

(e) (f)

Figure B.5: Test sequence results with the first scenario at night.



(a) (b)

(c) (d)

(e) (f)

Figure B.6: Test sequence results with the second scenario at night.

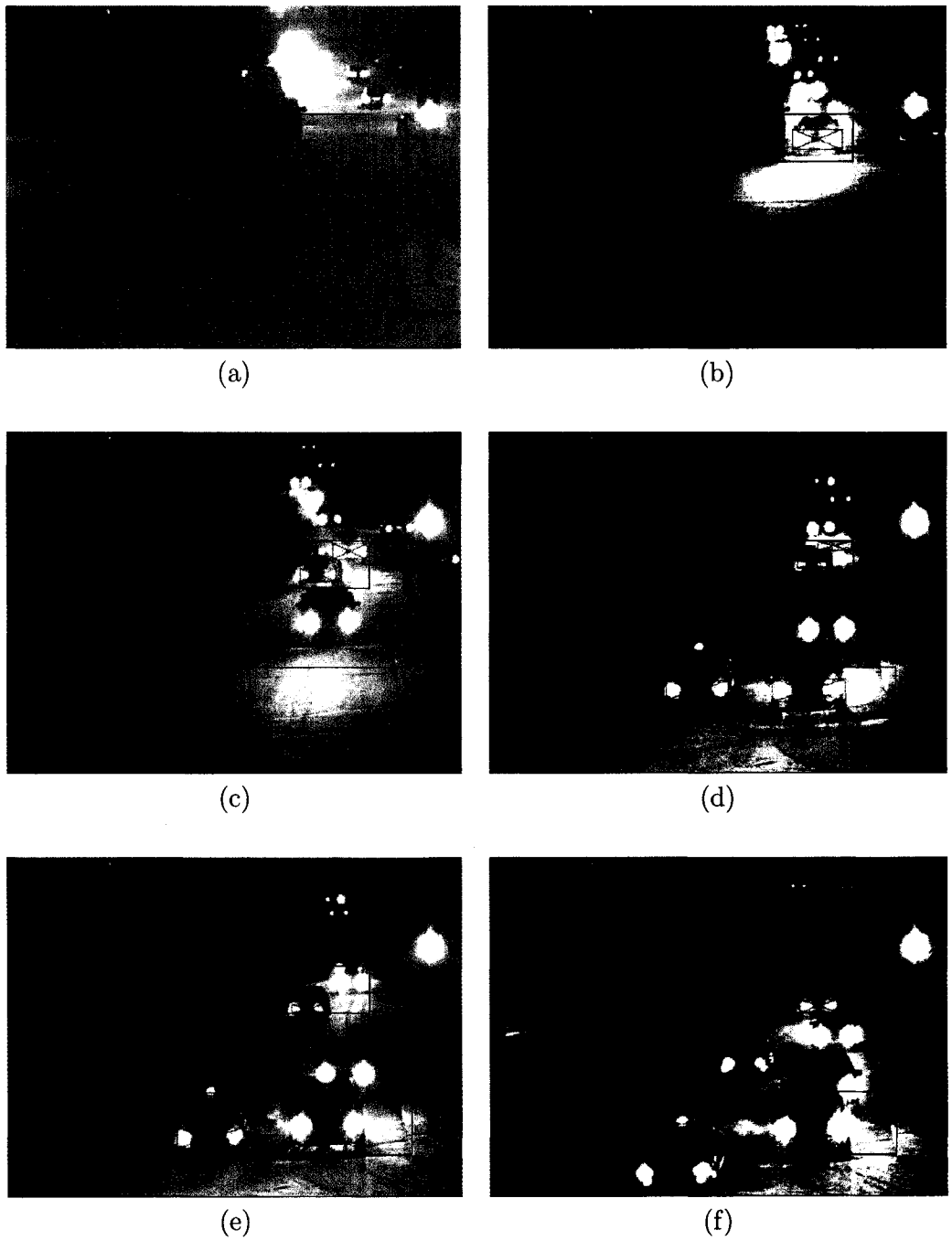
## B.4 Test Sequence Results at Night under Snowy Conditions

A snowy night is a typical bad weather for vehicle detection. At that time, all the objects on the street are covered by snow. The luminosity of reflections and headlights is stronger than under any other weather conditions.

For the first scenario, we start the test from a clear background image that does not contain any moving objects, as shown in figure B.7($a$). In figures B.7($b$) to B.7($d$), all vehicles are detected when they pass through the observation windows. Later on, when the cars stop in the front window , they are no longer treated as moving objects, as shown in figure B.7($f$).

In the second scenario, there are two stopped cars in the initial front window of figure B.8($a$). Later on when they start to move, they are detected by the system as shown in figures B.8($c$) to B.8($f$). We observe that when the headlights of the truck go out of the first window, the detection stops as only headlights are tracked under these conditions.

These two cases demonstrate the robustness of our system in front of extremely difficult weather conditions.

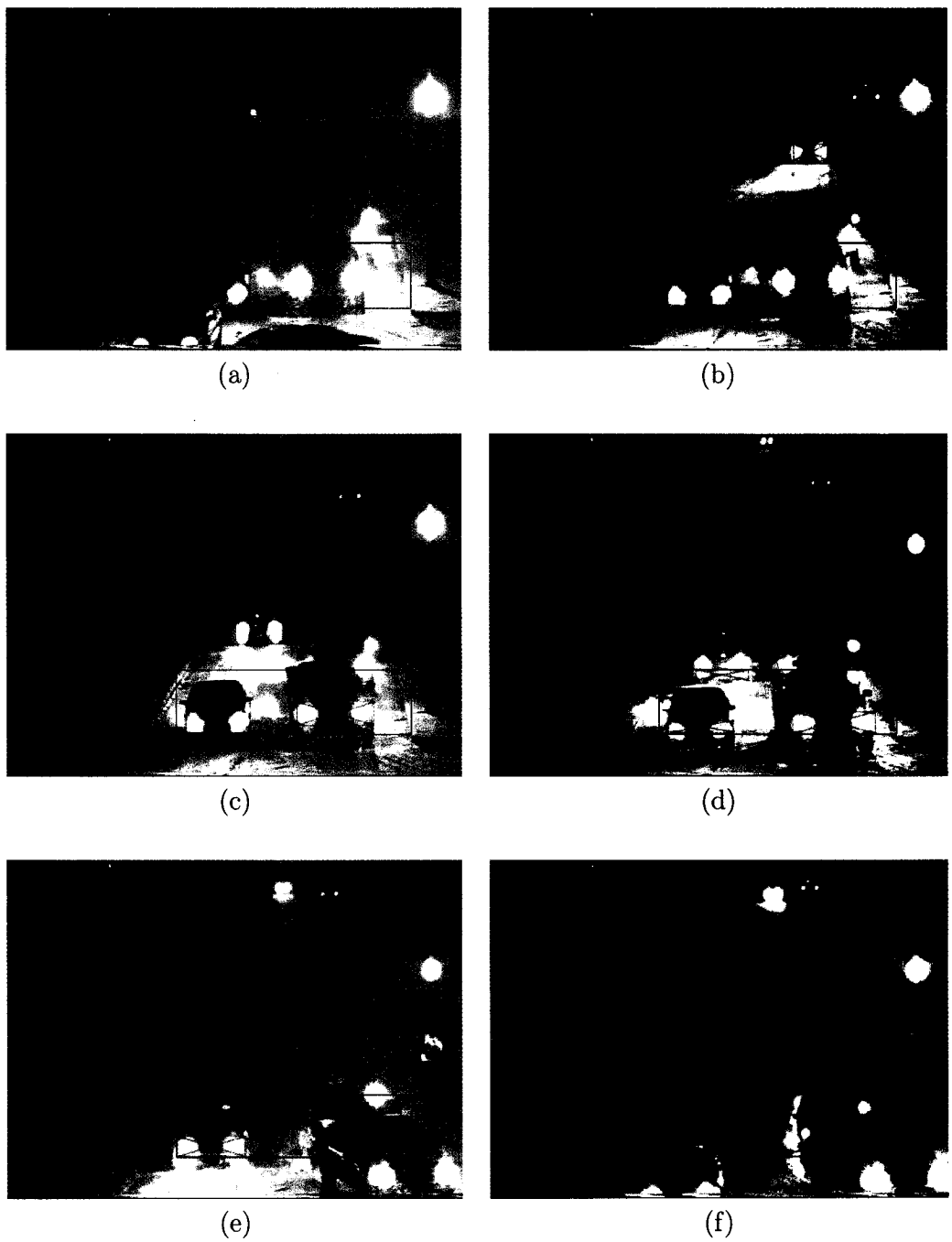


(a) (b)

(c) (d)

(e) (f)

Figure B.7: Test sequence results with the first scenario during a snowy night.



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

Figure B.8: Test sequence results with the second scenario during a snowy night.

# Appendix C

# Sequences Showing Motion Segmentation and Vehicle Verification Results under Different Weather Conditions

## C.1 Test Sequence Results under Sunny Conditions

Figure C.1 presents a sequence of images taken under sunny conditions. Here, we present both the segmentation and the vehicle verification results. As only list a part of the images from the original sequences is shown for compactness, they are identified by a number to show the interval between two successive images in the sequence. The shadow removal function works well, as shown in figure $image218(a)$. The system successfully removes the shadow on the side of the vehicle. However, since some pixels have similar characteristics to those of shadows in the HSV representation, it happens that some of them are mistreated as shadows and then removed from the motion mask. Some holes or gaps in the object mask might then appear, as shown

in figures $image224(a)$ to $image266(a)$. But the vehicle verification function working under the full feature mode efficiently makes up for this limitation, as shown in figures $image224(b)$ to $image266(b)$, ensuring that the detection is accurate.

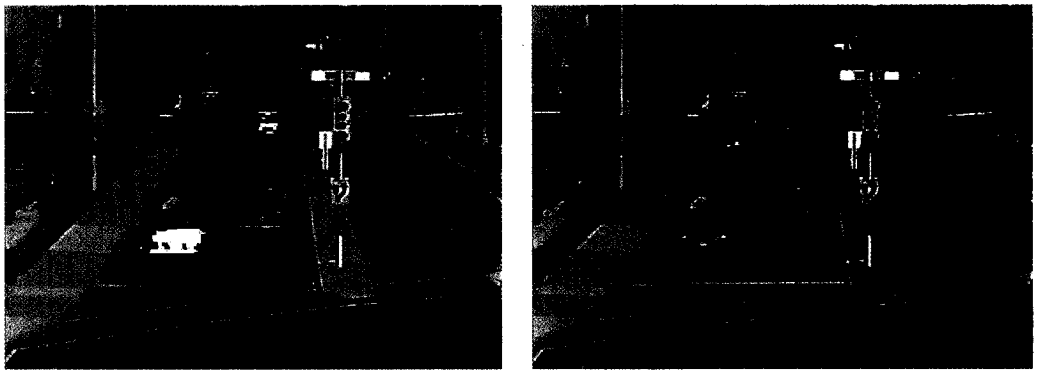


image 218 (a) (b)

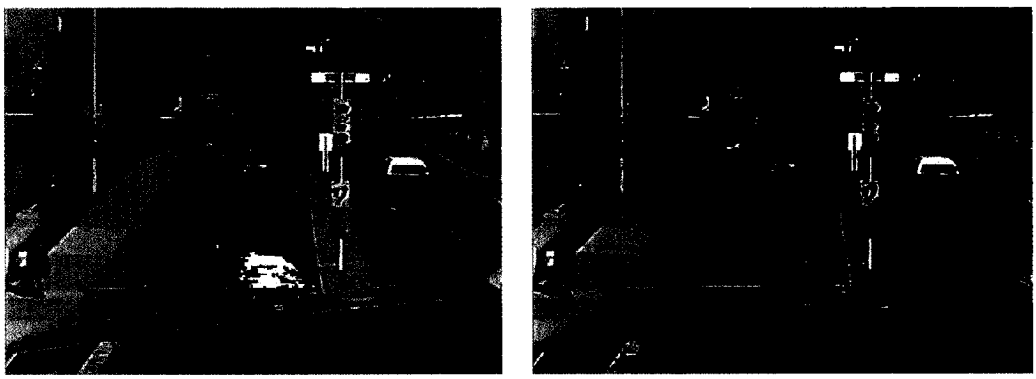


image 224 (a) (b)

image 244 (a) (b)

Figure C.1: Vehicles detection applied on a sequence of images under sunny conditions. a) mask images representing moving pixels; b) detected vehicles represented by bounding boxes.

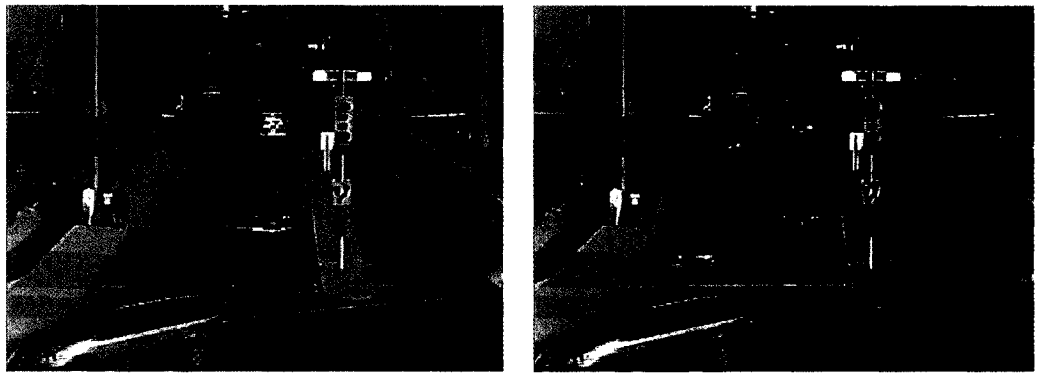


image 260 (a) (b)

image 264 (a) (b)

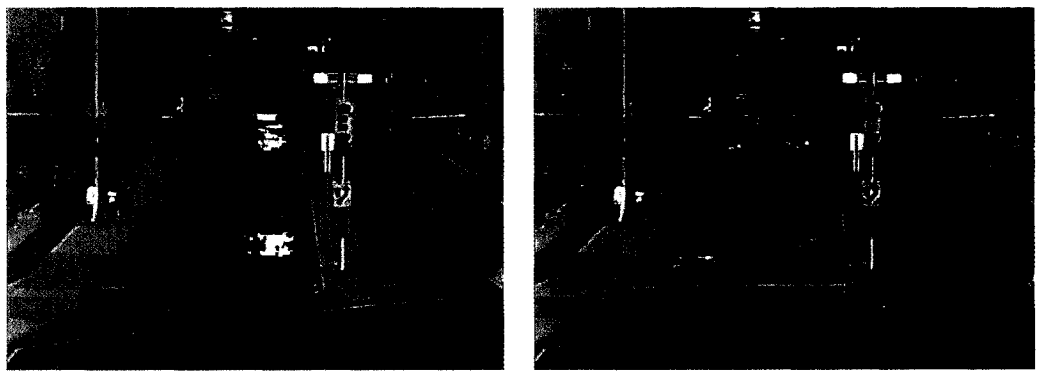


image 266 (a) (b)

Figure C.1: Vehicles detection applied on a sequence of images under sunny conditions. a) mask images representing moving pixels; b) detected vehicles represented by bounding boxes.

## C.2 Test Sequence Results under Cloudy Conditions

This section presents segmentation and verification results obtained on a sequence of images taken during a cloudy day. Figures *image*522($a$) to *image*598($a$) show the detected moving pixels in the images that might include some holes or separated segments as a result of the action of the shadow removal function. But once corrected with the rectangle merging function, the green bounding boxes of the final detection results perfectly correspond to the detected vehicles, as shown in figures *image*522($b$) to *image*598($b$).

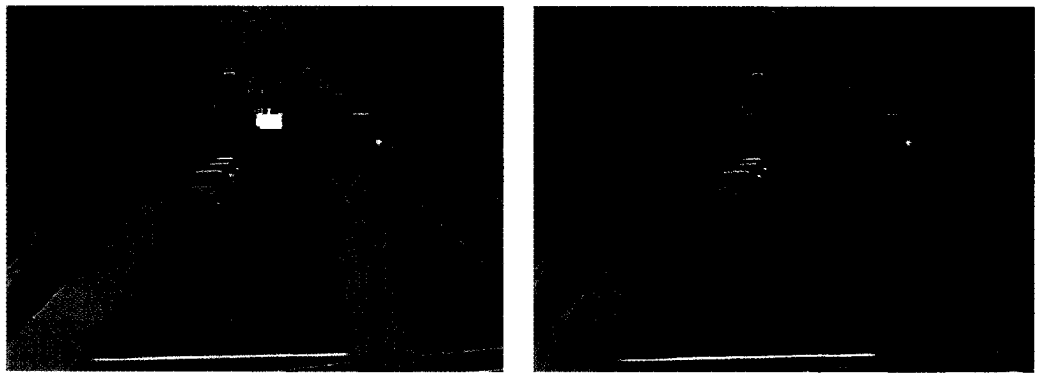


image 522 (a) (b)

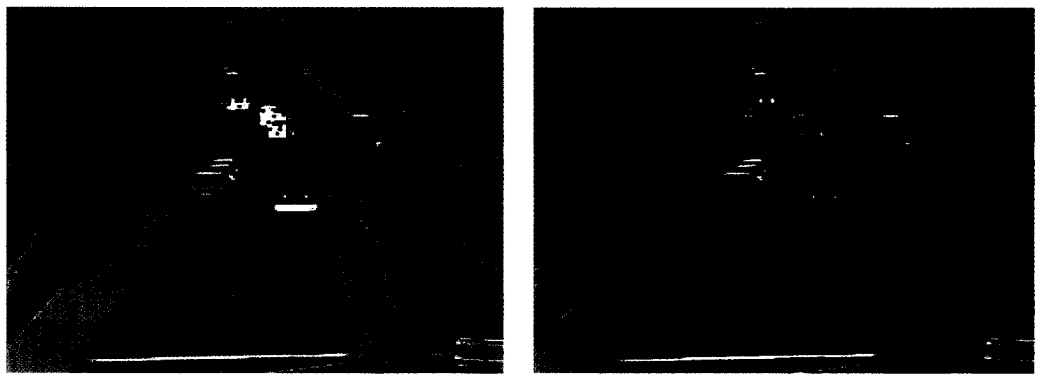


image 546 (a) (b)

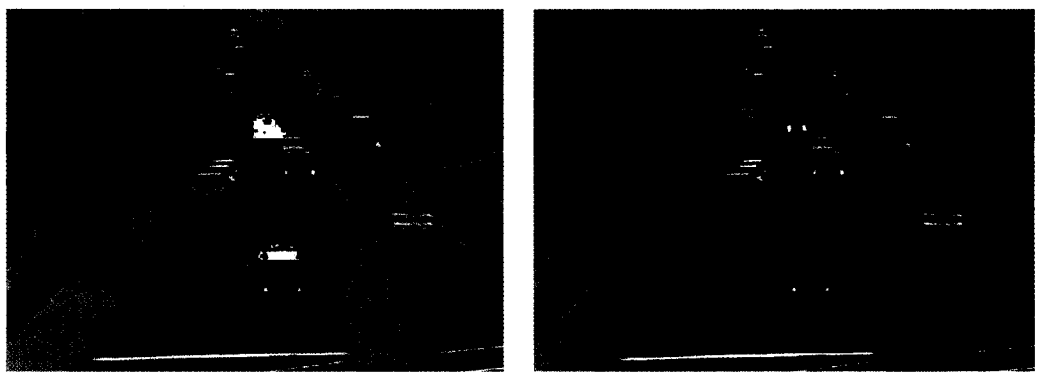


image 560 (a) (b)

Figure C.2: Vehicles detection applied on a sequence of images under cloudy conditions. a) mask images representing moving pixels; b) detected vehicles represented by bounding boxes.

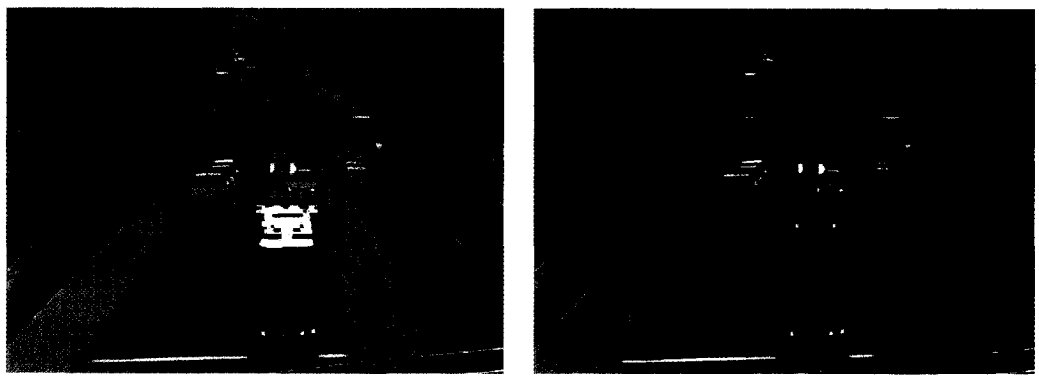


image 575 (a) (b)

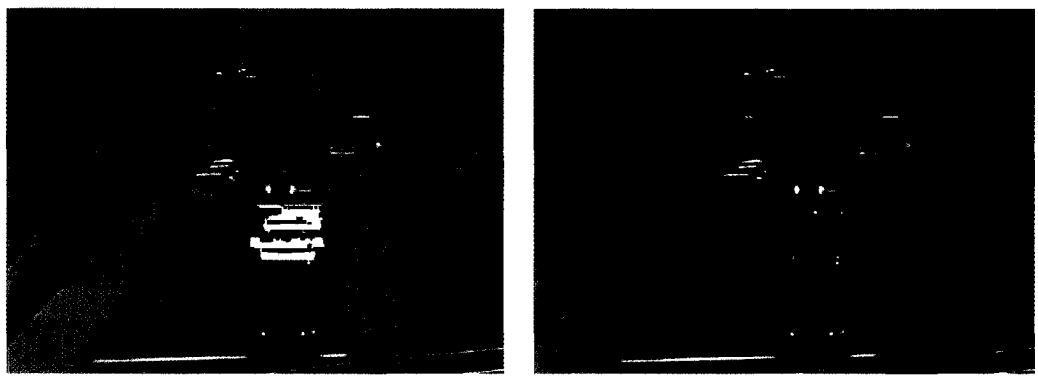


image 583(a) (b)

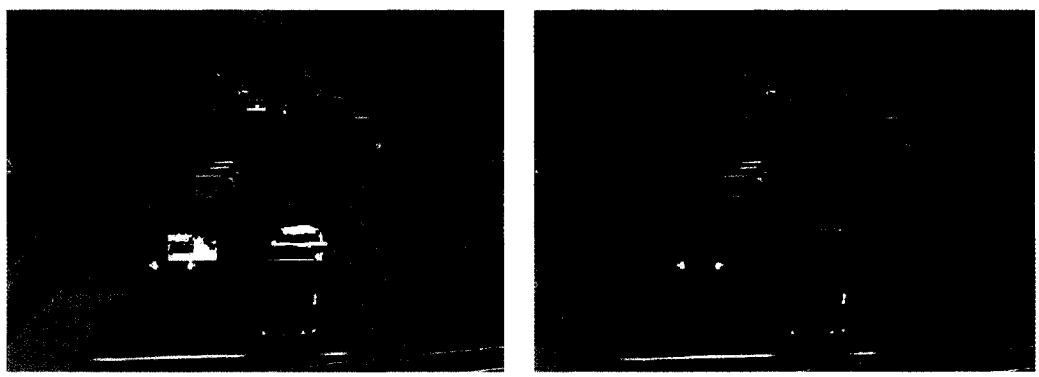


image 598 (a) (b)

Figure C.2: Vehicles detection applied on a sequence of images under cloudy conditions. a) mask images representing moving pixels; b) detected vehicles represented by bounding boxes.

## C.3 Test Sequence Results at Night

Figure C.3 shows a set of test results based on a sequence of images taken during night with clear weather conditions. The ground reflections are the problematic features which could significantly interfere with the detection of headlights. A large reflection on the ground surface could still be regarded as a moving object after motion detection and feature extraction, as shown in figures $image272(a)$, $image289(a)$ and $image323(a)$. But with the help of the circle shape verification and the circle pairing functions, they can be successfully separated from the headlights. These detection results demonstrate the good performance of the proposed vehicle verification algorithm under headlight-based detection mode.

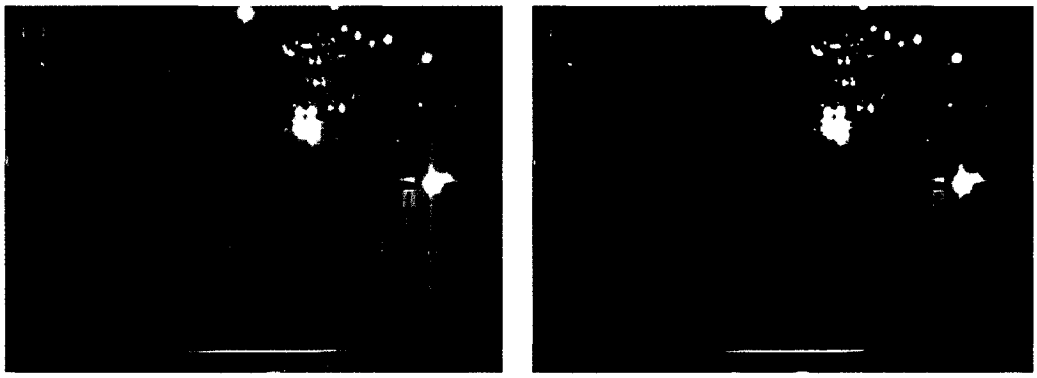


image 241 (a) (b)

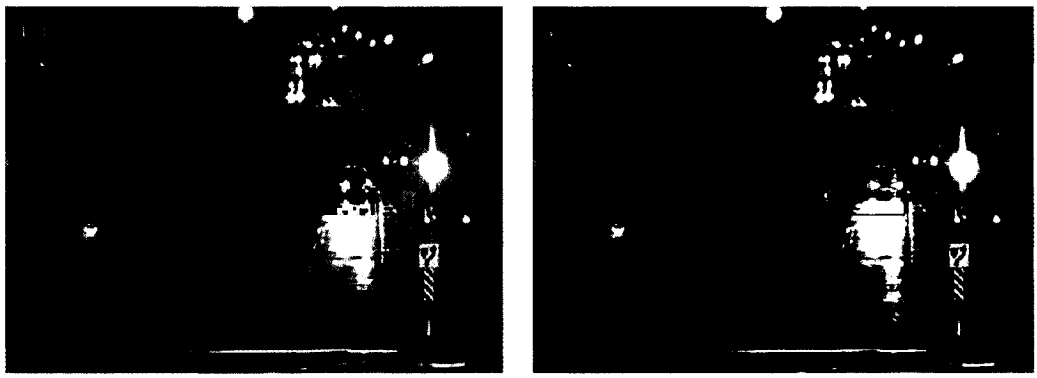


image 272 (a) (b)

image 289 (a) (b)

Figure C.3: Vehicles detection applied on a sequence of images under night conditions. a) mask images representing moving pixels; b) detected vehicles represented by bounding boxes.

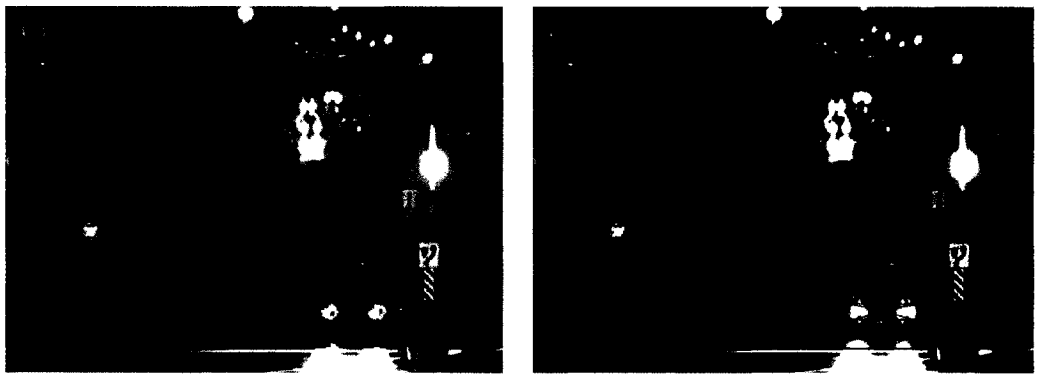


image 301 (a) (b)

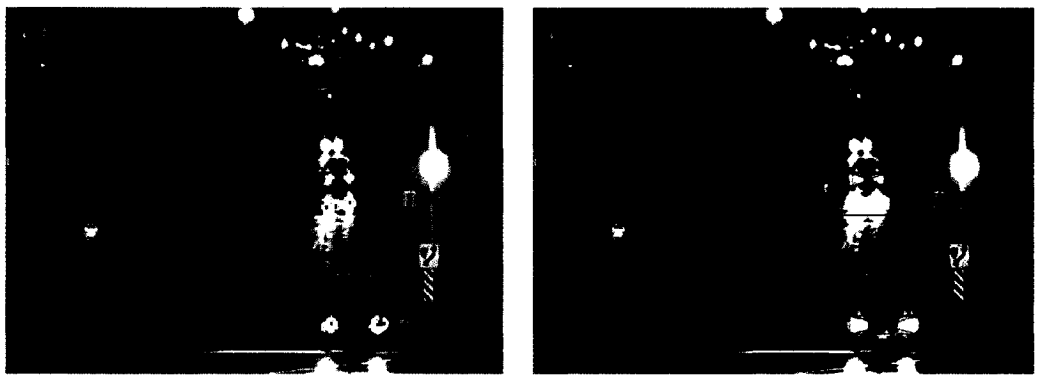


image 323 (a) (b)

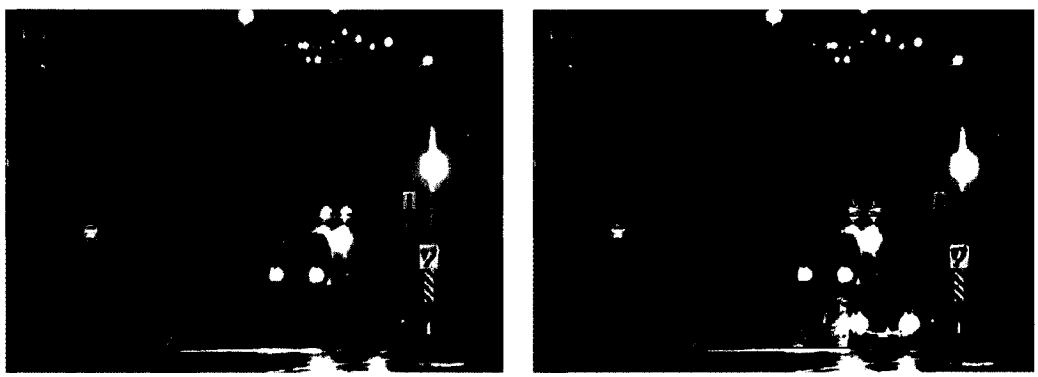


image 354 (a) (b)

Figure C.3: Vehicles detection applied on a sequence of images under night conditions. a) mask images representing moving pixels; b) detected vehicles represented by bounding boxes.

## C.4 Test Sequence Results at Night under Snowy Conditions

Figure C.4 shows a set of test results based on a sequence of images taken at night under snowy conditions. In order to show the evolving procedure of the detection of the headlights, an observation window has been set in the middle of the scene such that a moving car is included in the initial background image, as shown in figure *image*0. Since the initial background image is not a pure one, the moving car needs a short period of time to fade out from the background representation. As a consequently, the background differentiation progressively reveals moving pixels. As shown in figures *image*1($a$) to *image*7($a$), the size of the detected headlights is progressively increasing. This results from the continuous background update. The corresponding vehicle verification results are presented in figures *image*1($b$) to *image*7($b$). This illustrates the fact that only when the detected headlights completely appear, can the verification properly work. This test case demonstrates that the background image can be updated in a fast and adaptive way even under bad weather conditions. Also it confirms that the vehicle verification algorithm is efficient under headlight-based detection mode.

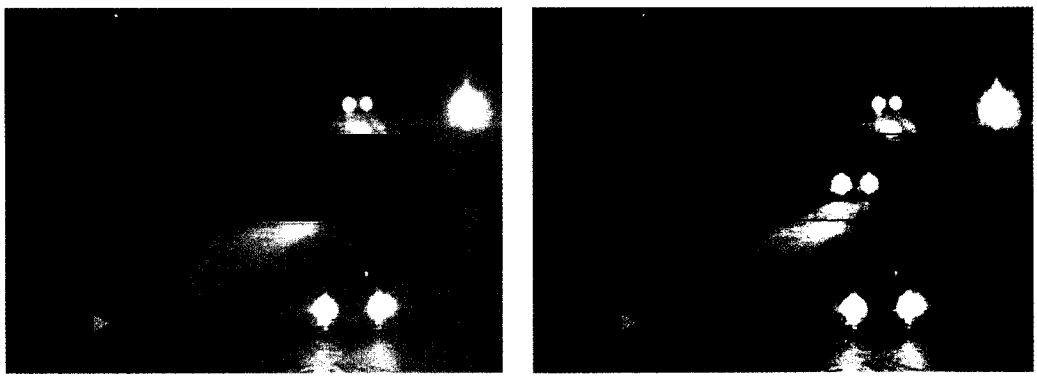


image 0 (a) (b)

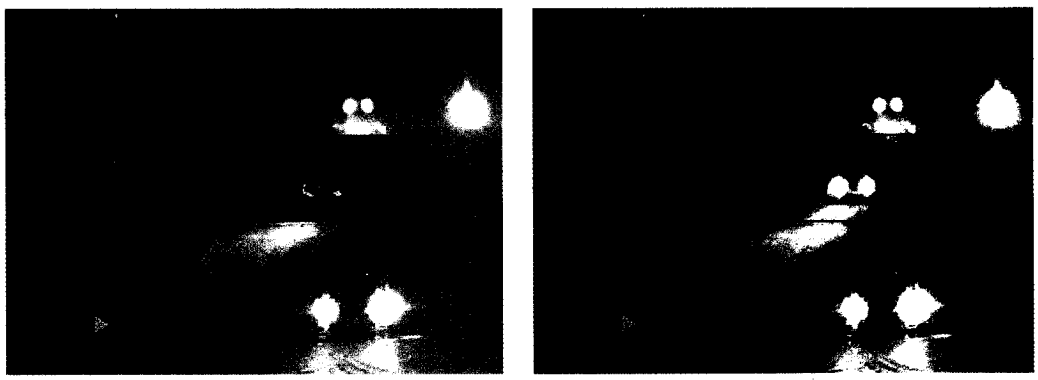


image 1 (a) (b)

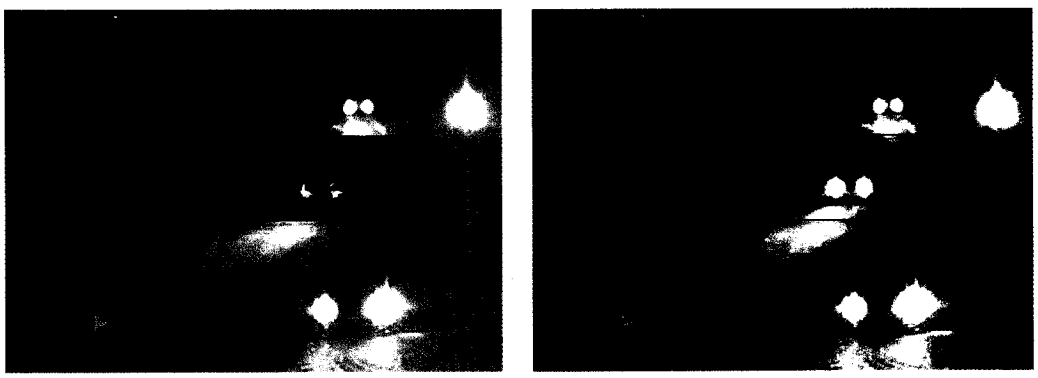


image 2 (a) (b)

Figure C.4: Vehicles detection applied on a sequence of images under snowy night conditions. a) mask images representing moving pixels; b) detected vehicles represented by bounding boxes.

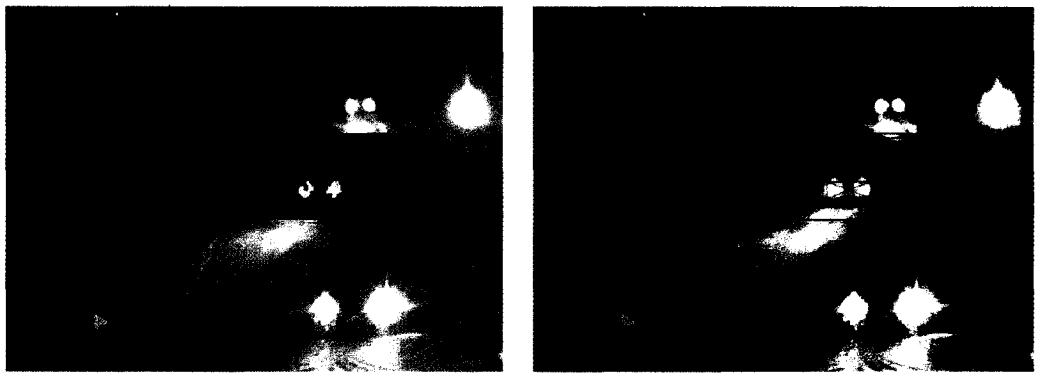


image 3 (a) (b)

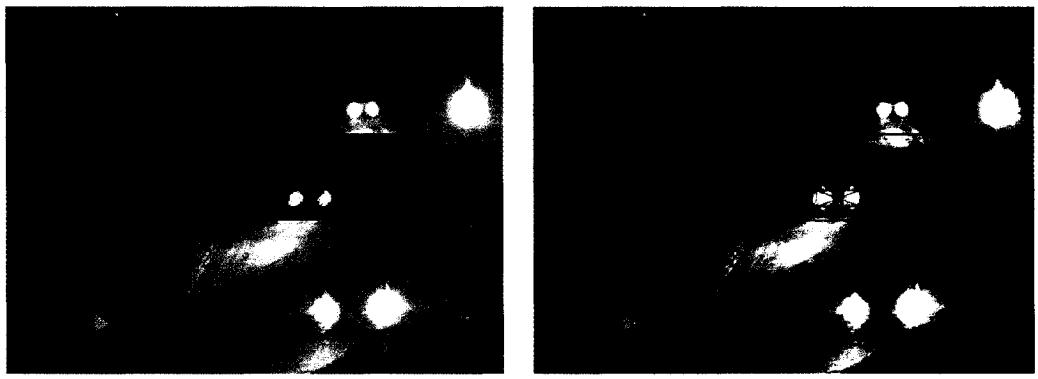


image 7 (a) (b)

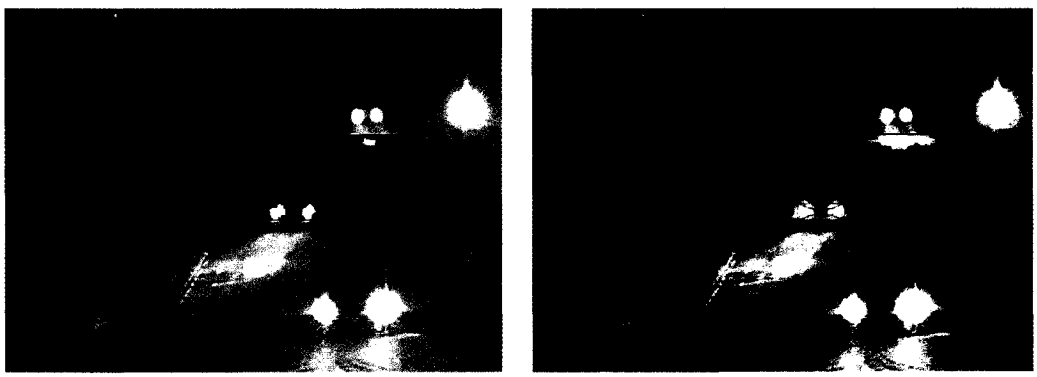


image 10 (a) (b)

Figure C.4: Vehicles detection applied on a sequence of images under snowy night conditions. a) mask images representing moving pixels; b) detected vehicles represented by bounding boxes.

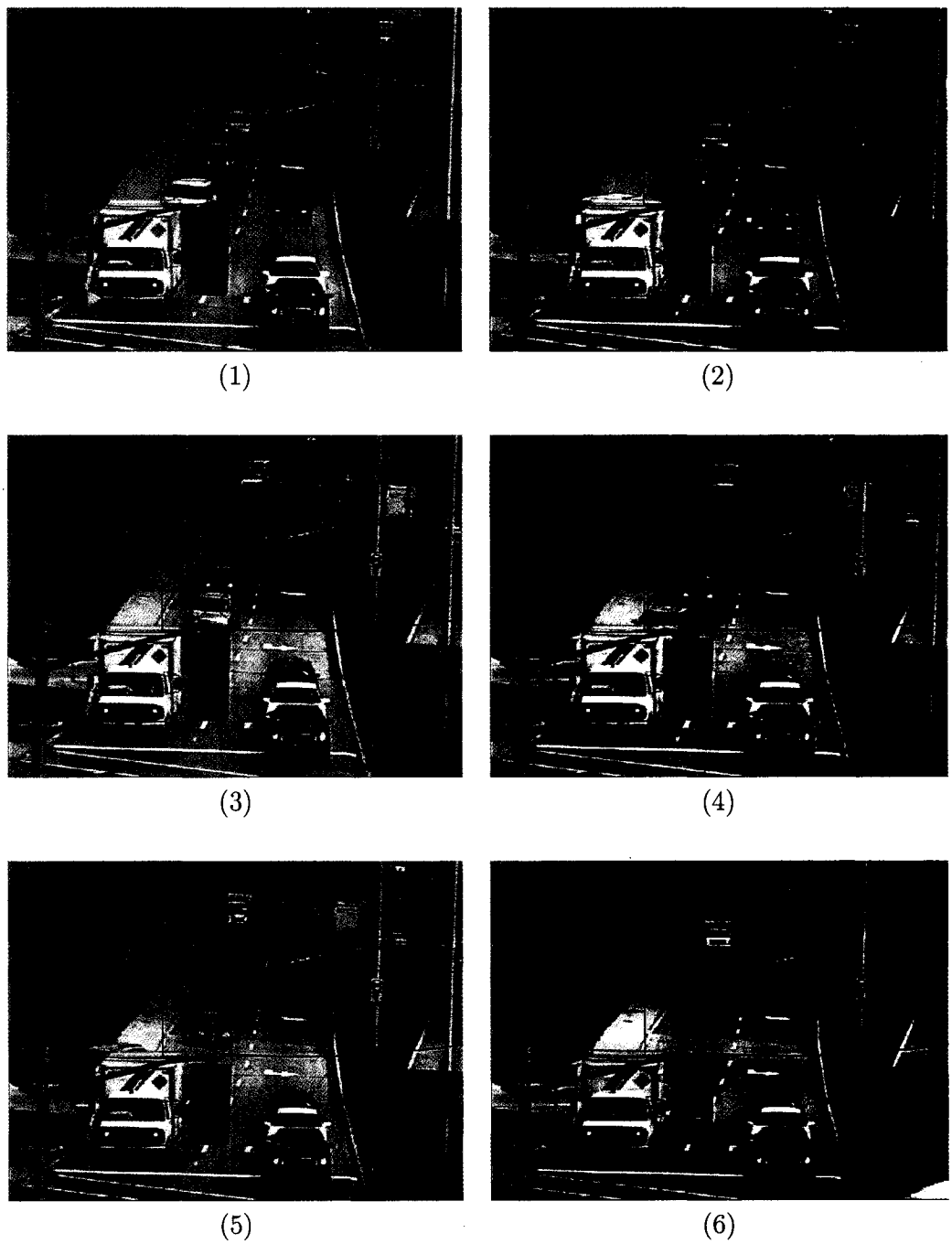
# Appendix D

# Supplementary Tests on Other Sites at Day and Night Time

Supplementary test have been performed on sequences collected in other locations in order to demonstrate the flexibility and the generality of the proposed approach. Figure D.1 and figure D.2 show two sets of test results based on the sequences taken at the intersection of King Edward Street South and Templeton Street (site2) under day and night conditions, respectively. Figure D.3 and figure D.4 present the testing results obtained for the intersection of King Edward Street North and Mann Ave (site3) at day and night. Without any change of our parameters in the program, the successful detection results demonstrate the robustness of our algorithm to different places.

In addition, as the system allows to easily set one or two windows of interest in the scene during the system's initialization phase. To cover one or multiple lanes, experiments have been made to evaluate this functionality. In figure D.2, a narrow window that covers only one lane, allowed us to focus the observation on vehicles driving straight.

These tests demonstrated that the system's performance are not degraded with the relative sizing of observation windows. This convenience can provide the system extra adaptability to different roads and streets in traffic monitoring applications.

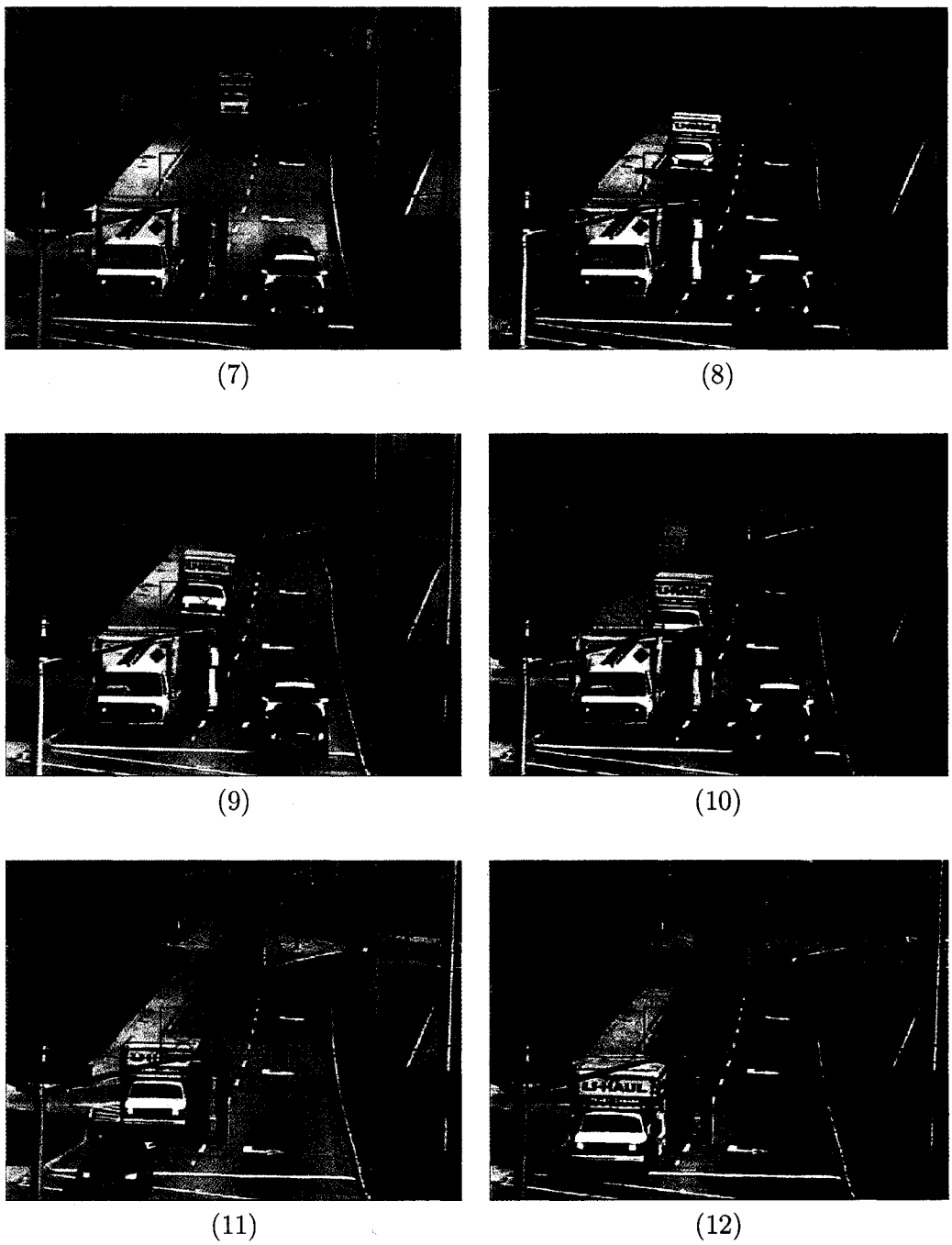


(1) (2)

(3) (4)

(5) (6)

Figure D.1: Test sequence results on King Edward Street South direction at day time.



(7) (8)

(9) (10)

(11) (12)

Figure D.1: Test sequence results on King Edward Street South direction at day time.

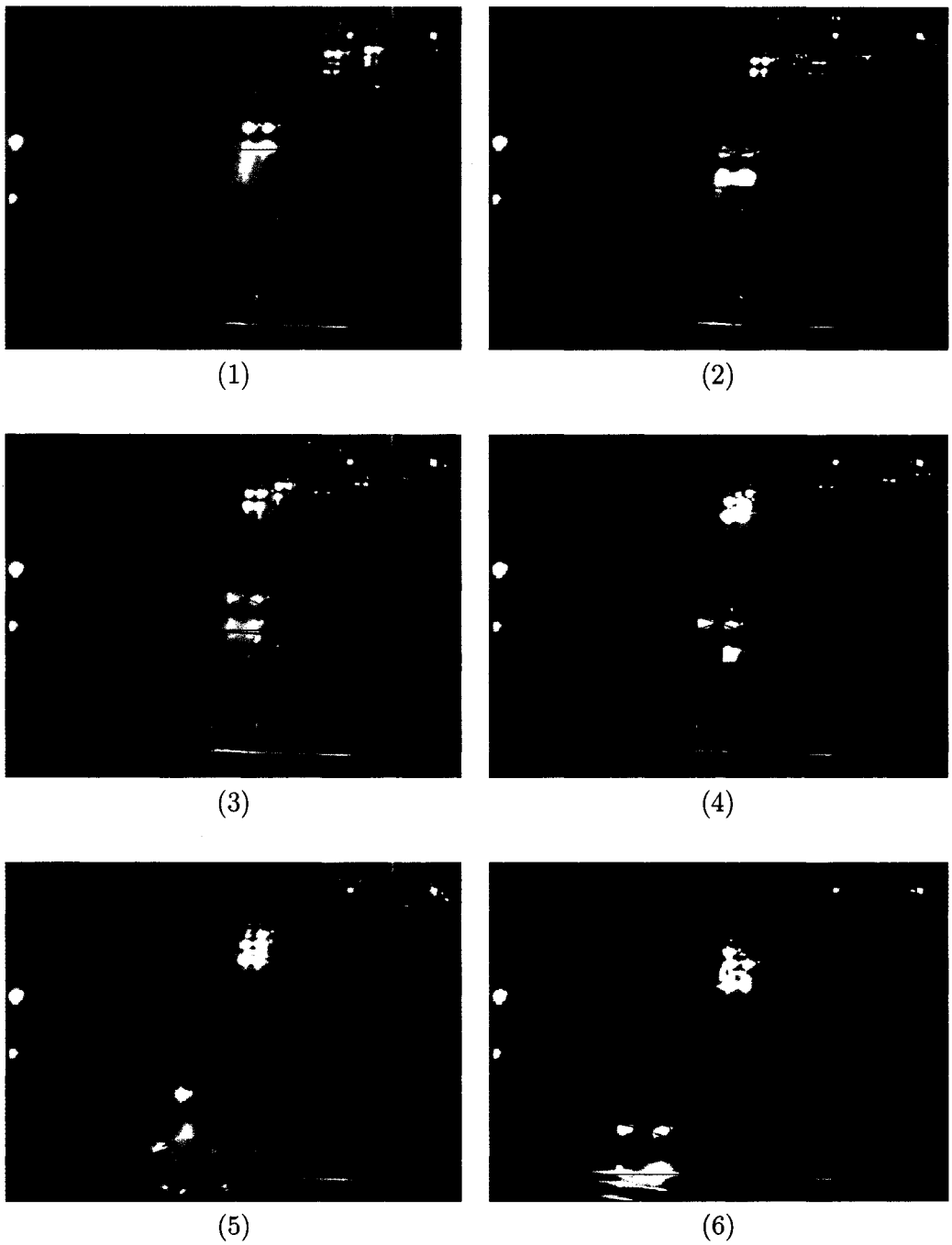
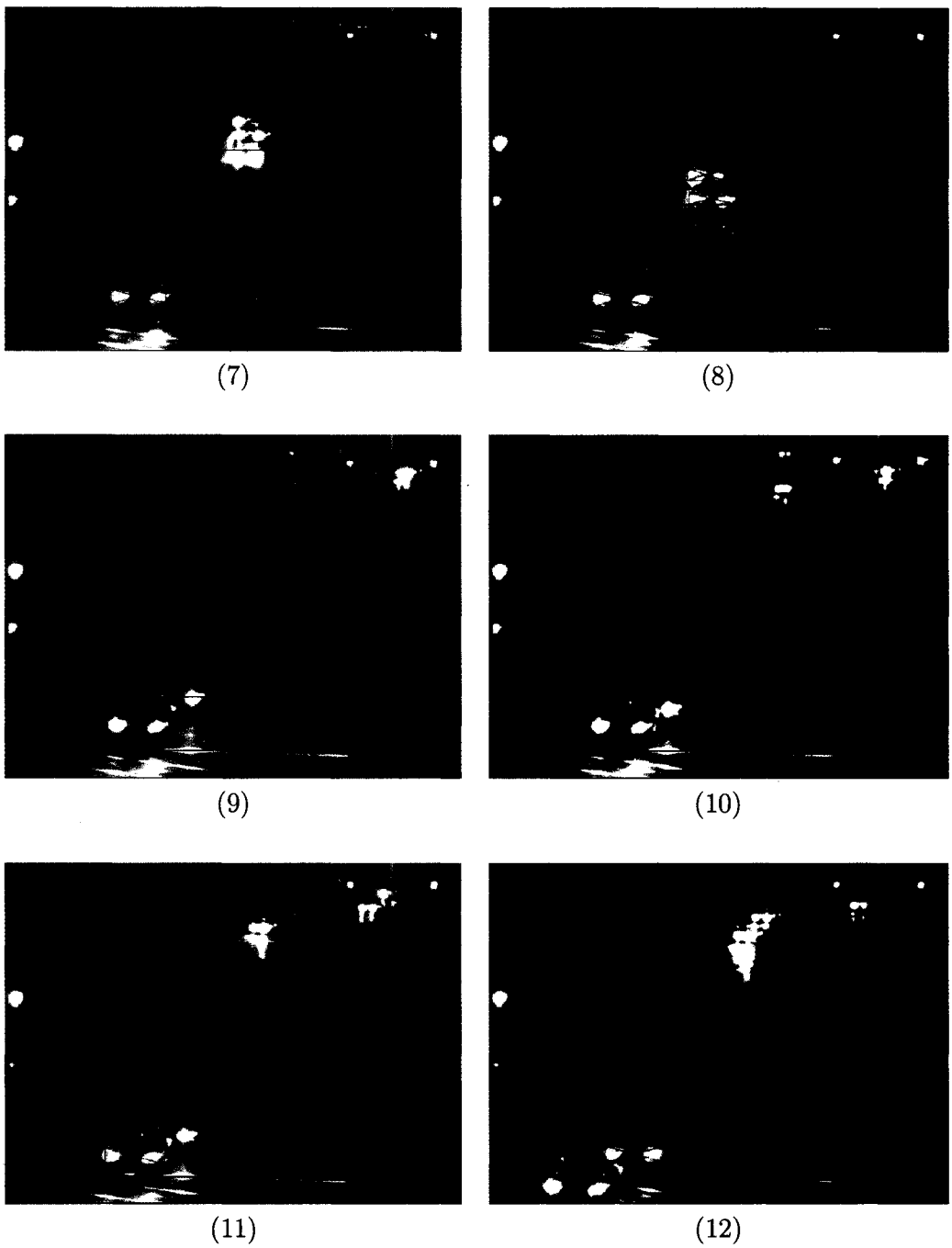


Figure D.2: Test sequence results on King Edward Street South direction at nighttime.

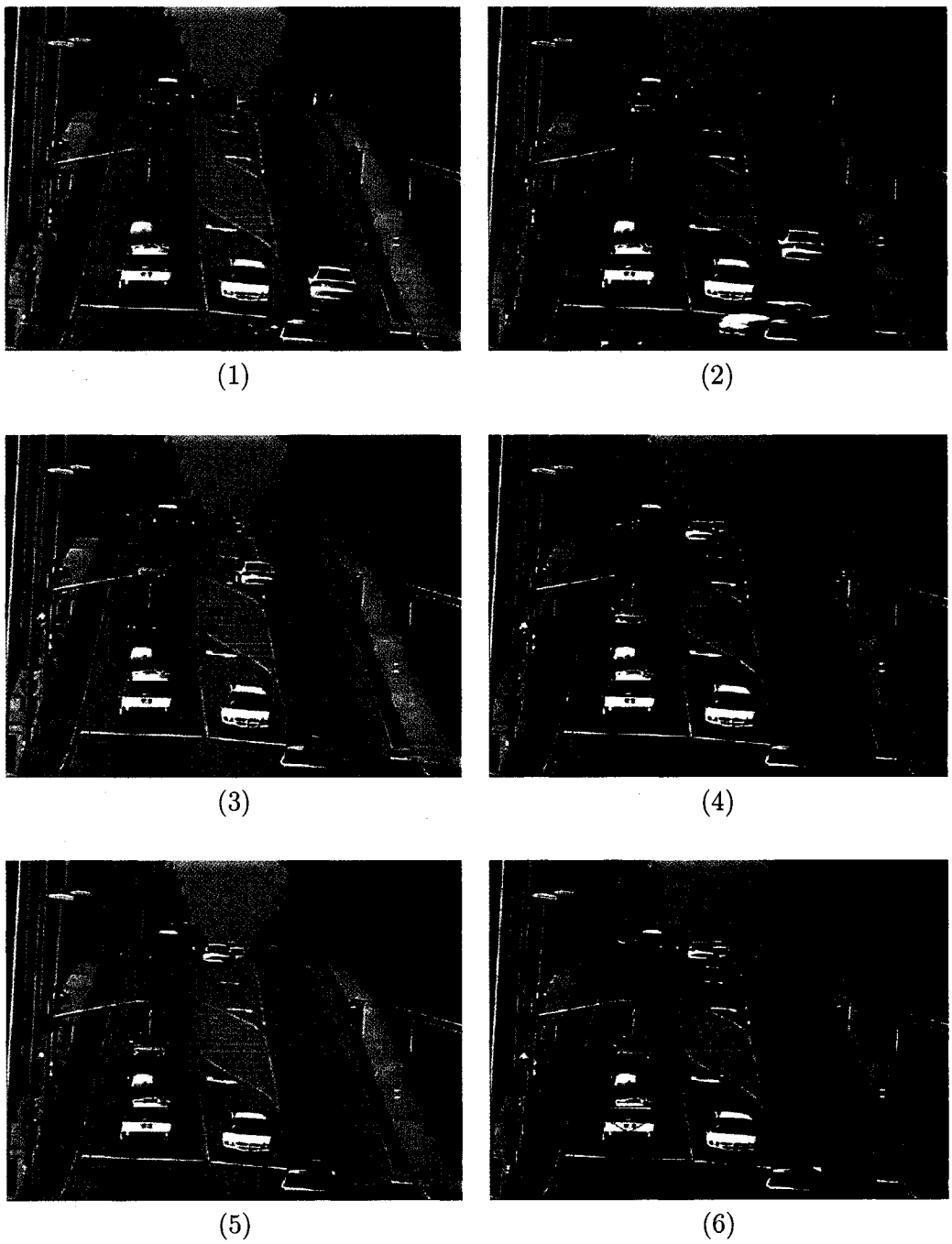


(7) (8)

(9) (10)

(11) (12)

Figure D.2: Test sequence results on King Edward Street South direction at nighttime.

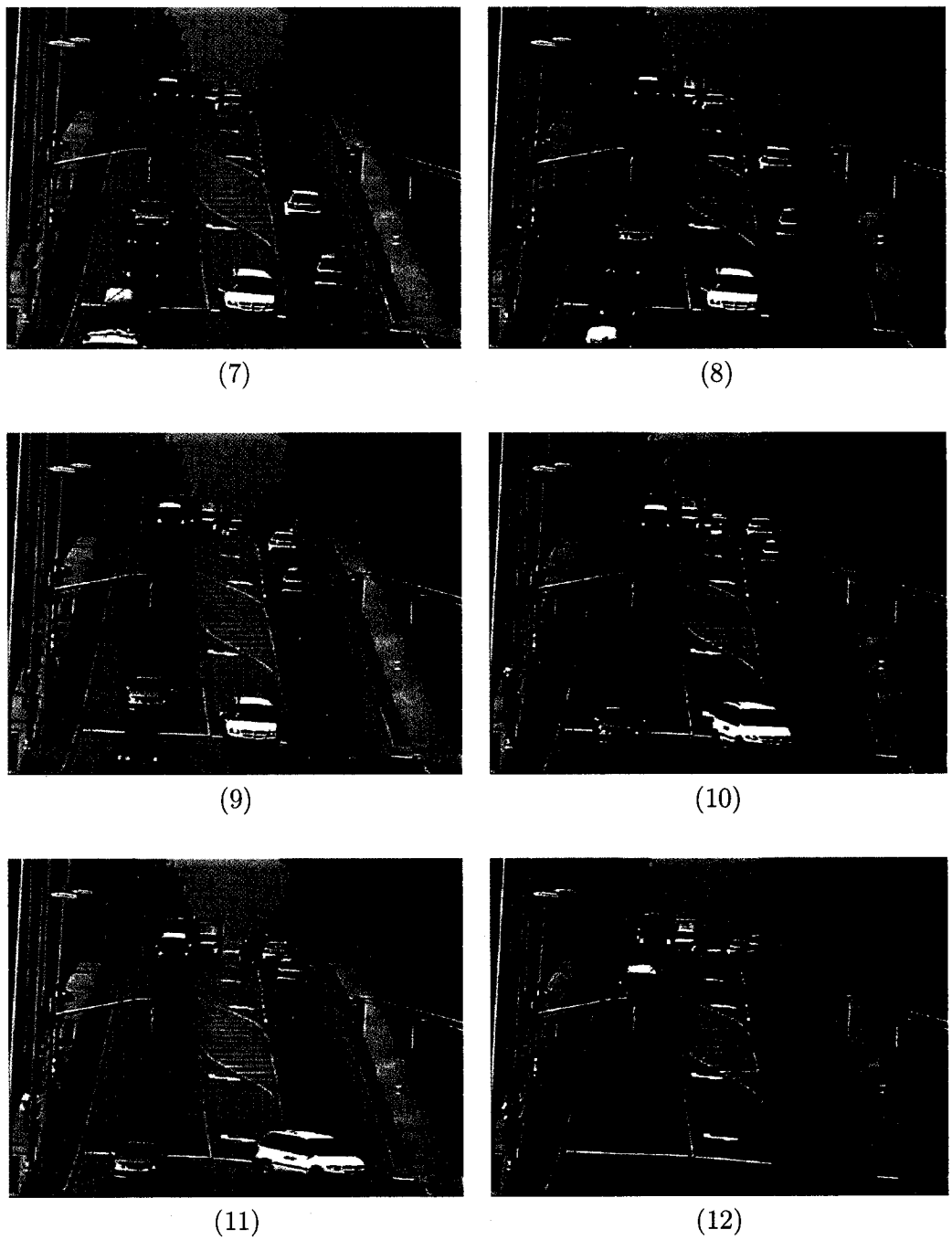


(1) (2)

(3) (4)

(5) (6)

Figure D.3: Test sequence results on King Edward Street North direction at day time.

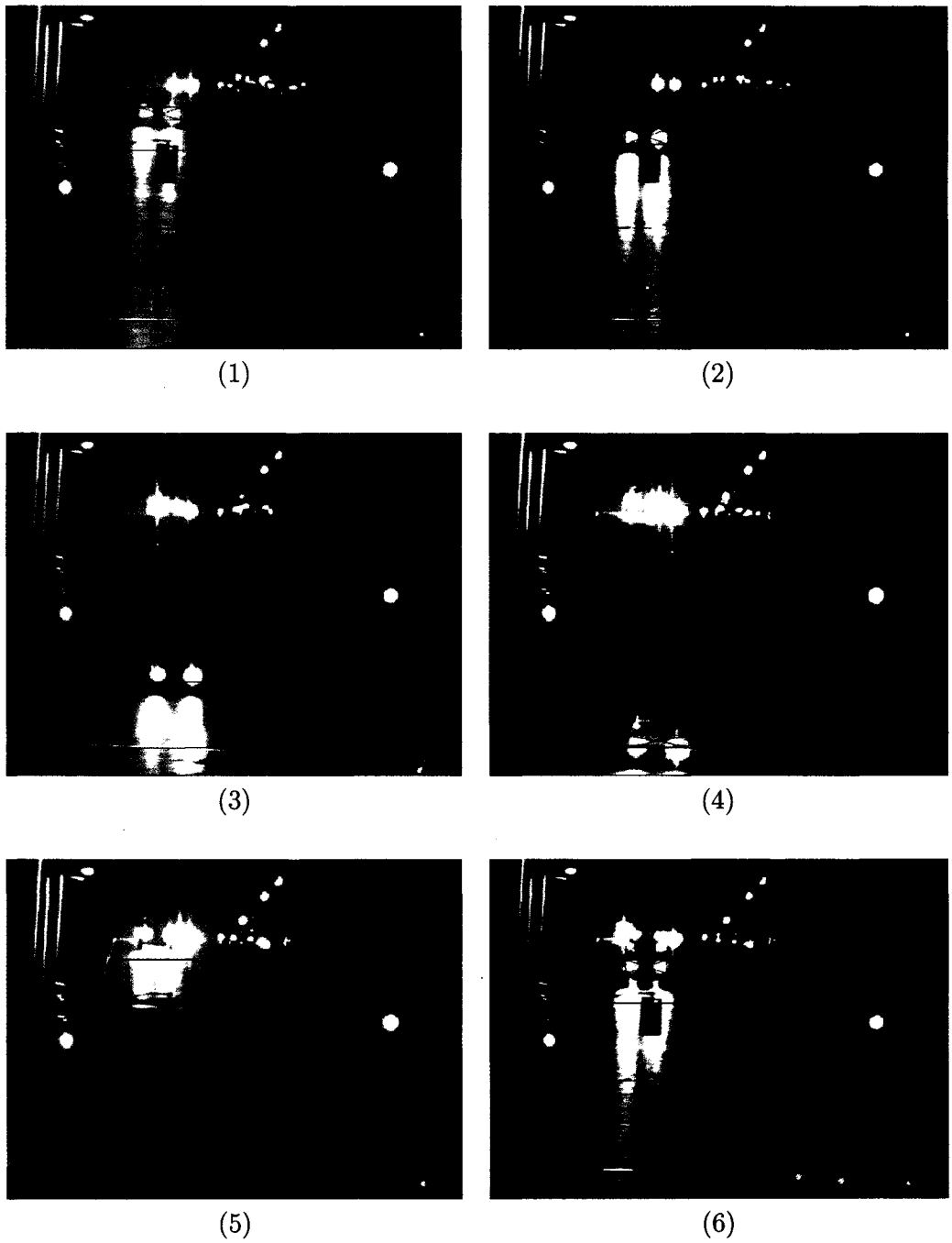


(7) (8)

(9) (10)

(11) (12)

Figure D.3: Test sequence results on King Edward Street North direction at day time.



(1) (2)

(3) (4)

(5) (6)

Figure D.4: Test sequence results on King Edward Street North direction at night-time.

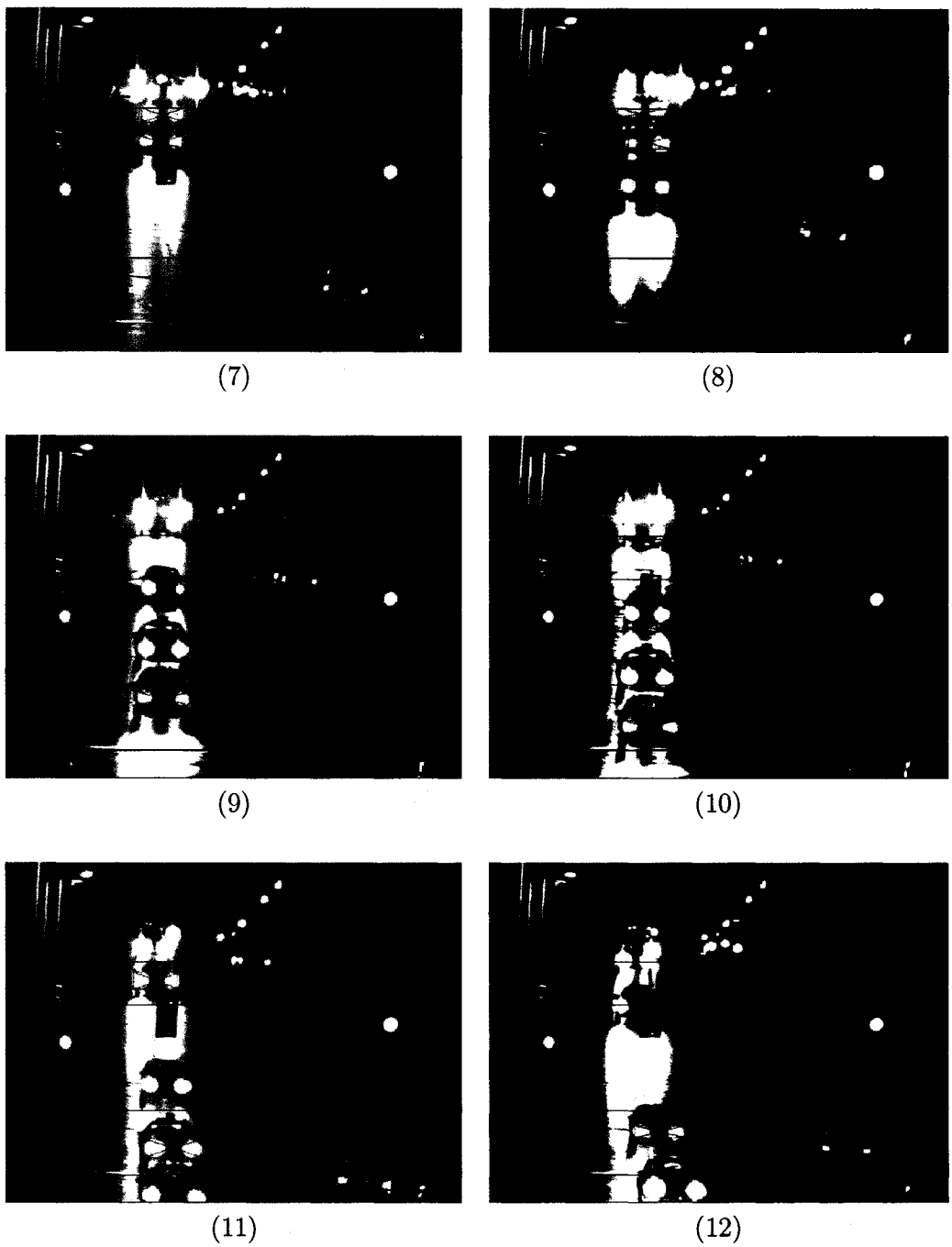



Figure D.4: Test sequence results on King Edward Street North direction at night-time.

# Bibliography

[1] Z. Zhu, B. Yang, G. Xu, D. Shi, "A Real-Time Vision System for Automatic Traffic Monitoring Based on 2D Spatio-Temporal Image," *in Proceedings of the IEEE Workshop on Applications of Computer Vision*, pp. 162-167, Dec. 1996.

[2] T. Nakanishi, K. Ishii, "Automatic Vehicle Image Extraction Based on Spatio-Temporal Image Analysis," *in Proceedings of the 11th IAPR International Conference of Computer Vision and Applications on Pattern Recognition*, vol.1, pp. 500-504, 1992.

[3] M.-P. Dubuisson, A.K. Jain, "Object Contour Extraction Using Color and Motion," *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 471-476, 1993.

[4] N. Oha, "A Statistical Approach to Background Subtraction for Surveillance Systems," *in Proceedings of the IEEE International Conference on Computer Vision*, vol.2, pp. 481-486, 2001.

[5] C. Ridder, O. Munkelt, H. Kirchner, "Adaptive Background Estimation and Foreground Detection using Kalman-Filtering," *in Proceedings of International Conference on Recent Advances in Mechatronics*, Istanbul, 1995.

[6] R. Cucchiara, C. Grana, M. Piccardi, A. Pratti, "Statistic and Knowledge-Based Moving Object Detection in Traffic Scene," *in Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pp. 27-32, Dearborn, MI, Oct. 2000.

[7] A. Lai, N. Yung, "A Fast and Accurate Scoreboard Algorithm for Estimating Stationary Backgrounds in an Image Sequence," *in Proceedings of the 1998 IEEE International Symposium on Circuits and Systems*, vol.4, pp. 241-244, 1998.

[8] S. Kamijo, Y. Matsushita, K. Lkeuchi, M. Sakauchi, "Traffic Monitoring and Accident Detection at Intersections," *in Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, pp. 703-708, 1999.

[9] W.E.L. Grimson, C. Stauffer, R. Romano, L. Lee, "Using Adaptive Tracking to Classify and Monitor Activities in a Site," *in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 22-29, 1998.

[10] P. Parodi, G. Piccioli, "Feature-Based Recognition Scheme for Traffic Scenes," *in Proceedings of the Intelligent Vehicles'95 Symposium*, pp. 229.234, 1995.

[11] M. Watanabe, N. Takeda, K. Onoguchi, "Moving Object Recognition Method by Optical Flow Analysis," *in Proceedings of the 13th International Conference on Pattern Recognition*, vol.1, pp. 25-29, 1996.

[12] N. Friedman, S. Russell, "Image Segmentation in video images: A probabilistic approach," *Annual Conference on Uncertainty in Artificial Intelligence*, pp. 175-181, 1997.

[13] R. Taktak, M. Dufaut, R. Husson, "Vehicle Detection at Night Using Image Processing and Pattern Recognition," *in Proceedings of the IEEE International Conference of Image Processing*, vol.2, pp. 296-300, 1994.

[14] S. Gupte, O. Masoud, R.F.K. Martin, N.P. Papanikolopoulos, "Detection and Classification of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol.3, pp. 37-47, 2002.

[15] M. Tsuchikawa, A. Sato, H. Koike, A. Tomono, "A Moving Object Extraction Method Robust Against Illumination Level Changes for a Pedestrian Counting

System," *in Proceedings of the International Symposium on Computer Vision*, pp. 563-568, 1995.

[16] H. Hase, K. Miyake, M. Yoneda, "Real-time Snowfall Noise Elimination," *in Proceedings of the International Conference on Image Processing*, vol.2, pp. 406-409, 1999.

[17] S. Kyo, T. Koga, K. Sakurai, S. Okazaki, "A Robust Vehicle Detecting and Tracking System for Wet Weather Conditions Using the IMAP-VISION Image Processing Board," *in Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, pp. 423-428, 1999.

[18] R. Cucchiara, M. Piccardi, P. Mello, "Image Analysis and Rule-Based Reasoning for a Traffic Monitoring System," *IEEE Transaction of Intelligent Transportation Systems*, vol.1, pp. 119-130, 2000.

[19] A. Soto, A. Cipriano, "Image Processing Applied to Real-Time Measurements of Traffic Flow," *in Proceedings of the IEEE Southeastern Symposium on System Theory*, pp. 312-316, 1996.

[20] J. Versavel, "Road Safety Through Video Detection," *in Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, pp. 753-757, 1999.

[21] Image Sensing Systems Inc., Autoscope Products, *http://www.imagesensing.com*, 2003.

[22] Teknovation Corp., TrafficVision Traffic Analysis Software, *http://www.teknovation.com*, 2003.

[23] Iteris Inc., Vantage Video Detection, *http://www.iteris.com*, 2003.

[24] D. Koller, K. Danilidis, H.H. Nagel, "Model-Based Object Tracking In Monocular Images Of Road Traffic Scenes," *International Journal of Computer Vision*, pp. 257-281, 1993.

[25] R. Cucchiara, C. Grana, M. Piccardi, A. Pratti, S. Sirotti, "Improving Shadow Suppression in Moving Object Detection with HSV Color Information," *in Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pp. 334-339, 2001.

[26] C.-S. Fuh, P. Maragos, "Affine Models for Image Matching and Motion Detection," *International Conference on Acoustics, Speech, and Signal Processing*, vol.4, pp. 2409 -2412, 1991.

[27] G.J. Awcock, R. Thomas, "Applied Image Processing," Basingstoke, Hampshire: Macmillan, 1995.

[28] L.G. Shapiro, G.C. Stockman, "Computer vision," Upper Saddle River, NJ: Prentice Hall, 2001.

[29] T.S. Huang, A.N. Netravali, "Motion and Structure from Feature Correspondences: A review," *in Proceedings of the IEEE*, vol.82, no.2, pp.252-268, February 1994.

[30] J.S Christopher, "Applications of Computer Vision to Road-traffic Monitoring," *PhD thesis, Department of Computer Science, University of Bristol*, 1997.

[31] Y. Liu, P. Payeur, "Vision-Based Detection of Activity for Traffic Control," *(Best Student Paper Award) in Proceedings of the Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 1347-1350, Montreal, May 2003.

[32] P. Payeur, Y. Liu, "Video Traffic Monitoring for Flow Optimization and Pollution Reduction," *(invited paper) in Proceedings of the 2nd IEEE/AEI International Workshop on Advanced Environmental Monitoring Technologies*, Como, Italy, July 2003.

[33] Y. Liu, P. Payeur, "Robust Image-Based Detection of Activity for Traffic Control," *in the Canadian Journal of Electrical and Computer Engineering*, vol. 28, no. 2, pp. 63-67, April 2003.