COMPUTER VISION TECHNIQUES FOR TRAFFIC DATA COLLECTION AND ANALYSIS

Xidong Yuan

A Thesis
in
The Department
of
Civil Engineering

Presented in Partial Fulfilment of the Requirements
For the Degree of Doctor of Philosophy
Concordia University
Montreal, Quebec, Canada

November 1994

ISBN 0-612-01276-X

ABSTRACT

COMPUTER VISION TECHNIQUES FOR TRAFFIC DATA COLLECTION AND ANALYSIS

Xidong Yuan, Ph.D.

Concordia University, 1994

Eight new models have been developed and presented in this thesis to process and analyze digitized monochrome and color images. The functions of these models include moving object detection, vehicle signal light detection, noise removal, overall vehicle dimension estimation, and vehicle classification. These models are then integrated to form four algorithms, each dedicated to a specific function: measurement of traffic volume and vehicle speed; detection and count of vehicles intending to turn; classification of vehicles; and measurement of pedestrian flow.

In order to verify the accuracy of the proposed algorithms and their associated software, field studies were carried out using software that has been developed to extract associated traffic data from video tapes. These field studies consisted of apprcximate ten hours of video tapes which had recorded the actual traffic scenes of several locations in downtown Montreal and a main highway in the Montreal area. Subsequently, these video records were utilized for collection of traffic data pertaining to traffic volume, vehicle speed, vehicle classification and pedestrian volume. Correlation was observed between the traffic data generated by the algorithms and those collected by human observers from a video monitor. The accuracy level obtained in all cases was higher than 90%.

The newly developed models and algorithms provided a new method for increasing the capability and reducing the detection error of currently used video traffic detection systems. With further improvement, the models and algorithms can be used to offer a wide variety of possible applications in intelligent vehicle-highway systems (IVHS). Potential applications include automatic incident detection, automatic queue detection, electronic toll collection, statistical data collection from installed cameras or videotape sequences, and automatic control unit or system for variable-message sign applications in tunnels and on motorways.

Dedicated to my parents

Mr. Bao-Zong Yuan

Ms. Shan-Zhu Yu

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Dr. Yean-Jye Lu of Department of Civil Engineering and co-supervisor Dr. O. Moselhi of Center for Building Studies for their guidance, advice, and support throughout the course of this research.

I wish to thank Professors Dr. S. Cheung and Dr. S. Sarraf of Department of Civil Engineering, and Dr. C.Y. Suen and Dr. Y. Tang of Department of Computer Science, and Dr. J.C.Y. Wang of Center for Building Studies for their suggestions and comments. I am also thankful to Ms. Qing Yang and Mr. Changde Yan for their help in the experiments. I would acknowledge also those people in the Department of Civil Engineering who provided me with a stimulating environment during my studies and research.

Also, I wish to express my gratitude to my father, Professor Bao-Zong Yuan, for his helpful advice concerning computer vision techniques. I would like to thank my wife, Ms. Yan Gu. This thesis could not have been completed without her immeasurable love, meticulous care, constant encouragement and wholehearted support.

Finally, I take special pleasure in thanking Concordia University which awarded me the Concordia University International Student Fee Remission to enable me to study at this University. I would like to acknowledge the research grants received from the National Science and Engineering Research Council of Canada.

CONTENTS

CHAPTER 3

CHAPTER 4

LIST OF FIGURES AND TABLES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

1.1.1 Traffic Data Requirements

Traffic data are required for a variety of activities including those related to research, planning, design, accident analysis, traffic operation, and enforcement. For example, traffic data in terms of speed and headway in advance of intersections on high-speed highways have been used in research to determine the length of yellow-clearance and all-red intervals as well as the need for flashing yellow "signal ahead" signs (ITE 1982). Classification counts provide data for estimating the proportion of trucks and other vehicle types for intersections, individual links, systems, etc. Information on deceleration rates on speed-change lanes and ramps at signalized intersections has provided insight into the demand for pavement friction and required friction or skid numbers (Rizenbergs 1976). Data on traffic volumes may be the most significant item for transportation planning, with the possible exception of road mileage itself. Continuing traffic volume, turning movement, speed, classification, and weight in motion data provide the basic information needed to design new highways; relocate, reconstruct, and modify existing ones; and resurface highway pavements and redeck highway bridges. Examples include geometric design, timing of signals, and channelization of intersections (AASHTO 1984).

Real-time traffic data in large-scale street and highway networks are required for the Intelligent Vehicle-Highway Systems (IVHS), especially for two of its six functional areas: Advanced Traffic Management Systems (ATMS), and Advanced Traveller Information Systems (ATIS). Inherent to the concept of IVHS is the use of this newly available information to integrate the traveller, vehicle, and roadway infrastructure into a comprehensive system. The traffic information distributed from the ATIS would help travellers both before and during the course of their trips to make fuller use of existing modes of transportation. The major assumption of the ATIS is that accurate and timely information on travel conditions becomes factored into a person's decision on where to travel, when to travel and how to travel (Khattak 1993; O'Neill 1993). The ATIS is able to mitigate traffic congestion problems by either informing an en route driver of traffic incidents ahead and providing alternative routes or allowing pretrip travel decisions to factor in the benefits of delayed trip starts, or alternative routes and modes. The ATMS is another important portion of the IVHS. The ATMS is most often identified with incident detection and congestion management. One of the management functions of the ATMS is to control the traffic signal systems for street networks. Determining the proper cycle length, splits and offsets of traffic signals requires information on traffic volumes, platoon speeds, turning movements, and other data. Based on the ongoing collection of traffic data or the time of day associated with certain traffic conditions, the signal timing is changed in an appropriate fashion.

1.1.2 Data Collection Methods and Problems

During the early part of this century, most traffic data were obtained by manual means. Manual data acquisition requires one or more observers, properiy equipped, to record the pertinent traffic information. Such a manual method is flexible and allows data to be collected on the basis of complicated categories. However, this method incurs a high labor cost, particularly when massive quantities of traffic data are required. In addition, the data collected by an observer is sometimes inaccurate because it is difficult for an observer to pay attention to a street scene for a long period. Although traffic data can be obtained by videotaping traffic situations and then analyzing these permanent records in the laboratory (List 1989; Young 1989), this method is also time consuming because the extraction of data from the video is still carried out by human observers.

Presently, most traffic data are obtained and processed by automatic devices and computers and transmitted directly to central offices. The vehicular detectors commonly used are inductive loop detectors, magnetic detectors and other types such as pressured pads, radar and sonic detectors (Moore 1981; Kell 1982).

Detectors such as pressur d pads, radar and sonic detectors were the earliest types to be used in traffic data collection. While these units are still available, their use is very limited. Pressure and rad detectors are becoming less common as they can only provide passage data, are expensive to install, and relativeiy complex to maintain. Sonic detectors have similar problems. Although they are capable of detecting both presence and passage, they are relatively expensive to purchase and install, sensitive to environmental conditions, and somewhat inaccurate in congested situation.

Magnetic detectors are divided into three types: the standard magnetic detector, the directional magnetic detector, and the magnetometer. All three types consist of two components, an in-road sensor and an amplifier unit. For these detectors to sense a change in magnetic field, the vehicle must be in motion. Vehicles travelling less than 10 km/h are generally not detected. Consequently, magnetic detectors can provide the equivalent of passage or motion data, but not occupancy or presence data.

Inductive loop detectors are the most commonly used today. This detector consists of one or more turns of insulated wire embedded in the road where vehicles are to be detected. The ends of the loop are connected by cable to an electronic amplifier usually located in the controller. A vehicle passing over or resting in the loop will unbalance a turned circuit which is sensed by the amplifier. Such detectors are thereby able to detect either presence or passage of vehicles. Unfortunately, the drawbacks of inductive loop detectors limit the range of their usage in the IVHS. Their principal shortcomings are discussed below.

(1). "Blind" detection. Inductive loop cannot measure some important traffic parameters and accurately assess traffic conditions. This is because the technology employed represents a "blind" type of detection. Traffic data such as speed and traffic composition queue length must be derived from vehicle presence and passage and require multiple detections, which increases cost and exacerbates the reliability problem.

(2). Inflexible functions. There are many different designs of inductive loop to suit various detection purposes. A loop detector designed for vehicles cannot be used for pedestrians or even motorcycles, and vice versa. Furthermore, the location and configuration of a loop detector is dependent on many conditions. These conditions include the type and capability of traffic signal controller, signal control mode, traffic variable to be measured, geometry of the intersection and approaches, and traffic flow characteristics (e.g., volume, speed, etc.). Any change in these conditions affects the detector adversely. This kind of fixed-point detection is a major disadvantage for traffic control since detection points should vary with speed, volume and control objective. With respect to reliability, it was noted that most cities with mature systems in the United States report that at any time 25-30% of such detectors were not functional or not operating properly (Lyles 1983).

(3). Disruptive installation and maintenance. Inductive loop detectors are embedded in the road. Installation and maintenance require digging grooves in the road, thus producing traffic disturbances. Adverse weather conditions or pavement reconstruction present additional challenges for maintaining loop detectors.

(4). The cost of loop detection for streets is highly sensitive to the distance of these detectors from the nearest signalized intersection. To minimize the traffic detection cost, it would be desirable to locate the loops near the stop line, but to implement the traffic adaptive features of ATMS requires installing the detectors at least several hundred feet away.

In summary, as the instrumentation and data requirements increase, conventional devices become inadequate, expensive and often unreliable for advanced applications such as vehicle guidance and navigation, adaptive control of congested street networks and freeway corridors, real-time forecasting of traffic demand patterns, etc. For example, although loop detectors deployed in sufficiently high numbers can identify the location of unusual congestion, they cannot help control center personnel in determining congestion causes: accidents, disabled vehicles, spilled loads, construction or maintenance activities, police or fire operations, or just heavier than normal traffic.

The use of remotely controlled CCTV (closed-circuit television) can give the control center visual access to the site of traffic flows and the operation of traffic signals. Also, many important traffic data can be obtained from the video. Unfortunately, extraction of traffic data from the video is carried out manually and is time consuming at present. Typically, it takes an operator five to ten hours to analyze a one-hour recording, depending on the complexity of the scene and the amount of information to be extracted (List 1989; Young 1989). Hence, it is impossible to provide real-time traffic data in this way. To address this problem, computer vision systems have a great potential to extract real-time traffic data from CCTV.

1.1.3 Computer Vision System for Traffic Data Collection and Analysis

Computer vision techniques lead the way to extract automatically the required traffic data in the viewing field of the camera. The computer vision system (CVS) includes four basic parts as shown in Fig. 1.1. In image acquisition, the traffic scenes shot by the video camera are digitized and stored in a computer frame by frame. Then these digital images are processed and analyzed by the computer until all the necessary traffic data have been extracted. The advantages of the CVS are listed below.

(1) It is able to obtain the data that cannot be easily or accurately obtained by conventional detection devices, such as turning movements and vehicle classification by body shape.

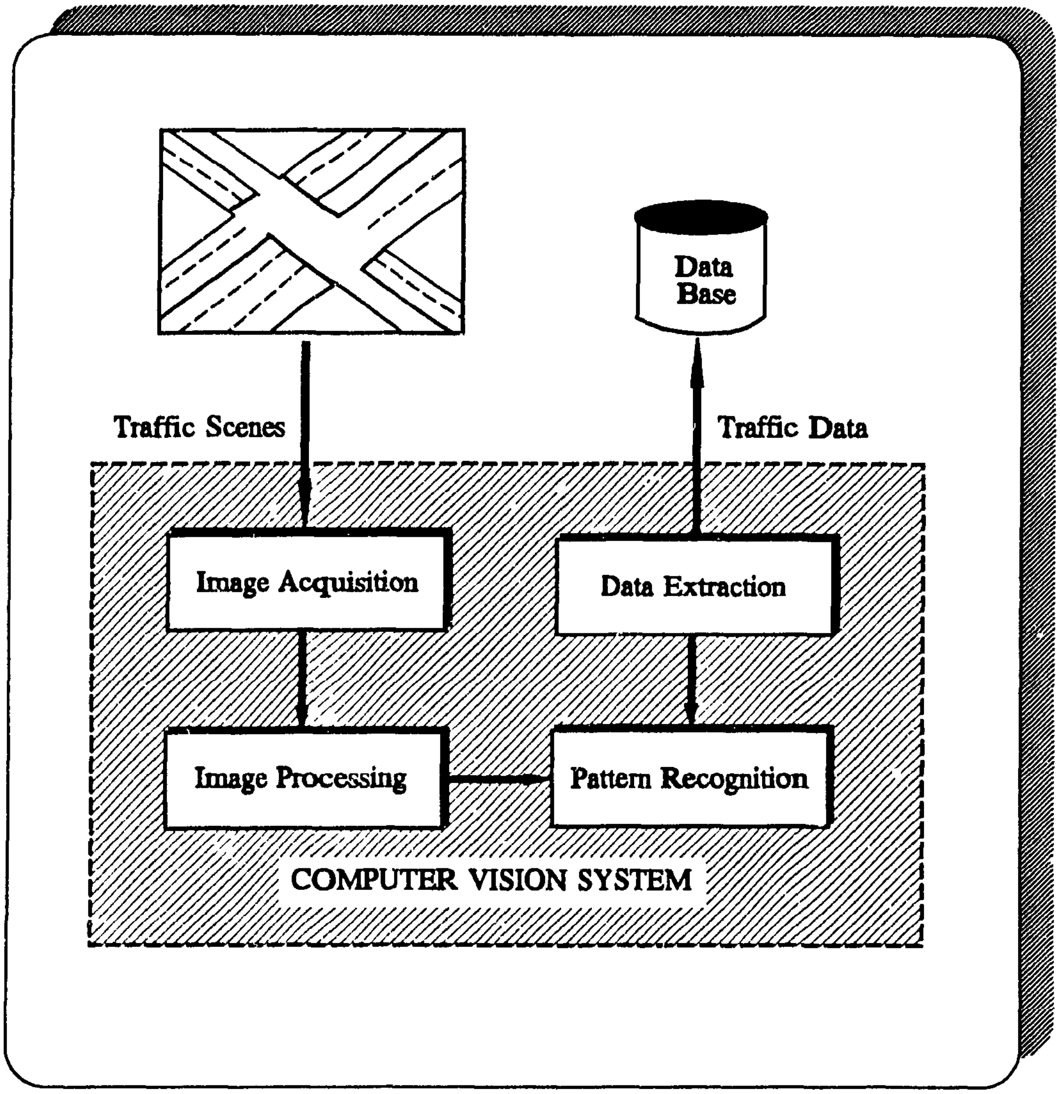


Fig. 1.1 Basic parts of a computer vision system

(2) It also has multitasking capabilities, i.e., while it performs its basic detection function, it can simultaneously derive traffic measurements locally (using a microprocessor) or at a central location, perform surveillance functions, act as a vehicle counting and classification station, detect incidents, and recognize special vehicles

(ambulances, fire trucks, buses, etc.).

(3) The CVS can detect traffic at multiple points on a roadway within the field of the camera's view thereby becoming cost-effective. For instance, it was estimated by Michalopoulos et al. (1986, 1990) that in Minnesota the cost of fully instrumenting an intersection with a CVS (40 cameras, one microprocessor) would be lower than that of loop detectors assuming that at least 3 loops per approach were required. It was also estimated that a CVS would save 35% in maintenance costs and reduce the man-hours required for maintenance by about 70% compared with a loop system. The savings can further increase if the same microprocessor also performs control functions, thereby eliminating the need for a separate controller.

(4) The CVS does not use any sensor embedded in the road, instead, it employs CCTV cameras to overlook the traffic scenes. Therefore, installation of CVS does not disturb the pavement and shoulder, resulting in an undisturbed traffic flow and improved reliability, especially during reconstruction operations. Once the basic hardware is on site, the detection zones can be easily placed in minutes in any configuration using a mouse and a television monitor. The zones can also be re-positioned to adapt to changing traffic control or data collection requirements. Once these detection zones are placed, the system generates presence and passage signals compatible with loops, measures speeds, classifies vehicles, and generates essential traffic parameters such as volumes, speed, and occupancy.

(5) The system allows visual inspection of the detection results along with the actual conditions for validation purposes and for optimizing the detector placement.

Using a CVS for traffic data collection becomes more and more feasible for the following reasons. Firstly, the established trend of falling prices for video equipment has led to an enormous increase in the use of video equipment for traffic monitoring and control, surveillance of motorways, tunnels and urban streets. Secondly, image processing hardware costs have fallen while computational power has increased, making automatic video scene analysis a realistic possibility.

1.2 LITERATURE REVIEW

In this section, a brief review is presented of research and development in the application of computer vision systems (CVS) for traffic monitoring and data collection. Detailed reviews of particular systems will be presented in chapters 3, 4, 5, and 6, respectively.

Research on the CVS for traffic monitoring and data collection has been in progress in Europe, Japan and the United States for about two decades. Most of this research work, dating back to the 1970's, was referred to in Hilbert (1978), Dickinson (1984) and Inigo (1985).

The earliest research on the application of image-based systems for traffic monitoring was performed at the University of Tokyo in 1973 (Onoe 1973, 1976). At that time, the developed system was limited to a crude estimate of vehicle speeds. In the early 1980's, at the same university, experiments were conducted to demonstrate that traffic-flow data could be obtained in real time from TV data (Takaba 1982). In 1989, Shimizu and Shigehara of the Metropolitan Expressway Public Corporation (Shimizu 1989) reported on their evaluation of five image-processing systems, all demonstrating comparable performance. As a result of these experiments, the authors started to investigate the methods for utilizing these systems in their traffic control system.

Over the past ten years, the British have been very active in research directed toward the application of image processing to the traffic monitoring problem (Dickinson 1989; Hoose 1989; Rourke 1989; Wan 1989). The British researchers have not restricted themselves to PC/AT compatible systems but rather have investigated the use of different processing architectures and r ore complex algorithms (Wan 1989). They have also explored vehicle classificati'n (Houghton 1989; Rourke 1989) and tracking (Houghton 1989). Researchers in other European countries (Burkhard 1987; Blosseville 1989; Enkelmann 1989; Versavel 1989) have also explored the use of image-based systems for traffic surveillance. Versavel (1989) described a product, Camera and Computer-Aided Traffic Sensor (CCATS), 'or vehicle length and speed measurement. The CCATS, which was developed in conjunction with the Belgian Government and the University of Leuven, was produced by the Traficon NV (previously named Devlonics) company. Blosseville, et al. (1989) developed another experimental traffic sensor system called TITAN with the goal of fully automating the process, from lane detection to vehicle detection. In the case of light traffic conditions, TITAN was able to measure traffic volume and vehicle speed. But in the case of congested traffic, only flow speed could be obtained. No quantitative performance data have been reported, but the processed imagery provided in the papers (Blosseville 1989) clearly indicates that the TITAN system performs quite well.

In the mid-1970's, the U.S. Department of Transportation made the first significant effort in the United States to apply imaging technology for automatic traffic monitoring. The Federal Highway Administration (FHWA) funded research at the Jet Propulsion Laboratory (JPL) in Pasadena to develop a breadboard system called a Wide Area Detection System (WADS), for vehicle detection and vehicle speed measurement (Hilbert 1978). The system's name was later changed to SCAN (Sensor for Control of Arterials and Networks). The program concluded with a field test of WADS in the Los Angeles vicinity along the Santa Monica Freeway (Schlutsmeyer 1982). For more detail on other efforts in the United States see (Inigo 1985; Michalopoulos 1990). The Institute of Transportation Studies at the University of California at Berkeley performed a research project to review, evaluate, select, and test detectors for freeway surveillance and control (Labell 1989; Spencer 1989). This was sponsored by the California Department of Transportation (Caltrans) and the FHWA. As part of the project, the California Polytechnic State University, San Luis Obispo, completed an evaluation of eight image processing traffic monitoring systems (three are commercially available, and the remaining five are prototypes) for Caltrans in 1991 (Chatziioanou 1991; Hockaday 1991). The result of the evaluation indicated that accuracy of measurement was over 95% for "ideal" conditions, but lighting transition periods, shadows and reflections may each reduce accuracy by up to 20-30 percent. Therefore, further studies were needed to improve the algorithms.

Artificial neural networks (Krose 1991; Maren 1990; Zornetzer 1990) are an emerging computational technology which can significantly enhance computer vision systems. They offer the advantages of learning from example, self-organization, fault tolerance, fast data processing, and ease of insertion into existing and newly developed systems. Neural networks achieve these abilities by using large numbers of simple, interconnected processing units which operate in logical parallelism.

In recent years, more and more researchers are focused on the applications of neural works to solving problems in transportation engineering, such as traffic control problems (Funabiki 1993), incident detection (Hsiao 1994), gap acceptance at stop-controlled intersections (Pant 1994), urban noise prediction (Cammarata 1993), pavement-crack detection (Kaseko 1994), and vehicle license character recognition (Lisa 1993), etc.

Neural network technology could be considered when developing a CVS for traffic data collection and analysis. It can help to solve complex problems which may be poorly structured and cannot be adequately described by a set of rules or equations. For instance, it is difficult for traditional pattern recognition techniques to locate a vehicle license plate or to determine the number of axles of a vehicle from video images. In these cases, the ability of a neural network to detect and locate objects in images (Vaillant 1993) could be considered. The main idea of the neural network method is to train itself to detect the presence or absence of an object by scanning all possible locations in the image. By using the appropriate neural network architecture (Maren 1990; Vaillant 1993), this process can be very efficiently accomplished without it being necessary to recompute the entire network state at each location. In order to solve problems in vehicle tracking or high-density pedestrian flow measurement, a neural network with recurrent architecture similar to that developed by Elman (1990) could be considered. This is because the Elman recurrent net has the simplest architecture and can predict the motion of moving objects (Meng 1993).

The reviewed literatures indicate that a variety of computer vision systems have been developed for traffic data collection. However, few systems are able to extract certain higher level traffic information such as vehicle count by body shape, count of vehicles intending to turn, and measurement of pedestrian flow at intersections. In addition, previous systems mostly suffered from problems of accuracy reduction caused by shadows and reflections from vehicles or environment. In order to meet practical requirements, improvements are necessary to address these problems.

1.3 SCOPE AND OBJECTIVES

The primary objective of this research is to develop theoretical models and algorithms which could be considered in advancing of existing computer vision systems in some particular areas of traffic data collection and analysis.

The new models and algorithms are limited to the processing of two-dimensional video images that are taken in daytime situation. Three dimensional images and images taken at night are not considered in this study. The new algorithms intend to generate real-time traffic data on vehicle count, speed, volume and classification as well as pedestrian count and volume measurement. These algorithms also intend to increase the detection accuracy by reducing the errors that may result from the effect of shadows. The errors caused by occlusion will not be dealt with.

The specific objectives and scopes are described below.

1. To increase the detection accuracy by solving shadow problem. One of the problems the existing video traffic detection systems suffer from is accuracy reduction caused by shadows from vehicles. Such reduction could be up to 20-30 % due to the inherent weakness of the systems to deal with this problem (Chatziioanou 1991). To increase the detection accuracy, color images are utilized in this study instead of monochrome images. A new method is developed for video vehicle detection on the basis of color image processing techniques. But its capability is limited to the processing of images in which color is synthesized with a weighted sum of three principle colors: red, green and blue.

2. To expand the capability for blinking signal lights detection. New models and algorithm are developed to detect turning signal lights in the front of vehicles which are within about 30 meters from a video sensor. They are not intended to provide the detection of the lights in motion or too far from the video sensor. The detection of other kinds of blinking lights such as ones on emergency vehicles or police cars is not provided although the models and algorithm can be extended for this application.

3. To increase the capability of vehicle classification. The capability of the existing systems is limited in classifying vehicles into several simple categories (Houghton 1989; Lu 1991; Pan 1991; Rourke 1989). The newly developed models and algorithm intend to extract vehicle information not only on length but also on height, width, and shape in order to make complicated classification. The information on vehicle shape includes the front shape of vehicle and the number of units of a vehicle, the number

of axles is not included.

4. To introduce a new algorithm for estimating the number of pedestrians at intersections. Currently, pedestrian count is usually performed manually. Manual counting is expensive and not suitable for a large volume of pedestrians. Another way to count pedestrian is the usage of automatic counter, which consists of detector pads laid on the sidewalk and connected to a counting device (Cameron 1977). This device is probably the best volume determination system currently available, but it cannot be used at intersections due to the presence of vehicles. Also, this device is inflexible after installation and incapable of measuring other pedestrian flow data such as speed and walking direction. In this research, on the basis of computer vision techniques, a new algorithm is developed for automatically counting the number of pedestrians at intersections. The algorithm is able to differentiate pedestrians from vehicles, and detect pedestrians moving in both directions on a crosswalk under "ideal" conditions. It doesn't consider conditions such as lighting transition periods, shadows and reflections.

1.4 THESIS ORGANIZATION

In the following chapters, detailed descriptions of the newly developed models and algorithms will be presented for traffic data collection and analysis using the computer vision system.

In Chapter 2, four general methods are presented for moving object detection, color detection, noise removal and perspective projection, respectively. These methods are employed in each of the four algorithms developed in chapters 3 to 6.

Chapter 3 presents the development of a new algorithm to measure traffic volume and speed. The algorithm, based on the new models, will overcome the disadvantages of previous approaches and improve detection accuracy.

In Chapter 4, two models are developed to detect vehicle signal lights and recognize turning-signal lights, respectively. Using these models, a new algorithm is formed for detection of vehicles intending to turn at intersections. The algorithm will process color image sequences in order to distinguish signal lights and the reflections of objects as well as differentiate turning signal lights from other vehicle lights.

In Chapter 5, four models are developed to form a tree type of vehicle classifier. The new models can be used to obtain vehicle data not only of length but also of height, width and type of vehicle shape. Using these new kinds of data as classification parameters, the classifier is able to classify vehicles into many complicated categories. Therefore, the gap between the requirement for vehicle classification data and its availability can be greatly reduced.

Vehicles and pedestrians are two primary elements of traffic. Besides vehicle data, pedestrian flow data are also heavily in demand to facilitate the geometric and operational design of intersections. Unfortunately, the literature review indicated that no device is available that can automatically collect and analyze pedestrian flow data at intersections. To make use of computer vision techniques, Chapter 6 presents a new method that can automatically measure the pedestrian volume at intersections. The most important function of the new approach is its capability to measure the multidirectional flow on a crosswalk without being affected by vehicles passing through the detection spot.

Concluding remarks and recommendations for future work are presented in Chapter 7.

CHAPTER 2

METHODOLOGY

2.1 INTRODUCTION

A computer vision system which is illustrated in Fig. 2.1 is proposed in this thesis for the purpose of traffic data collection. The method developed to support this system utilizes video images as input. These images are digitized to facilitate computer processing of traffic information included in the images. The method consists of eight newly developed models which are NR model, PP model, DOS model, DOB model, LM model, WHE model, PCE model, and TTC model (See Fig. 2.2). These models are used to develop four algorithms, each dedicated to a specific function of traffic data collection.

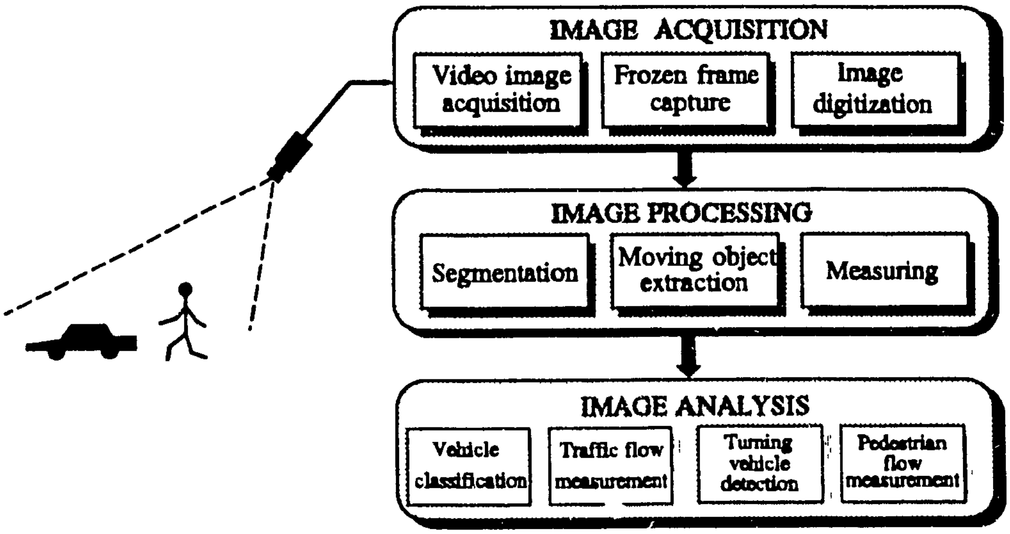


Fig. 2.1 A Computer Vision System for traffic data collection and analysis

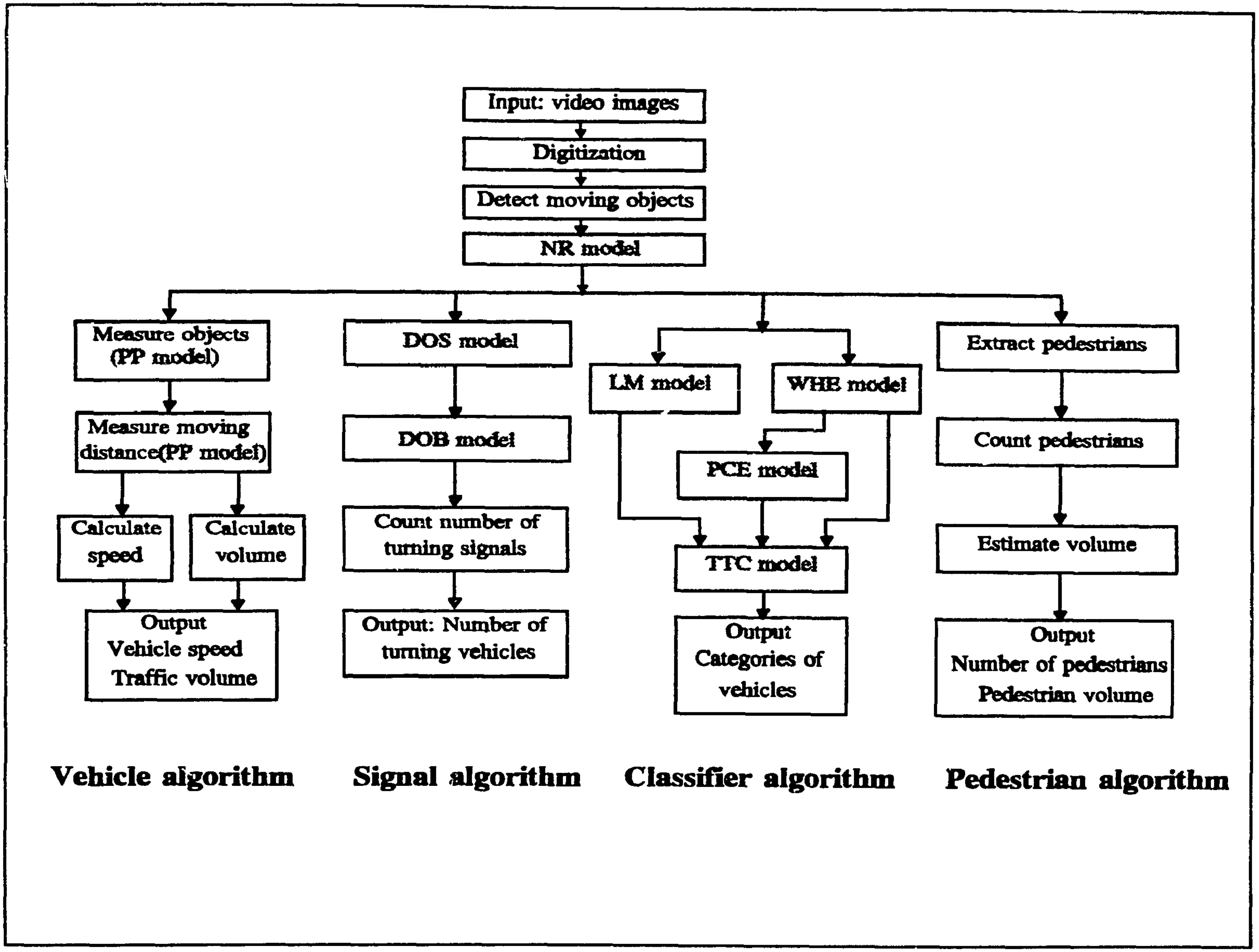


Fig. 2.2 Models and algorithms

In this chapter, general methods for image processing will be developed and presented. These approaches are moving object detection, color detection, noise removal and perspective projection which are common to all the processes in the following chapters.

2.2 MOVING OBJECT DETECTION

The purpose of moving object detection is to separate all moving objects from stationary backgrounds in the traffic images. Since the stationary background such as road surface or marks on the pavement surface will be overlapped while a moving object passes by, moving objects can be segmented by testing color changes when the scene shifts from background to a moving object.

Suppose that a *reference image*, say $\boldsymbol{P_R}$, which contains only stationary components is stored in the computer. It is compared with an image having the same environment but including moving objects such as vehicles or pedestrians, say $\boldsymbol{P}$, which is called *measured image*. As a result, a *difference image* is obtained as

$$\boldsymbol{P_D} = | \boldsymbol{P} - \boldsymbol{P_R} | \qquad (2.1)$$

In the *difference image*, $\boldsymbol{P_D}$, only moving objects remain because the stationary background is removed due to the difference between the two images.

The *difference image* is still a grey-level or color image. In order to make geometric measurements within the image, it is necessary to convert the *difference image* into a *binary image* in which each pixel has one of only two status, considered as *of* *interest* or *background.* The operation to convert a grey-level or color image into a *binary image* is called *segmentation.* There were two major approaches to image segmentation: edge based and region based (Young 1986). In edge-based methods, the local discontinuities are detected first and then connected to form longer, hopefully complete, boundaries. In region methods, areas of image with homogeneous properties (such as intensity or color) are found, which in turn give the boundaries. The two methods are complementary and one may be preferred over the other for some applications. In the case of traffic image segmentation, *thresholding* (one of region segmentation techniques) is preferred.

A *binary image* is obtained by thresholding:

$$P_B = \begin{cases} 1, & \text{if } P_D > \varepsilon \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

where P_B denotes a *binary image.* ε denotes a predetermined threshold. All pixels above ε are treated as black and all below as white. The system treats black as *moving object* and white as *background.*

2.3 COLOR DETECTION

A method is developed in this section to detect the occurrence of a particular color in images. Before the method is presented, a brief description of color precedes in the following section.

2.3.1 Color Representation in Digital Image

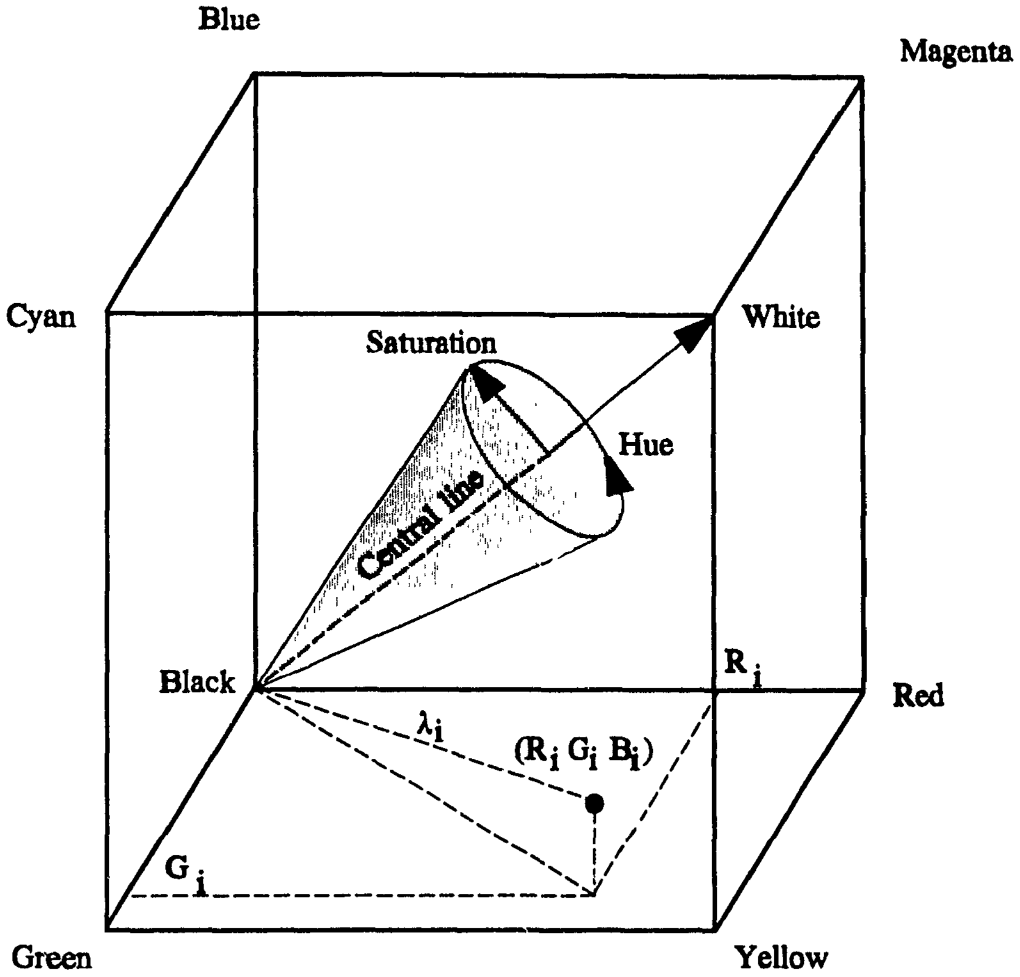


Fig. 2.3 RGB color cube

Color in a digital image used in this study is synthesized with a weighted sum of three principle colors: red, green and blue. Let ***R***, ***G*** and ***B*** denote unit vectors corresponding to the three principal colors; then every color is represented by components of these vectors. These three values are called *tristimulus values* that range from the weakest 0, indicating the dimmest level, to the strongest, say 31 or 256 etc., indicating the brightest. Any color in the digital image is represented by a vector (R, G, B). If one thinks of the origin-corner of a cube, vectors $\boldsymbol{R}$, $\boldsymbol{G}$ and $\boldsymbol{B}$ extend along each of the three edges as shown in Fig. 2.3. White light is represented by a line equidistant from all three edges, i.e. the line which joins the furthest corner to the origin. This line is called *central line*. The stronger the light the longer the line, indicating a brighter degree of light. If the light is colored, the line inclines more towards one of the edges or lies somewhere between two of them. The direction in which it inclines represents the *hue*, defining what kind of color it is, an^d the extent to which it departs from the central line is the *saturation*, the purity of the color; if it coincides with an edge, it is fully saturated in that color. In a color image, each pixel with a particular color can locate itself in this cube.

As shown in Fig. 2.3, suppose pixel i locates itself at point (R_i, G_i, B_i) in the color cube, then its brightness is defined as the length of the line joining the origin to the point. Let λ_i denote the brightness of pixel i and be calculated as

$$\begin{gathered}\lambda_i = \sqrt{R_i^2 + G_i^2 + B_i^2} \\ R_i,\ G_i,\ B_i \geq 0\end{gathered} \tag{2.3}$$

The brightness λ_i ranges from 0.00, indicating black, to a maximum value, indicating white, when all the values of R_i, G_i, and B_i are equal to the maximum value. The values between them are the brightness of the other colors in the color cube.

2.3.2 Color Detection Functions

To identify the occurrence of a particular color in an R-G-B image, one can select all the pixels which have precisely the required values of ***R*, *G*** and ***B***. This represents a single point in the color cube.

In the real world there are variations in the perceived color of objects which have the same color. These variations are due to illumination conditions, surface texture or orientation effects as well as the electronic noise from the video capture system. It is therefore realistic to look for ranges of color combinations rather than single-point colors.

Let (c_R, c_G, c_B) denote a color with variation limits of r_R, r_G, *and* r_B in R, G, and B, respectively. In order to define the region of this color, the variations must be estimated. This can be done by sampling the pixel brightness over a training area to give three histograms, one for each hue. Meanwhile, c_i and r_i, where i = R, G, and B, can be calculated as follows.

$$c_i = \frac{1}{N} \sum\sum_{(x,y) \in \Omega} i(x,y) \qquad (i = R,\ G,\ B) \tag{2.4}$$

and

$$r_i = \frac{z}{2} \sqrt{\frac{\sum\sum_{(x,y) \in \Omega} [\ i(x,y) - c_i\ ]^2}{N-1}} \qquad (i = R,\ G,\ B) \tag{2.5}$$

where (x,y) are coordinates of a point in an image; *R(x,y), G(x,y), and B(x,y)* are the ***tristimulus values*** of a pixel *p(x,y)* in the training area; Ω denotes the training area which includes N pixels; and z is a statistic based upon the desired "confidence" in the value of r: 1.96 for 95% confidence, and 3.0 for 99.7% confidence.

In order to test an input pixel against a required color, each of its *tristimulus* *values*, *R(x,y)*, *G(x,y)* and *B(x,y)*, is tested separately. If, for example, the value of *R(x,y)* is located within the region $[c_R - r_R, c_R + r_R]$, the region of variation of a color, then the pixel is regarded as 1/3 match of the color. Otherwise, if the value of *R(x,y)* is not within the region $[c_R - r_R, c_R + r_R]$, then the pixel is regarded as 1/3 different from the color. The above procedure can be generally expressed by a function as

$$f_i(x,y) = \begin{cases} \frac{1}{3}, & \text{if } (c_i - r_i) \leq i(x,y) \leq (c_i + r_i) \\ 0, & \text{otherwise} \end{cases} \qquad (i = R,\ G,\ B) \qquad (2.6)$$

where $f_R(x,y)$, $f_G(x,y)$ and $f_B(x,y)$ are *match indexes* for each of the *tristimulus values*, *R(x,y)*, *G(x,y)* and *B(x,y)*, respectively.

For determining the color of the pixel, a function is defined as

$$f_D(x,y) = f_R(x,y) + f_G(x,y) + f_B(x,y) \qquad (2.7)$$

where $f_D(x,y)$ denotes a *color detection function* which has four values: 0, 1/3, 2/3, and 1. Each value presents a level of color match. The value of 1 means a 100% match of two colors while the value of 0 a complete difference.

2.3.3 Application of Color Detection Functions

The color detection function can be used to detect the asphalt pavement surface and the shadow of vehicle. To detect the asphalt pavement surface, a study was made of its color and inherent variation. The result showed that the values of ***R***, ***G*** and ***B*** were in approximately equal balance representing the color of asphalt pavement surface. Therefore, this color is normally found in a ball-shaped range with radius *r* and center (*c*,

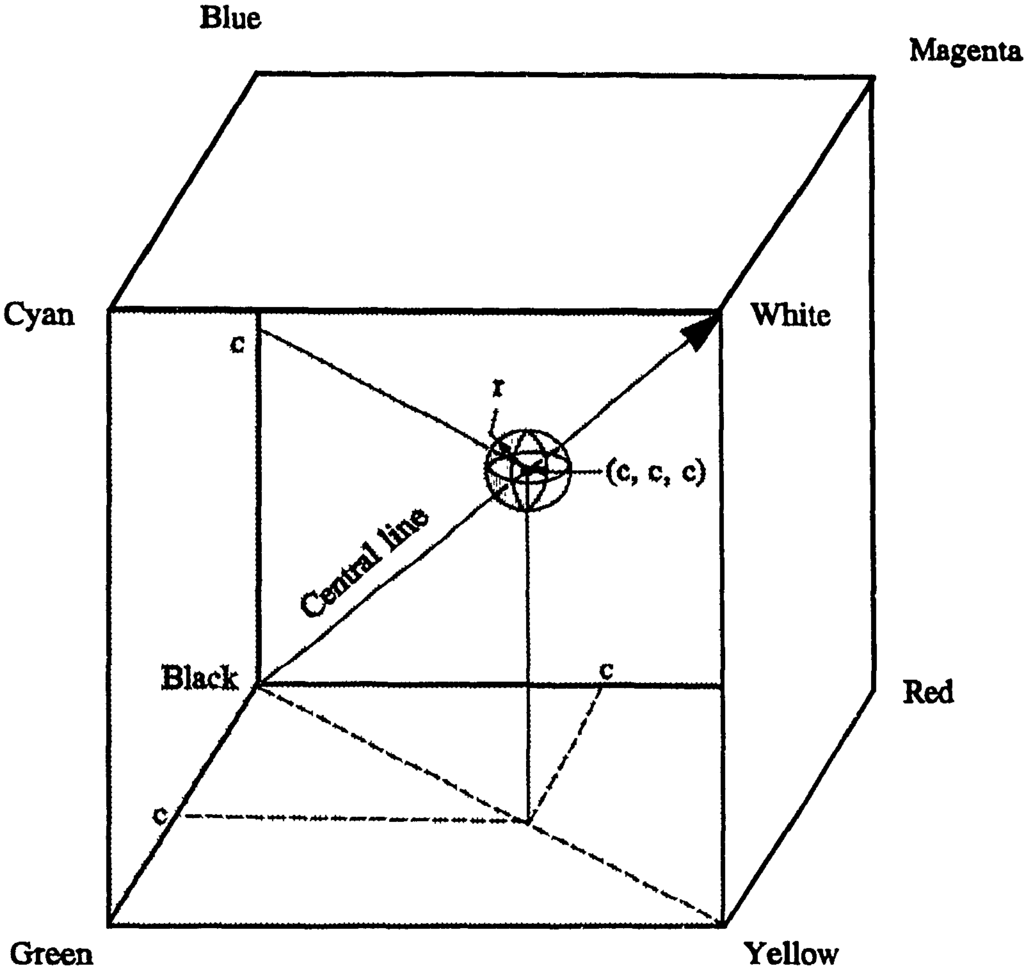


Fig. 2.4 Color in the color cube

c, c) in the color cube as shown in Fig. 2.4. Variation of the illumination condition on the pavement surface changes the location of this range along the *central line*. The brighter the pavement surface, the higher the position of the range and the dimmer, the lower. This is because the surface of asphalt pavement has a fixed hue and saturation under sunlight condition. Its *tristimulus values* shift on a line, the *central line*, when the brightness varies.

The position of the color of asphalt pavement surface can be found by computing the value of c as follows:

$$c = \frac{1}{3} \left(c_R + c_G + c_B \right) \tag{2.8}$$

where c_R, c_G, *and* c_B are obtained by Equation (2.4).

The radius r is then determined to form a ball-shaped range that includes most of pixels in Ω. r is expressed as

$$r = \frac{1}{3} \left(r_R + r_G + r_B \right) \tag{2.9}$$

where r_R, r_G, *and* r_B are obtained by Equation (2.5).

Sequentially, the *match indexes* can be calculated with the reference of parameters c ai·d r, and

$$f_i(x,y) = \begin{cases} \frac{1}{3}, & \text{if } (c-r) \leq i(x,y) \leq (c+r) \\ 0, & \text{otherwise} \end{cases} \qquad (i = R, G, B) \tag{2.10}$$

Hence, from Equation (2.7), the *color detection function* can be computed.

Shadows on the pavement surface can also be detected by the color detection function. The difference existing between the colors of the pavement surface with and without shadow is the change of brightness. If the color of the pavement surface without shadow is (c, c, c) with variation limit of r, then the color of the pavement surface with shadow is equivalent to a color shift from (c, c, c) to $(\omega_R c, \omega_G c, \omega_B c)$. Consequently, the *match indexes* will have the following form

$$f_i(x,y) = \begin{cases} \frac{1}{3}, & \textit{if} \;\; (\omega_i c - r) \leq i(x,y) \leq (\omega_i c + r) \\ 0, & \textit{otherwise} \end{cases} \qquad (i = R,\, G,\, B) \quad (2.11)$$

where ω_R, ω_G, *and* ω_B denote weights that will be discussed below.

Discussion of ω_R, ω_G, and ω_B

The values of ω_R, ω_G, and ω_B define the quantitative relationship between the colors of the pavement surface with and without shadow. If the color of the pavement surface without shadow is at point $(c,\ c,\ c)$ in the color cube, then it will shift to $(\omega_R c,\ \omega_G c,\ \omega_B c)$ when the pavement surface is covered with the shadow of vehicle or pedestrian. The values of ω_R, ω_G, and ω_B vary from greater than zero to less than one, depending upon the variation of brightness of the pavement surface while being covered with shadow. The more remarkable the variation, the smaller the values of ω_R, ω_G, and ω_B meaning the further the distance between points $(c,\ c,\ c)$ and $(\omega_R c,\ \omega_G c,\ \omega_B c)$.

The values of ω_R, ω_G, and ω_B are obtained as follows:

$$\omega_i = \sum_{(x,y)\in\Omega}\sum \frac{i(x,y)}{cN} \qquad (i = R,\ G,\ B) \qquad (2.12)$$

where, similar to the previous definitions, Ω denotes the training area which includes N pixels; *R (x,y), G(x,y), and B(x,y)* are the *tristimulus values* of the pixel *p(x,y)*.

In practice, Ω can be obtained by using a *training image* where only the color of pavement surface under shadow is known to occur.

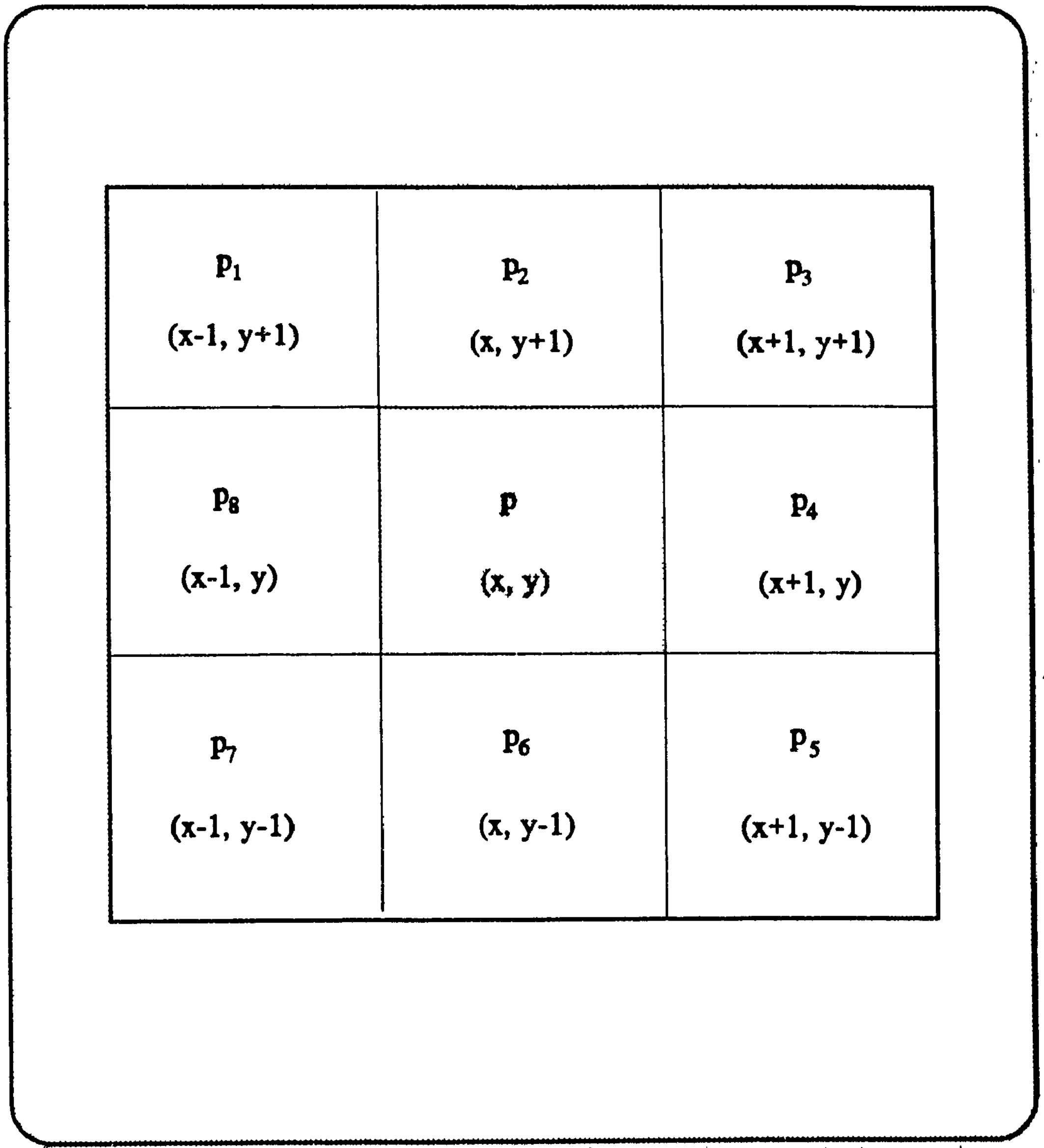
Since the brightness on the pavement surface varies hour by hour with light and weather conditions, it is necessary to update periodically the values of ω_R, ω_G, and ω_B for satisfactory detection of shadow on the pavement surface over an extended period of time. Normally, the values of ω_R, ω_G, and ω_B are updated after a constant time interval, say one minute, if no sudden luminance change occurs on the detected pavement surface during this period of time. But they must be updated immediately after any major change of luminance due to weather conditions.

2.4 NOISE REMOVAL MODEL (NR model)

There are two kinds of noises that usually occur in images after the segmentation operation. One belongs to *fine noise*. This kind of noise is usually an isolated point. In other words, each noise point has non-noise neighbours. The other kinds of noises are called *coarse noises* because each of them is composed of a cluster of points. The method for *coarse noises* removal is application dependent because it needs to distinguish the *coarse noises* from clusters of interest before the elimination. For example, in some cases (Lu 1988), the size, width, and height of a cluster are used as classification parameters to make differentiations. The removal of the *coarse noises* will not be discussed here. The model developed in this section is for *fine noise* removal.

A 3 x 3 pixel scanning window can be used for *fine noise* detection and elimination. Figure 2.5 shows the scanning window that contains the scanned pixel (x,y) itself and its 8-neighbour pixels.

The 8 neighbours of a pixel $p(x,y)$ is defined by a set of pixels $\{p_1(x-1,y+1)$,



p_1 (x-1, y+1)	p_2 (x, y+1)	p_3 (x+1, y+1)
p_8 (x-1, y)	p (x, y)	p_4 (x+1, y)
p_7 (x-1, y-1)	p_6 (x, y-1)	p_5 (x+1, y-1)

Fig. 2.5 3 x 3 pixel scanning window

$p_2(x,y+1)$, $p_3(x+1,y+1)$, $p_4(x+1,y)$, $p_5(x+1,y-1)$, $p_6(x,y-1)$, $p_7(x-1,y-1)$, $p_8(x-1,y)\}$ and denoted by $N_8(x,y)$, e.g. $N_8(x,y) = \{p_1, p_2, ..., p_8\}$.

With reference to the neighbourhood arrangement $N_8(x,y)$, the point $p(x,y)$ is regarded as a *fine noise* depending on whether or not the following condition is satisfied.

Condition

Let $\{p_1, p_2, ..., p_8\}$ be eight neighbours of the pixel $p(x,y)$.

If $\quad p(x,y) \neq 0 \cap \exists i \ (p_i = 0)$,

or $\quad p(x,y) = 0 \cap \exists i \ (p_i \neq 0)$, where $p_i \in N_8(x,y)$,

then this pixel is a noise point. This is illustrated in Fig. 2.6.

If a pixel is a fine noise point, then its value will be replaced by the weighted average of the values of 8-neighbour pixels, and

$$p(x,y) = \frac{1}{8} \sum_{i=1}^{8} W_i \, p_i \tag{2.13}$$

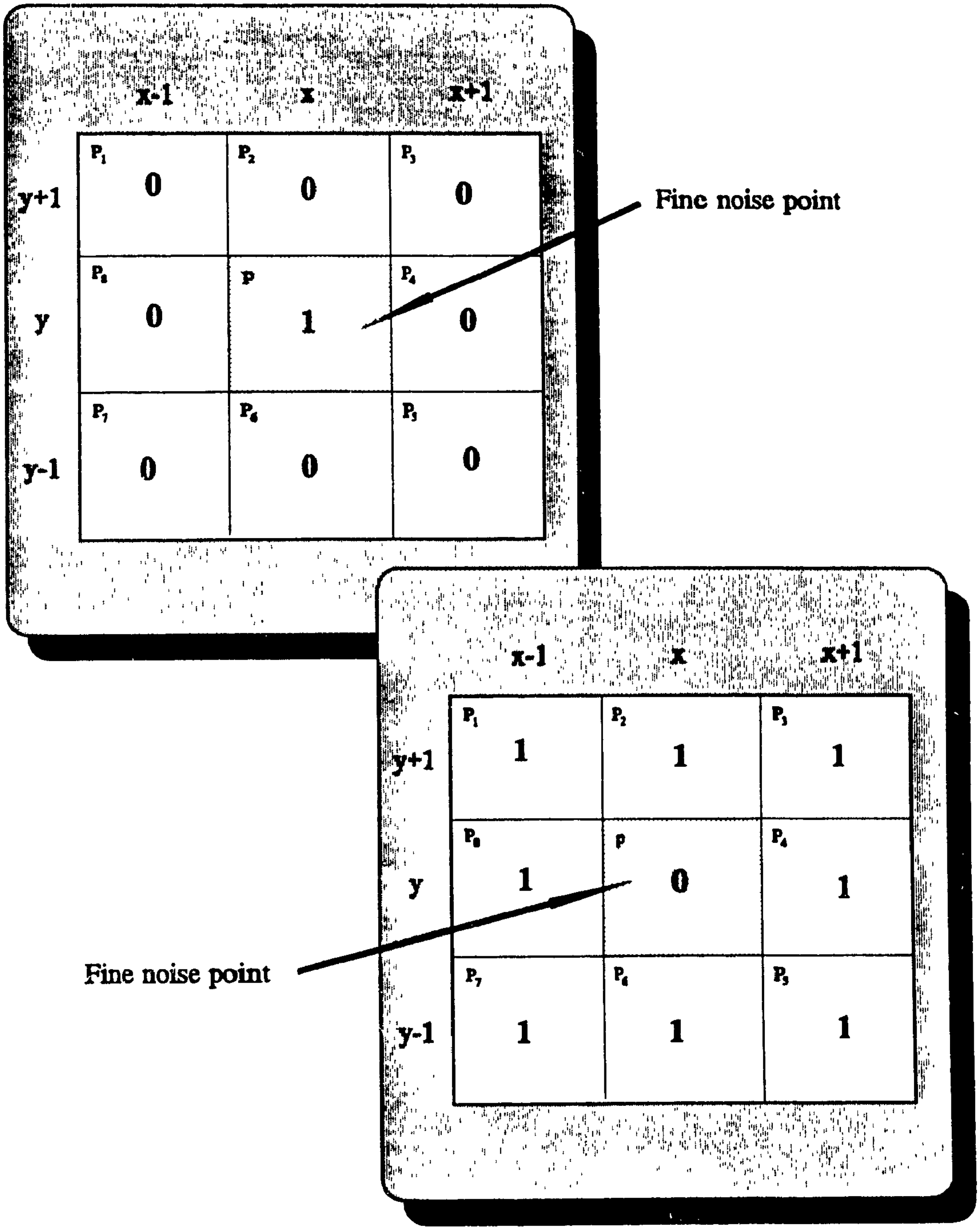
where W_i is the weight of the neighbour pixel p_i, and $p_i \in N_8(x,y)$.

2.5 PERSPECTIVE PROJECTION MODEL (PP model)

The purpose of the perspective projection model (PP model) is to transform images from image plane to horizontal plane. As exemplified in Fig. 2.7, if point A_1 is an image of a space point A and its location in an image plane is known, the PP model can locate another image of the point, A_2 in a horizontal plane defined in the figure.

2.5.1 Principle of Perspective Projection

The relationship between a camera and a scene defines the geometry of the picture-taking process. The simplest model of the picture-taking process is the pinhole-camera model that is usually used to derive the direct perspective transformation that



	x-1	x	x+1
y+1	P_1 0	P_2 0	P_3 0
y	P_8 0	P 1	P_4 0
y-1	P_7 0	P_6 0	P_5 0

	x-1	x	x+1
y+1	P_1 1	P_2 1	P_3 1
y	P_8 1	p 0	P_4 1
y-1	P_7 1	P_6 1	P_5 1

Fig. 2.6 Find a noise point

shows how points on an object are projected onto points in an image. A basic pinhole-

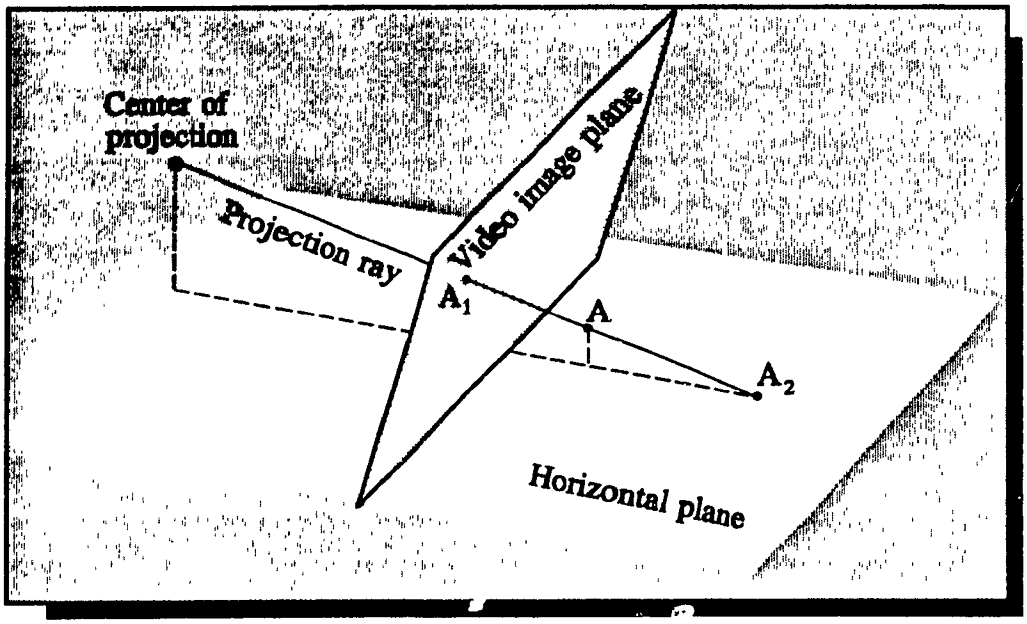


Fig. 2.7 Perspective Projection

camera model is shown in Fig. 2.8. The camera is represented by a pinhole lens together with an image plane lying a distance of *d* behind the lens. The actual lens on a real photographic or television camera can always be replaced by a mathematically equivalent pinhole lens at an appropriate distance from the image plane. The image of a point *A* in three-dimensional space is determined by the intersection of the image plane with the projecting ray defined by the line that joins *A* and the lens centre. The intersection point A_I is the image point corresponding to the object point *A*.

While th^e model of Fig. 2.8 gives a mapping from the scene to the image plane in which the images are flipped left to right and upside down, it is more convenient to intercept the projecting ray with a front image plane as shown in Fig. 2.9. The process illustrated in Fig. 2.9 is called *central projection*, the lens point being the *center of*

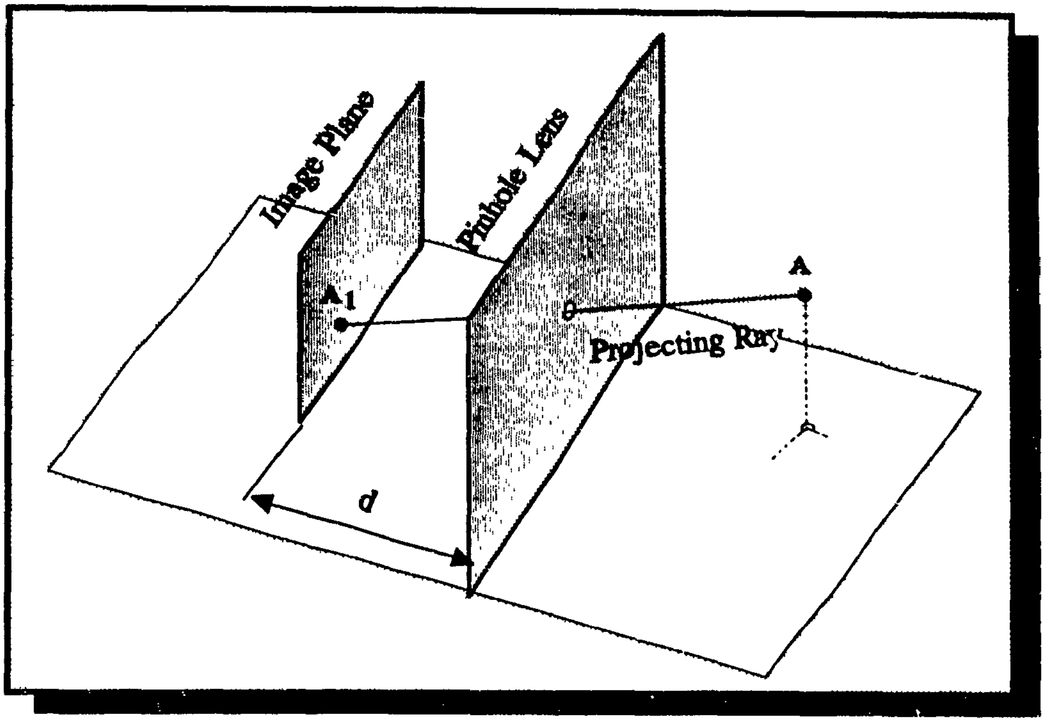


Fig. 2.8 Basic pinhole-camera model

projection.

For simplicity, the center of the image plane is defined as the origin, and the Z-axis as the line that intersects the pinhole and the origin. Thus the pinhole is at (0, 0, -*d*) and points on the image plane are at $(x_A, y_A, 0)$.

The direct perspective transformation for the simple case depicted in Fig. 2.9 can be derived intuitively. By inspection, the answer involves only similar triangles. If A = (x, y, z) denotes a point in three-dimensional space and $A_I = (x_A, y_A, 0)$ a corresponding point in image plane, then clearly

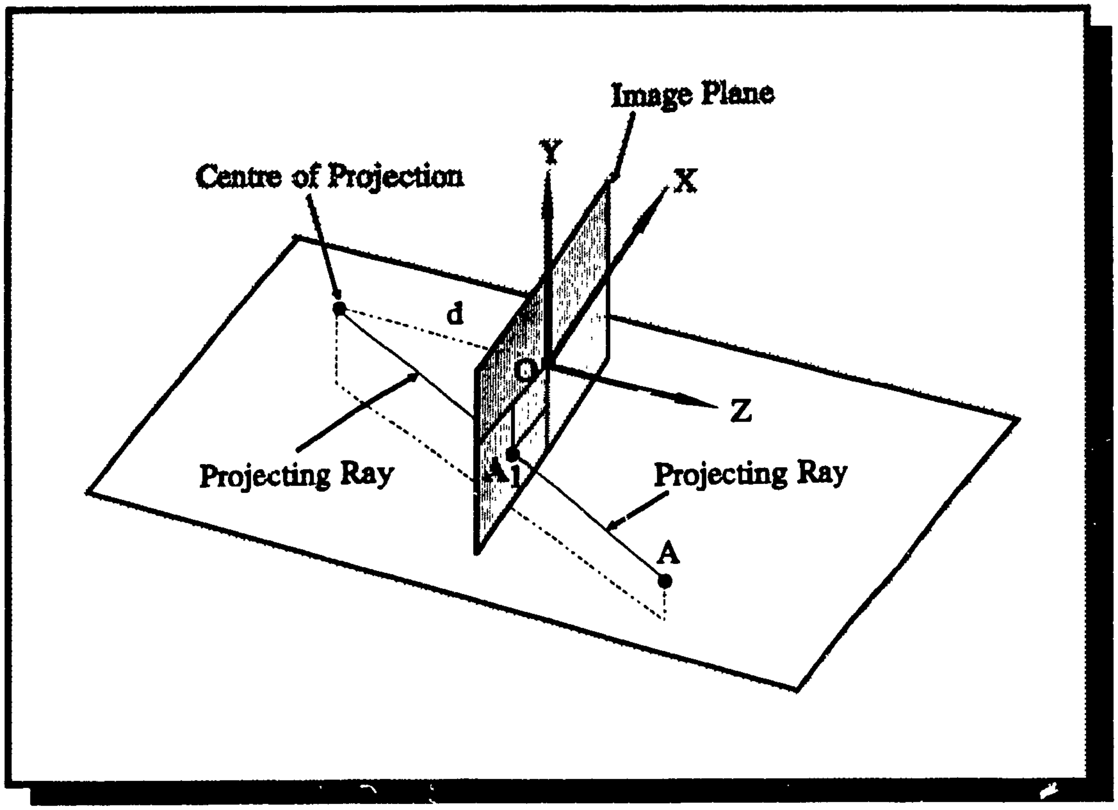


Fig. 2.9 Central projection model

$$\frac{x_A}{d} = \frac{x}{d+z}$$

$$\frac{y_A}{d} = \frac{y}{d+z}$$

or

$$x_A = \frac{d}{d+z}\,x$$
$$y_A = \frac{d}{d+z}\,y \qquad (2.14)$$

Equation (2.14) maps an object point expressed in three-dimensional coordinates into an image point on an image plane.

Equivalently, each point in the image plane can be mapped onto a line in three-dimensional space by a *back projection*:

$$x = \frac{(d+z)}{d} x_A$$
$$y = \frac{(d+z)}{d} y_A \qquad (2.15)$$

These equations represent a line passing from the center of projection through the image point A_I, where z (depth) is a free parameter. In other words, all points on this line are projected to the image point $A_I = (x_A, y_A, z_A)$.

2.5.2 PP Model

An *image plane* is defined as a plane that is perpendicular to the projection line (Fig 2.9). A *horizontal plane* is defined as a plane that is horizontal in three-dimensional space.

In practice, the transformations given in the previous section are hard to use because of the awkwardness of the coordinate system. It is true that the single coordinate system of Fig. 2.9 is very convenient for locating points in the *image plane*. But it is notably inconvenient for locating points in the *horizontal plane*, since it is constrained to measure distances to a set of axes whose position is determined by the camera. In other words, the system used in the last section is "camera centric," and this is often unnatural and bothersome. In order to meet the situation for traffic data collection, it is required to project points from the *image plane* onto the *horizontal plane* as illustrated in Fig. 2.7.

In a *global coordinate system*, a projection of a point from (ξ, η, ζ) to (ξ_l, η_l, ζ_l) can be derived intuitively by inspection of Fig. 2.10, and

$$\begin{aligned} \xi_l &= \frac{H}{H - \zeta}\,\xi \\ \eta_l &= \frac{H}{H - \zeta}\,\eta \qquad (2.16) \\ \zeta_l &= 0 \end{aligned}$$

The above equation requires that the points be expressed as (ξ, η, ζ) by the *global coordinate system*. Unfortunately, in practice, the points in the *image plane* are expressed as (*x*, *y*, *z*) by the *image coordinate system*. Therefore, a transformation between the two coordinate systems are required as shown in Fig. 2.11. Mathematically, this transformation can be represented in the following form:

$$\begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{bmatrix} x-x_0 \\ y-y_0 \\ z-z_0 \end{bmatrix} \qquad (2.17)$$

where

x, y, z = coordinates in the *image coordinate system*;

ξ, η, ζ = coordinates in the *global coordinate system*.

It will be a one-to-one mapping, if the determinant of matrix T does not equal to 0, as

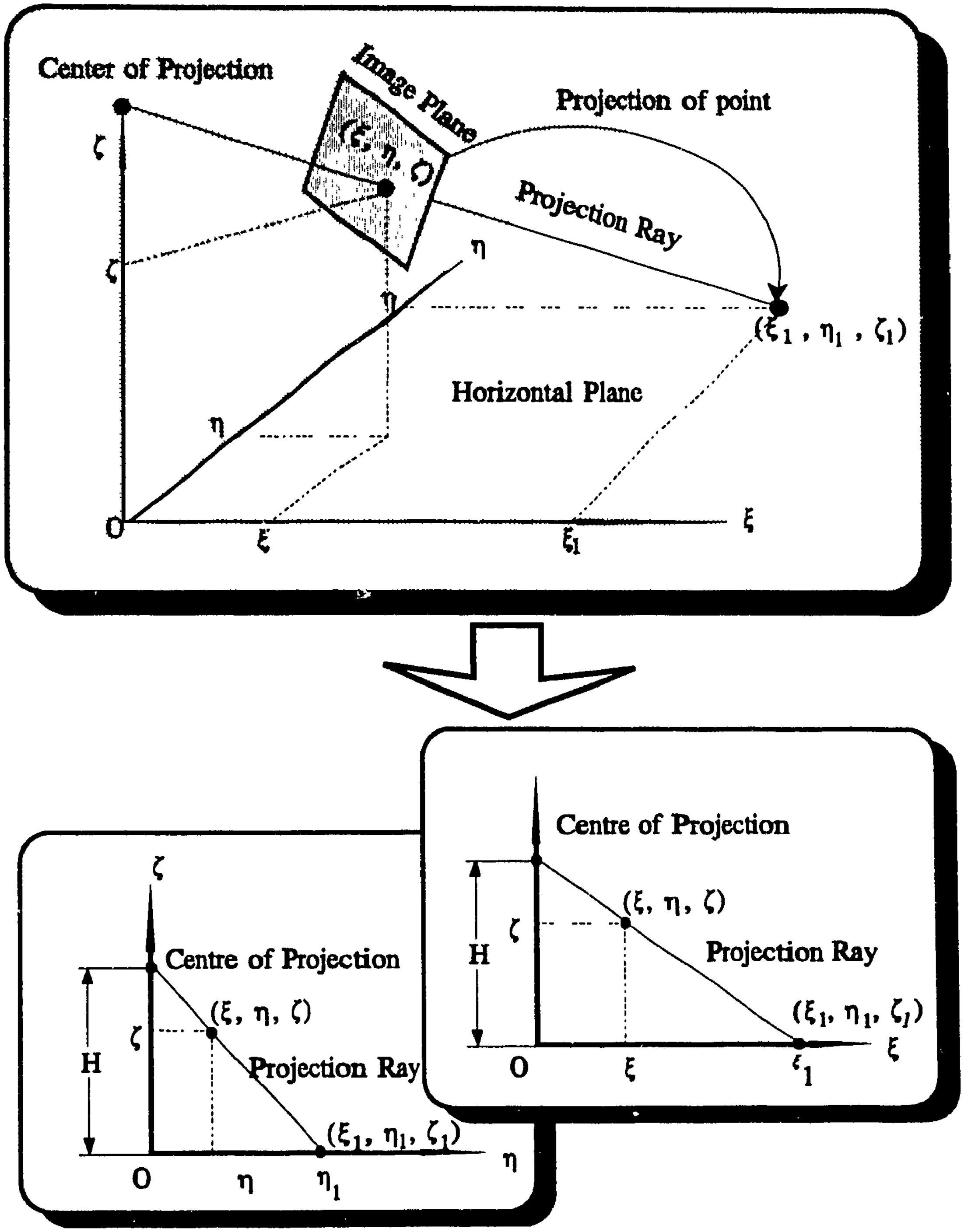


Fig. 2.10 Projection of point in global coordinate system

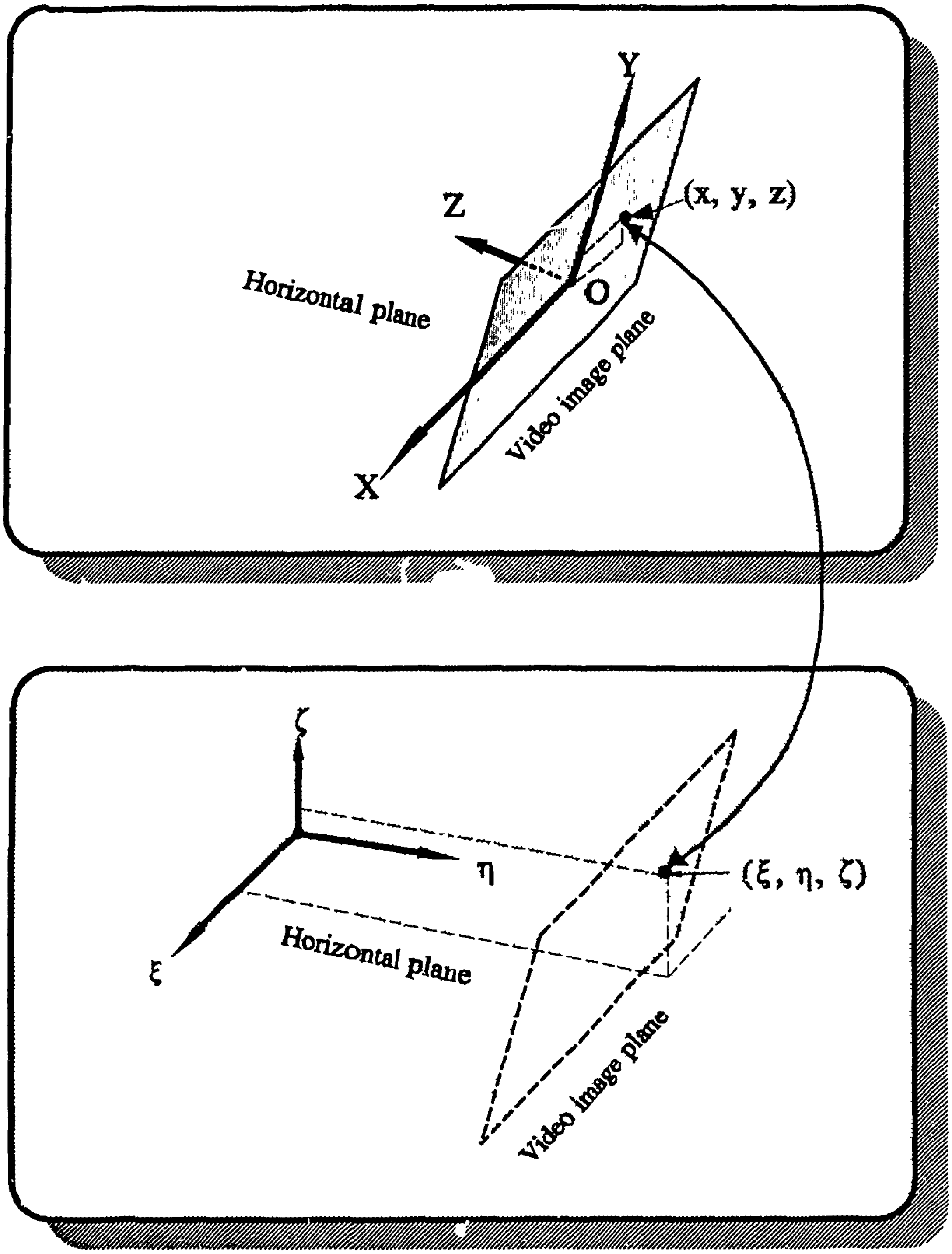


Fig. 2.11 Coordinates transformation

$$| \boldsymbol{T} | = \begin{vmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{vmatrix} \neq 0 \qquad (2.18)$$

The transformation $\boldsymbol{T}$ is a combination of translation, scaling, and rotatiºn and can be represented as

$$\boldsymbol{T} = \boldsymbol{R} * \boldsymbol{S} * \boldsymbol{E} \qquad (2.19)$$

where $\boldsymbol{E}$, $\boldsymbol{S}$, and $\boldsymbol{R}$ are transformation matrixes for the operations of translation, sca'ing, and rotation, respectively. They are discussed below.

Translation

The transformation matrix for translation is defined as $\boldsymbol{E}$, and

$$\boldsymbol{E} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (2.20)$$

The following equation performs a translation mapping as shown in Fig. 2.12.

$$\begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x-x_0 \\ y-y_0 \\ z-z_0 \end{bmatrix} \qquad (2.21)$$

Scaling

The transformation matrix for scaling is defined as $\boldsymbol{S}$, and

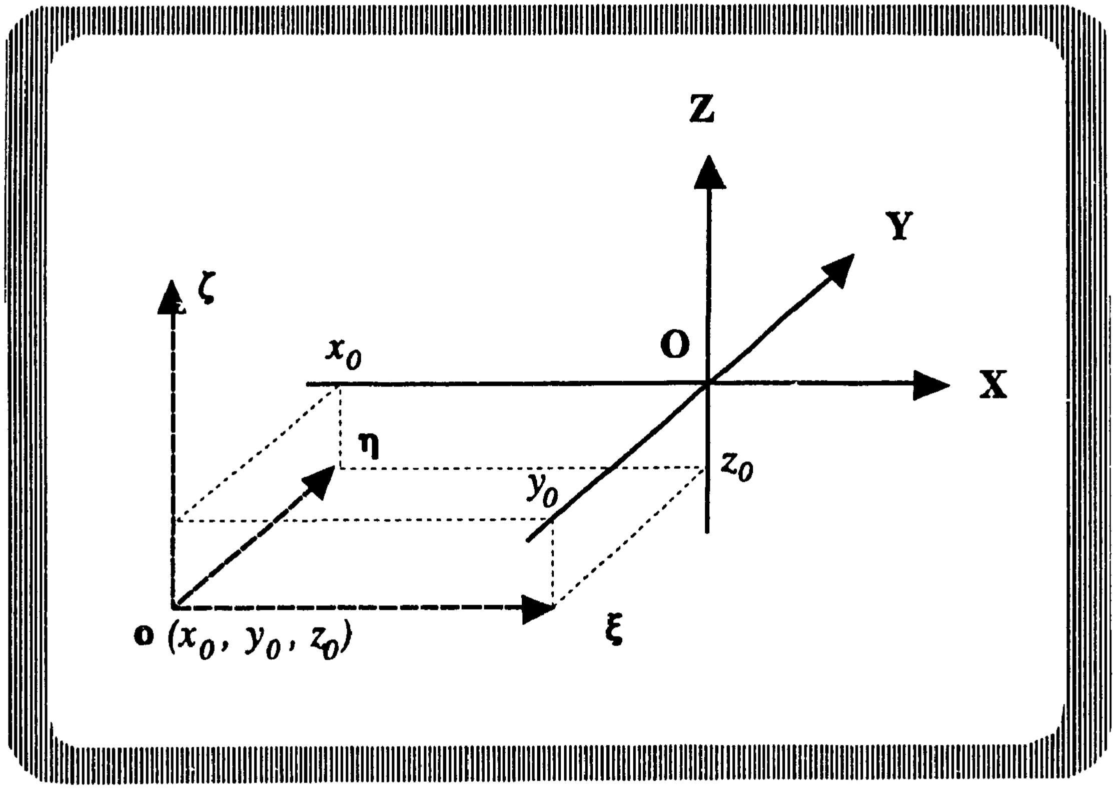


Fig. 2.12 Translation

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \tag{2.22}$$

The following equation performs a translation mapping as shown in Fig. 2.13.

$$\begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.23}$$

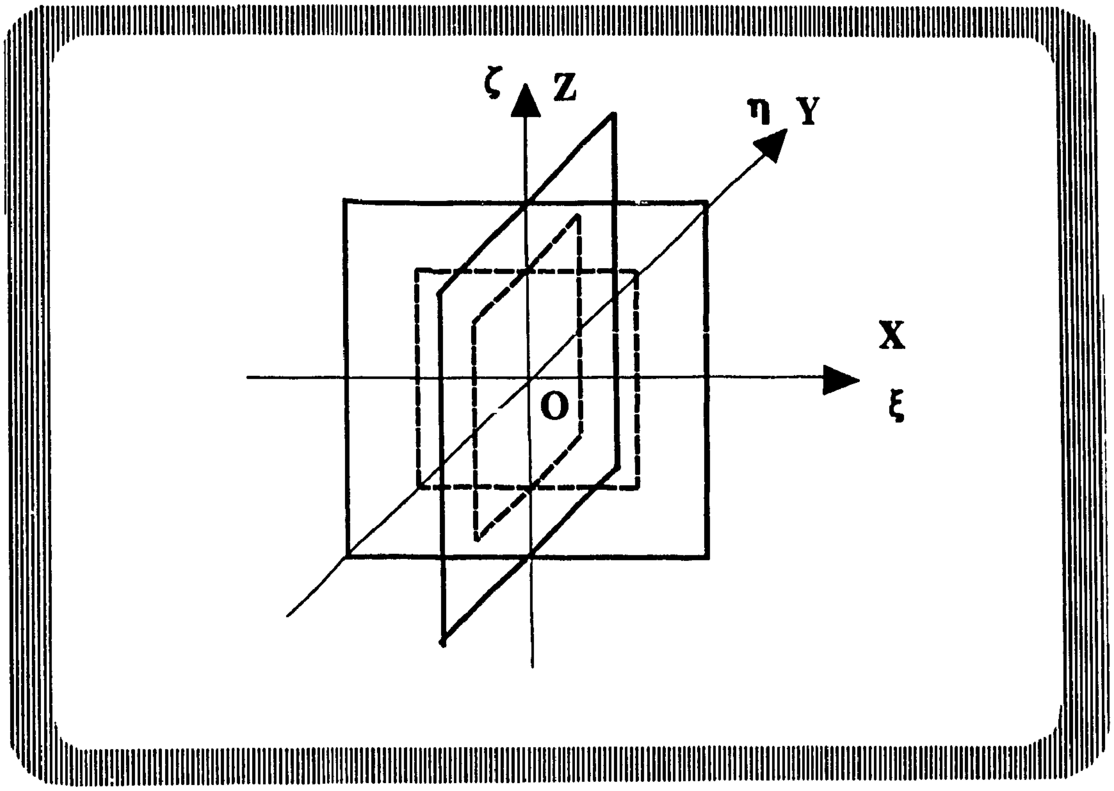


Fig. 2.13 Scaling

Rotation

Due to the camera mount tilt and camera mount pan, the transformation involves two rotation operations. Refer to Fig. 2.14, for the tilt rotation about the X axis with an angle θ (counterclockwise is positive, $-\pi/2 \le \theta \le 0$); we have

$$\begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.24}$$

where the transformation matrix for the tilt rotation is expressed as

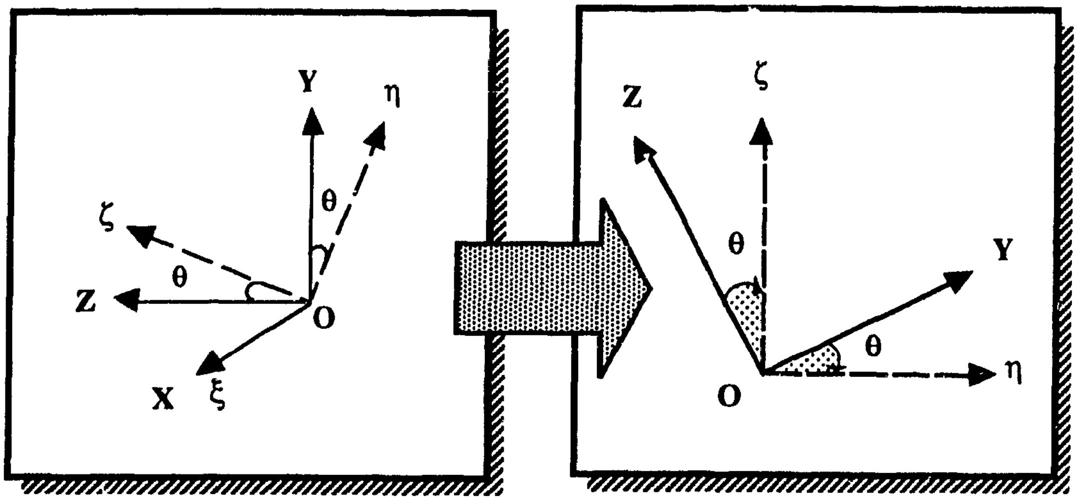


Fig. 2.14 Tilt rotation about X-axis

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{2.25}$$

Similarly, a transformation matrix can be derived for pan rotation about the Y axis with an angle α (counterclockwise is positive, $-\pi/2 \le \alpha \le \pi/2$) as (see Fig. 2.15)

$$R_y = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{2.26}$$

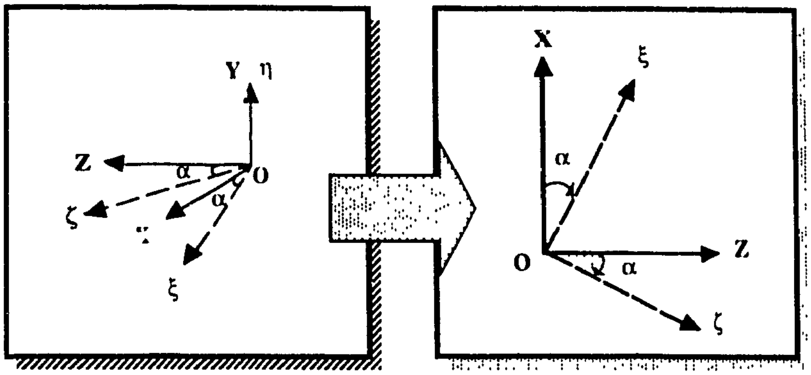


Fig. 2.15 Pan rotation about Y-axis

The product of $\boldsymbol{R}_y$ and $\boldsymbol{R}_x$ integrates the pan rotation about Z axis and tilt rotation about X axis. This product is $\boldsymbol{R}$ and computed as

$$\boldsymbol{R} = \boldsymbol{R}_y * \boldsymbol{R}_x = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{2.27}$$

$$= \begin{bmatrix} \cos\alpha & -\sin\alpha\sin\theta & \sin\alpha\cos\theta \\ 0 & \cos\theta & \sin\theta \\ -\sin\alpha & -\cos\alpha\sin\theta & \cos\alpha\cos\theta \end{bmatrix}$$

Get back to the transformation $\boldsymbol{T}$ which is the product of transformation matrixes $\boldsymbol{E}$, $\boldsymbol{S}$, and $\boldsymbol{R}$ for translation, scaling, and rotation, respectively. The transformation matrix $\boldsymbol{T}$ is computed as

$$\boldsymbol{T} = \boldsymbol{R} * \boldsymbol{S} * \boldsymbol{E}$$

$$= \begin{bmatrix} \cos\alpha & -\sin\alpha\sin\theta & \sin\alpha\cos\theta \\ 0 & \cos\theta & \sin\theta \\ -\sin\alpha & -\cos\alpha\sin\theta & \cos\alpha\cos\theta \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.28}$$

$$= \begin{bmatrix} s_x\cos\alpha & -s_y\sin\alpha\sin\theta & s_z\sin\alpha\cos\theta \\ 0 & s_y\cos\theta & s_z\sin\theta \\ -s_x\sin\alpha & -s_y\cos\alpha\sin\theta & s_z\cos\alpha\cos\theta \end{bmatrix}$$

Therefore, a general transformation of a point from the *image coordinate system* to the *global coordinate system* can be represented in the following form:

$$\begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \boldsymbol{T} * \begin{bmatrix} x-x_0 \\ y-y_0 \\ z-z_0 \end{bmatrix} \tag{2.29}$$

or

$$\begin{aligned} \xi &= xs_x\cos\alpha - ys_y\sin\alpha\sin\theta + zs_z\sin\alpha\cos\theta \\ &\quad - x_0s_x\cos\alpha + y_0s_y\sin\alpha\sin\theta - z_0s_z\sin\alpha\cos\theta \\ \eta &= ys_y\cos\theta + zs_z\sin\theta - y_0s_y\cos\theta - z_0s_z\sin\theta \\ \zeta &= -xs_x\sin\alpha - ys_y\cos\alpha\sin\theta + zs_z\cos\alpha\cos\theta \\ &\quad + x_0s_x\sin\alpha + y_0s_y\cos\alpha\sin\theta - z_0s_z\cos\alpha\cos\theta \end{aligned} \tag{2.30}$$

Note tha' the transformation of points from the *image plane* is a special case of the above transformation due to the fact that z is equal to zero. In this case, the above equation is simplified as

$$\begin{aligned}
\xi &= xs_x\cos\alpha - ys_y\sin\alpha\sin\theta - x_0s_x\cos\alpha + y_0s_y\sin\alpha\sin\theta - z_0s_z\sin\alpha\cos\theta \\
\eta &= ys_y\cos\theta - y_0s_y\cos\theta - z_0s_z\sin\theta \qquad (2.31) \\
\zeta &= -xs_x\sin\alpha - ys_y\cos\alpha\sin\theta + x_0s_x\sin\alpha + y_0s_y\cos\alpha\sin\theta - z_0s_z\cos\alpha\cos\theta
\end{aligned}$$

Substituting the above equations into Equation (2.16), the mapping of the points from the *image plane* onto the *horizontal plane* is given as

$$\xi_1 = \frac{H(xs_x\cos\alpha - ys_y\sin\alpha\sin\theta - x_0s_x\cos\alpha - y_0s_y\sin\alpha\sin\theta - z_0s_z\sin\alpha\cos\theta)}{H + xs_x\sin\alpha + ys_y\cos\alpha\sin\theta - x_0s_x\sin\alpha - y_0s_y\cos\alpha\sin\theta + z_0s_z\cos\alpha\cos\theta}$$

$$\eta_1 = \frac{H\,(\,ys_y\cos\theta - y_0s_y\cos\theta - z_0s_z\sin\theta)}{H + xs_x\sin\alpha + ys_y\cos\alpha\sin\theta - x_0s_x\sin\alpha - y_0s_y\cos\alpha\sin\theta + z_0s_z\cos\alpha\cos\theta} \qquad (2.32)$$

$$\zeta_1 = 0$$

In practice, the above equations can be further simplified by properly creating the coordinate systems. This will be discussed in the following section.

2.5.3 A Special Case of Perspective Projection

If an *image coordinate system* and the *global coordinate system* have relations as illustrated in Fig. 2.16, the general expression of the mapping of a point from the *image plane* onto the *horizontal plane* can be expressed in a simpler equation than Equation (2.32). This simpler equation is derived by the following procedure.

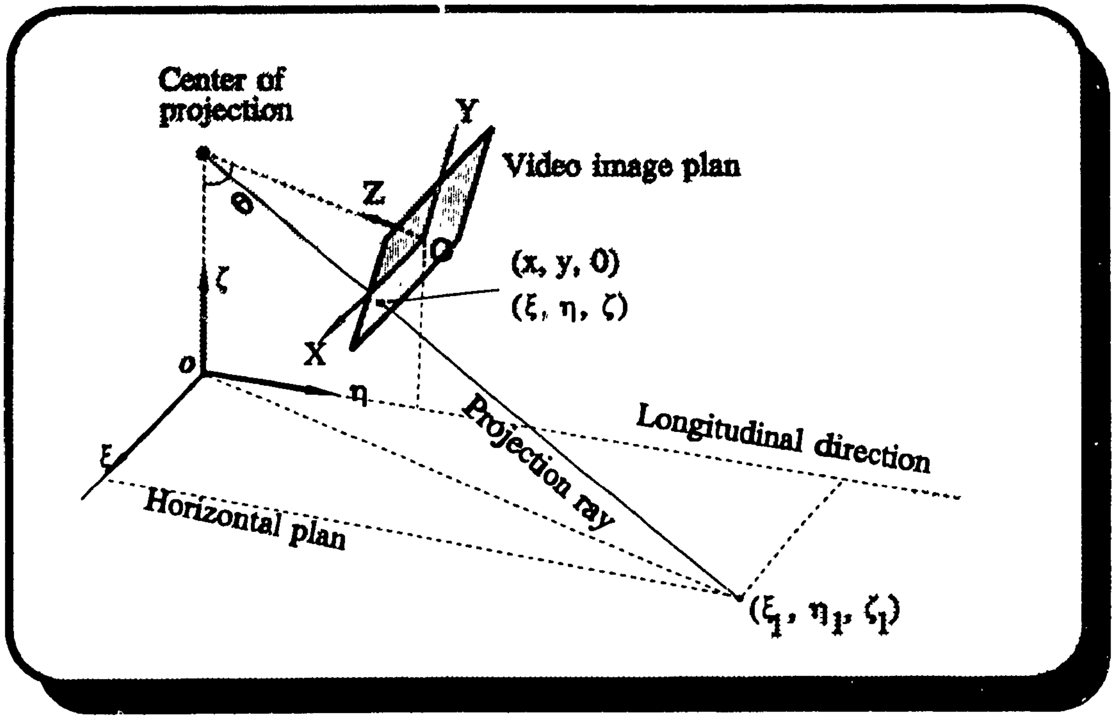


Fig. 2.16 Two coordinate systems

Rotation

As shown in Fig. 2.16, there is only the camera mount tilt but no camera mount pan. In other words, the tilt rotation about X axis is $-\pi/2 \leq \theta < 0$ while the pan rotation about Y axis is $\alpha = 0$. Substitute $\theta \neq 0$ and $\alpha = 0$ into Equation (2.32), it becomes

$$\xi_1 = \frac{H(xs_x - x_0 s_x)}{H + ys_y\sin\theta - y_0 s_y\sin\theta + z_0 s_z\cos\theta}$$

$$\eta_1 = \frac{H\ (\ ys_y\cos\theta - y_0 s_y\cos\theta - z_0 s_z\sin\theta)}{H + ys_y\sin\theta - y_0 s_y\sin\theta + z_0 s_z\cos\theta} \qquad (2.33)$$

$$\zeta_1 = 0$$

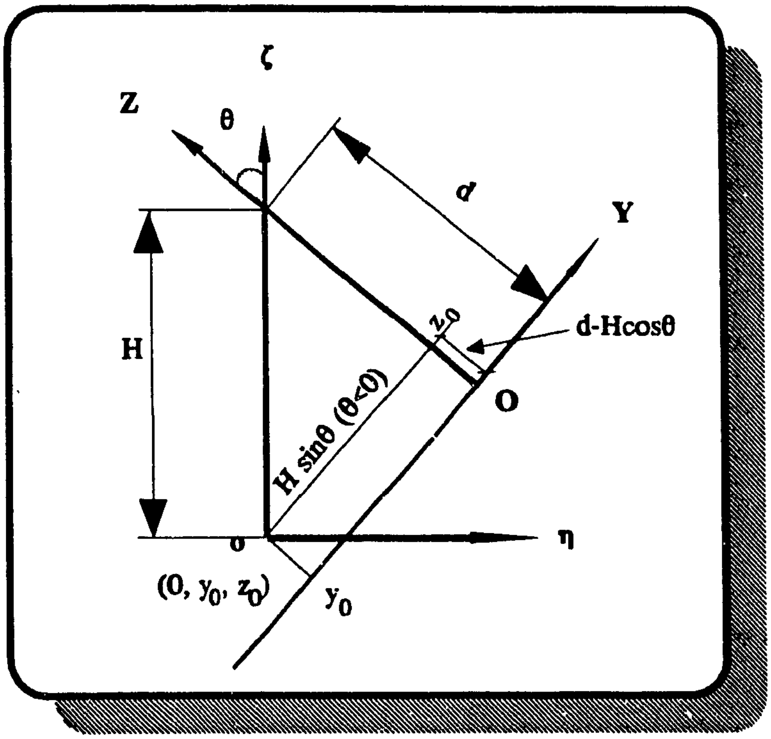


Fig. 2.17 Translation

Translation

As illustrated in Fig. 2.17, the original of the *global coordinate system* is expressed as (x_0, y_0, z_0) in the *image coordinate system*, and these coordinates can be derived intuitively. By inspection, the answer involves only similar triangles. Therefore, x_0, y_0, and z_0 are derived as

$$x_0 = 0$$
$$y_0 = \frac{H\ sin\theta}{s_y} \tag{2.34}$$
$$z_0 = \frac{d\ -\ H\ cos\theta}{s_z}$$

where s_y and s_z are scaling coefficients. *H* and *d* are distances as defined in Fig. 2.17.

Substituting the above equations for x_0, y_0, and z_0 respectively in Equation (2.33), it gives

$$\xi_1 = \frac{H\,x\,s_x}{ys_y sin\theta + dcos\theta}$$

$$\eta_1 = \frac{H(ys_y cos\theta - dsin\theta)}{ys_y sin\theta + dcos\theta} \tag{2.35}$$

$$\zeta_1 = 0$$

Scaling

Distance in the image plane (XOY plane) is measured in the number of pixels while the distance in the horizontal plane ($\xi o \eta$ plane) is measured in meters. The scaling parameters s_x and s_y can be determined as follows.

$$s_x = \frac{2tan\frac{\phi_x}{2}}{N_x}d = k_x d$$

$$\tag{2.36}$$

$$s_y = \frac{2tan\frac{\phi_y}{2}}{N_y}d = k_y d$$

where, as illustrated in Fig. 2.18, ϕ_x and ϕ_y are the view angles of the camera respectively in X and Y direction. N_x and N_y are the number of pixels in a row and a column respectively.

Substitute above equations for s_x, and s_y respectively in Equation (2.35), then it becomes

$$\xi_1 = \frac{H\,x\,k_x}{yk_y sin\theta + cos\theta}$$

$$\eta_1 = \frac{H(yk_y cos\theta - sin\theta)}{yk_y sin\theta + cos\theta} \qquad (2.37)$$

$$\zeta_1 = 0$$

where, as illustrated in Fig. 2.17 and Fig. 2.18, H is thr distance between the video camera and *horizontal plane*; θ denotes the angle ot the camera mount tilt, ($-\pi/2 < \theta < 0$); ϕ_x, ϕ_y denote the view angles of the camera in X and Y direction respectively, ($0 < \phi_x, \phi_y < 2\pi$); and k_x, k_y are coefficients that are given as the following equations:

$$k_x = \frac{2tan\frac{\phi_x}{2}}{N_x} \qquad k_y = \frac{2tan\frac{\phi_y}{2}}{N_y} \qquad (2.38)$$

Equation (2.37) can be easily used for mapping a point in the *image plane* onto the *horizontal plane*.

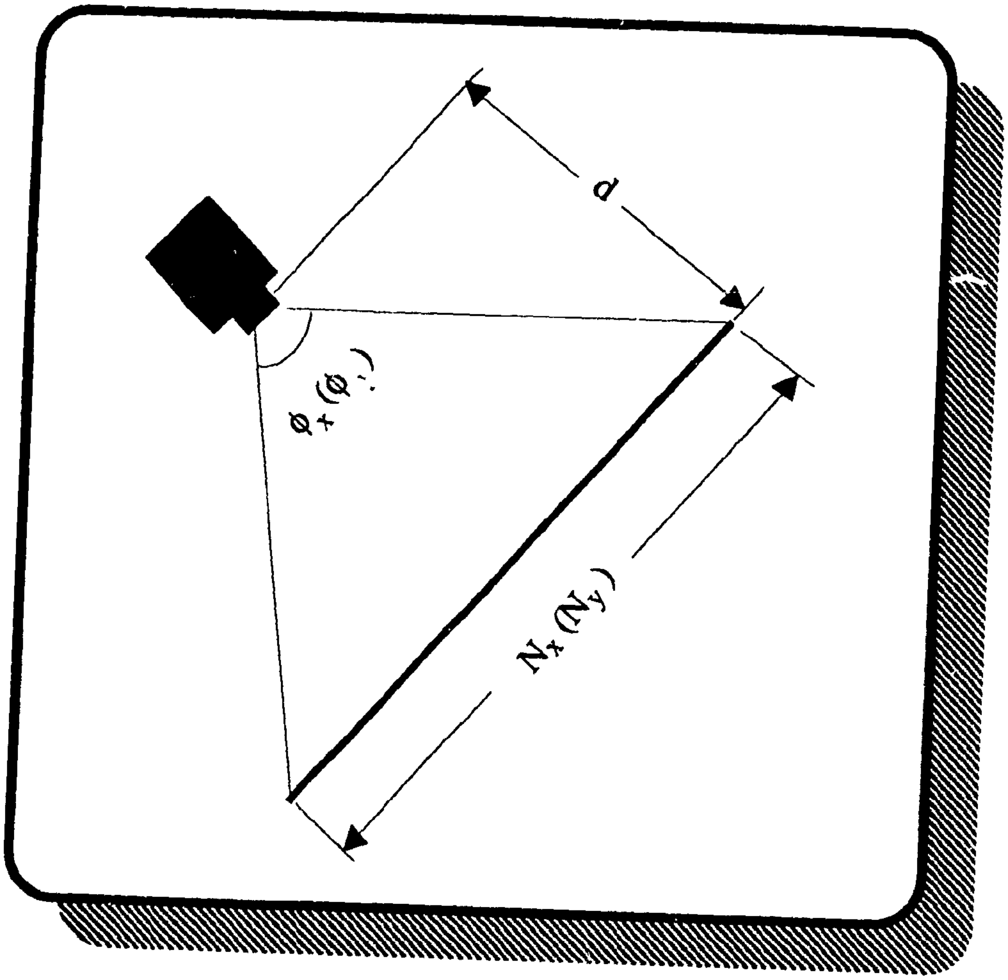


Fig. 2.18 View angles of the camera

CHAPTER 3

VEHICLE ALGORITHM

3.1 INTRODUCTION

Vehicle detection is fundamentally important for traffic planning, maintenance and traffic control. Unfortunately, vehicle detection is the weakest link in traffic surveillance and control.

Published literature was reviewed relating to topics on vehicle detection. The result indicated that in the last two decades a variety of new techniques was studied to overcome the drawbacks of the commonly used inductive loop detectors. For instance, Braston (1975) tried to use a time-lapse photography method to measure speeds and headway. Ashworth (1976) and Polus (1978) proposed a video recording system to obtain data on speeds, occupancies, and volumes. Also, Ashworth (1976) tried to use an ultrasonic detector to measure vehicle occupancy. These methods didn't use computer vision techniques. The data were still collected by human observers. Almost at the same time, image processing techniques were introduced for traffic data collection and a variety of image processing systems have been developed since 1970's. The majority of the systems were reviewed by Dickinson (1984a, 1984b, 1989), Inigo (1985, 1989) and Gilbert (1991).

A survey of existing video image processing systems shows that the majority are derived from monochrome images. This is because mo^n^ochrome video equipment is inexpensive and already widely used for traffic monitoring purposes. The storage capacity required by a monochrome image is only about one-third of that needed by a color image. However, monochrome image limits accuracy due to its low resolution. For example, shadows of vehicles usually make detection false alarms (Inigo 1989; Schlutsmeyer 1982), because it is difficult to distinguish vehicle shadows from some vehicles due to the similarity of their grey levels.

Color image analysis, once a costly process, now offers an effective way to overcome such problems. With the falling prices of color video equipment and computer memory, the application of color image analysis for traffic monitoring has become a realistic possibility.

In this chapter, a new method, called vehicle algorithm, has been developed to process color images in order to reduce detection errors caused by shadow or similarity of vehicle color and pavement color. To verify the accuracy of the detection, the algorithm have two kinds of real-time output: vehicle speed and volume. In section 3.2, a method for reducing the data is presented. A vehicle detection method is presented in section 3.3. Individual vehicle speeds and traffic volumes are derived by using the information extracted from video images. This procedure is presented in section 3.4. In section 3.5, experimental results and analysis are discussed.

3.2 SET UP DETECTION LINE

The practical problem faced by the real-time algorithm of image processing is the large number of data to be processed. For instance, a grey-level picture with resolution of 512 x 512 pixels contains 262,144 data while the color one will contain three times more data than this number (i.e., 786,432). It was observed that the data could be reduced because it was only necessary to test for changes (i.e., vehicle absence / presence) at selected spatial regions, or windows, in the scene in order to count all the vehicles within that scene and estimate their velocities.

Instead of processing all the pixels in a frame, the vehicle algorithm only processes a number of pixels chosen from each frame to make the process suitable for real-time purposes. These pixels form a number of detection lines. Each line, used for vehicle detection in one lane, is one pixel wide. The length and location of the line can be arbitrarily determined by users on a TV screen depending upon the requirements. Figure 3.1 shows the examples of the detection lines set up horizontally and vertically. Detection lines work like sensors embedded under the pavement for recording vehicle presence and passage. When these particular detection lines are overlaid by vehicles, pixels in those lines manifest the colors of those vehicles respectively By means of detecting the color of these pixels, the algorithm can determine the presence or absence of vehicles.

Suppose the position of a pixel is described by (x,y,t) in an image sequence, where x and y are coordinates in the *image coordinate system* defining the position of the pixel in a frame; t defines the position of the frame in a sequence of consecutive frames. A pixel line that contains n adjacent pixels p_1, p_2, ..., p_n is expressed as a matrix, and

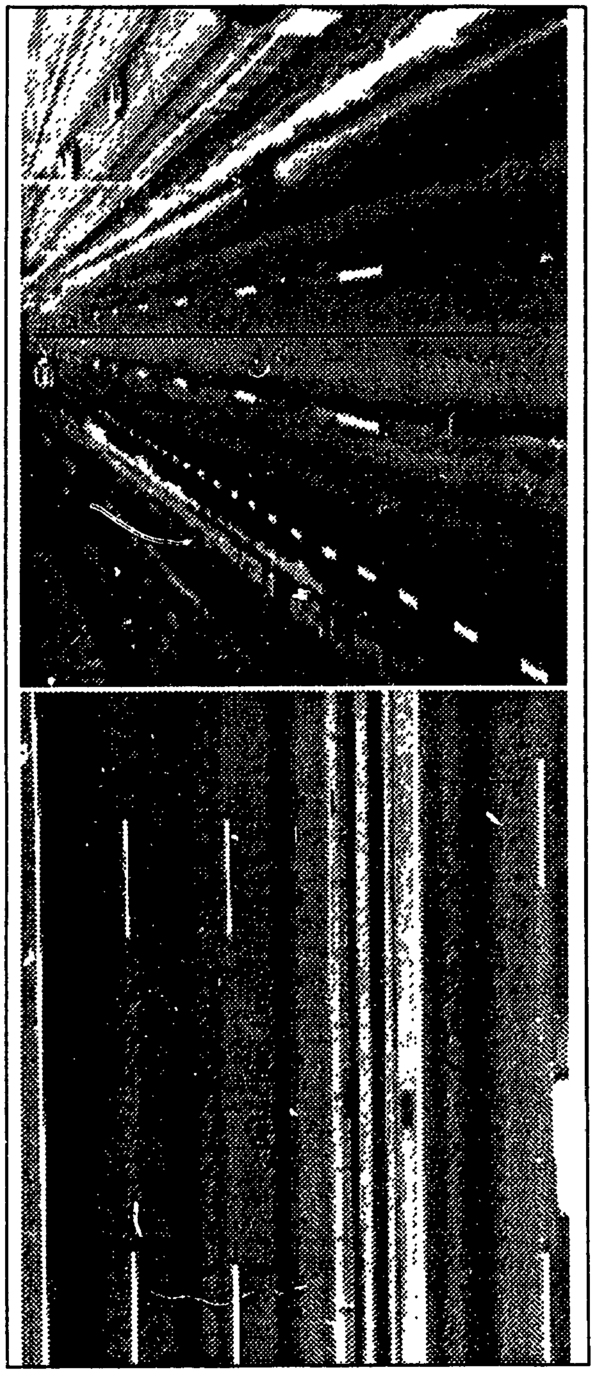


Fig. 3.1 Detection lines

$$\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} (x_1,\ y_1,\ t) \\ (x_2,\ y_2,\ t) \\ \vdots \\ (x_n,\ y_n,\ t) \end{bmatrix} \qquad (3.1)$$

Define the value of the pixel at (x,y,t) as a vector $P(x,y,t)$, and

$$P(x,y,t) = (\ R(x,y,t),\ G(x,y,t),\ B(x,y,t)\ )$$

where $R(x,y,t)$, $G(x,y,t)$ and $B(x,y,t)$ denote the ***tristimulus values*** of the pixel. The values of the pixel line are expressed as

$$\begin{bmatrix} P(p_1) \\ P(p_2) \\ \vdots \\ P(p_n) \end{bmatrix} = \begin{bmatrix} R(p_1) & G(p_1) & B(p_1) \\ R(p_2) & G(p_2) & B(p_2) \\ & \vdots & \\ R(p_n) & G(p_n) & B(p_n) \end{bmatrix} \qquad (3.2)$$

3.3 VEHICLE DETECTION

3.3.1 Extraction of Object

Information carried by pixels is classified into three categories: *object*, *shadow*, and *pavement surface*. *Object* in this context refers to something whose color is significantly different from that of pavement surface and shadows in the detection area. In order to determine in which category a pixel belongs, two *color detection functions* f_1 and f_2 are introduced according to Equation (2.7) of Chapter Two. The function f_1 is for detection of pavement surface without shadow, and function f_2 is for detection of shadows. The objects in an image can be extracted by the following procedure.

(1) Determine the color of the pavement surface without shadow;

(2) Test pixel p_i, $p_i \subseteq \{p_1, p_2, ..., p_n\}$, by f_1:

if $f_1(p_i) \geq 2/3$, then the pixel is regarded as *pavement surface*; otherwise,

if $f_1(p_i) < 2/3$, then the pixel is regarded as *object* or *shadow*;

(3) Determine the color of the shadow on the pavement;

(4) Test the pixel by f_2:

if $f_2(p_i) \geq 2/3$ the pixel is *shadow*; otherwise,

if $f_2(p_i) < 2/3$ the pixel is *object*.

Since only the *object* will be used for further analysis, both the *pavement surface* and *shadow* are regarded as *background* and neglected. Therefore, the processed pixels are classified into either *object* or *background* category. Instead of the original vector value of a pixel $P(p_i)$, a new value $P_B(p_i)$ will be used to indicate in which category the pixel belongs. The value of $P_B(p_i)$ is either 0 or 1. A pixel with a value of 0 is interpreted as a white point representing the background, and a pixel with a value of 1 is interpreted as a black point representing the object. Referring to the above decision rules, the value of $P_B(p_i)$ is determined as follows:

$$P_B(p_i) = \begin{cases} 1, & \textit{if and only if } [f_1(p_i) < \frac{2}{3}] \cap [f_2(p_i) < \frac{2}{3}] \\ 0, & \textit{if } [f_1(p_i) \geq \frac{2}{3}] \cup [f_2(p_i) \geq \frac{2}{3}] \end{cases} \tag{3.3}$$

The image with pixels represented by values of P_B is called *binary image* in which every pixel has one of only two options, either *object* or *background*, for the purposes of the analysis at hand.

3.3.2 Vehicle Detection

The objects extracted from an image are further classified into two categories: *vehicle* and *noise*. A *noise* refers to any object on the pavement surface other than a vehicle. Noise can be distinguished from vehicle by measuring the size of the object. The size of an object in a binary image is defined as the number of pixels in a cluster in which all the pixel values are equal to 1. The procedure for differentiating vehicles from noises is as follows.

1. Detection of object boundaries in the line image

The boundaries of the objects in a line image can be obtained through the following steps:

Step 1. Locate left boundary of the first object in a line image. Suppose the image includes n pixels, $p_1, p_2, ..., p_n$. The values of these pixels are $P_B(p_1), P_B(p_2), ...,$ $P_B(p_n)$.

If a pixel $p_{a_1} \in [p_1, p_2, ..., p_n]$ satisfies the following conditions, then it is the left boundary of the first object.

$$P_B(\ p_{a_1}\ ) = 1 \ \cap \ \sum_{p_i = p_1}^{p_{a_1}-1} P_B(\ p_i\ ) = 0 \qquad (3.4)$$

Step 2. Locate the right boundary of the first object. If a pixel $p_{b_1} \in [p_{a_1}, p_{a_1}+1,$

..., p_i, ..., P_n] satisfies all of the following conditions, then it is another boundary of the first object.

$$\prod_{p_i=p_{a_1}}^{p_{b_1}} P_B(\ p_i\ ) = 1 \ \cap \ P_B(\ p_{b_1}+1\ ) = 0 \tag{3.5}$$

Step 3. Locate the left boundaries of the other objects. Let a_k denote the left boundary of the kth object, where $a_k = a_2, a_3, \cdots\cdots$. For example, a_2 denotes the left boundary of the second object, and a_3 the third one. A pixel at a_k should make all of the following equations true

$$p_{a_k} \in [p_{b_{k-1}}+1, p_n] \ \cap \ P_B(\ p_{a_k}\ ) = 1 \ \cap \ \sum_{i=b_{k-1}+1}^{a_k-1} P_B(\ p_i\ ) = 0 \tag{3.6}$$

Step 4. Locate the right boundaries of the other objects. Let b_k denote the right boundary of the kth object, where $b_k = b_2, b_3, \cdots\cdots$. For example, b_2 denotes the right boundary of the second object and b_3 the third etc. After the left boundary of an object is found at pixel a_k, its right boundary must exist at b_k that makes all of the following equations true

$$p_{b_k} \in [p_{a_k}, p_n] \ \cap \ P_B(\ p_{b_k}+1\ ) = 0 \ \cap \ \prod_{i=a_k}^{b_k} P_B(\ p_i\ ) = 1 \tag{3.7}$$

2. Object measurement

The real size of an object can be obtained if two of its boundaries are found in the image. Let L denote the real length of an object in term of meters, and

$$L = \sqrt{L_x^2 + L_y^2} \tag{3.8}$$

where L_x, L_y = lengths of the projections of L in horizontal and vertical directions, respectively. Referring to Fig. 3.2, L_x and L_y can be obtained according to the PP model (Equation (2.37)) as follows.

(1) Determine L_x:

$$\begin{aligned} L_x &= \left| \xi(x_b, y_b) - \xi(x_a, y_a) \right| \\ &= \left| \frac{H\, k_x\, x_b}{y_b k_y \sin\theta + \cos\theta} - \frac{H\, k_x\, x_a}{y_a k_y \sin\theta + \cos\theta} \right| \end{aligned} \tag{3.9}$$

where

(x_a, y_a) = coordinates of the left boundary of the object;

(x_b, y_b) = coordinates of the right boundary of the object;

H = distance between the video camera and the *horizontal plane*;

θ = angle of the camera mount tilt ($-\pi/2 < \theta < 0$);

k_x, k_y = coefficients that are determined by Equation (2.38):

$$k_x = \frac{2tan\frac{\phi_x}{2}}{N_x} \qquad k_y = \frac{2tan\frac{\phi_y}{2}}{N_y} \tag{3.10}$$

where

ϕ_x , ϕ_y = view angles of the camera in X and Y directions respectively ($0 < \phi_x$, ϕ_y

$< 2\pi$);

N_x, N_y = total number of pixels respectively in a row and a column of the frame.

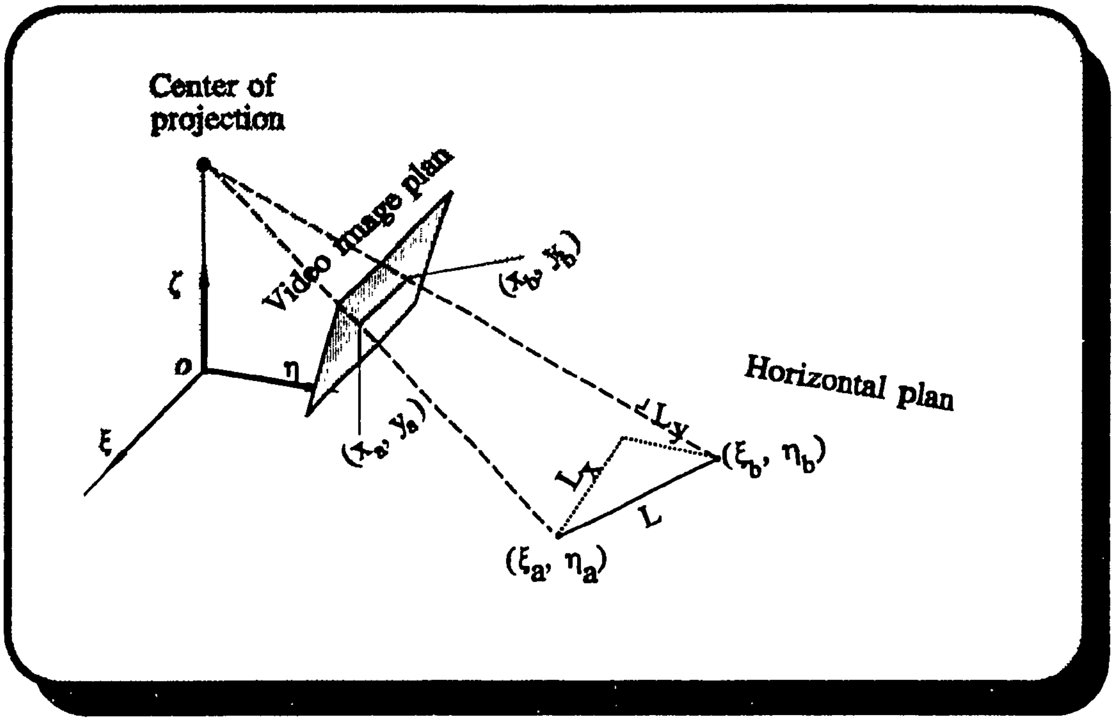


Fig. 3.2 Length projection

(2) Determine L_v: similarly, L_v is calculated as

$$L_v = \left| \eta(x_b, y_b) - \eta(x_a, y_a) \right|$$

$$= \left| \frac{H(y_b k_y \cos\theta - \sin\theta)}{y_b k_y \sin\theta + \cos\theta} - \frac{H(y_a k_y \cos\theta - \sin\theta)}{y_a k_y \sin\theta + \cos\theta} \right| \qquad (3.11)$$

3. Removal of noise object

Assume that a detection line is carefully placed on an area of asphalt pavement containing as few surface color variations as possible. This reduces the size of the noises to a minimum. In other words, it is assumed that the size of ***noise*** is smaller than that of a vehicle. A predetermined threshold, β, gives the maximum acceptable size of noises in a binary image. If the size of an object is larger than threshold β, the object is regarded as a vehicle; otherwise, it is regarded as a noise and removed from the image. This operation is expressed as follows.

a. Determine the number of objects in image t;

b. Compare the size of each object L with the threshold, β.

c. If $L < \beta$, then it is determined that the object is a noise. The pixels between two boundaries are set the value of 0.

3.4 MEASUREMENT OF SPEED AND VOLUME

In order to determine the vehicle movement between two consecutive images, *a reference point* should be determined. If the detection line is drawn horizontally in the images, then either a left or right boundary of a vehicle can be defined as a *reference point.* But if the detection line is drawn vertically, then only the bottom boundary could be used as a reference point shown in Fig. 3.3. The location change of a reference point in two images is the distance that the vehicle moved during a time interval.

Let m denote the number of consecutive images in which a *reference point* can be observed. As exemplified in Fig. 3.4, if the left boundary of the vehicle is used as a

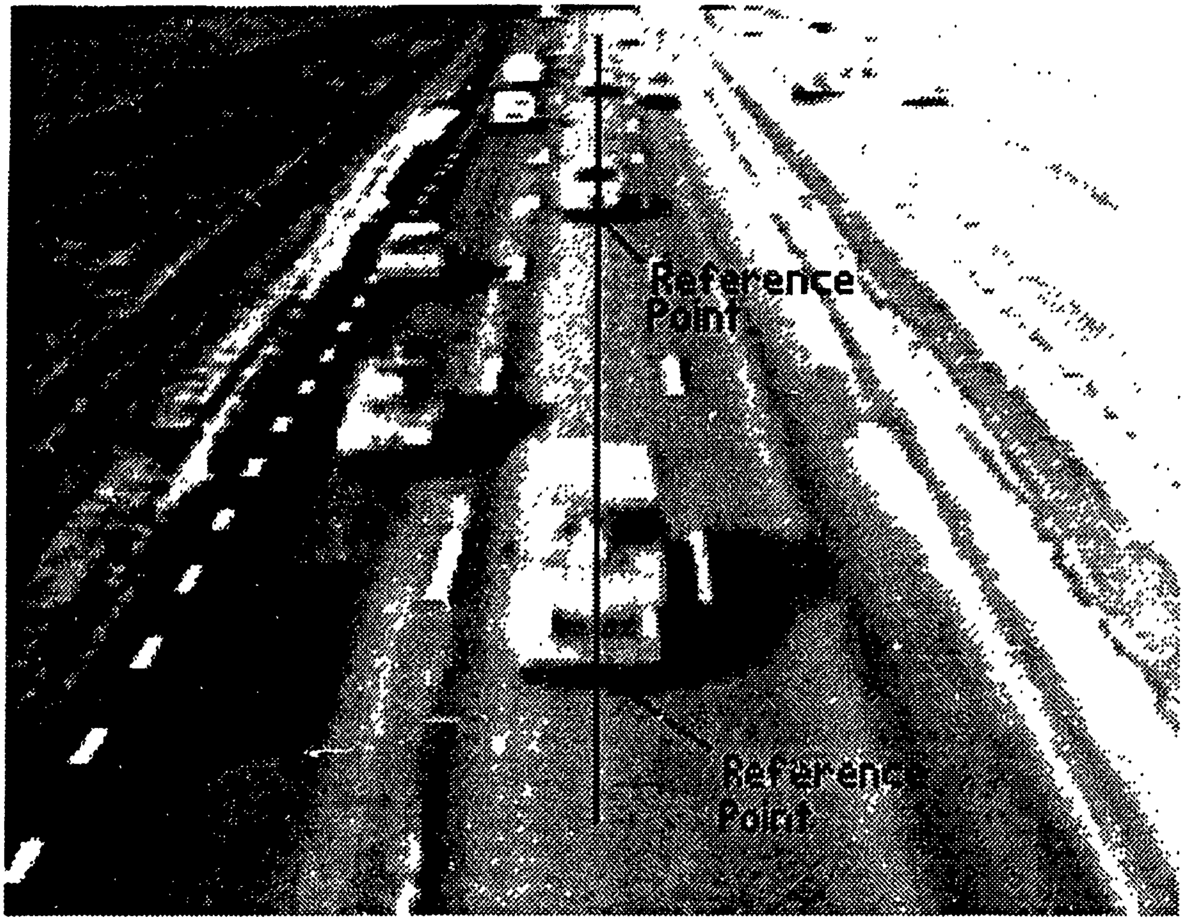


Fig. 3.3 Reference point

reference point, then the value of m is 6 because the reference point was observed in six successive images. Normally, the value of m is determined depending upon the average speed of vehicles: the slower the speed, the greater the value of m and the faster the speed, the smaller the value of m.

3.4.1 Speed Measurement

Denote p_t as a reference point of a vehicle in image t, and $p_t = (x_t, y_t)$. It is obvious that the variation of the positions of a reference point in two successive images

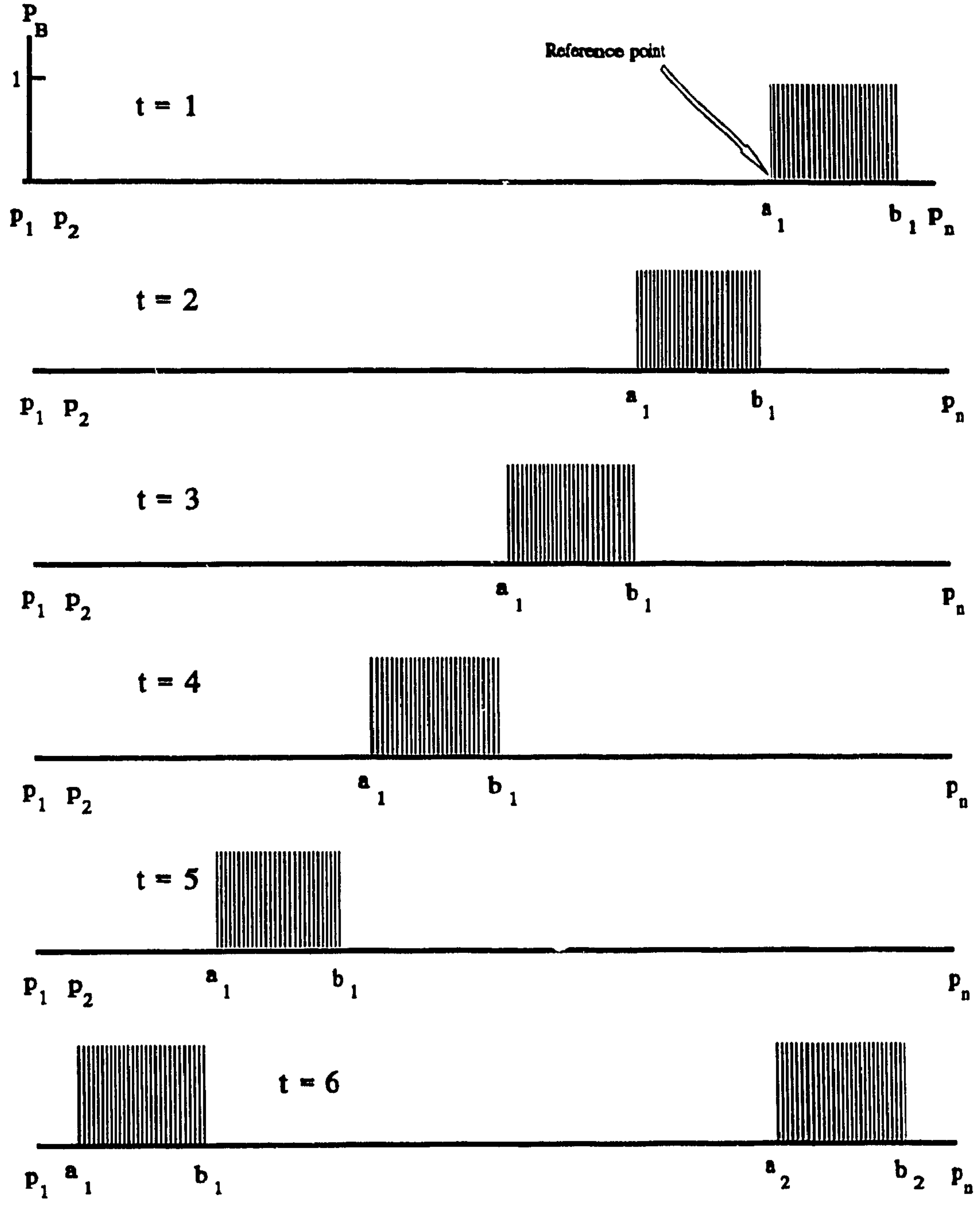


Fig. 3.4 Sequence of binary images

is the distance that a vehicle passed during the time interval between image *t* and image *t+1*. If the positions of a reference point in two consecutive images are measured as p_t and p_{t+1} respectively in the ***image plane***, then the real distance that point moves in the horizontal plane can be calculated as

$$D_t = \sqrt{\Delta\xi_t^2 + \Delta\eta_t^2} \tag{3.12}$$

where

D_t = real distance between point p_t and p_{t+1};

$\Delta\xi_t$, $\Delta\eta_t$ = distances in directions ξ and η, respectively.

Similar to Equations (3.9) and (3.11), $\Delta\xi_t$ *and* $\Delta\eta_t$ can be computed below.

$$\begin{aligned} \Delta\xi_t &= \left| \xi(x_{t+1}, y_{t+1}) - \xi(x_t, y_t) \right| \\ &= \left| \frac{H\ k_x\ x_{t+1}}{y_{t+1}k_y\sin\theta + \cos\theta} - \frac{H\ k_x\ x_t}{y_t k_y\sin\theta + \cos\theta} \right| \end{aligned} \tag{3.13}$$

and

$$\begin{aligned} \Delta\eta_t &= \left| \eta(x_{t+1}, y_{t+1}) - \eta(x_t, y_t) \right| \\ &= \left| \frac{H(y_{t+1}k_y\cos\theta - \sin\theta)}{y_{t+1}k_y\sin\theta + \cos\theta} - \frac{H(y_t k_y\cos\theta - \sin\theta)}{y_t k_y\sin\theta + \cos\theta} \right| \end{aligned} \tag{3.14}$$

Suppose the time interval between image t and image $t+1$ (t = *1, 2, 3, ..., m-1*) is T_t seconds, then the average speed of a vehicle passing through the detection line is calculated as

$$Speed\ (km/h) = \frac{3.60 \sum_{i=1}^{m-1} D_i}{\sum_{i=1}^{m-1} T_i} \tag{3.15}$$

where the average speed of a vehicle is measured during time interval T_m-T_1, and 3.60 is a coefficient that translates the speed unit from meters per second to kilometres per hour.

3.4.2 Traffic Volume Measurement

Traffic volume is defined as the number of vehicles passing a given point during a specified period of time, or the number of vehicles that pass over a given section of a lane or a roadway during a specified period of time. If N vehicles are counted in the image sequence of m frames, the traffic volume is computed as

$$Volume\ (veh/h) = \frac{3600\ N}{\sum_{i=1}^{m-1} T_i} \tag{3.16}$$

3.5. EXPERIMENTAL RESULTS

The developed algorithm was tested based on videotaped data recorded in real traffic situations under a variety of environmental conditions. The main purpose in collecting these data was to capture as many different conditions as possible in order to achieve high performance from the algorithms. These videotaped data were recorded in different months in 1990 and 1991 in Montreal, Canada. Data covered a number of weather conditions including clear, cloudy, rain and snow.

In order to determine the accuracy of the detection, vehicle speeds measured by the algorithm were verified because the accuracy of the speed measurement is very sensitive to the accuracy of vehicle boundary detection. An accurate speed measurement must be a result of accurate boundary detection. To find approximate accuracy of the algorithm in speed measurement, an experiment took place in May 1991 on Decarie Expressway in Montreal, involving two cars. The drivers were asked to pass through the detection area repeatedly and record their speeds, as indicated on their respective speedometers. Traffic flow during the experimental period was recorded using a video recorder. The speeds of those two cars were measured by the algorithm in the laboratory using the developed software. In the experiment, error of the algorithm was defined as the deviation between the actual speed and the one generated by the algorithm. With a sampling of 76 readings, the goodness of fit test indicated that errors fell within a normal distribution with an estimated mean of 1.2 km/h and an estimated standard deviation of 4.9 km/h at the level of significance 0.05. According to the characteristic of the normal distribution, 95.4% of the samples fall in the range [x±2s]. Therefore, the errors should belong in ±9.8 km/h with 95.4% probability. Since the speed obtained from speedometer is known to have a small error, in the order of ±3%, the above results may have to be slightly adjusted to indicate the actual accuracy of the algorithm. In addition, to get the actual accuracy, a large number of speed data is required to be obtained from other more accurate ways such as using radar guns. Error analysis indicated that errors in vehicle boundary detection can be easily reduced by increasing the resolution of the image. The errors in speed measure;.ıent is reduced linearly with the accuracy increase in boundary detection. For example, the error can be decreased to ±4.5 km/h by merely changing the resolution of the detection line from 25€ to ɔ12 pixels.

The above accuracy of ±9.8 km/h was obtained when the speed of the traffic flow was between 80-110 km/h. Other experiments were alsᴐ performed in orJer to test the accuıacy in other speed ranges. The results show that the accuracy of the algorithm becomes higher when the vehicle speed is lower than 80 km/h. But when the vehicle speed exceeds 110 km/h, the accuracy of the algorithm is lower. This is due to the limitation of the hardware speed. When the speed of a vehicle is high, its front and rear boundaries become hᴊzy in the images as exemplified in Fig. 3.5. With the help of high-speed image digitizing devices, this problem could be solved and the accuracy improved.

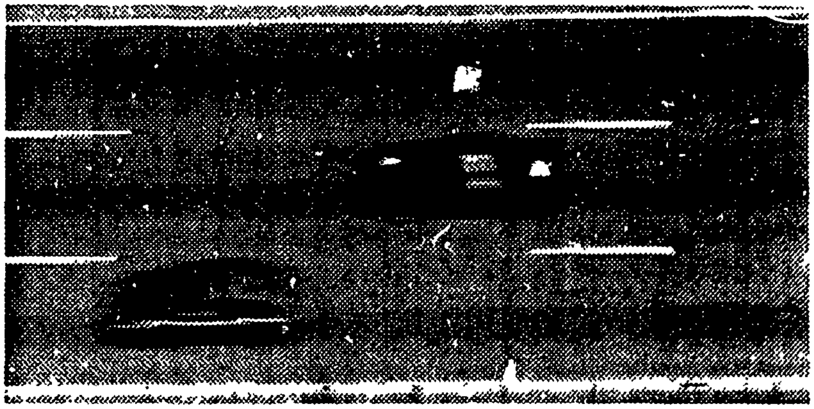


Fig. 3.5 Hazy boundaries of a vehicle

In conclusion, color images can be used in real-time video traffic detection. This method is able to reduce detection errors caused by shadow. It can be considered as an improvement to existing video detection systems. The algorithms develop in this thesis is incapable of dealing with vehicle occlusions and its accuracy in speed measurement requires further verification.

CHAPTER 4

SIGNAL ALGORITHM

4.1 INTRODUCTION

Besides speed and volume, one of the other factors which affects the capacity of intersections is the presence of turning vehicles. Traffic problems such as traffic congestion, fuel consumption, air pollution, and noise pollution usually occur at intersections, particularly when heavy left-turning traffic is present.

The left-turning volume is important in the study of any traffic system because it significantly affects vehicle speeds and intersection delays. If the data relative to turning demands are available, the traffic-responsive control systems can become more effective in reducing traffic delay and increasing intersection capacity. Unfortunately, present detection systems are incapable of detecting turning vehicles before they enter the intersections. This is because these detectors perform a "blind" type of detection, i.e. only the presence or absence of a vehicle over the sensors can be detected.

To find a vehicle that intends to turn before its turning movement, computer vision techniques are used to detect the situation ot the vehicle's turn indicator. The literature review shows that little work has been done in this area. Lu et al (1988) proposed an algorithm for detection of left-turning vehicles at intersections during the daytime. On the basis of assuming that the signal lights of left-turning vehicles will likely be the brightest spots in the pictures, Lu's algorithm tried to search for these signal lights in order to detect the turning movements. In other words, the algorithm utilized the image analysis techniques to extract the turning lights from pictures. The accuracy of the algorithm was about 80% which indicated that improvements were needed. However, it is extremely difficult for the algorithm to attain greater accuracy because of the following difficulties.

1. The algorithm distinguished reflections from signal lights by examining the size of bright clusters in the images. When the sizes and shapes of the reflections were similar to those of the signal lights, the algorithm was incapable of separating them.

2. Variation of the size of signal lights was another challenge to the algorithm. In Lu's algorithm, the signal lights were extracted using a simple threshold. This method made the accurate recognition of signal lights difficult because their sizes vary widely due to the different types of vehicles and their distance from the camera.

3. The blink of these turning lights made it even more difficult to detect them. This is because such blinking not only affects the size of the bright clusters in images but also affects the grey values of these clusters. When an image shows a blinking light at its dimmest level, the algorithm will completely fail to extract this turning light because it analyzed only one picture.

In this chapter, two models will be developed respectively for recognizing vehicle signal lights and determining turning-signal lights. These models are used to form a new algorithm, signal algorithm, to detect vehicles intending to turn at intersections. There are two significant differences between the new algorithm and that of Lu: (1) instead of processing a grey-level image, the signal algorithm will process color images in order to distinguish signal lights from bright reflections of objects; (2) instead of an isolated image, a sequence of images will be analyzed by the signal algorithm. Problems such as variation in size of the signal lights and the fact that they are blinking can be solved by the algorithm. However, the algorithm is incapable of detecting moving flashing lights or flashing lights in a distance of 30 meters away from a video sensor.

There are six sections in this chapter. Section 4.2 discusses the characteristics of vehicle turning lights. These characteristics will be used in the detection models to distinguish turning lights from other objects or other types of vehicle light (such as parking lights, headlights, etc.). The third section will develop a new model (DOS model) to segment signal lights in the images. The method to remove the noises remaining in the segmented images is presented in section 4.4. In section 4.5, another model, DOB model, will be developed to distinguish blinking lights from non-blinking ones. In section 4.6, DOS and DOB models are integrated to form the signal algorithm for detection of vehicles intending to turn. The experiment results will also be discussed in this section.

4.2 CHARACTERISTICS OF TURNING LIGHTS

The characteristics of turning lights are discussed below.

1. Signalling: Driver's manuals in Canada state that a driver "should signal his intentions continuously and for a sufficient distance before making a turn" (Driver's Handbook 1983). The purpose of signalling before making a turn, for drivers in Canada,

is the same as in other countries -- safety. Thus, turning vehicles which do not signal will be ignored in this study.

2. Blinking: When signal lights are on indicating the intention of a vehicle to turn, they blink from fifteen to twenty times every 10 seconds. This blinking is an important characteristic of turning lights, and is used in the new algorithm to make the appropriate distinction between turning and non-turning vehicles.

3. Color: The color of turning lights in front of vehicles is mostly yellow which wavelength varies from 570μm to 600 μm. This is another important characteristic of turning lights. This feature can be used to distinguish turning lights from other objects in the images.

4.3 SIGNAL LIGHT DETECTION MODEL (DOS model)

In this section, the DOS model was developed to segment signal lights (including turning and parking lights) in the digitized video images.

Color and brightness are the two most sensitive parameters for differentiating turning lights from other objects. The color of all turning lights in the front of vehicles is yellow, which is different from the colors of most other objects. Even if yellow objects appear in images, they can easily be distinguished because their brightness is different from that of illuminated turning lights. Based on these two parameters, the following DOS model is proposed:

$$\boldsymbol{P}(x,\ y,\ t) = f_c{*}f_b{*}\boldsymbol{P}_0(x,\ y,\ t) \tag{4.1}$$

where

$\boldsymbol{P}(x, y, t)$ = output value of pixel (x, y, t);

$\boldsymbol{P}_0(x, y, t)$ = input value of pixel (x, y, t);

f_c = color filter;

f_b = brightness filter.

The filters f_c and f_b are discussed as follows:

The color filter f_c masks out the colors that are not within the acceptable range of yellow. The filter f_c is expressed as

$$f_c = \begin{cases} 0, & \textit{if } [B(x,y,t) > G(x,y,t)-\varepsilon_g] \cup [B(x,y,t) > R(x,y,t)-\varepsilon_r] \cup [G(x,y,t) > R(x,y,t)]; \\ 1, & \textit{others} \end{cases} \quad (4.2)$$

where

R, G, and B = the *tristimulus values* of the pixel;

ε_r = threshold of minimum differences required between red and blue in mixture;

ε_g = threshold of minimum differences required between green and blue in mixture.

The f_c value of 1 indicates that the chromaticity value (R, G, B) produces the color of turning lights. Otherwise, the f_c value of 0 indicates that the color is not in the acceptable range. The two thresholds ε_r and ε_g mainly depend upon the colors of turning lights and satisfy conditions below:

$$\varepsilon_r, \varepsilon_g \geq 0, \text{ and } \varepsilon_r \geq \varepsilon_g$$

The values of ε_r and ε_g can be determined by outputting and analyzing the (R, G, B)

values of several turning lights when initialization.

The brightness filter f_b is used to discard pixels with a low brightness. It is expressed below.

$$f_b = \begin{cases} 0, & if\ [R(x,y,t) < \beta_r] \cap [G(x,y,t) < \beta_g]; \\ 1, & others \end{cases} \tag{4.3}$$

where

β_r = brightness threshold for red component;

β_g = brightness threshold for green component.

If either R or G is less than the corresponding threshold, the value of f_b will be 0 indicating that the pixel is not a turned-on signal light due to the low brightness. The component of blue is not necessarily considered in this filter because the color with a large value of blue component (e.g. $B > R$ or $B > G$) is not possible among the acceptable colors and can be masked out by the color filter. The values of the thresholds β_r and β_g are determined by comparing the chromaticity values (R, G, B) of signal lights at on-and-off situations.

Figure 4.1 shows a traffic image of four cars in a queue with their left-turning lights on. The image became what is shown in Fig. 4.2 after passing through the color filter f_c. Then, it was further filtered by the brightness filter. The result is shown in Fig. 4.3 where the background in Fig. 4.1 was discarded.

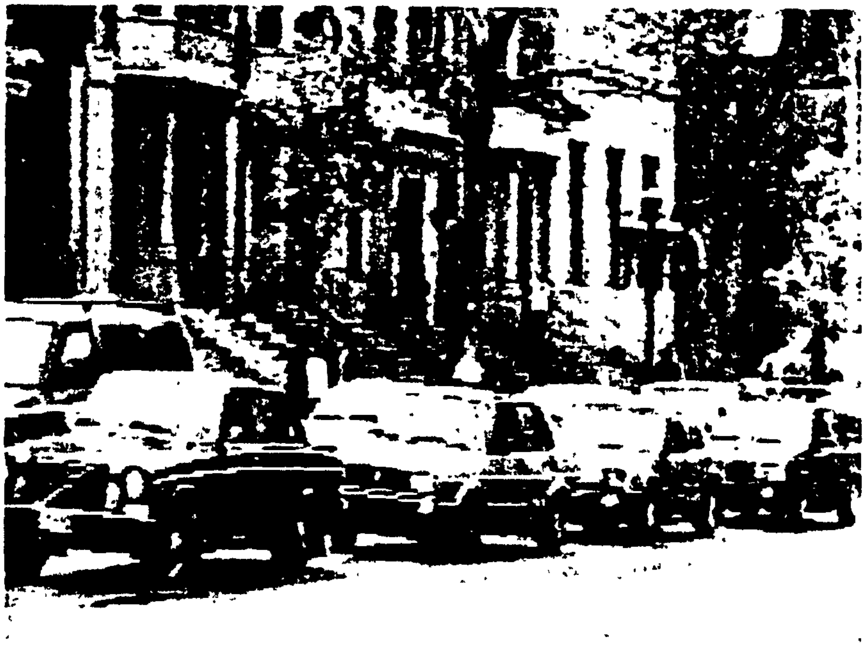


Fig. 4.1 A traffic image

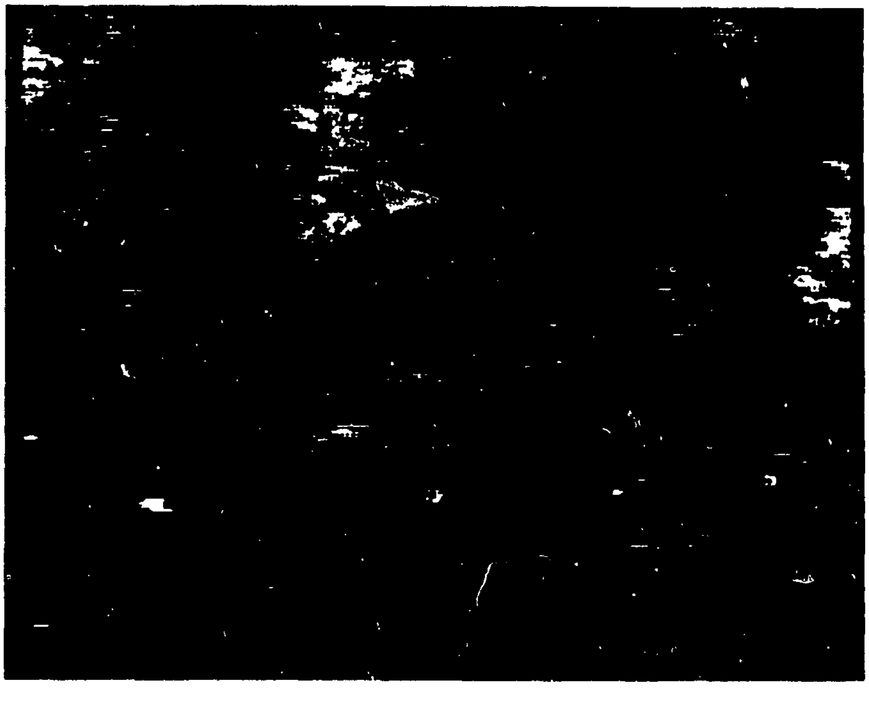


Fig. 4.2 Image after color filtering

Fig. 4.3 Image after brightness filtering

4.4 NOISE REMOVAL

Although the background was discarded after an image was processed by the DOS model, noises still remained in the image along with turning lights as shown in Fig. 4.3. The noises must be removed before the turning lights can be detected.

Usually, noises in the images are isolated pixels or short pixel lines. To detect such kinds of noise, a 5-pixel scanning window, as a special case of 3 x 3 window defined in Chapter 2, is introduced. This 5-pixel scanning window shown as Fig. 4.4 includes the scanning pixel (*x*, *y*, *t*) and its 4-neighbour pixels, say (*x*-*1*, *y*, *t*), (*x*+*1*, *y*, *t*), (*x*, *y*-*1*, *t*) and (*x*, *y*+*1*, *t*).

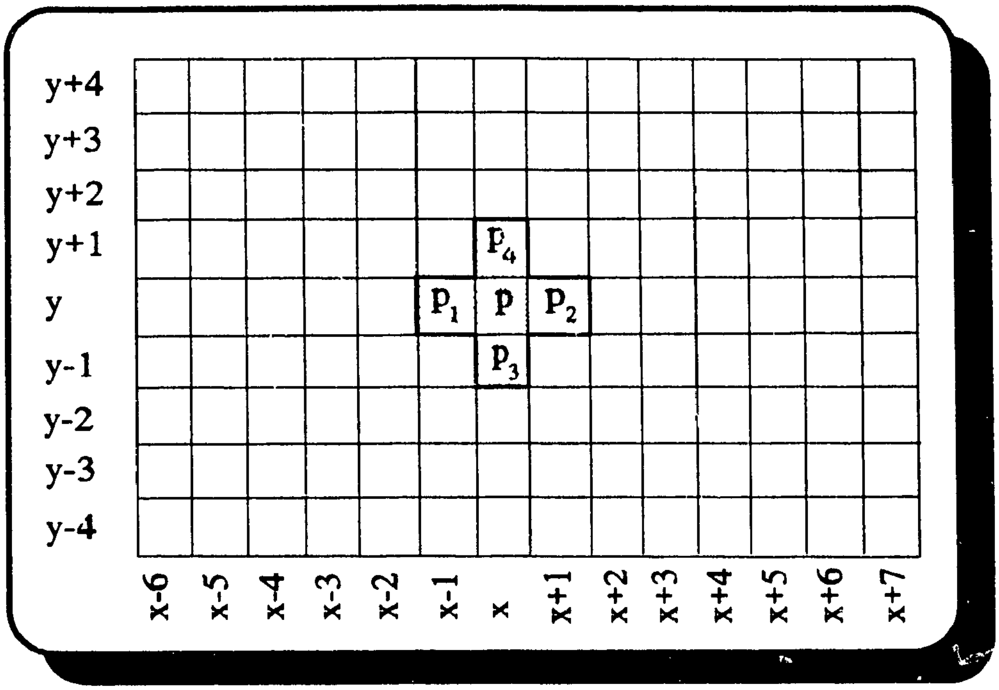


Fig. 4.4 5-pixel scanning window

Let vectors $\boldsymbol{P}$, $\boldsymbol{P_1}$, $\boldsymbol{P_2}$, $\boldsymbol{P_3}$, and $\boldsymbol{P_4}$ denote the chromaticity values of these pixels respectively, anc

$$\begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \boldsymbol{P}_3 \\ \boldsymbol{P}_4 \end{bmatrix} = \begin{bmatrix} R(x,\ y,\ t) & G(x,\ y,\ t) & B(x,\ y,\ t) \\ R(x-1,\ y,\ t) & G(x-1,\ y,\ t) & B(x-1,\ y,\ t) \\ R(x+1,\ y,\ t) & G(x+1,\ y,\ t) & B(x+1,\ y,\ t) \\ R(x,\ y-1,\ t) & G(x,\ y-1,\ t) & B(x,\ y-1,\ t) \\ R(x,\ y+1,\ t) & G(x,\ y+1,\ t) & B(x,\ y+1,\ t) \end{bmatrix} \tag{4.4}$$

If the following condition exits,

$$[\ \boldsymbol{P} \neq (0,0,0)\ ] \cap [\ (\boldsymbol{P}_1 = \boldsymbol{P}_2 = (0,0,0)\ ) \cup (\boldsymbol{P}_3 = \boldsymbol{P}_4 = (0,0,0)\ )\ ] \tag{4.5}$$

then the pixel (x, y, t) is determined as a noise and eliminated by setting $P(x, y, t) = (0, 0, 0)$. Figure 4.5 shows an image obtained after eliminating noises on the image of Fig.4.3.

Fig. 4.5 Image after noise removal

4.5 BLINK DETECTION MODEL (DOB model)

Blinking lights cannot be detected from an isolated image. However, they can be detected from a series of consecutive images as shown in Fig. 4.6. The DOB model is developed in this section for the detection of blinking lights by means of processing an image sequence.

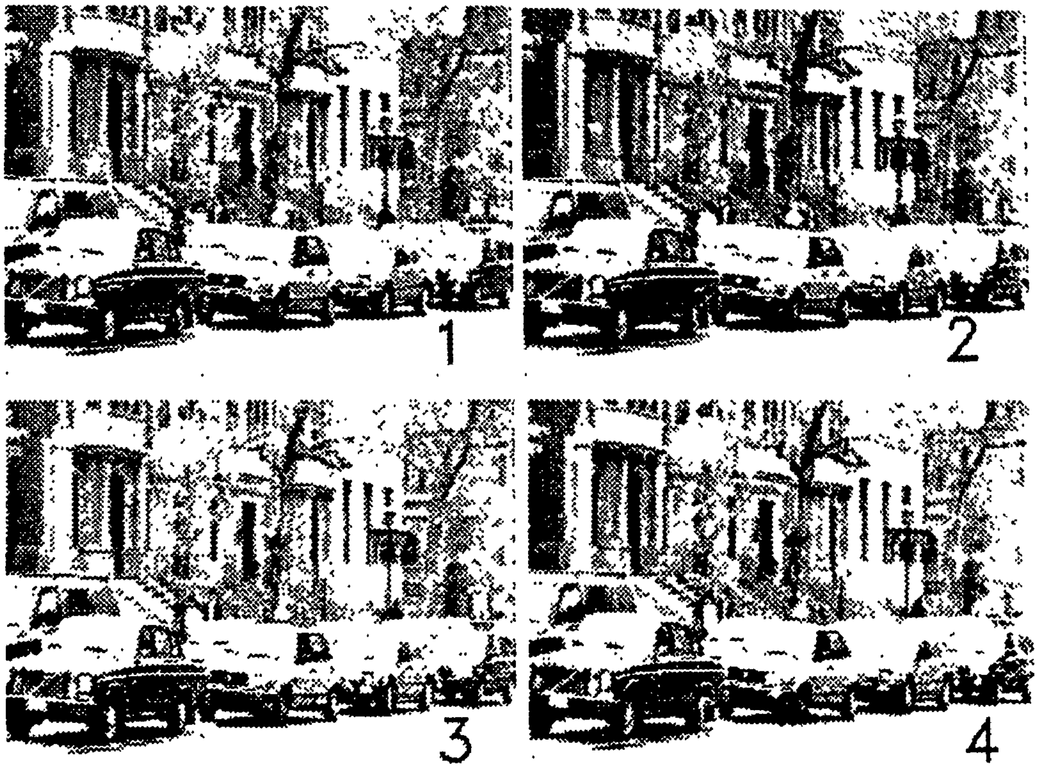


Fig. 4.6 An image sequence

Assume that the signal lights are all that remains in the images after removal of background and noises. In this kind of image as exemplified in Fig. 4.5, the bright clusters are the signal lights.

In order to determine whether these lights blink or not, two parameters are set up. These two parameters are size and brightness of a cluster. The size of a cluster, denoted as s, is the number of pixels contained in the cluster; and the brightness of a cluster, denoted as λ_c, is the summation of the brightness of all pixels contained in the cluster. The size of a cluster in each image is mathematically expressed as follows:

$$s(t) = \sum_{(x,y)\in\Omega_t}\sum F\ [\boldsymbol{P}(x,\ y,\ t)] \tag{4.6}$$

where

$s(t)$ = size of a cluster in image t of the sequence;

Ω_t = an area in image t that just includes one cluster;

$\boldsymbol{P}(x,\ y,\ t)$ = chromaticity value of pixel $(x,\ y)$ in image t;

F = 0-1 value function, and

$$F\ [\boldsymbol{P}(x,\ y,\ t)] = \begin{cases} 0, & \text{if } \boldsymbol{P}(x,\ y,\ t) = (0,\ 0,\ 0); \\ 1, & \text{if } \boldsymbol{P}(x,\ y,\ t) \neq (0,\ 0,\ 0). \end{cases} \tag{4.7}$$

The variation of the size of a light during a period of time is defined as

$$\Delta s = \frac{s_{\max} - s_{\min}}{s_{\max}} \tag{4.8}$$

where

s_{max} = max{ $s(t)$ | t = 1, 2, ..., m };

s_{min} = min{ $s(t)$ | t = 1, 2, ..., m };

m = number of images in a sequence.

The brightness of a cluster is expressed as

$$\lambda_c(t) = \sum_{(x,y)\in\Omega_t}\sum \lambda(x,\ y,\ t) \tag{4.9}$$

where

$\lambda_c(t)$ = brightness of the cluster in image t of the sequence;

$\lambda(x, y, t)$ = brightness of pixel (x,y) in image t;

Ω_t = an area in image t that just includes one cluster.

The variation of the brightness of a light is defined as

$$\Delta\lambda = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max}} \tag{4.10}$$

where

$\lambda_{max} = \max\{\ \lambda_c(t)\ |\ t = 1, 2, ..., m\ \}$;

$\lambda_{min} = \min\{\ \lambda_c(t)\ |\ t = 1, 2, ..., m\ \}$;

m = number of images in a sequence.

Considering the variations of the *size* and the *brightness*, a decision function for blink detection is proposed as follows.

$$u = \frac{\Delta s + \Delta\lambda}{2} \tag{4.11}$$

where

u = decision function, and $0 \le u \le 1$;

Δs = variation of the size;

$\Delta\lambda$ = variation of the brightness.

The decision function u synthetically reflects the change of the blinking lights in *size* and *brightness*. If the change is significant, then the detected light is blinking; otherwise, if the change is slight, then the light is not blinking. A threshold β is set up to determine whether the signal light is blinking or not. The decision rule is as follows:

if $u > \beta$, then the light is blinking; otherwise,

if $u \leq \beta$, then the light is not blinking.

The value of β is recommended to be 0.25 ~ 0.30 according to experiments of this study.

4.6 SIGNAL ALGORITHM AND EXPERIMENTAL RESULTS

4.6.1 Signal Algorithm

The signal algorithm, which is developed to detect vehicles intending to turn at intersections, is divided into four stages as depicted in Fig. 4.7 and discussed below.

Stage 1: Image Acquisition

This stage consists of two steps as follows:

Step 1: Digitize video images. A digitizer fetches frozen frames from the video camera and digitizes them into two-dimensional arrays of pixels by converting the analog signals into digital signals. The value of a pixel is a vector of three components (*R*, *G*, *B*).

Step 2: Fetch the relevant parts of the images into the RAM of a computer. The relevant parts are the image portion containing the left-turning vehicles and their surrounding background as shown in Fig. 4.8. This step significantly reduces the size of the traffic images.

Stage 2: Image Segmentation

In this stage, the algorithm segments the images by using the DOS model developed in the preceding section. In the segmented images, the signal lights are separated from other objects in the images. Two steps contained in this stage are

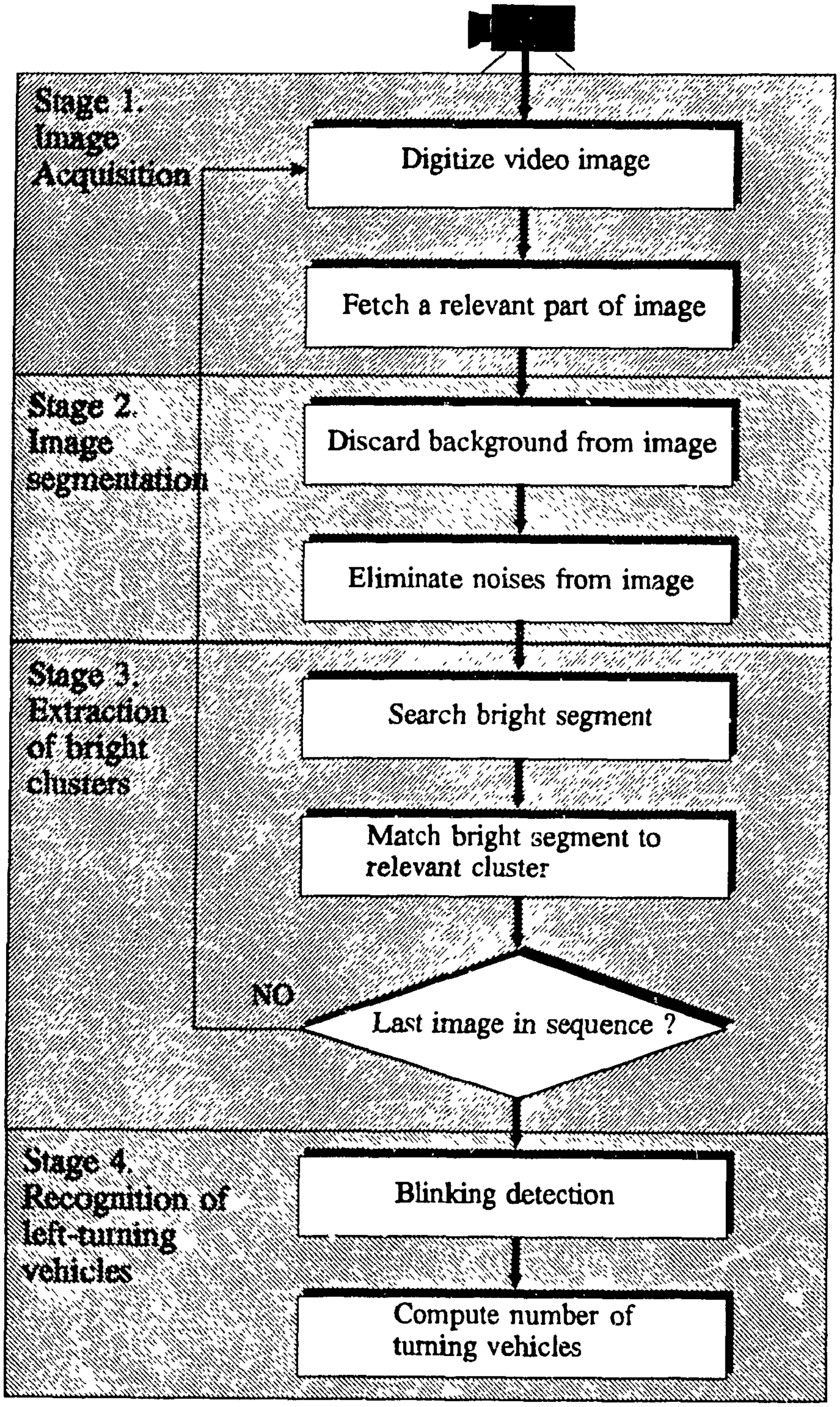

Stage 1. Image Acquisition
Digitize video image
Fetch a relevant part of image
Stage 2. Image segmentation
Discard background from image
Eliminate noises from image
Stage 3. Extraction of bright clusters
Search bright segment
Match bright segment to relevant cluster
NO
Last image in sequence ?
Stage 4. Recognition of left-turning vehicles
Blinking detection
Compute number of turning vehicles

Fig. 4.7 Flow Chart of the signal algorithm

discussed below.

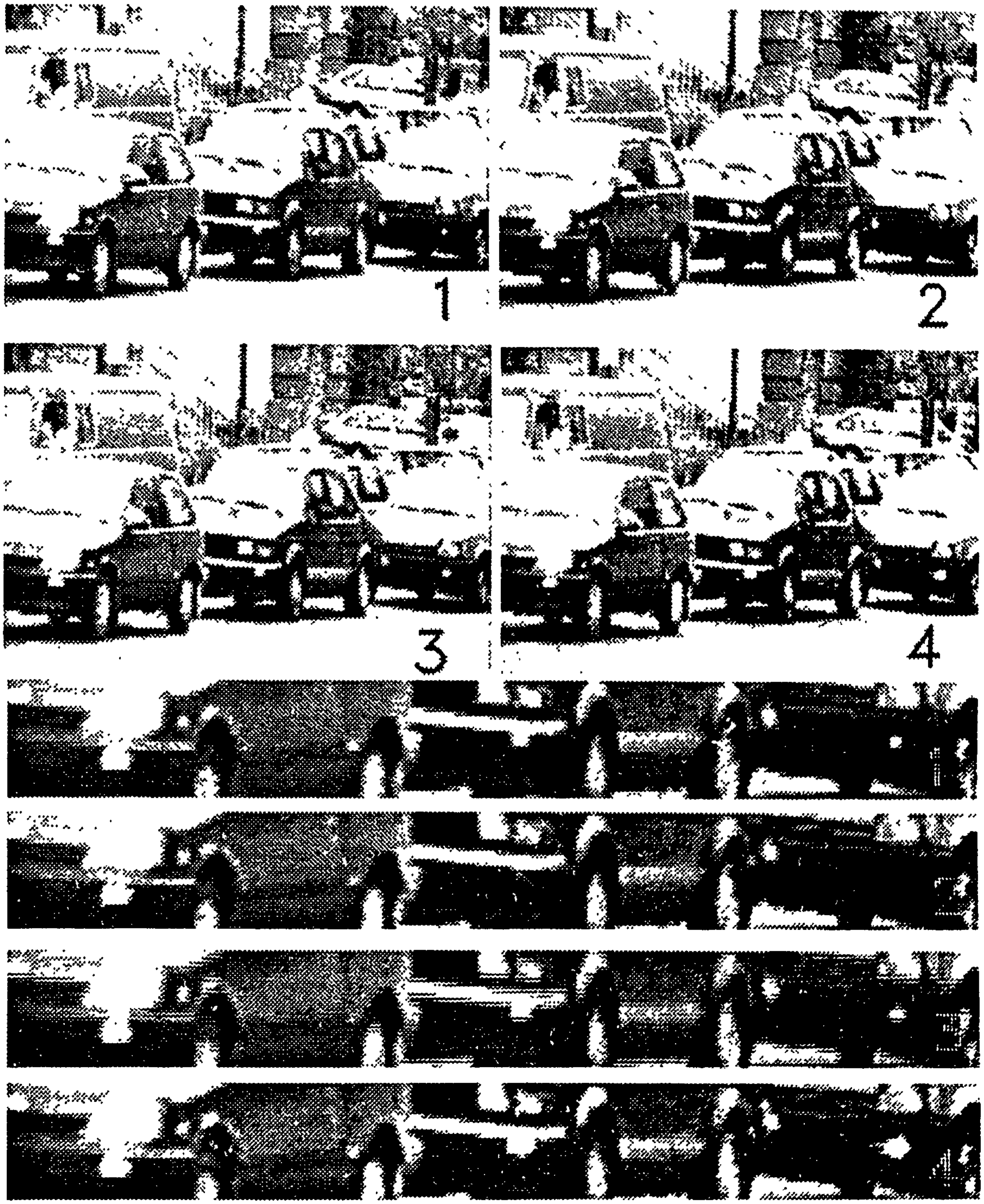


Fig 4.8 Relevant parts of the images

Step 1: Discard background from images. The background filter (Equation 4.1) developed in the preceding section is used for this purpose. The turned-on signal lights of vehicles are separated from other surrounding objects. Consequently, those surrounding objects are removed from the images and segmented images are obtained.

Step 2: Remove noises from the segmented images. The noise filter (Equation 4.5) developed in the preceding section is used in this step in order to obtain a noise-free image.

Stage 3: Extraction of Bright Clusters

A bright cluster is a group of adjacent pixels. In this stage, the segmented images are scanned in order to locate the bright clusters that are likely to be signal lights or other bright objects. Three major steps are included in this stage and presented below.

Step 1: If any bright segment (a group of horizontally adjacent pixels in a row) is found at the current scanned row of the image, proceed to step 2. Otherwise, continue to scan pixel by pixel until the end of the row. When the present row is scanned and a bright pixel is found, this pixel is considered as the left boundary point of a bright segment. Then the algorithm searches for the right boundary point of the bright segment. A pixel is considered as a right boundary point if the following two pixels are not bright.

Step 2: After a bright segment is found in step 1, step 2 examines the possibility that the segment may belong to any existing cluster. If so, the segment will be added to that existing cluster. Otherwise, a new cluster will be created for this segment.

Figure 4.9 illustrates the algorithm for determining whether the bright segment

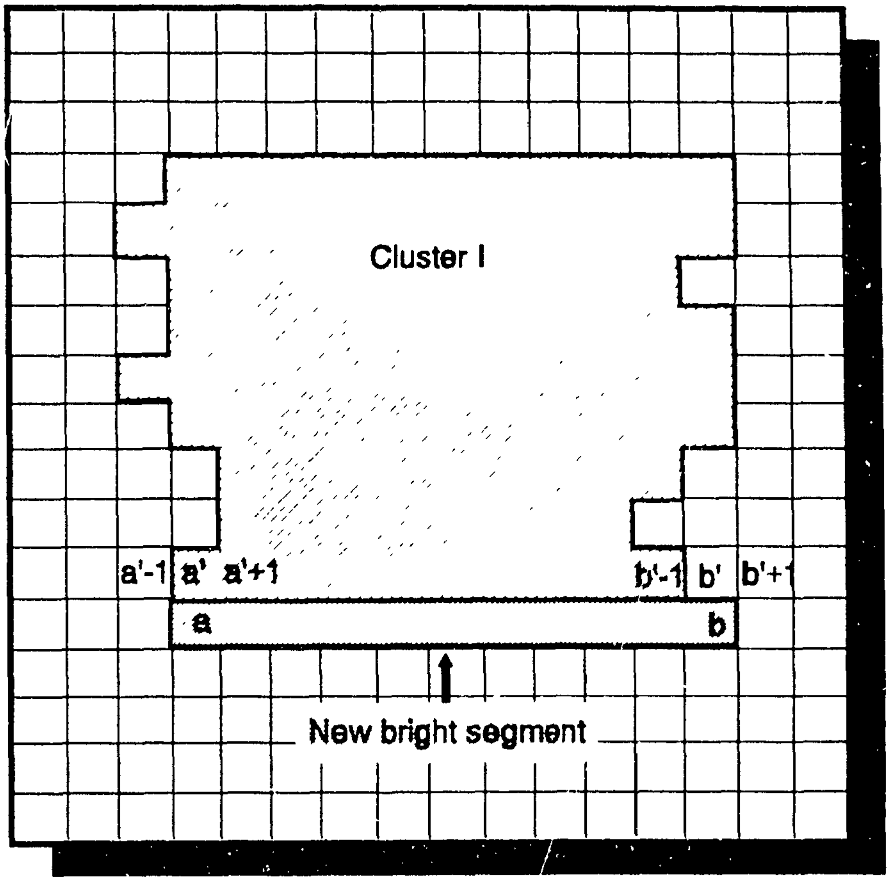


Fig. 4.9 New segment and existing cluster

belongs to any existing cluster. This algorithm does not search every existing cluster for making such determination. Instead, it ascertains whether the pixels corresponding to this bright segment belong to any existing cluster. In Fig. 4.9, the light-shadowed squares are the pixels in the existing cluster *I*, and the dark-shadowed squares are the pixels of a new segment in the current scanning line. Pixels *a* and *b* are the two boundary points of the new segment. In addition, pixels *a'* and *b'* correspond to *a* and *b*, respectively. Pixels *a'-1* and *a'+1* are the two pixels adjacent to pixel *a'*, as are pixels *b'-1* and *b'+1* adjacent to pixel *b'*.

Since the images obtained are high resolution with 512x512 pixels per image, any bright object in the image will have a smooth boundary line. Therefore, this algorithm checks if any of pixels *a'-1, a'*, and *a'+1* or any of pixels *b'-1, b'*, and *b'+1* belong to an existing cluster. If so, the new segment belongs to the same cluster. If not, a new cluster will be created for the new segment. For instance, in Fig. 4.9, pixels *a', a'+1*, and *b'-1* belong to cluster *1*. Thus, the new segment, with two boundary pixels *a* and *b*, belongs to cluster *1*.

Step 3: This step checks if the currently scanned line is the last row of the image. If so, go back to stage 1 for the left images of the sequence, or proceed to stage 4 if this is the last image of the sequence. Otherwise, scan the next row of the image and go back to step 1 of this stage.

Stage 4: Recognition of Left turning Vehicles

In this stage, the blink detection model (DOB model)is utilized to exam the bright clusters for recognition of left-turning lights of vehicles. This stage includes two steps as follows:

Step 1: Detection of blink. The status of the bright clusters in the images are classified into either blinking or not blinking category by the DOB model. Those clusters that are not blinking lights will be discarded in this step.

Step 2: Count the number of left-turning lights. After step 1 is completed, the remaining clusters are left-turning lights. The number of clusters are counted in this step in order to estimate the number of left-turning vehicles.

4.6.2 Experimental Results

A series of experiments have been conducted to verify the signal algorithm. The conditions of the experiments were as follows:

1. The traffic scenes of two intersections in downtown Montreal were videotaped by a video recorder;

2. The digitized images were color ones, and the chromaticity value (R, G, B) ranged from (0, 0, 0) to (255, 255, 255);

3. The microprocessor used was PC 386/25; and

4. The software was written in the TURBO PASCAL computer language.

About 50 image sequences, each lasts about one minute, were used in the experiments. The results indicated the following conclusions:

1. The DOS model is sensitive to the thresholds. Therefore, the proper selection of the thresholds is important when using the DOS model.

2. The noise removal model is very effective for removing isolated or fine noises.

3. The accuracy of the DOB model for distinguishing blinking lights from non-blinking ones was about 94%. The error analysis indicated that errors usually occurred because some turning lights blinked so fast that no image showing the off moment of the blinking circle was fetched in the image sequences.

4.6.3 Discussion

The accuracy of 94% is a significant increase compared to the previous algorithm with an accuracy of 80%. This is because the newly developed DOS and DOB models employed new criteria in the detection such as color, characteristic of blink, and variation in size and brightness of the signal lights, they can solve the problems that are very difficult for the previously developed algorithm (Lu 1988). The signal algorithm no longer suffers from the difficulties of accurately recognizing the signal lights with variation in size, or detecting them from among the reflections of small objects whose sizes and shapes are similar. Also, the algorithm is able to distinguish turning signal lights from turned-on parking lights of vehicles. This function was not available in the previous algorithm. The signal algorithm was developed for detection of vehicles intending to turn at intersections. It is, particularly, useful in the case where there is no special turning lane. Although the algorithm may not provide an accurate practical estimate of the number of turning vehicles, simply due to a fact that some drivers do not use their turning indicators, it provides a method for detecting emergency vehicles such as ambulances and police cars.

CHAPTER 5

CLASSIFIER ALGORITHM

5.1 INTRODUCTION

Vehicle classification can be defined as observation of highway vehicles and the subsequent sorting of the resulting data into a fixed set of categories (*Traffic 1985*). In practice, vehicle-classification data are extremely important because they are involved in most aspects of transportation and traffic engineering such as pavement design, pavement-maintenance scheduling, commodity flow analysis, highway-capacity analysis, weight enforcement, and environmental analysis.

The number of available categories varies widely depending on the type of sensing devices. For instance, vehicles can be categorized into four groups based entirely on overall length (raw data obtained from loops), or vehicles can be divided into eight groups based on the number of axles and axle spacings (raw data obtained from axle sensors). Due to the fact that no currently used detector uses inputs from both presence and axle sensors at the same time, there appears to be a significant gap between what available systems can deliver and what users desire.

In this chapter, four models were developed to form a new vehicle classification algorithm, called classifier algorithm. The new algorithm intends to extract data not only on length but also on height, width and profile features of vehicles. These data are usually not available with other sensors. With these detailed vehicle data, the classifier algorithm is able to divide vehicles into many complicated categories, thereby reducing the gap between the requirement and the availability. Similar to current methods which are used in video traffic detection systems such as AUTOSCOPE (1995) and CCATS (1995), the algorithm does not intend to detect the number of axles directly because that may be impossible for computer vision techniques under occlusion situations. However, the number of axles may approximately be estimated using vehicle length and number of units. In section 5.2, commonly used detectors for vehicle classification are reviewed. Section 5.3 presents the requirements of the new tree type classifier. Four models and the classifier algorithm are presented in section 5.4. The experimental results are discussed in Section 5.5. Finally, a summary of this chapter is introduced in Section 5.6.

5.2 DETECTOR TECHNOLOGIES FOR VEHICLE CLASSIFICATION

The detector technologies that can be employed for traffic data collection include (1) loop detectors, (2) piezo electric detectors, (3) infrared detectors, (4) microwave detectors, (5) ultrasonic detectors, and (6) video image analysis.

Loop Detectors

The inductive loop detector is currently the most widely used detector for traffic control and data collection. Although vehicle classification is not the strongest function of such kinds of detectors, some of them are able to provide the length of passing vehicles. This is usually determined by detecting the speed of the vehicle and measuring the time taken to pass the site. Vehicle classification is then undertaken by considering a combination of parameters that loop detectors can provide. *Lyles and Wyman* (1983) pointed out the accuracies of loop detectors. The minimum error for detectors using overall length as a vehicle classification parameter was 5 - 8% whereas the maximum was 82% if there was an incorrect tuning of loop detectors. *Pursula and Kosonen* (1989) once discussed the use of analog signals produced by loop detectors in vehicle classification. That was performed according to the features (the length, height, and form) of the analog signal generated by the passing vehicles. But their design required that vehicles travel at uniform speed, an operational condition not found in most practical situations.

Piezo Electric Detector

Piezo electric cable is used to provide an accurate axle detector for traffic data collection, especially vehicle classification. But a disadvantage is the fact that the detectors operating on the piezo electric principle cannot provide detailed information about size of vehicles since outputs are provided as an axle crosses the detector, and there is no overall vehicle detection zone. Therefore, systems using such axle sensors experienced numerous errors ranging from a low of 15 - 20% to a maximum of about two thirds of all non-two-axle vehicles observed being misclassified (*Lyles 1983*).

Infrared and Microwave Detectors

Both the infrared and microwave detectors represent above ground vehicle sensors that utilize, respectively, the infrared and microwave wavelengths of the electromagnetic spectrum. Pan (1991) developed a vehicle classification system using a photo-electronic sensor. The sensor used for vehicle detection and dimension measurement consists of two parts: an infrared transmitter and a light sensitive receiver. By placing the transmitter and receiver side-by-side, opposite each other, an uninterrupted beam from the transmitter to the receiver will form. The output signal from the receiver changes when the continuous beam is interrupted by the passage of a vehicle. Once the measurements are taken, vehicles are classified on the basis of these measurements.

Ultrasonic Detectors

Miyasako (1989) proposed an ultrasonic vehicle classifier with which vehicles were classified by comparing their profiles with a library of standardized profile data. An overhead ultrasonic transducer was used to measure the profiles of vehicles while they passed under it. The application of Miyasako's design was limited by its accuracy.

From the above overview of existing detectors, it can be seen that the detector techniques developed for vehicle classification are based on sensors either for measuring vehicle length or for counting the number of axles, wheelbases, and axle spacings. Both length and axle sensors used in these systems perform a "blind" type of detection, that is, only the presence or absence of a vehicle over the sensors can be detected. Two fundamental problems exist in such systems. The problems are: (1) any single sensing device cannot provide adequate information for comprehensive classification of vehicles, (for example, buses cannot be accurately separated from trucks; vans, pickup trucks, and other two-axle four-tire vehicles overlap with passenger cars; and motorcycles cannot be reliably detected or differentiated from other vehicles); (2) portable sensing devices are prone to both purposeful and accidental damage.

To solve these problems and improve vehicle classification systems, video image analysis techniques have been introduced in this area.

Video Image Analysis

Video image analysis techniques have great potential to improve vehicle classification systems because video images contain much more detailed information about passing vehicles than the other detectors can provide. In other words, a lot of data, such as width, height, and profile characteristics of vehicle (or type of vehicle shape) that are not easily obtained by other kinds of detector, can be extracted by image analysis. Therefore, many complicated categories could be obtained.

In recent years, video-image-analysis techniques have been applied with varying degrees of success to automatically detect and measure the presence and speed of vehicles in real time (Taylor and Young 1988). However, vehicle classification, as one of several difficult tasks for applying image analysis techniques in the area of traffic-data collection, has not been fully developed. Blosseville (1990) proposed a traffic sensor called TITAN based on image processing techniques. Although the feasibility of image processing for vehicle classification was proved, the significant advantage of this technique was not shown in their study because vehicles can only be classified into either light or heavy categories. Two other systems were developed separately by two research groups (Pan 1991; Lu 1992) using infrared images for vehicle classification. However, these researchers stayed with the same classification parameters in their studies, that is, number of axles and axle spacings. In addition, the use of the infrared system is usually more expensive than the use of the ordinary camera system. This is because purchase and installation of new infrared systems will cost more than modifying the ordinary camera systems that already exist. A method is needed for automatically performing a comprehensive classification of vehicles at low cost.

5.3 REQUIREMENTS OF THE VEHICLE CLASSIFIER

There is a wide variation of vehicle classification emanating from different government authorities and traffic engineers. Based on their needs, they can define their own classification scheme. For instance, the Federal Highway Administration (FHWA) has classified vehicles into 13 categories (*Traffic 1985*). However, in Canada, the classification scheme is different from the FHWA scheme because of different needs and laws. For example, there are 20 categories used in the province of Quebec and 18 in the province of Ontario. Traffic engineers usually classify vehicles into four categories for the purposes of capacity analysis. These four categories are passenger cars, trucks, recreational vehicles and buses. Many vans and small panel trucks may be classified as passenger cars. This classification scheme is based on the number of wheels of a vehicle and for what it is used. But for the purpose of pavement design and maintenance scheduling, only heavy vehicles are considered. Therefore, vehicles are only divided into two groups: light and heavy vehicles. Heavy vehicles are further divided into several categories by the combinations of the size, weight and number of axles. In this kind of classification scheme, the size, weight and number of axles of vehicles become the most important parameters for the classification.

A classifier algorithm, which is a tree type classifier, will be developed in this chapter. In order to enable the classifier to classify vehicles into as many categories as possible, it will meet the following requirements in two categories: functional requirements and performance criteria.

Functional Requirements

The sensors of the classifier should be able to provide information about dimension of vehicles and type of vehicle shapes. The dimension of vehicles includes length, width and height. The type of vehicle shapes, which will be called *profile characteristics* of vehicles in this thesis, include (1) the front shape of the vehicle: flat front or projecting front; and (2) the number of units of which a vehicle is composed. profile characteristics passing the detection site.

The classification schemes that are available with the classifier algorithm are formed by the combination of a vehicle's length, height, and profile characteristics. In other words, the classifier algorithm should be able to segregate vehicles on the basis of different combinations of their size and profile characteristics. Consequently, the algorithm can meet different application requirements and users can classify vehicles according to their own classification scheme.

Performance Criteria

The classifier algorithm should meet the following performance requirements.

It is important that the algorithm does not require adjustment after initial set up.

The algorithm should be able to work at all typical road driving speeds, not those necessarily at or below the national speed limit. Any vehicle in a video image may be measured and classified no matter what position it is in the image.

The algorithm model should be able to differentiate buses from trucks, and separate single unit trucks from single trailer trucks.

The errors in estimations of width and height of vehicles must be less than 0.2m and 0.4m, respectively. The number of units of vehicles should be obtained at an accuracy of more than 95%.

5.4 MODELS AND CLASSIFIER ALGORITHM

A vehicle classification algorithm, called classifier algorithm, is proposed in this section based on four models as shown in Fig. 5.1 Each model will be presented in the following sub-sections.

5.4.1 Model Of Length Measurement (LM Model)

The length of a vehicle is defined as the length from its front bumper to its rear one. The model of length measurement (LM model) is developed to measure the actual length of vehicles from video images.

A coordinate system as shown in Fig. 5.2 is set up in the images for measurement of vehicle length. The origin of the coordinate system is located at the center of the frame. To measure the length of vehicles in a lane, a line of pixels is considered as a detector that is parallel with the movement of the vehicles and located in the middle of the lane. Vehicles moving in the lane can be detected by means of checking the value

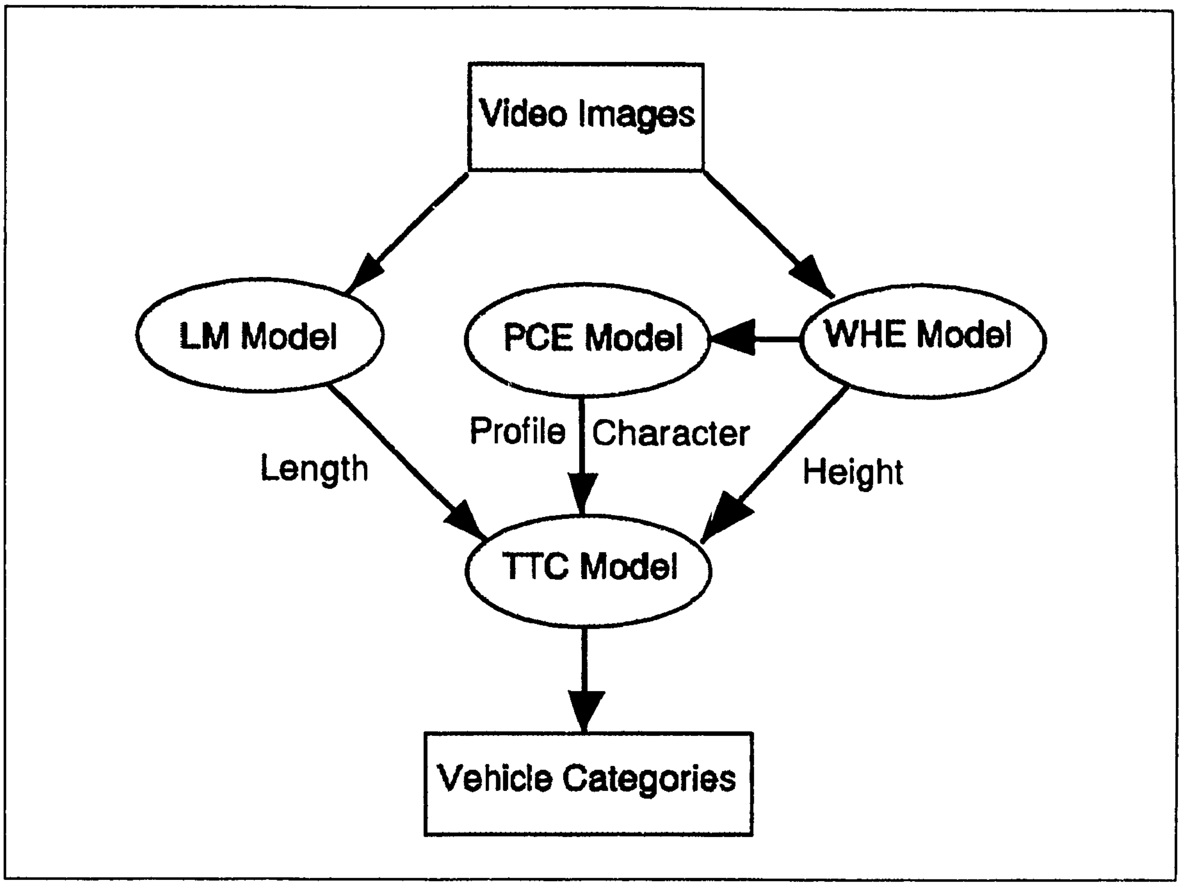


Fig. 5.1 Flow chart of the classifier algorithm

variations of the pixels in this detector. Details of the algorithm for the vehicle detection were described in Chapter Three. Following is a brief summery of the algorithm:

Step 1. Extraction of object. Two *color detection functions* f_1 and f_2 (Equation 2.7) are employed in order to differentiate moving objects from background and shadows. Then the original color image is transformed to a binary image (Equation 3.3)

Step 2. Measure the size of the vehicle in each frame. The length of the vehicles can be obtained in this step. Because the detection line for length measurement is placed horizontally in the image plane, Equation (3.9) can be expressed in the following simple way:

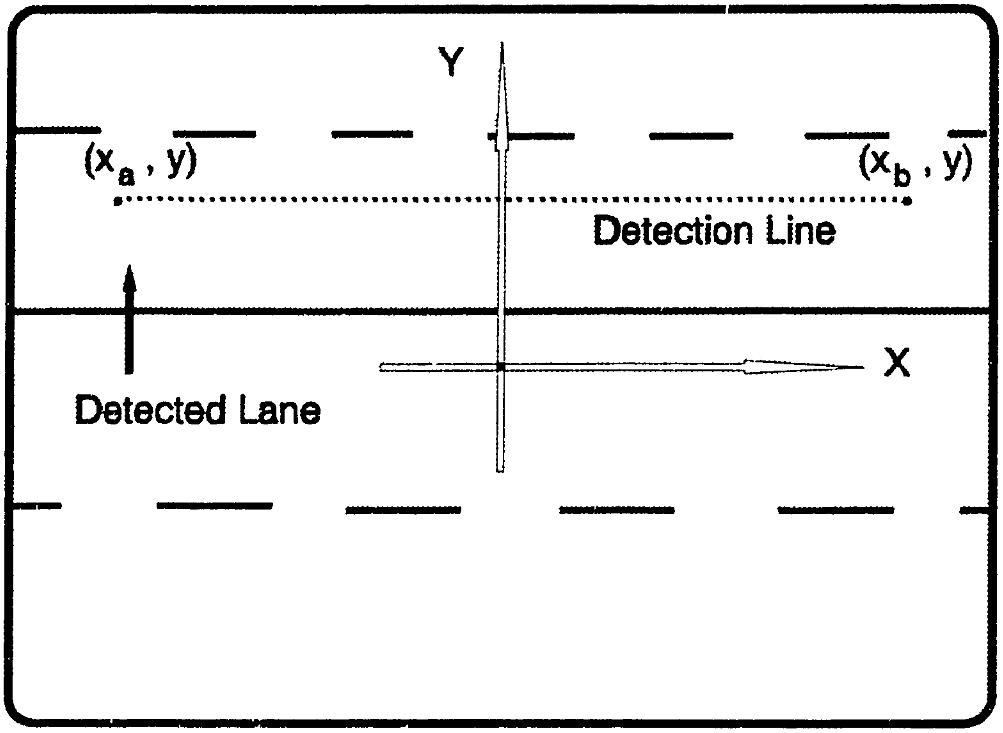


Fig. 5.2 Vehicle length measurement

$$L_t = \left| \xi(x_b, y) - \xi(x_a, y) \right|$$

$$= \left| \frac{H\ k_x\ x_b}{yk_y \sin\theta + \cos\theta} - \frac{H\ k_x\ x_a}{yk_y \sin\theta + \cos\theta} \right| \qquad (5.1)$$

$$= \left| \frac{H\ k_x\ (x_b - x_a)}{yk_y \sin\theta + \cos\theta} \right|$$

where

L_t = length of the vehicle measured in image t, in term of meters;

(x_a, y) = coordinates of the left boundary of the vehicle, y = constant;

(x_b, y) = coordinates of the right boundary of the vehicle, y = constant;

H = distance from the video camera to the *horizontal plane*;

θ = angle of the camera mount tilt ($-\pi/2 < \theta < 0$);

k_x, k_y = coefficients as determined by Equation (2.38).

Step 3. Determine the overall length of the vehicle. Note that L_t is not necessarily the overall length of the vehicle, since it may represent the length of only a part of the vehicle while it enters or leaves the detection area. Therefore, the overall length of a vehicle is obtained only from those images that include the whole vehicle. In other words, the overall length is represented as

$$L = \max \{ L_t \mid t = 1, 2, ..., m \} \tag{5.2}$$

where

L = overall length of a vehicle, in meters;

L_t = length of the vehicle measured in frame t;

m = number of consecutive images showing the vehicle.

Figure 5.3 shows an example of a group of consecutive images in which a vehicle is measured when passing through the detector.

5.4.2 Model of Width And Height Estimation (WHE Model)

The appearance of vehicles in a frame is exemplified in Fig. 5.4. Since both the width and height of a vehicle are projected in the vertical direction of the image plane, determining the width and length separately is a complicated process. The WHE model is proposed to estimate the width and height of vehicles according to the measures of the

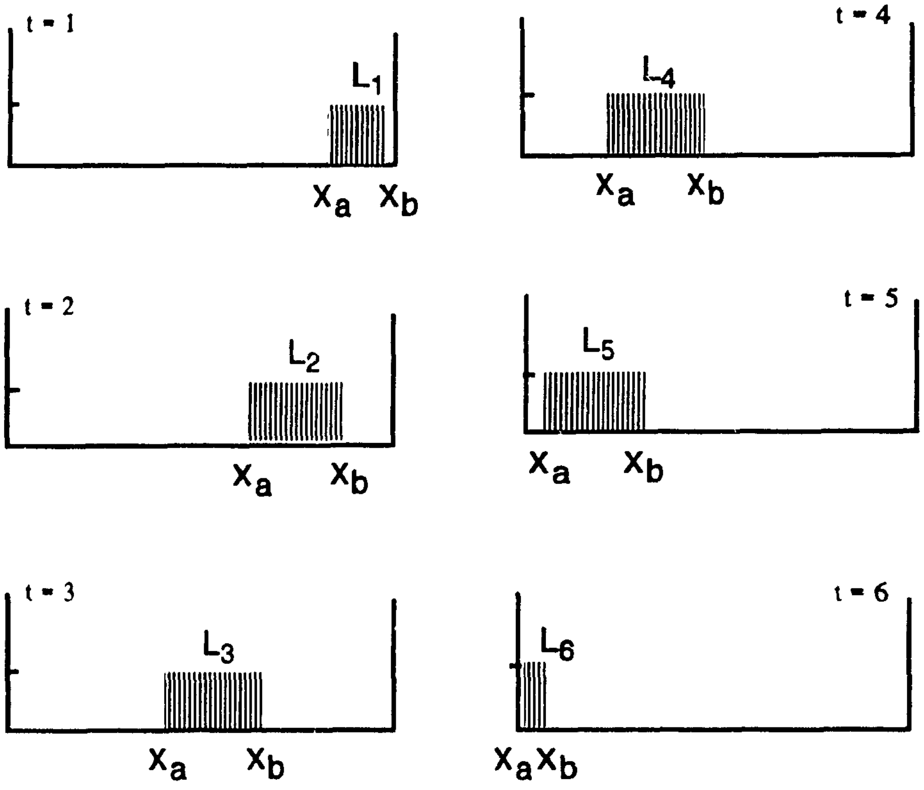


Fig. 5.3 Vehicle Length Measurement

distance from the lower to upper boundaries of the vehicles in the video images. The block diagram of the WHE model is shown in Fig. 5.5. Details of the WHE model are described below.

1. Determine the lower and upper boundaries in the images

To estimate the width and height of a vehicle, the lower and upper boundaries of the vehicle must be detected from the video images. As in a similar method used for the LM model, only one line of pixels placed perpendicular to the movement of the vehicles is employed as a detector, as exemplified in Fig. 5.6. By means of checking the

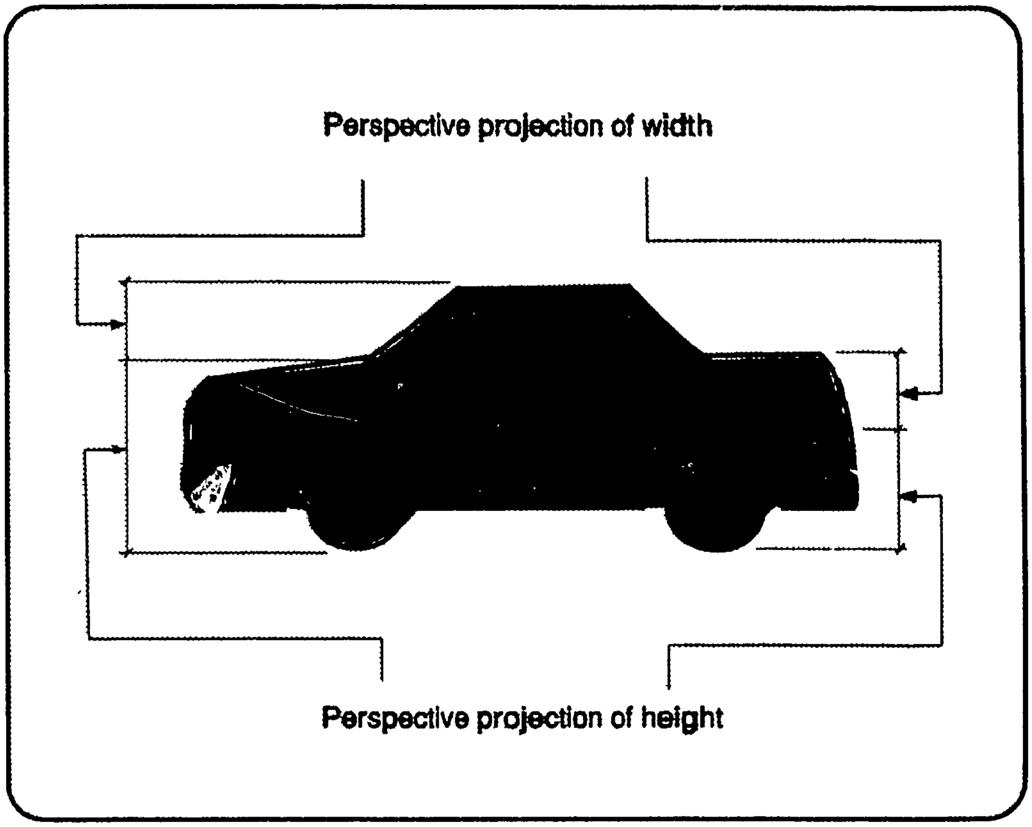

Perspective projection of width
Perspective projection of height

Fig. 5.4 Appearance of vehicle in a frame

variations of the pixel values, both boundaries of a vehicle can be determined.

2. Project the boundaries into the horizontal plane

Any point in the image plane has a perspective projection in the horizontal plane. If the position of a point is determined in the image plane, its position in the horizontal plan can be calculated by using the perspective projection model (Equation 2.37). As illustrated in Fig. 5.7, two boundary points of a vehicle in the image plane have two corresponding points in the horizontal plane. Their coordinates can be derived as

$$(\xi, \eta_a)^T = \boldsymbol{T} [(x, y_a)]$$

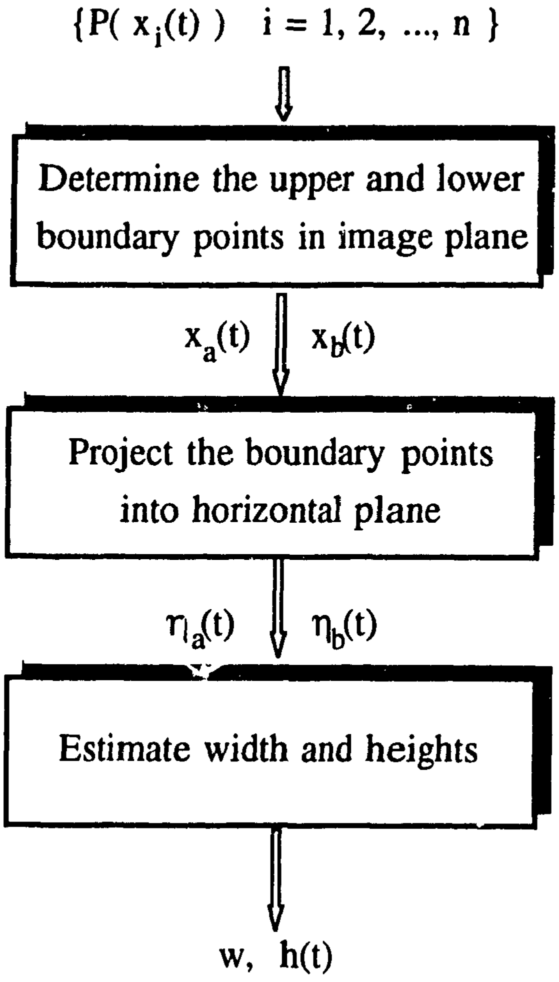

{P(x_i(t)) i = 1, 2, ..., n }
Determine the upper and lower boundary points in image plane
x_a(t)
x_b(t)
Project the boundary points into horizontal plane
η_a(t)
η_b(t)
Estimate width and heights
w, h(t)

Fig. 5.5 Flow chart of the WHE model

$$(\xi,\ \eta_b)^T = \boldsymbol{T}\,[\ (x,\ y_b)\ ] \tag{5.3}$$

where

$(x,\ y_a)$ = coordinates of the lower boundary in the image plane;

$(x,\ y_b)$ = coordinates of the upper boundary in the image plane;

$(\xi,\ \eta_a)$ = coordinates of the lower boundary in the horizontal plane;

$(\xi,\ \eta_b)$ = coordinates of the upper boundary in the horizontal plane;

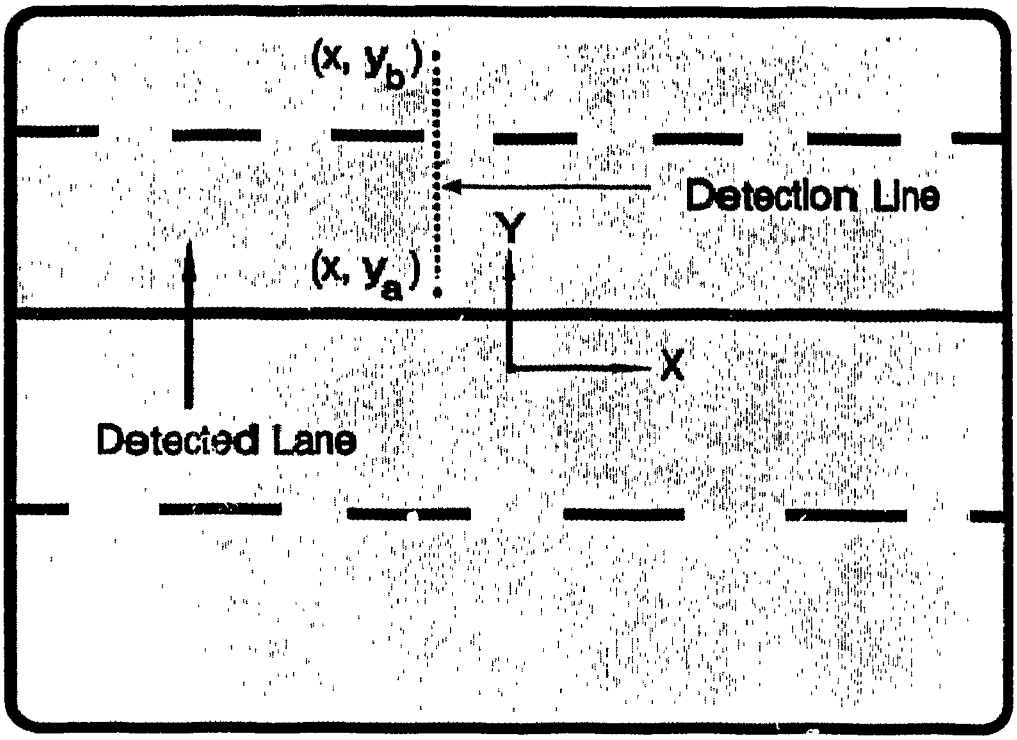


Fig. 5.6 Detection line for the WHE model

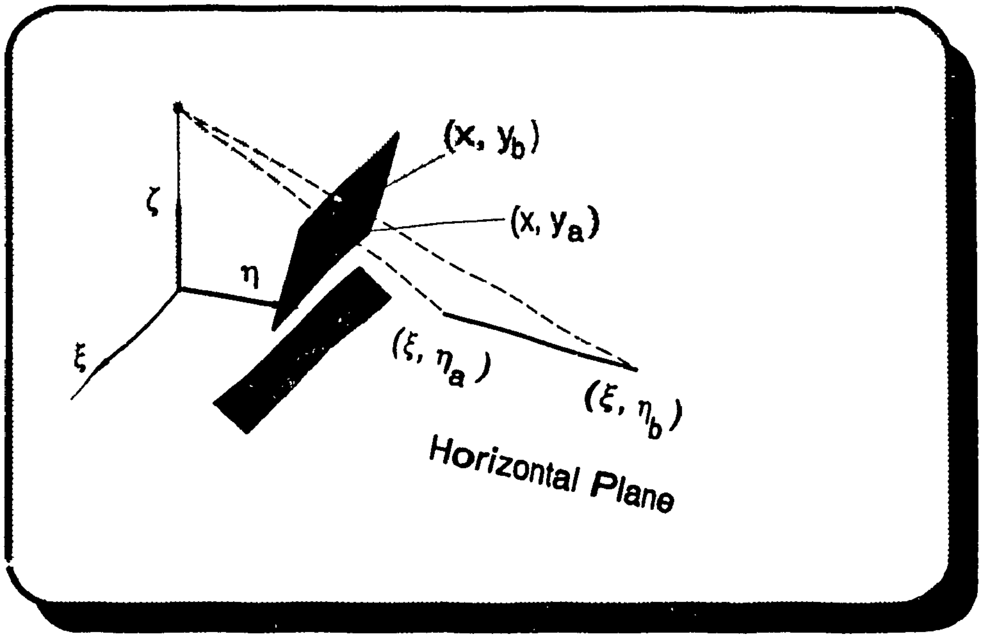


Fig. 5.7 Perspective projection of points

T = transformation function of the perspective projection.

Since x and ξ are constant and have no relation to the measurement, they are therefore neglected. Considering only the transformation between y and η, we have

$$\eta_a = \frac{H\,(n_x \sin\theta + y_a \tan\frac{\phi_x}{2} \cos\theta)}{n_x \cos\theta - y_a \tan\frac{\phi_x}{2} \sin\theta}$$

$$\eta_b = \frac{H\,(n_x \sin\theta + y_b \tan\frac{\phi_x}{2} \cos\theta)}{n_x \cos\theta - y_b \tan\frac{\phi_x}{2} \sin\theta} \qquad (5.4)$$

where

ϕ_x = vertical visual angle of the camera;

ϕ_y = horizontal visual angle of the camera;

θ = rotation angle from camera's lens axle to vertical line, in degrees;

H = vertical distance from video camera to the horizontal plane;

n_x = number of pixel rows in a video image;

n_y = number of pixel columns in an image.

3. Estimate the width and height

If a vehicle overlays the detector and its two boundaries are transformed into the *horizontal plane*, then the width and height of the vehicle can be estimated by the WHE model. The details of the WHE model are described below.

In a frame t, suppose that the coordinates of the two boundaries are extracted from the *video image plane* and projected onto the *horizontal plane*. The corresponding

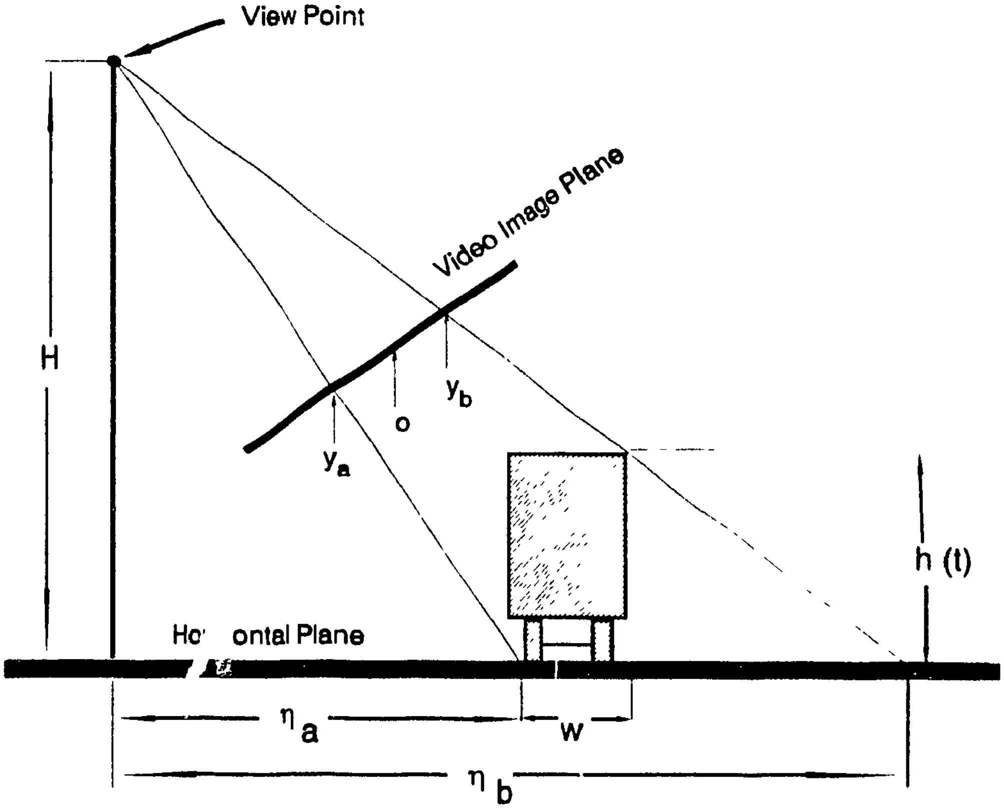


Fig. 5.8 Estimation of width and height of vehicle

coordinates obtained in the *horizontal plane* are $\eta_a(t)$ and $\eta_b(t)$ respectively representing the lower and upper boundary. By inspection of Fig. 5.8, it is easily verified that

$$\frac{h(t)}{H} = \frac{\eta_b(t)-\eta_a(t)-w(t)}{\eta_b(t)}$$

or (5.5)

$$h(t) = \frac{H}{\eta_b(t)}[\eta_b(t)-\eta_a(t)-w(t)]$$

where

h(t) = height of vehicle in image *t*, in meters;

w(t) = width of the vehicle in image *t*, in meters;

$\eta_a(t)$ = coordinate of the lower boundary in the horizontal plane;

$\eta_b(t)$ = coordinate of the upper boundary in the horizontal plane;

H = distance from camera to the horizontal plan, in meters.

Because there is only one equation with two unknown variables (h and w), one variable must be estimated. Regarding $w(t)$ as an independent variable, it can first be estimated on the base of following two assumptions:

Assumption 1: The width of vehicles varies from 1.6m to 2.6m.

Assumption 2: The lowest part of vehicles is 0.6m (such as the hood of a car) and the highest part is 4.0m (such as the roof of a heavy truck).

From the above equation, it is easy to conclude that a smaller estimation than actual width of a vehicle will result in a larger estimation than actual height. On the contrary, a larger estimation in width will cause a smaller estimation in height. Therefore, estimation of width is limited not only by assumption one but also by assumption two.

With these assumptions, the minimum estimated width of a vehicle can be obtained as

$$w_l = \max\{\ 1.6\ ,\ w'_l\ \} \tag{5.6}$$

where

w'_l = minimum estimation of the width, in meters;

1.6 = minimum possible width of vehicle, in meters;

w'_l = result of following equation:

$$w_1' = \Delta\eta_{\max} - \frac{4.0\ \eta_b(\Delta\eta_{\max})}{H} \tag{5.7}$$

where

$\Delta\eta_{max} = \max\ \{\ \eta_b(t)\text{-}\eta_a(t)\ |\ t\text{=}1,2,...,m\ \}$;

$\eta_b(\Delta\eta_{max})$ = coordinate of the upper boundary corresponding to $\Delta\eta_{max}$;

H = distance from camera to the horizontal plan, in meters;

4.0 = the largest possible height of vehicle, in meters;

In the same way, the maximum estimation of a vehicle's width can be obtained as w_2 and

$$w_2 = \min\{\ 2.6\ ,\ w'_2\ \} \tag{5.8}$$

where

w_2 = maximum estimation of the width of a vehicle, in meters;

2.6 = the largest possible width of vehicle, in meters;

w'_2 = result of following equation:

$$w_2' = \Delta\eta_{\min} - \frac{0.6\ \eta_b(\Delta\eta_{\min})}{H} \tag{5.9}$$

where

$\Delta\eta_{min} = \min\ \{\ \eta_b(t)\text{-}\eta_a(t)\ |\ t\text{=}1,2,...,m\ \}$;

$\eta_b(\Delta\eta_{min})$ = coordinate of the upper boundary corresponding to $\Delta\eta_{min}$;

H = distance from camera to the horizontal plane, in meters;

0.6 = minimum possible height of vehicle, in meters.

The actual width of the vehicle must be somewhere between w_1 and w_2, and

$$w = w_1 + \delta(w_2 - w_1) \tag{5.10}$$

where

w = estimated width of the vehicle, in meters;

w_1 = minimum estimation of the width, in meters;

w_2 = maximum estimation of the width, in meters;

δ = coefficient given by

$$\delta = \frac{w_2'-w_2}{(w_1-w_1')+(w_2'-w_2)}\ , \qquad 0 \le \delta \le 1 \tag{5.11}$$

where w_1 w_2 w'_1, and w'_2 are defined in previous equations.

Substituting w for $w(t)$ in Equation (5.5), the estimated height of the vehicle in image t can be calculated as

$$h(t) = \frac{H}{\eta_b(t)}\,[\eta_b(t) - \eta_a(t) - w] \tag{5.12}$$

4. A case study

Following is a case of using the WHE model to estimate width and height of vehicles in a real case.

Traffic scenes of Decarie Expressway were videotaped in 1993. The video camera was mounted at a height of 8m and tilted at an angle of 30 degree from the horizontal line as shown in Fig. 5.9. Figure 5.10 is a sample of the scene taken by the video camera.

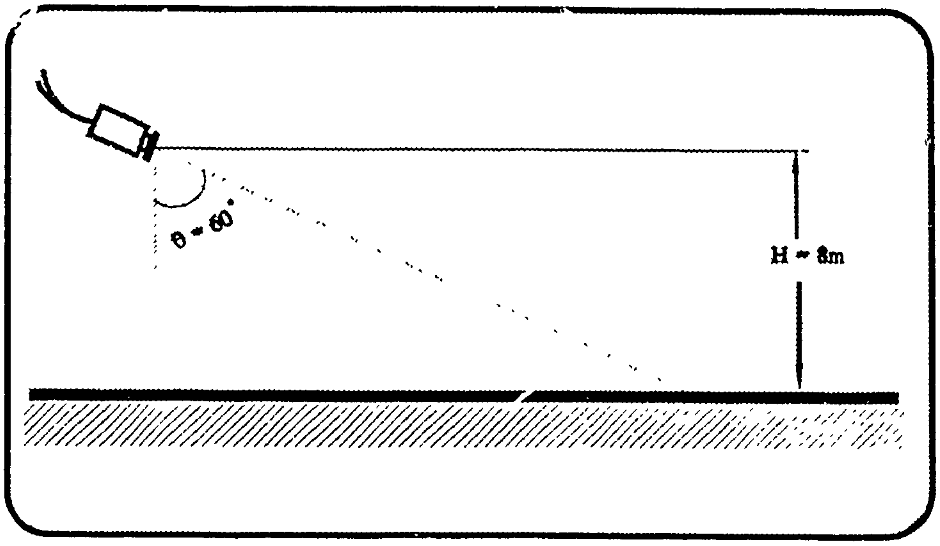


Fig. 5.9 Set up of video camera

TABLE 5.1 Coordinates of the perspective projections of a vehicle

t	$y_a(t)$	$y_b(t)$	$\eta_a(t)$	$\eta_b(t)$	$\Delta\eta(t)$
1	45	99	16.4	20.7	4.3
2	45	102	16.4	21.0	4.6
3	45	111	16.4	21.9	5.5
4	45	123	16.4	23.2	6.8
5	45	123	16.4	23.2	6.8
6	45	123	16.4	23.2	6.8

Table 5.1 shows the results of the perspective projections from the video image into the horizontal plane. Column one shows the index number of the images. Columns two and three list the coordinates of the lower and upper boundaries of the vehicle in the image plane, respectively. Columns four and five illustrate the mapping results by using Equation (5.4). The parameters employed in this case are listed below.

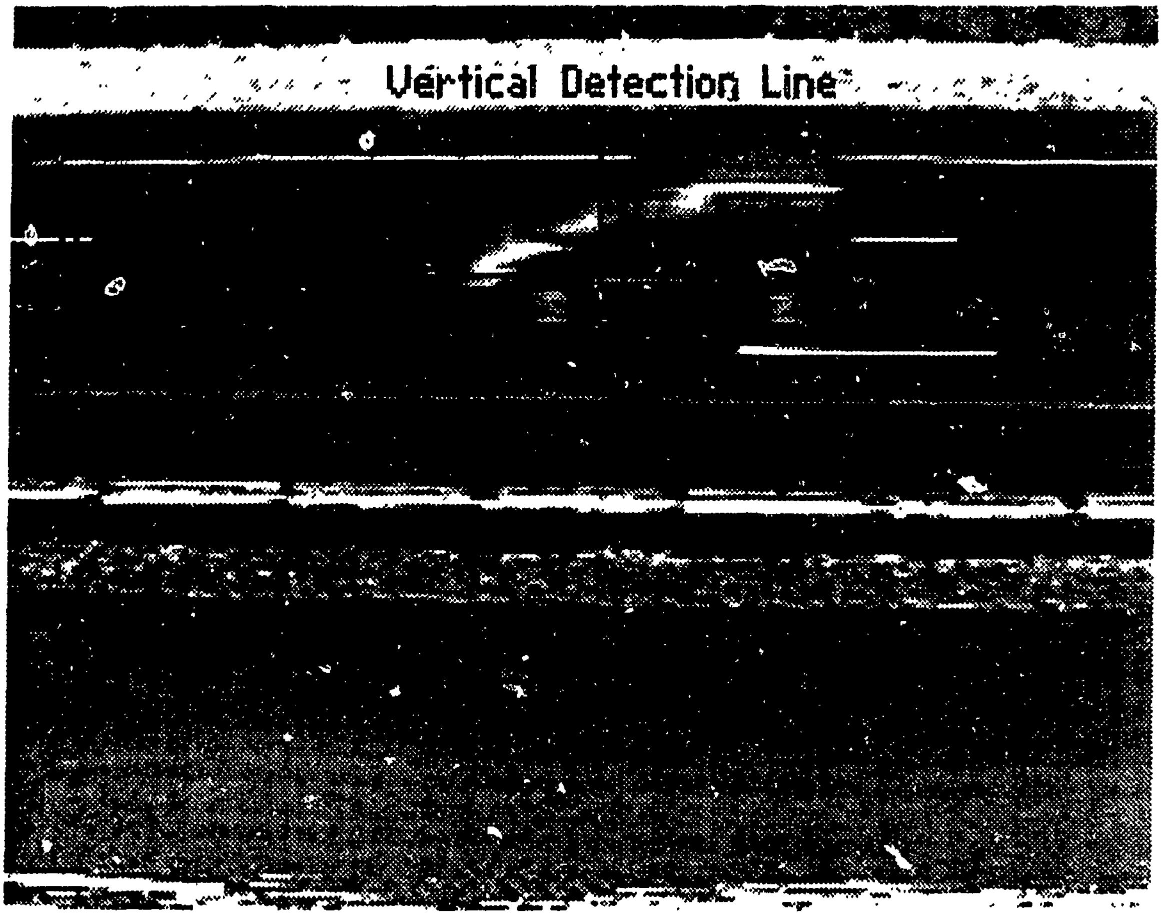


Fig. 5.10 A frame of video image

H = 8m (vertical distance from video camera to the horizontal plane);

ϕ_x = 35° (vertical visual angle of the camera);

ϕ_z = 35° (horizontal visual angle of the camera);

θ = 60° (rotation angle between X'-Z' plane and the horizontal plane);

n_x = 400 (number of pixel rows in an image).

Column six shows the distance between two boundary points. From this column, it was found that $\Delta\eta_{min} = \Delta\eta_b(1) = 4.3$ is the smallest one and $\Delta\eta_{max} = \Delta\eta_b(4) = 6.8$ is the largest one. Therefore, the corresponding $\eta_b(\Delta\eta_{min})$ and $\eta_b(\Delta\eta_{max})$ are 20.7 and 23.2, respectively.

By substituting these data into Equation (5.6) through Equation (5.11), the width of the vehicle was estimated as 1.6m which is very close to the actual width of the detected vehicle (TOYOTA Previa/92 seven-passenger van). Table 5.2 shows the results from Equation (5.6) to Equation (5.11).

TABLE 5.2 Results of the estimation of width

w_1'	w_2'	w_1	w_2	δ	w
-4.8	2.7	1.6	2.6	0.02	1.6

With the estimated width of the vehicle, the height series of this vehicle was obtained by using Equation (5.12). The results are shown in Table 5.3.

TABLE 5.3 Height series of the vehicle

$h(t)$	$h(1)$	$h(2)$	$h(3)$	$h(4)$	$h(5)$	$h(6)$
Value(m)	1.0	1.1	1.4	1.8	1.8	1.8

5.4.3 Model For Extracting The Profile Characteristics Of Vehicle (PCE Model)

The model created in this section is able to extract two profile characteristics of a vehicle. The first is the front shape of a vehicle: flat front or projecting front. The second is the number of units of which a vehicle is composed. Figure 5.11 shows the block diagram of the PCE model.

The model works on the basis of two assumptions.

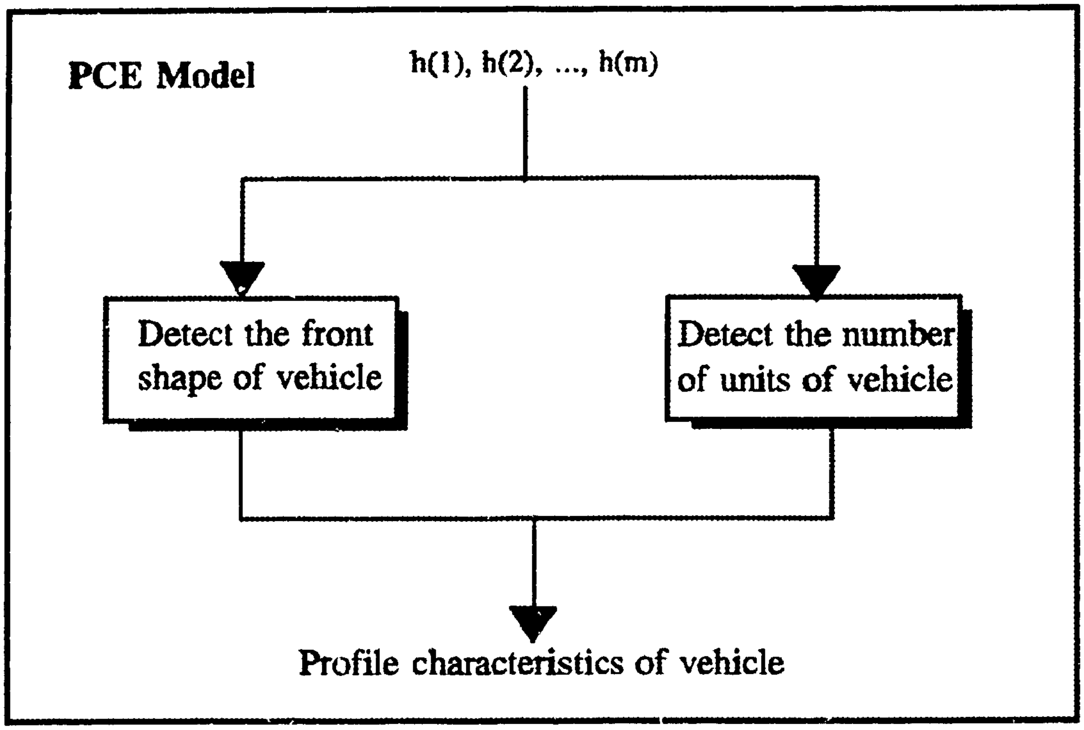

PCE Model
h(1), h(2), ..., h(m)
Detect the front shape of vehicle
Detect the number of units of vehicle
Profile characteristics of vehicle

Fig. 5.11 Flow chart of the PCE model

Assumption 1: The time interval between any two consecutive video images is so short that a vehicle can only move less than 1.0m during this period of time.

Assumption 2: The characteristic parts of a vehicle, such as the projecting part of a projecting-front vehicle or the space between the trailer and its tractor of a trailer truck, are longer than 1.0m.

Thes› assumptions ensure that if a fixed area is predetermined in each frame of an image sequence which records the movement of a vehicle, then any characteristic part of the vehicle can be seen in at least one frame at that fixed area. Therefore, the profile characteristics of a vehicle concerning both its shape and the number of units are reflected in a height series. This series is composed of the heights of a vehicle at different sections. Let *h(1), h(2), ..., h(t), ..., h(m)* denote a height series which can be obtained from *m* consecutive video frames by using the WHE model. As shown in Fig. 5.12, *h(1)* is the height of some point in the front part of the vehicle and *h(m)* is the height of some point in the rear part. Let h_{max} denote the largest value among the height series, and $h_{max} = \max \{ h(t) \mid t = 1, ..., m \}$. The profile characteristic of the shape of a vehicle is extracted under the following rule:

$$IF \quad \frac{h(1)}{h_{\max}} < \beta_1 \qquad (5.13)$$

THEN** the vehicle has a **projecting front

ELSE** the vehicle has a **flat front

where β_1 = a predetermined threshold, $0.7 \le \beta_1 < 1.0$

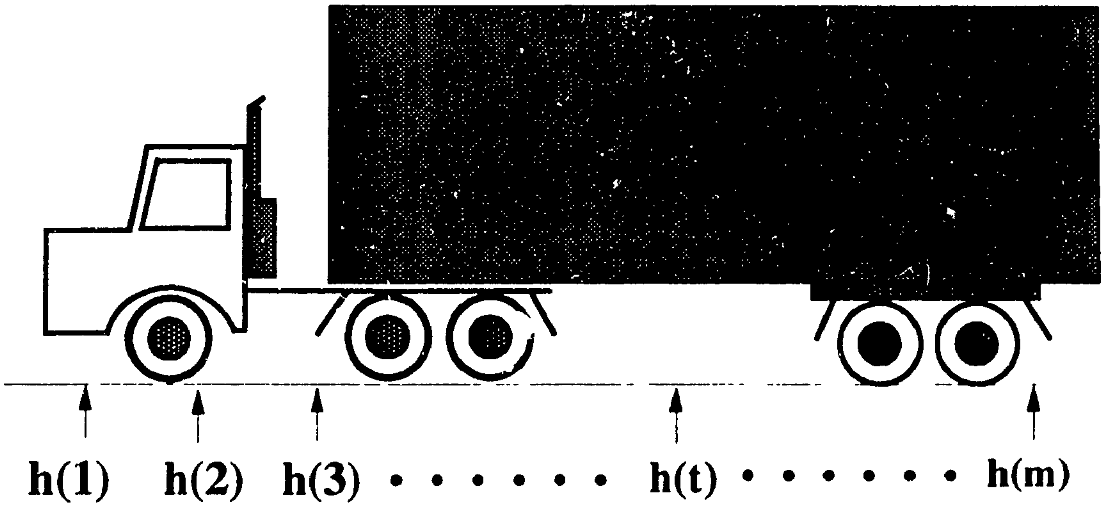


Fig. 5.12 Height series of a vehicle

This rule can be verified intuitively. If a vehicle has a projecting front, then the difference betwecn h_{max} and $h(1)$ should be relatively large because of the lower height of the projecting part. Therefore, the ratio of $h(1)$ and h_{max} is small. Otherwise, if a vehicle has a flat front, the values of h_{max} and $h(1)$ should be close, making the ratio of $h(1)$ and h_{max} large. For instance, the values of the ratio $h(1)/h_{max}$ for a pickup truck and van are 0.7 and 1.0, respectively. If the threshold β_1 is 0.8, then the above rule is able to classify the pickup into the category of projecting-front vehicle while the van into the category of flat-front vehicle.

The profile characteristic regarding the number of units can be obtained by the following procedure.

Step 1: Calculate gradient of each height. If the height series is regarded as a function $h(t)$, a gradient h' is defined as

$$h'(t) = \frac{dh}{dt} = \frac{h(t+\Delta t) - h(t)}{\Delta t} \tag{5.14}$$

For discrete height series with Δt = 1, Equation (5.14) can be implemented digitally by

$$h'(t) = h(t+1) - h(t) \qquad t = 1, 2, ..., m-1$$

Step 2: Find the boundaries of each unit of the vehicle. Since there must be a sharp drop in height at the end of each unit, a greater value of rate can be expected on the corresponding boundary. Thus, the number of units of a vehicle is determined by counting the number of h's whose values are greater than a predetermined threshold β_2. The threshold β_2 is empirically determined as $2.0 \leq \beta_2 \leq 3.0$. It is based on the assumptions that the height of trailers is more than 3.5m but smaller than 4.0m, and the height of the link between units is more than 1.0m but smaller than 1.5m. For example, a height series of a truck is obtained as $h(1)$=1.8m, $h(2)$=1.8m, $h(3)$=2.6m, $h(4)$=2.6m, $h(5)$=1.2m, $h(6)$=3.9m, $h(7)$=3.9m, $h(8)$=3.9m, $h(9)$=3.9m, $h(10)$=3.9m. Observing its gradients which are $h'(1)$=0, $h'(2)$=0.8, $h'(3)$=0, $h'(4)$=-1.4, $h'(5)$=2.7, $h'(6)$=0, $h'(7)$=0, $h'(8)$=0, $h'(9)$=0, it can be determined that the vehicle is a two-unit truck by using the above rule with the threshold $\beta_2 = 2.0$.

5.4.4 Tree Type Classifier (TTC Model)

A tree type classifier (TTC), proposed in this section, is a two-level classifier that is able to classify vehicles into user defined categories. At the first level, vehicles are divided into several categories by a combination of various lengths and heights. The height of a vehicle is defined as the measure extending from the ground to its highest roof. At the second level, vehicles are further divided according to their profile characteristics, such as buses are differentiated from trucks; and single unit trucks, single unit trailer trucks and multi-trailer trucks are differentiated from each other.

First level classification: Classifying vehicles by length and height. An algorithm based on k-nearest-neighbour rule (kNN rule) (Fukunaka 1972) was developed in this level. In order to utilize the kNN rule, vehicles are expressed as pattern vectors whose components are the vehicle's length and height. A pattern vector is defined as

$$\boldsymbol{e} = (L,\ h) \tag{5.15}$$

where

$\boldsymbol{e}$ = pattern vector of a vehicle;

L = length of the vehicle, in meters;

h = height of the vehicle, in meters.

A certain number of vehicles with known vehicle types (e.g. cars, vans, buses, trucks, and trailer trucks, etc) are chosen to form a prototype space. If a given vehicle i is represented by a pattern vector $\boldsymbol{e}_i$ in the prototype space, such as

$$\boldsymbol{e}_i = (L_i\,,\ h_i) \tag{5.16}$$

then the prototype space forms a set of all given pattern vectors, and this set can be expressed as

$$S = \{e_1\,,\ e_2\,,\ \ldots,\ e_i\,,\ \ldots,\ e_n\} \tag{5.17}$$

where

S = a set of vectors representing the prototype space;

n = number of the known vehicles.

Since the types of the n vehicles are known, the prototype space can be divided into several clusters of the pattern vectors. Each cluster containing several different types of vehicles forms a subset within S. The pattern vectors within the same subset represent, therefore, the vehicles of the same group. Let S_1, S_2, ..., and S_q denote q subsets in S. Then the following relations exist:

$$S = S_1 \cup S_2 \cup \ldots \cup S_q \tag{5.18}$$

and

$$S_1 \cap S_2 = \ldots = S_i \cap S_j = \ldots = S_q \cap S_1 = \Phi \tag{5.19}$$

where $i, j = 1,\ 2,\ \ldots,\ q$ and $i \neq j$.

The purpose of the kNN approach is to determine to which subset a new pattern vector, $\boldsymbol{e}_{new}$ where $\boldsymbol{e}_{new} \not\subset S$, belongs. This pattern vector represents a detected vehicle in the video image. In other words, the algorithm will classify detected vehicles into several user defined groups.

With the modified *k*NN rule in mind, the algorithm searches for the k (k is a predetermined integer) nearest neighbours of the unknown vector e_{new} within the prototype space. Then, e_{new} is classified into one of the subsets in the prototype space. The chosen subset is the one that contains more of these k neighbours than the other ones. Details of the algorithm are described through an easy example as follows.

Table 5.4 shows fourteen vehicles with known vehicle types and sizes. These vehicles are chosen to form a prototype space as shown in Fig. 5.13.

TABLE 5.4 Vehicles to form a prototype space

Vehicle type (1)	Length(m) (2)	Height(m) (3)	Vehicle type (1)	Length(m) (2)	Height(m) (3)
car	4.1	1.5	bus	12.0	3.0
pickup	6.0	2.1	truck	7.4	3.0
pickup	5.1	1.7	truck	8.0	2.8
pickup	5.6	1.8	truck	9.5	3.9
van	4.3	1.9	truck	15.0	3.9
van	5.1	2.0	truck	12.0	3.9
van	4.5	2.5	truck	9.5	3.2

Step 1: Define groups of vehicles by users. There is a wide variation of vehicle classifications emanating from different government authorities and traffic engineers. Based on their needs, users can define their own vehicle classification scheme by the

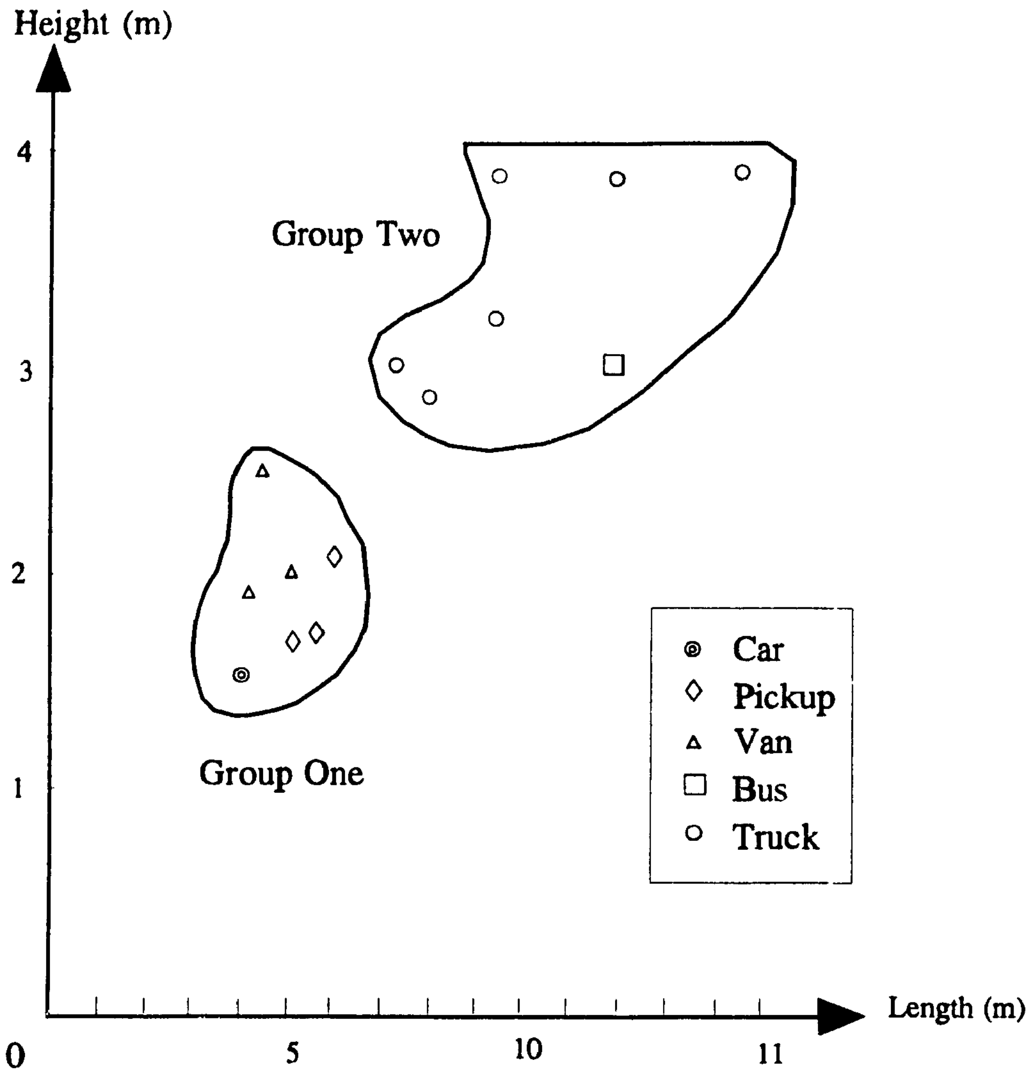


Fig. 5.13 A prototype space

combinations of length and height of vehicles. Suppose two groups are defined in this sample case. Group one includes all types of cars, vans, and pickups while group two includes buses, ambulances, and all types of trucks.

Step 2: Predetermine the value of integer *k*. Methods for choosing *k* can be found in pattern recognition reference (Duda 1973). In this study, $k = 3$ was chosen.

Step 3: Calculate the distance, d_i. Let d_i denote the distance between pattern vectors $\boldsymbol{e}_{new}$ and $\boldsymbol{e}_i$, and be defined as

$$d_i = \sqrt{(L_{new} - L_i)^2 + (h_{new} - h_i)^2} \qquad i = 1, 2, ..., m_1 \qquad (16)$$

Suppose there is an unknown vector $\boldsymbol{e}_{new}$ = (7.0, 2.5) representing a detected vehicle. The distance from $\boldsymbol{e}_{new}$ to every pattern vector $\boldsymbol{e}_i$ is calculated and shown in Table 5.5.

TABLE 5.5 Distance from e_{new}(7.0, 2.5) to every e_i

Vector e_i (1)	Distance d_i (2)	Group No. (3)	**Vector e_i** (1)	Distance d_i (2)	Group No. (3)
(4.0, 1.5)	3.2	1	(12.0, 3.0)	5.0	2
(6.0, 2.1)	1.1*	1	(7.4, 3.0)	0.6*	2
(5.1, 1.7)	2.1	1	(8.0, 2.8)	1.0*	2
(5.6, 1.8)	1.6	1	(9.5, 3.9)	2.9	2
(4.3, 1.9)	2.8	1	(15.0, 3.9)	8.1	2
(5.1, 2.0)	2.0	1	(12.0, 3.9)	5.2	2
(4.5, 2.5)	2.5	1	(9.5, 3.2)	2.6	2

* The three nearest neighbours of the unknown vector e_{new}(7.0, 2.5).

Step 4: Select the k smallest d_is. The values of d_is are compared, and the k smallest d_is are selected. Then k corresponding pattern vectors in the prototype space to these smallest d_is are found. These k pattern vectors may come from different subsets. Each of these k pattern vectors is assigned a name of the corresponding subset. In the sample case, the values of d_is shown in Table 5.5 are compared and three (k = 3) smallest ones are selected as 0.6, 1.0, and 1.1. Then the three corresponding pattern vectors, (7.4, 3.0), (8.0, 2.8), and (6.0, 2.1) in the prototype space to these three smallest d_is are found as the nearest neighbours of the unknown vector e_{new}(7.0, 2.5). This is shown in Table 5.5 and Fig. 5.14.

Step 5: Determine to which group e_{new} belongs. The vector e_{new} is classified into the subset most frequently named among the k pattern vectors. In this case, e_{new} is classified into group two because there are two of its three nearest neighbours coming from group two. But if two or more subsets are named with the same number of pattern vectors in these k nearest neighbours, then the sums of their corresponding distances are calculated and compared. If only one subset exists having the smallest sum of such distances, then e_{new} belongs to that subset. Otherwise, e_{new} is unidentified.

Second level classification: Classifying vehicles by the profile characteristics. Based on the profile characteristics of a vehicle, the algorithm can differentiate buses from trucks, single unit trucks from trailer trucks, and single trailer trucks from multi-unit trailer trucks in this level of classification. Details of the algorithm are as follows.

Step 1: Differentiate buses from trucks. The profile shape of the front part of a vehicle is used to separate buses from trucks. If a vehicle in a group that includes both buses and trucks is found to have not only a similar size as a bus but also a flat front, then it is classified as a bus (note that school buses were excluded in this study). Otherwise, the vehicle is classified as a truck with a projecting front.

Step 2: Differentiate trailer trucks from single unit trucks. The number of units in a truck is used in this separation. If a truck is found to be in the group that contains

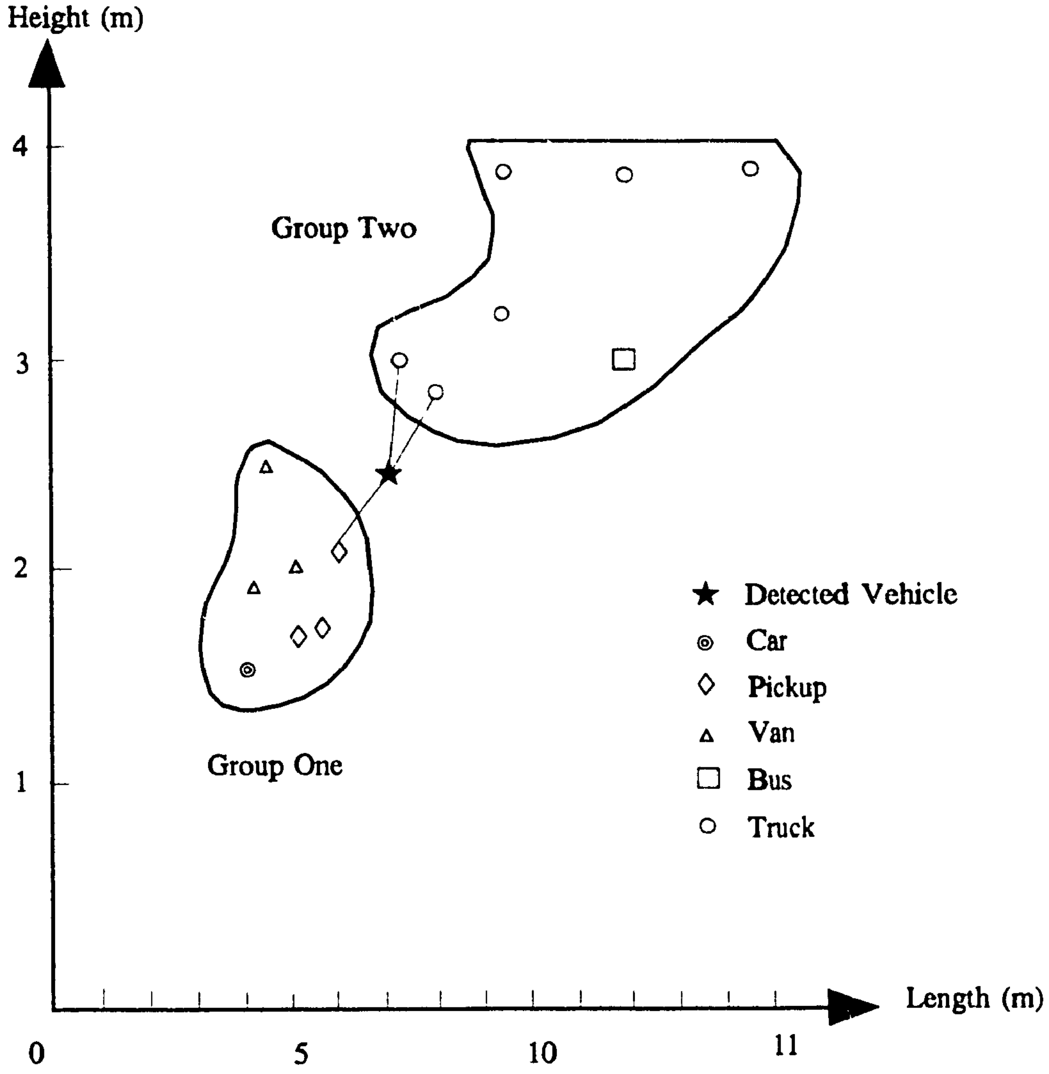


Fig. 5.14 The nearest neighbours of the unknown vector

single unit trucks and trailer trucks and has two or more than two units, then it is classified as a trailer truck. Otherwise, if a truck is found to have only one unit, then it is classified as a single unit truck.

Step 3: Differentiate single trailer trucks from multi-trailer trucks. If a vehicle is classified as a trailer truck in step 2, the number of units will play a major role in further classification. If a trailer truck has two units, then it is classified as a single trailer truck. Otherwise, if the number of units of a trailer truck is more than two, it is classified as a multi-unit trailer truck.

Although the TTC model is ready to classify vehicles by the number of axles, the classifier algorithm does not intend to detect the number of axles directly from images because it may be impossible for computer vision techniques to detect the number of axles directly under occlusion situations. However, the number of axles may approximately be estimated using vehicle length and number of units.

5.5 EXPERIMENTAL RESULTS

5.5.1 Experiments and analysis

The four newly developed models were tested in the laboratory and their accuracies were evaluated. The traffic scene of Decarie Expressway in Montréal was videotaped in different seasons of 1992 and 1993. The video tapes were played back in the laboratory, and 580 vehicles with known sizes were selected from the tapes. The experimental results are presented below.

TABLE 5.6 Results of vehicle size measurement

	Sample size	Mean of errors	Std. deviations
Length	580	0.3	0.04
Width, height	580	0.2	0.03

Accuracy of length measurement

The selected vehicles were detected by the LM model. As a result of comparison of the measured sizes with the actual ones, it was found that the mean of errors in length measurement was 0.3m with standard deviations of 0.04m (see Table 5.6). Error analysis indicated that most errors occurred in vehicle boundary detection. For example, if the boundary of a vehicle is mis-detected with one pixel error on the video image plane, then its length will be miscalculated with an error of (25m/512) 0.05m under the condition that the length of the detection area is 25m and the resolution of the video image in a horizontal direction is 512 pixels. Therefore, it was concluded that the resolution of video images will effect the accuracy of length measurement. Furthermore, vehicle speed is another important facet affecting accuracy. This is because the boundary of vehicles becomes fuzzier in digitized video images at high speed, leading to more errors in boundary detection.

In order to reduce error in length measurement, two aspects of the image digitizer need improvement. First, the speed of digitizing should be increased in order to sharpen the edges of vehicles. Second, the resolution of the digitizer should be higher to reduce system errors.

Accuracy of width and height estimation

The 580 selected vehicles were also utilized in the test of the WHE model. The results showed that the mean of errors in estimation of width and height was 0.2m with standard deviations of 0.03m (see Table 5.6). This accuracy is good enough to meet the requirement of vehicle classification because no precise height is necessary for vehicle classification on the basis of the KNN rule. For example, a height estimation of 1.2m for a passenger car is regarded as the same as an estimation of 1.4m for the purpose of separating this car from a van which has a height of at least 1.6m.

TABLE 5.7 Results of profile characteristic extraction

	Sample size	Right	Wrong	Accuracy
Front shape	648	612	36	94%
Number of units	648	635	13	98%

Accuracy of profile characteristic extraction

The PCE model was also tested by means of detecting those 580 vehicles and another 68 selected trucks including both trailer trucks and single unit trucks. The experimental results (see Table 5.7) showed that 612 vehicles out of 648 were correctly classified into either the projecting-front or flat-front group. Therefore, the accuracy was about 94%. Error analysis indicated that the selection of the value of threshold β_l is very important. Although there is not any fixed value of β_l that can fit all kinds of vehicles, a properly selected value is able to effectively increase the accuracy of the PCE model. The experiments indicated that $\beta_l = 0.7$ was suitable for most kinds of vehicles chosen from the videotape for the test.

The experimental results also showed that the accuracy for counting the number of units was about 98% if the speed of vehicles was less than 70 km/h. But the accuracy was reduced when the speed of vehicles was over 70 km/h. This is due to the constraints of the hardware and software. Although the digitizer worked at a rate of 30 frames per second, the actual rate of image input to the PCE model was only less than 20 frames per second due to time consumption in image processing. As assumed previously, the space between the power unit and the following trailer unit of trailer trucks is only about 1m, but a vehicle travelling at a speed of 70 km/h may move about 0.97m during the time interval of taking two consecutive images (about 1/20 second). Thus, in such a situation, the space behind the power unit of a trailer truck is likely to be missed in detection. To increase the accuracy of the PCE model when the speed of vehicles is in the range of 100-150 km/h, more advanced hardware and software tools should be used in order to meet the speed ιcquirement.

TABLE 5.8 Results of the tree type classifier

	Sample size	Right	Wrong	Accuracy
First level	280	254	26	90%
Second level	30 (bus)	28	2	93%
	68 (truck)	64	4	94%

Results of classification

Table 5.8 shows the experimertal results of the tree type classifier. In order to test the first level classification, three vehicle groups were defined. Group I included two-axle vehicles such as passenger cars, vans, pickups, ambulances and single unit trucks; in group II were buses and recreation vehicles and motor homes; and group III single unit trucks, single trailer trucks and multi-trailer trucks. In each group, 10 vehicles of different types were selected to form a prototype space. Another 280 vehicles randomly chosen from the 580 vehicles were classified by the *k*NN method. The results showed that 254 vehicles out of the 280 were correctly classified. Therefore, the accuracy of the first level classification was over 90%. Most errors occurred due to the size overlap that happened among vehicles in groups II and III. For instance, 18 recreation vehicles in group II were classified into group III because their lengths and heights were similar to single unit trucks in Group III. Perhaps a more acute division of groups would reduce the size overlap and increase the accuracy.

The second level classification was tested with separating vehicles in group II into buses and trucks and classifying trucks in group III into three categories namely one-unit, two-unit, and three-unit trucks. In the experiment, 28 buses out of 30 in group II were differentiated from trucks and 64 trucks out of 68 in group III were correctly classified by the number of units. Error analysis indicated that when the vehicles moved at speed of over 70 km/h their profile characteristics usually cannot be extracted by the PCE model due to the constraints of the low hardware speed. Therefore, errors occurred. For instance, four single trailer trucks that consisted of two units, one of which was a power unit and the other was a trailer unit were misclassified as single unit trucks whose power part and trailer part were on a single frame.

5.5.2 FHWA Classification

The Federal Highway Administration (FHWA) has classified vehicles into 13 categories (*Traffic 1985*). That classification scheme was based on whether the vehicle is a passenger-carrying or non-passenger vehicle. Non-passenger vehicles were further subdivided by number of axles and number of units including both power and trailer units. The FHWA classification scheme with definitions is described in the following table.

TABLE 5.9 FHWA classification scheme

Categories	Type	Description
1	Motorcycles (Optional)	All two- or three-wheel motorized vehicles. This category includes motorcycles, motor scooters, mopeds, motor-powered bicycles, and three-wheel motorcycles.
2	Passenger cars	All sedans, coupes, and station wagons manufactured primarily for the purpose of carrying passengers and including those passenger cars pulling recreational or other light trailers.
3	Other two-axle, four-tire single unit vehicles	All two-axle four-tire vehicles, other than passenger cars. Included in this category are pickups, panels, vans, and other vehicles such as campers, motor homes, ambulances, hearses, and carryalls. Other two-axle, four-tire single unit vehicles pulling recreational or other light trailers are included in this category.
4	Buses	All vehicles manufactured as traditional passenger-carrying buses with two axles and six tires or three or more axles. This category includes only traditional buses (including school buses) functioning as passenger-carrying vehicles. All two-axle, four-tire minibuses should be classified as other two-axle, four-tire single unit vehicles. Modified buses should be considered to be a truck and be appropriately classified.

5	Two-axle, six-tire, single unit trucks	All vehicles on a single frame including trucks, camping and recreation vehicles, motor homes, etc., having two axles and dual rear wheels.
6	Three-axle single unit trucks	All vehicles on a single frame including trucks, camping and recreation vehicles, motor homes, etc., having three axles.
7	Four or more axle single unit trucks	All trucks on a single frame with four or more axles.
8	Four or less axle single trailer trucks	All vehicles with four or less axles consisting of two units, one of which is a tractor or straight truck power unit.
9	Five-axle single trailer trucks	All five-axle vehicles consisting of two units, one of which is a tractor or straight truck power unit.
10	Six or more axle single trailer trucks	All vehicles with six or more axles consisting of two units, one of which is a tractor or straight truck power unit.
11	Five or less axle multi-trailer trucks	All vehicles with five or less axles consisting of three or more units, one of which is a tractor or straight truck power unit.
12	Six-axle multi-trailer trucks	All six-axle vehicles consisting of three or more units, one of which is a tractor or straight truck power unit.
13	Seven or more axle multi-trailer trucks	All vehicles with seven or more axles consisting of three or more units, one of which is a tractor or straight truck power unit.

Due to several constraints, this study was not intended to determine the accuracy of the classifier algorithm to classify vehicles into a set of FHWA (the Federal Highway Administration) classifications. The main constraints in this study include: (*1*) A limited number of vehicle types with vehicle sizes were known; and (2) no predetermined set of vehicle classifications. However, the capability of the classifier algorithm to classify vehicles into the FHWA classification scheme is discussed below.

1. For the first category -- motorcycles (optional) -- this study did not have enough samples to make a definite claim. However, because the LM model is able to reliably obtain length of motorcycles and WHE model is able to estimate approximate the height and width of motorcycles, the TTC model may be able to differentiate motorcycles from oth^er small cars by considering width as one of classifying parameters.

also due to the significant difference between motorcycles and other vehicles, .

2. For passenger cars, the experiments showed that the TTC model is able to separate passenger cars from most other vehicles in the third category due to their lower height and profile characteristic of projecting front. The errors mostly occurred when differentiating passenger cars from small pickups with a similar height.

3. Although the TTC model is able to differentiate small vehicles in the third category such as vans and pickups from vehicles in the fifth category, it will not separate other single unit vehicles in the third category from the single unit trucks in the fifth category due to their similar heights and lengths.

4. Buses defined in the fourth category can be reliably distinguished by the TTC model. This is because the WHE model and the PCE model provide the information about height and profile characteristics for classification. Other available detector systems cannot properly differentiate buses from trucks due to lack of such information.

5. The FHWA scheme divided trucks and trailers into categories ranging from the fifth to the thirteenth category by the number of axles and number of units. Although the number of units can be obtained by the PCE model, the number of axles is not directly measured by the algorithm. However, the number of axles could be estimated by both

the number of units and overall length of vehicles. Therefore, the TTC model is able to classify vehicles among these categories. This study did not have enough samples to make a definite claim.

5.6 SUMMARY

In this chapter, four models have been proposed to form an automatic vehicle classifier. These models are: (1) the LM model for vehicle length measurement; (2) the WHE model for width and height estimation; (3) the PCE model that is able to extract two important profile characteristics of vehicles; and (4) the tree type classifier (TTC model). With the information obtained by the LM, WHE and PCE models, the TTC model classifies vehicles into user-defined categories by performing two levels of classification. At the first level, vehicles are separated by using height and length as classification parameters. At the second level, vehicles are further differentiated on the basis of their profile characteristics. Although the TTC model is presently a two-level classifier, its structure permits expansion to a multi-level classifier if more detailed information about vehicles is available.

The models were tested in the laboratory by means of measuring the sizes of 580 videotaped vehicles, extracting profile characteristics of 648 vehicles, and classifying 280 vehicles into a user-defined category. The performance of the models was satisfactory to meet all the functional and performance requirements.

There are two important functions of the new models. The first comes from the WHE model that provides an effective way to estimate width and height of vehicles. This capability is not available from other detector systems. The second is that the PCE model provides an approach to obtaining two important profile characteristics of vehicle. With these characteristics, not only can buses be differentiated from trucks, but also vans can be separated from cars.

CHAPTER 6

PEDESTRIAN ALGORITHM

6.1 INTRODUCTION

Statistics of pedestrian accidents show that 52.5 percent involve pedestrians crossing or entering the street at or between intersections (National Safety Council, 1986). Pedestrian control at intersections is one of the important ways to reduce the number and severity of traffic accidents involving pedestrians. On the other hand, pedestrian control is also needed to prevent reduction in intersection capacity. For example, pedestrian control at unsignalized intersections can prevent considerable reduction of vehicular capacity caused by a steady stream of pedestrians preempting crosswalks. At signalized intersections, special pedestrian signals can reduce the congestion caused by conflicts between vehicular turning movements and pedestrians.

In order to help bring pedestrian control to reality, pedestrian flow data are essentially demanded. Pedestrian flow data consist of characteristics such as volume, density, speed, and direction. Pedestrian volume, defined as the number of people passing a perpendicular line of sight across the width of a walkway during a specified period of time, is the most important when considering pedestrian control at intersections. This chapter will mainly concentrate on pedestrian volume measurement. Currently, measurement of pedestrian volume is often performed manually. Manual counting is expensive and not suited to a large volume of pedestrians. Application of computer vision techniques leads an effective way to collect automatically pedestrian volume data at traffic intersections. A literature review indicated that a full study in this field has not been performed. One of few studies was reported by Hwang and Takaba (1983). They placed a number of detection points on the surface of a path. Using image analysis techniques, they counted the number of pedestrians walking in a common direction under the assumption that some separation exists between the pedestrians. Instead of detection points, Lu et al (1990) proposed a different algorithm by laying a white grid over the observation surface. The number of pedestrians was obtained by measuring the number of black objects and their sizes in the processed image. The walking direction of pedestrians was determined by comparing two sequential images under the assumption that pedestrians walk either in a northbound or southbound direction.

Although the studies of Hwang (1983) and Lu (1990) were pioneering, the algorithms developed for measurement of pedestrian volume data at intersections were still limited. This is because the previous algorithms assumed that pedestrians were the only moving objects on the scene and moved in a certain direction. They did not study the other moving objects likely to appear within the detection area. The situation at an intersection is much more complicated because not only are there pedestrian movements but also vehicular ones at intersections. Furthermore, the directions of pedestrian movements are likely to vary on the crosswalk. Therefore, extraction of pedestrian information from scenes at intersections becomes more difficult.

This chapter will propose a new algorithm used in a computer vision system for measurement of pedestrian volume. From section 6.2 though 6.5, the details of each procedure used in the algorithm are presented respectively. These procedures are: (1) extraction of moving object from video image, (2) fine noise removal, (3) distinguish pedestrian from vehicle, and (4) pedestrian number estimation and volume calculation. With the integration of these procedures, the new algorithm is proposed in Section 6.6. In the same section, the experimental results and analysis are discussed. The new algorithm is able to measure pedestrian volume at intersections, which means that the new algorithm is capable of measuring multidirectional flow on a crosswalk without being affected by vehicles passing through the detection spot.

6.2 EXTRACTION OF MOVING OBJECT FROM VIDEO IMAGE

Pedestrians are not the only moving object in the scenes of an intersection. In order to extract pedestrians from the video images, all the moving objects must be extracted. There are alternative approaches to extracting moving objects from the scenes of intersections. The first one is based on the *color detection functions*, f_D (Equation 2.7) and the second is based on *image subtraction* (Equation 2.1).

1. Color detection method

If an object moves on a uniform-color asphalt pavement, then it can be extracted from the video image by using the *color detection* method.

Let $p(x, y, t)$ denote a pixel with a vector value of $\boldsymbol{P}(p) = (R(p), G(p), B(p))$ in an R-G-B image . Let $P_B(p)$ denote a binary value of pixel p after the following operation:

Step 1. Determine the color of the pavement surface by using the method introduced in Section 2.3.

Step 2. Calculate the value of color detection function (Equation 2.7):

$$f_D(p) = f_R(p) + f_G(p) + f_B(p)$$

where

$f_D(p)$ = *color detection function* which has four values: 0, 1/3, 2/3, and 1.

$f_R(p)$ = match index for value of $R(p)$;

$f_G(p)$ = match index for value of $G(p)$;

$f_B(p)$ = match index for value of $B(p)$;

To determine the match indexes, refer to Equation (2.6).

Step 3. Determine the binary value, $P_B(p)$.

$$P_B(p) = \begin{cases} 1, & if\ \ f_D(p) < \dfrac{2}{3} \\ 0, & if\ \ f_D(p) \geq \dfrac{2}{3} \end{cases} \tag{6.1}$$

After the above operation, if pixel p has the value of $P_B(p) = 1$, then it belongs to a moving object. Otherwise, if pixel p has the value of $P_B(p) = 0$, then it belongs to the background.

2. Image Subtraction Method

If an object does not move on the uniform-color pavement surface, then the *color detection function* will no longer work. Our previous work (Lu, Yuan and Yan 1991) showed that the *image subtraction* method will be an alternative in such situations. A binary image which shows moving object can be obtained by means of *image subtraction* as

$$P_B(p) = \begin{cases} 1 & if \;\left| \dot{P}(p) - P'(p) \right| > \varepsilon \\ 0 & otherwise \end{cases} \tag{6.2}$$

where

P(p) = value of pixel *p* in a *measured image*;

P'(p) = value of pixel *p* in the *reference image*;

ε = a predetermined threshold to determine whether a pixel belongs to moving object or not.

Note that the value of ***P*** could be an integer value for grey-scale images or a vector for color images. To process a grey-scale image, the difference of *P(p)* and *P'(p)* can be obtained directly by comparing their grey levels. Otherwise, to process a color image, the *tristimulus values* of a pixel should be transformed as:

$$P(p) = w_1R(p) + w_2G(p) + w_3B(p) \tag{6.3}$$

where

$\boldsymbol{w} = (w_1, w_2, w_3)$ is the optimal projection transformation. In order to save computation time, it is not necessary to process the whole image. Instead, only the pixels on a line that is perpendicular to the crosswalk are chosen to be analyzed. This pixel line is horizontal in the video image as shown in Fig. 6.1.

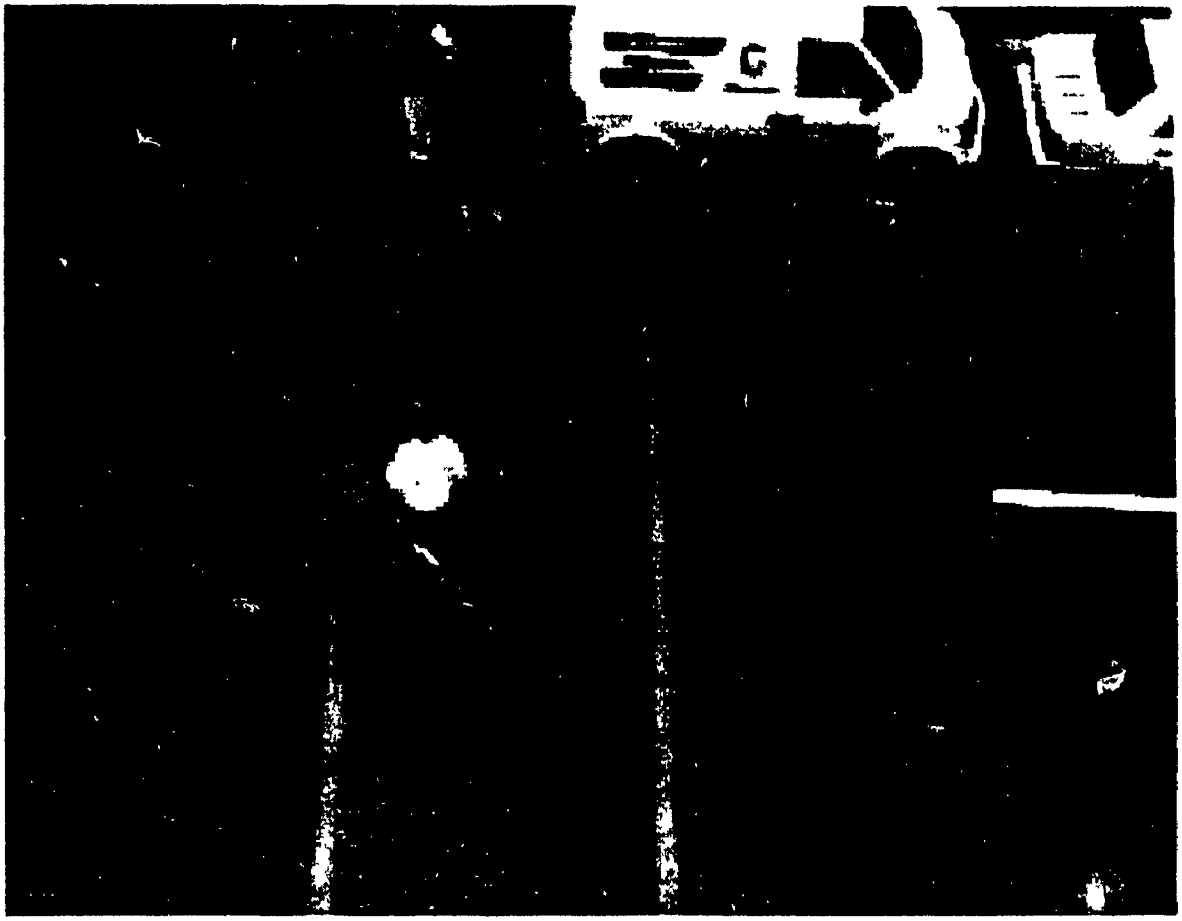


Fig. 6.1 A scene of intersection

6.3 FINE NOISE REMOVAL

A *fine noise* is usually an isolated point which has nonnoise neighbours. To detect the fine noises in a oixel line, a 3-pixel scanning window, as a special case of 3 x 3 window defined in Chapter 2, is utilized. This 3-pixel scanning window includes the scanning pixel $p = (x, y)$ and its 2-neighbour pixels, say $p_1 = (x-1, y)$ and $p_2 = (x+1, v)$, where y is a fixed constant determining where the pixel line is located in the image.

Let N_2 denote a set of neighbours of pixel $p(x, y)$, and

$$N_2(p) = \{p_1, p_2\} \tag{6.4}$$

where

$$\begin{bmatrix} p \\ p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} (x,\ y) \\ (x-1,\ y) \\ (x+1,\ y) \end{bmatrix} \tag{6.5}$$

If the following condition exits,

$$\begin{gathered} [P_B(p) = 1] \cap [\forall_i(P_B(p_i) = 0)], \\ p_i \in N_2(p). \end{gathered} \tag{6.6}$$

then the pixel p is determined as a fine noise and eliminated by setting $P_B(p) = 0$.

6.4 DISTINGUISH PEDESTRIAN FROM VEHICLE

Since a vehicle occupies more space in the detection area than a pedestrian, the number of adjacent pixels representing a vehicle must be larger than that representing a pedestrian. Therefore, a pedestrian can be distinguished by counting the number of adjacent pixels in the binary image. If the number is less than a threshold, say β, then the pixels possibly belong to a pedestrian. The threshold β gives the largest possible width of pedestrians in images. The value of β can be calculated as

$$\beta = Int\ [\ N_x D_P / D_x\ ] \tag{6.7}$$

where

Int = symbol means to round to the nearest integer;

N_x = number of pixels in a row of a frame;

D_P = maximum width of pedestrians, in meters;

D_x = real distance, in meters, covered by the camera's field of view in a horizontal direction. D_x can be computed by using the perspective projection model (Equation 2.37) as follows:

$$D_x = \frac{H\ k_x\ N_x}{yk_y sin\theta\ +\ cos\theta} \tag{6.8}$$

where

H = distance between the video camera and the *horizontal plane*;

θ = angle of the camera mount tilt ($-\pi/2 < \theta < 0$);

k_x, k_y = coefficients that are determined by the following equations:

$$k_x = \frac{2tan\frac{\phi_x}{2}}{N_x} \qquad k_y = \frac{2tan\frac{\phi_y}{2}}{N_y} \tag{6.9}$$

where

ϕ_x , ϕ_y = view angles of the camera in X and Y directions, respectively, ($0 < \phi_x$, $\phi_y < 2\pi$).

The width of a moving object in the image is determined by counting the number of adjacent pixels that are equal to 1. Let (a, y) and (b, y) denote respectively the first and last pixel equalling 1. The width of the moving object is W, and

$$W = | b - a | + 1 \tag{6.10}$$

Compare the values of W and β, and remove the vehicles from the image according to the result of the comparison:

if $W \leq \beta$, then the segment belongs to pedestrian; otherwise,

if $W > \beta$, then the segment belongs to vehicle.

In the above operation, all big segments of moving objects are regarded as vehicles and removed from the image. This simple threshold method may cause error when pedestrian density is high in the image. For instance, when the pedestrians are so closely passing one another that no space could be found between them in an image, the pixels belonging to these pedestrians may be removed from the image if the width of the pixel segment is larger than the threshold β. But this will not affect the calculation of pedestrian volume because the pedestrians will be counted according to an image sequence, not only one isolated image. Therefore, in an image sequence, there are images that show the same group of pedestrians with spaces appearing between them.

6.5 ESTIMATION OF PEDESTRIAN VOLUME

Suppose the detection line is composed of n pixels, $p_1, p_2, ..., p_n$. The values of these pixels can form a vector v_p, and

$$v_p = \begin{bmatrix} P_B(p_1) \\ \vdots \\ P_B(p_i) \\ \vdots \\ P_B(p_n) \end{bmatrix} \quad \circ \tag{6.11}$$

where

P_B = binary value of the pixel p_i, $P_B(p_i) = 0$ or 1.

If there are segments of pixels that equal 1 in a binary image, the center of each segment can be easily found. Suppose that points $(a_1, b), (a_2, b), \ldots, (a_q, b)$ are the centers of the q segments in an image and

$$(a_1, b), (a_2, b), \ldots, (a_q, b) \in \{ p_1, p_2, \ldots, p_n \} \tag{6.12}$$

Another vector l in which only the values of pixels $(a_1, b), (a_2, b), \ldots, (a_q, b)$ remain equalling 1 can be obtained from vector v_p by the following transformation:

$$l = E\, v_p = \begin{bmatrix} e_{11} & \cdots & e_{1n} \\ \vdots & e_{ij} & \vdots \\ e_{n1} & \vdots & e_{nn} \end{bmatrix} \begin{bmatrix} P_B(p_1) \\ \vdots \\ P_B(p_n) \end{bmatrix} \tag{6.13}$$

where

l = vector that records the centers of the pedestrian segments;

v_P = vector that records the segments of pedestrians;

E = transfer matrix defined as $E = [\, e_{ij} \,]$, where

$$e_{ij} = \begin{cases} 1 \, , & \textit{if} \;\; i = j \;\; \wedge \;\; i \in \{a_1, a_2, ..., a_n\}; \\ 0 \, , & \textit{others.} \end{cases} \tag{6.14}$$

$$i, j = 1, 2, ..., n \quad (\textit{Number of components in vector } \boldsymbol{P}\ )$$

Since the number of pedestrian passing through the detection line cannot be accurately estimated from an isolated image, therefore, a series of consecutive images must be used. Let *m* denote the number of consecutive images in an image sequence that covers a time interval *T*. For each image, there is a corresponding vector *l(t)*, where *t* = *1, 2 ..., m.* Therefore, during the time interval *T*, the image sequence corresponds to a series of vectors, *l(1), l(2), ..., l(m).* Let *L* denote a matrix that is defined as follows:

$$L = \begin{bmatrix} l^T(1) \\ l^T(2) \\ \vdots \\ l^T(m) \end{bmatrix} = \begin{bmatrix} L(1,1) & \cdots & L(1,n) \\ L(2,1) & \cdots & L(2,n) \\ \vdots & L(x,t) & \vdots \\ L(m,1) & \cdots & L(m,n) \end{bmatrix} \tag{6.15}$$

Matrix *L* records the pedestrian flow during the time period *T*. The number of pedestrians recorded in the matrix can be determined by the following procedure.

(1) Scan matrix *L* from left to right, and row by row, until an element *L(x,t)* that equals 1 is encountered, where *x* = *1, 2, ..., n, t* = *1, 2, ..., m* as shown in Fig. 6.2.

(2) It is obvious that the center of a pedestrian crossing the detection area remains approximately in the same position in every image. Therefore, if *L(x,t)* actually recorded the centre of a pedestrian in image *t*, elements below it in the following rows should have also recorded the centre of the same pedestrian. In other words, information about a pedestrian should be recorded by a group of elements within several consecutive rows of

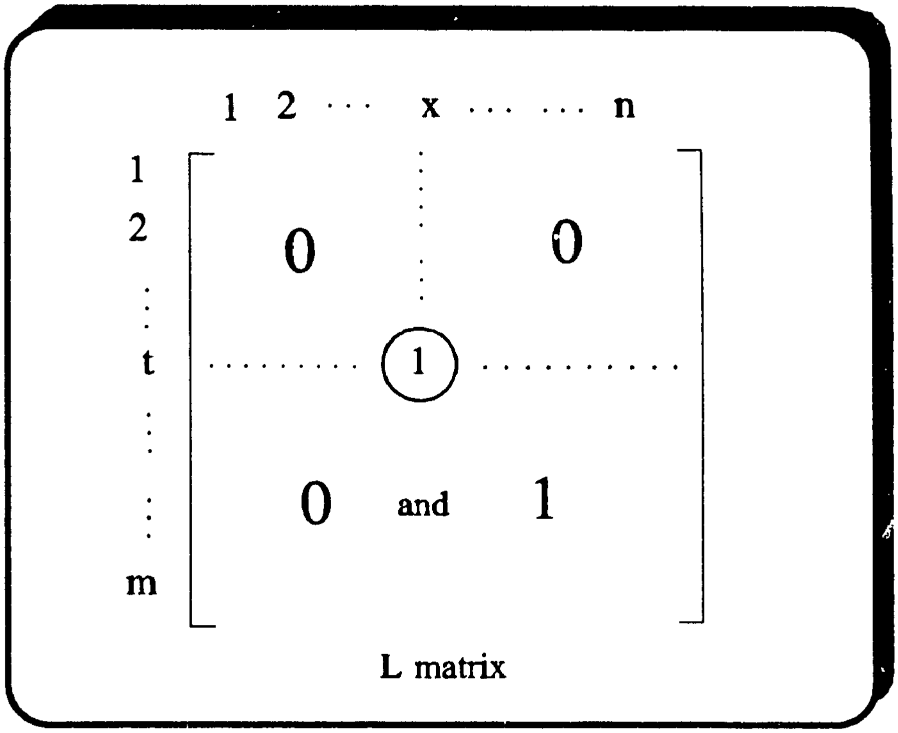


Fig. 6.2 Find first value of 1 in L matrix

matrix L. Hence, as shown in Fig. 6.3, a n_t x n_x-element window was utilized for checking and processing some specific neighbours of element $L(x,t)$.

The number of elements in each row of the window, n_x, is the maximum variation of the centre of a pedestrian. Since it was assumed that pedestrians cross the detection line perpendicularly, the variations of their centres should not exceed this range. n_x can be determined by using the following equation:

$$n_x = IntOdd\ [\ 0.5N_x\ /\ D_x\ ] \qquad (6.16)$$

where

IntOdd = symbol means to round to the nearest odd integer;

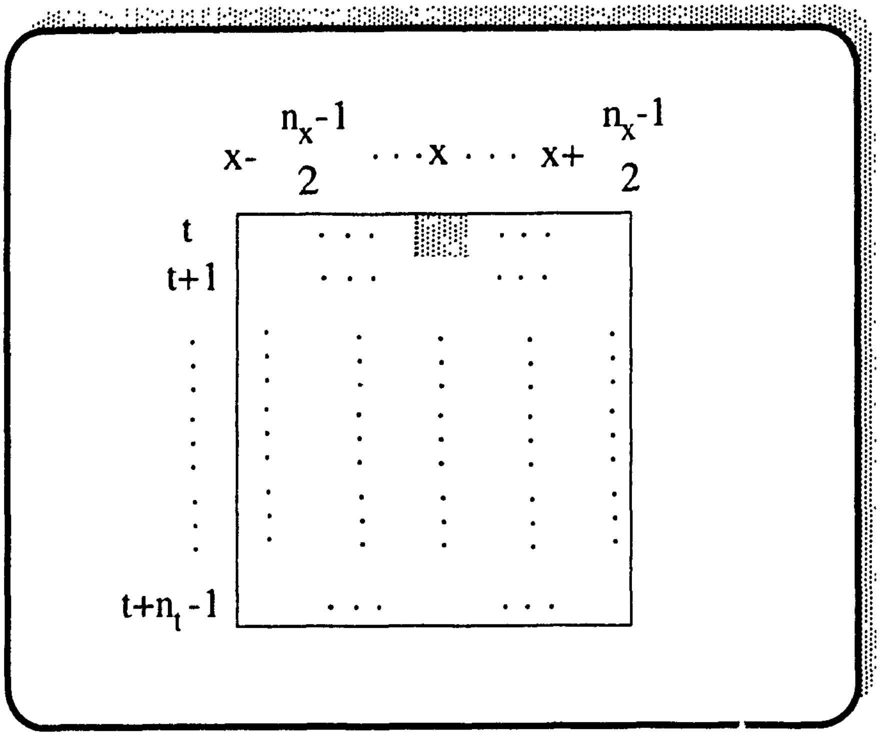


Fig. 6.3 n_txn_x-element window

0.5 = maximum variation of the centre of a pedestrian, in meters;

N_x = number of pixels in a row of a frame;

D_x = real distance, in meters, covered by the detection line and can be computed by using the Equation (6.8).

The number of rows of the window, n_t, should be equal to the number of consecutive images that cover the period of time in which a pedestrian can cross the detection line completely. Refer to Fig. 6.4, n_t can be approximated as

$$n_t = Int\ [\ N_F\ D\ /\ V\ ] \qquad (6.17)$$

where

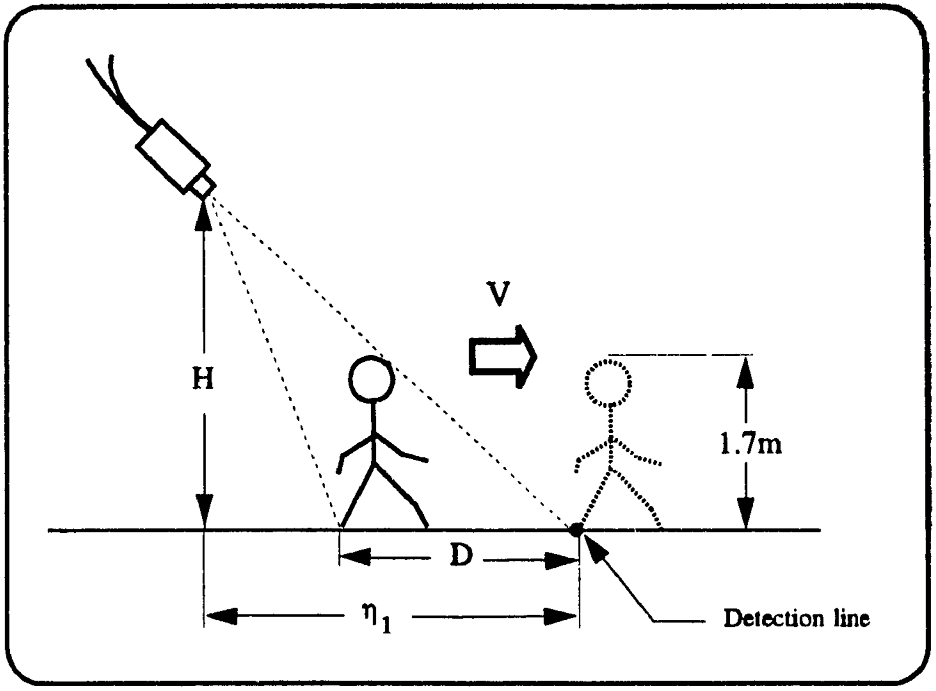


Fig. 6.4 A pedestrian crossing the detection line

N_F = number of frames that can be obtained in a second;

V = average walk speed of pedestrians (m/sec),

D = denotes a distance, in meters, as defined in Fig. 6.4.

From Fig. 6.4, D can be obtained easily by applying similar triangles as

$$D = 1.7\eta_1 / H \tag{6.18}$$

where

1.7 = average height of pedestrians;

H = distance between video camera and the horizontal plane;

η_l = coordinate of the detection line in the horizontal plane.

η_l can be obtained by mapping the detection line from the image plane onto the horizontal plane, and

$$\eta_l = \frac{H(bk_y cos\theta \ - \ sin\theta)}{bk_y sin\theta \ + \ cos\theta} \tag{6.19}$$

where

b = Y-coordinate of the detection line in the image plane;

θ = angle of the camera mount tilt, $(-\pi/2 < \theta < 0)$;

k_x, k_y = coefficients that are determined by using Equation (6.9).

(3) Create a 0-1 vector ***I***, called *index vector*. The value of ***I(j)*** is determined as follows:

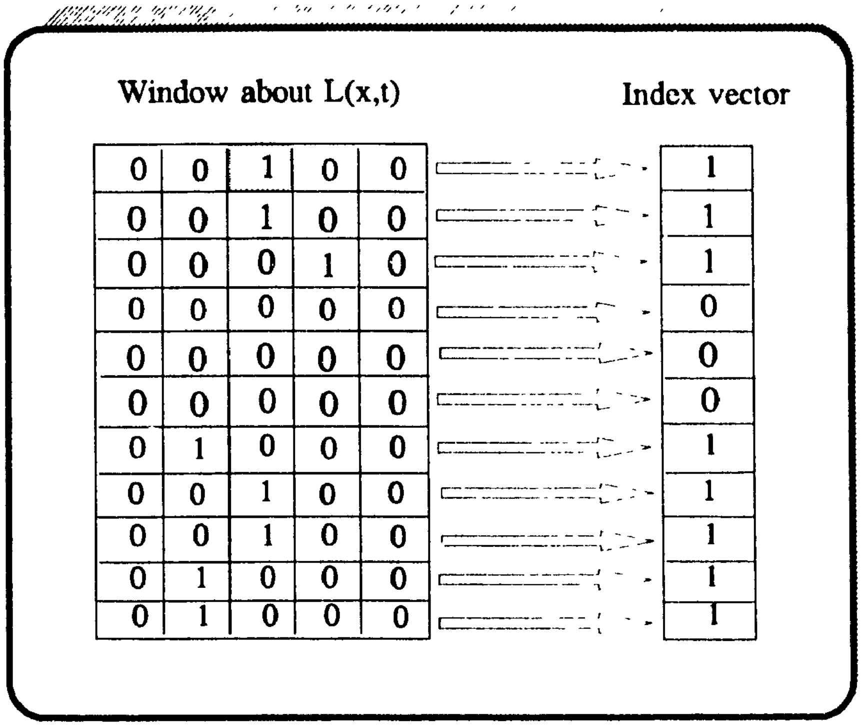
$$I(j) = \begin{cases} 0 \ , & if \ \ \forall_t \ (\ L(i, \ j{+}t) \ = \ 0, \ \ i \ = x{-}\dfrac{n_x{-}1}{2} \ , \ \dots \ , \ x, \ \dots \ , \ x{+}\dfrac{n_x{-}1}{2} \ ) \\ 1 \ , & if \ \ \exists_t \ (\ L(i, \ j{+}t) \ = \ 1, \ \ i \ = x{-}\dfrac{n_x{-}1}{2} \ , \ \dots \ , \ x, \ \dots \ , \ x{+}\dfrac{n_x{-}1}{2} \end{cases} \tag{6.20}$$

$$j \ = 0, \ 1, \ 2, \ \dots, \ n_t{-}1.$$

In other words, if all the elements in row j of the window are equal to zeros, ***I(j)*** is set to zero; otherwise, if any one of the elements in row j is equal to 1, ***I(j)*** is set to 1. Fig. 6.5 depicts an example of this procedure.

(4) As shown in Fig. 6.6, if the number of successive ***I(j)*** = *0* is equal to or less than two, then change the value of ***I(j)***'s from 0 to 1; otherwise no change.

(5) Scan the elements from ***I(0)*** to ***I(n***$_t$***-1)***. If the number of successive ***I(j)=1*** from



0	0	1	0	0	1
0	0	1	0	0	1
0	0	0	1	0	1
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	0	1
0	0	1	0	0	1
0	1	0	0	0	1
0	1	0	0	0	1

Fig. 6.5 Example of creating vector ***I***

$I(0)$ is equal to or greater than a threshold β_l, then add one to an accumulator of pedestrian number, indicating that a pedestrian is found. Meanwhile, set to zero the window elements that correspond to the successive $I(j)=1$ elements. This is shown in Fig. 6.7. Otherwise, if the number of successive $I(j)=1$ from $I(0)$ is less than β_l, then only change window element (x,t) itself from 1 to 0 as shown in Fig. 6.8, meaning that it is a noise. The threshold β_l represents the minimum number of consecutive images in which a pedestrian should appear, and

$$\beta_l = 0.6n_t \qquad (6.21)$$

where

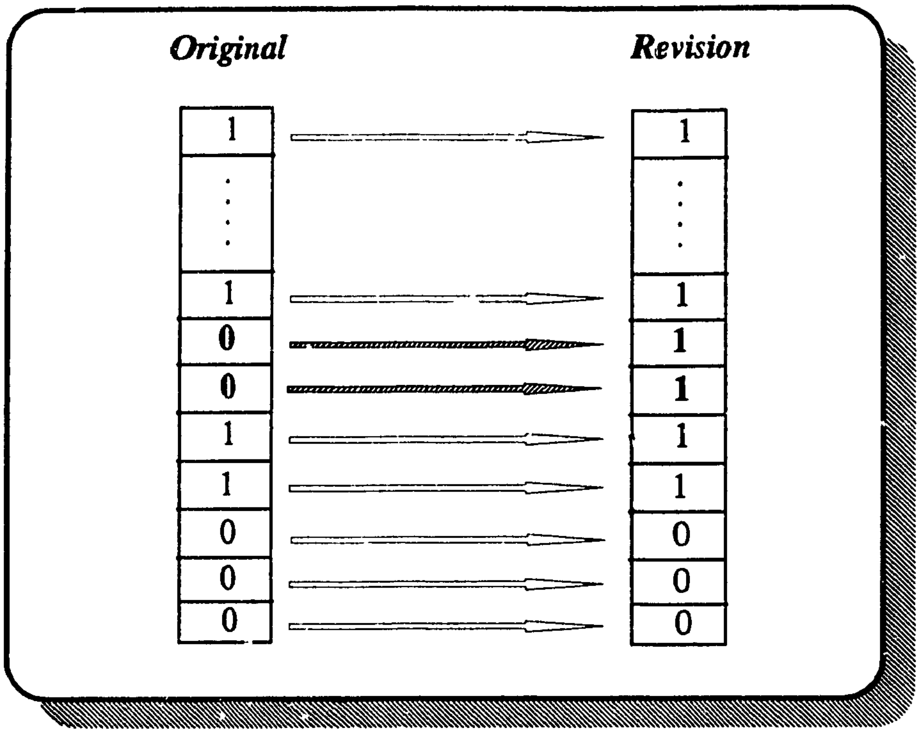
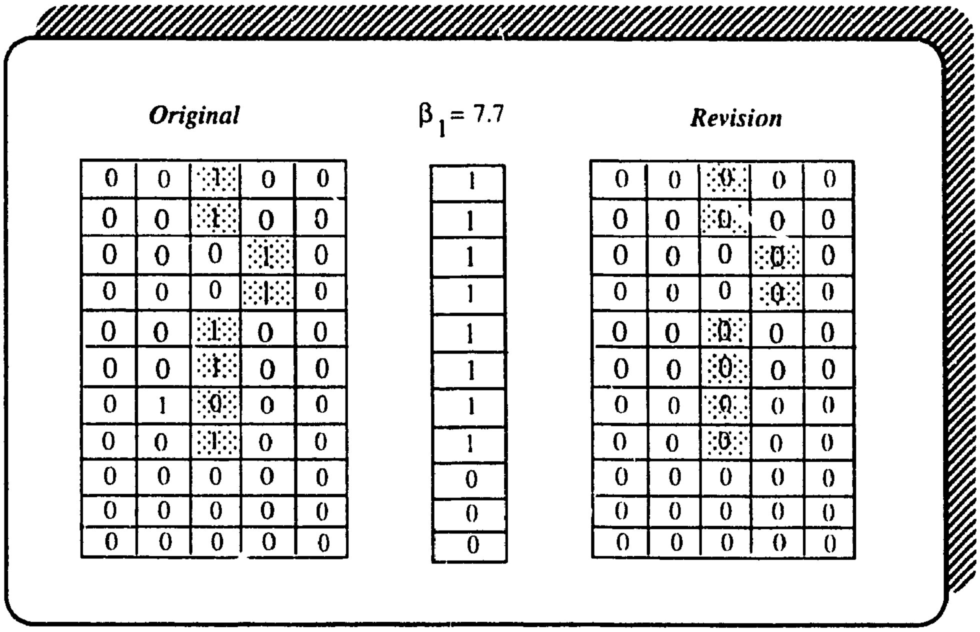


Fig. 6.6 Example of changing the value of $I(j)$

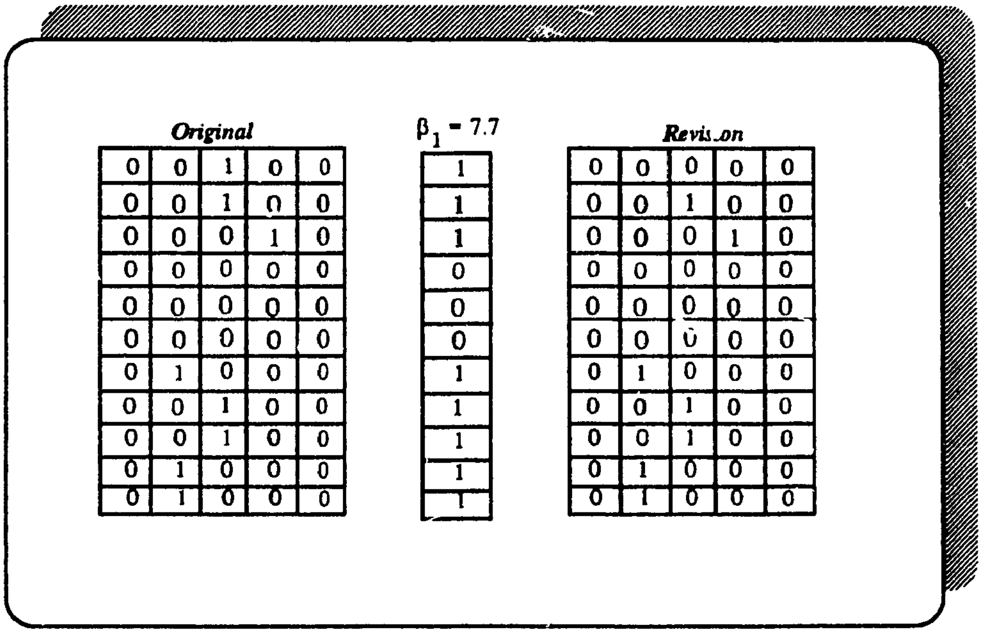
n_r = number of rows of the window and is determined by Equation (6.17).

Figure 6.9 shows an example of the above operation. Suppose an element $L(x,t)$ is found equalling 1 in L matrix (Fig. 6.9a), and a 3 x 10 window (Fig. 6.9b) is used for the processing. According to the values of the elements in the window, an index vector is obtained as Fig. 6.9c. Scanning the index vector, it is found that $I(2)$ which equals 0 has no same-value neighbours. Therefore, the value of $I(2)$ is switched to 1 as shown in Fig. 6.9d. In the revised index vector of Fig. 6.9d, $I(0)$, $I(1)$, ..., $I(5)$ are successively equal to 1. Since the number of successive equal-1 elements is *six* that is equal to β_l (β_l = 0.6 x 10 = 6 in this case), a pedestrian is detected, and all the elements in the first *six* rows of the window are set to 0 as illustrated in Fig. 6.9e. Consequently, the L matrix is updated (Fig. 6.9f).



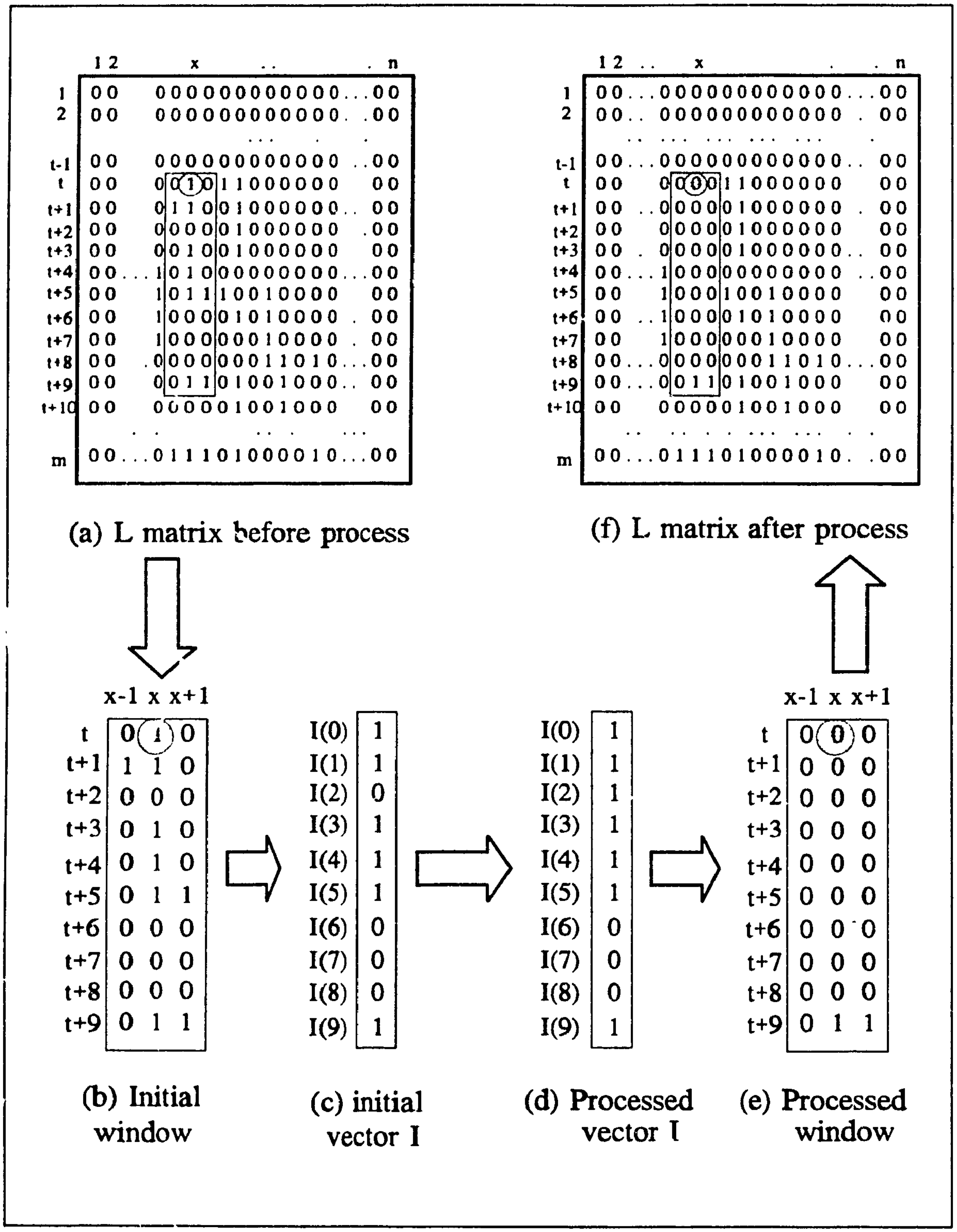
Original					β_1 = 7.7	Revision				
0	0	1	0	0	1	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Fig. 6.7 Example of changing elements in the window



Original					β_1 = 7.7	Revision				
0	0	1	0	0	1	0	0	0	0	0
0	0	1	0	0	1	0	0	1	0	0
0	0	0	1	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	1	0	0	0
0	0	1	0	0	1	0	0	1	0	0
0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	0	1	0	1	0	0	0
0	1	0	0	0	1	0	1	0	0	0

Fig. 6.8 Example of changing elements in the window



```
     1 2       x        ..       .  n
 1    00    0000000000000...00
 2    00    0000000000000. .00
                ...    .    .
t-1   00    0000000000000  ..00
 t    00    00(1)0110000000    00
t+1   00    0110010000000..  00
t+2   00    0000010000000   .00
t+3   00    0010010000000    00
t+4   00...1010000000000...00
t+5   00    1011100100000    00
t+6   00    1000010100000  . 00
t+7   00    1000000100000  . 00
t+8   00   .0000000110l0...00
t+9   00    0011010010000..  00
t+10  00    0000010010000    00
         .  .       ..  .     ...
 m    00...0111010000010...00
```

(a) L matrix before process

```
     1 2 ..    x                 .    . n
 1    00...0000000000000...00
 2    00.. 0000000000000  .  00
         ..  ...  ...  ...  ...     .
t-1   00...0000000000000..  00
 t    00    00(0)0110000000    00
t+1   00  ..0000010000000  .  00
t+2   00    0000010000000    00
t+3   00  . 0000010000000..  00
t+4   00...1000000000000  ..00
t+5   00    1000100100000    00
t+6   00  ..1000010100000    00
t+7   00    1000000100000    00
t+8   00...0000000110100...00
t+9   00...0011010010000    00
t+10  00    0000010010000    00
         ..   ..  .  .  ...  ...   .  .
 m    00...0111010000010.  .00
```

(f) L matrix after process

```
     x-1 x x+1
 t    0 (1) 0
t+1   1  1  0
t+2   0  0  0
t+3   0  1  0
t+4   0  1  0
t+5   0  1  1
t+6   0  0  0
t+7   0  0  0
t+8   0  0  0
t+9   0  1  1
```

(b) Initial window

```
I(0)  1
I(1)  1
I(2)  0
I(3)  1
I(4)  1
I(5)  1
I(6)  0
I(7)  0
I(8)  0
I(9)  1
```

(c) initial vector I

```
I(0)  1
I(1)  1
I(2)  1
I(3)  1
I(4)  1
I(5)  1
I(6)  0
I(7)  0
I(8)  0
I(9)  1
```

(d) Processed vector I

```
     x-1 x x+1
 t    0 (0) 0
t+1   0  0  0
t+2   0  0  0
t+3   0  0  0
t+4   0  0  0
t+5   0  0  0
t+6   0  0 `0
t+7   0  0  0
t+8   0  0  0
t+9   0  1  1
```

(e) Processed window

Fig. 6.9 Example of processing *L* matrix

(6) Continue to scan matrix L from the element (x,t) and return to (1) until all the elements in matrix L have been scanned.

The volume of pedestrians can be calculated as

$$Vol = N / 60T \tag{6.22}$$

w!.ere

Vol = volume of pedestrians, in pedestrians per minute;

T = a period of time, in seconds;

N_p = number of pedestrians counted ir time T.

6.6 ALGORITHM AND EXPERIMENTAL RESULTS

6.6.1 Pedestrian Algorithm

The algorithm developed for pedestrian volume measurement consists of ten steps. A flow chart of the algorithm is shown in Fig. 6.10. The details of each step are described below.

Step 1: Locate a horizontal detection line that includes pixles from point $p_1=(x_1\ ,\ b)$ to $p_n=(x_n\ ,\ b)$ in the images.

Step 2: Input an ımage and extract the values of pixels $\boldsymbol{P(p_1)}$, $\boldsymbol{P(p_2)}$, ..., $\boldsymbol{P(p_n)}$.

Step 3: Extract moving objects from p_1, p_2, ..., p_n and convert their values $\boldsymbol{P(p_i)}$ into binary ones $P_B(p_i)$ by using Equation (5.1) or (6.2).

Step 4: Remove the fine noises in the image by using 3_pixel noise

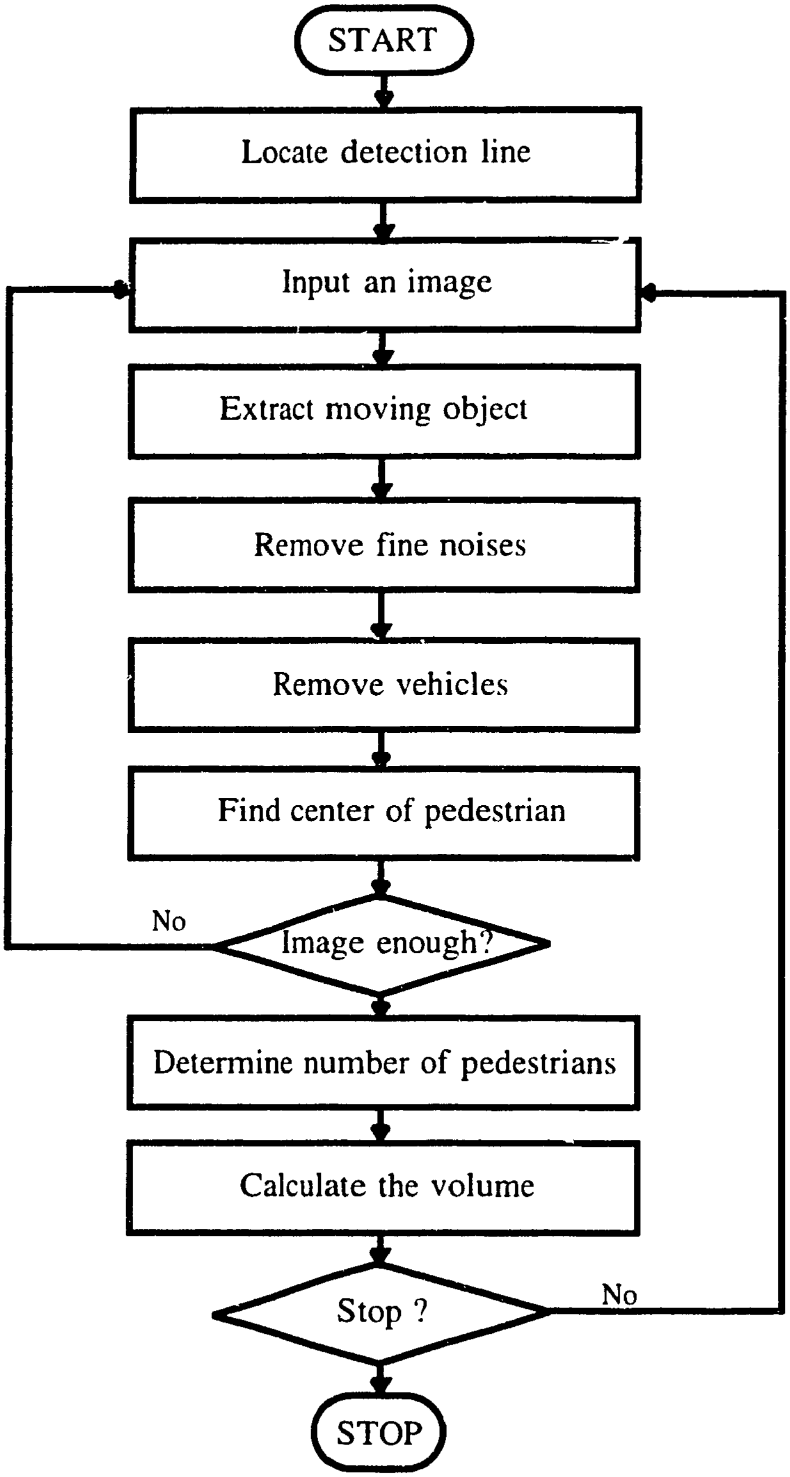


Fig. 6.10 Flow chart of the pedestrian algorithm

removal model (Equation 6.6).

Step 5: Remove vehicles in the image by using models introduced in Section 6.4.

Step 6: Find the center of each pedestrian in the image and store it in a 0-1 vector *l(i)* (See Equation 6.13).

Step 7: If this is not the last image in the sequence, then $i = i + 1$ and return to step 2; otherwise, go to step 8.

Step 8: Determine the number of pedestrians in the time period of T from matrix L, where $L = [l^T(i)]$, $i = 0, 1, ..., m$ (see Section 6.5).

Step 9: Calculate volume of pedestrian flow in time period T by using Equation (6.22).

Step 10: Set $i = 0$ and return to step 2, or stop the procedure under user's instruction.

6.6.2 Experimental Results and Analysis

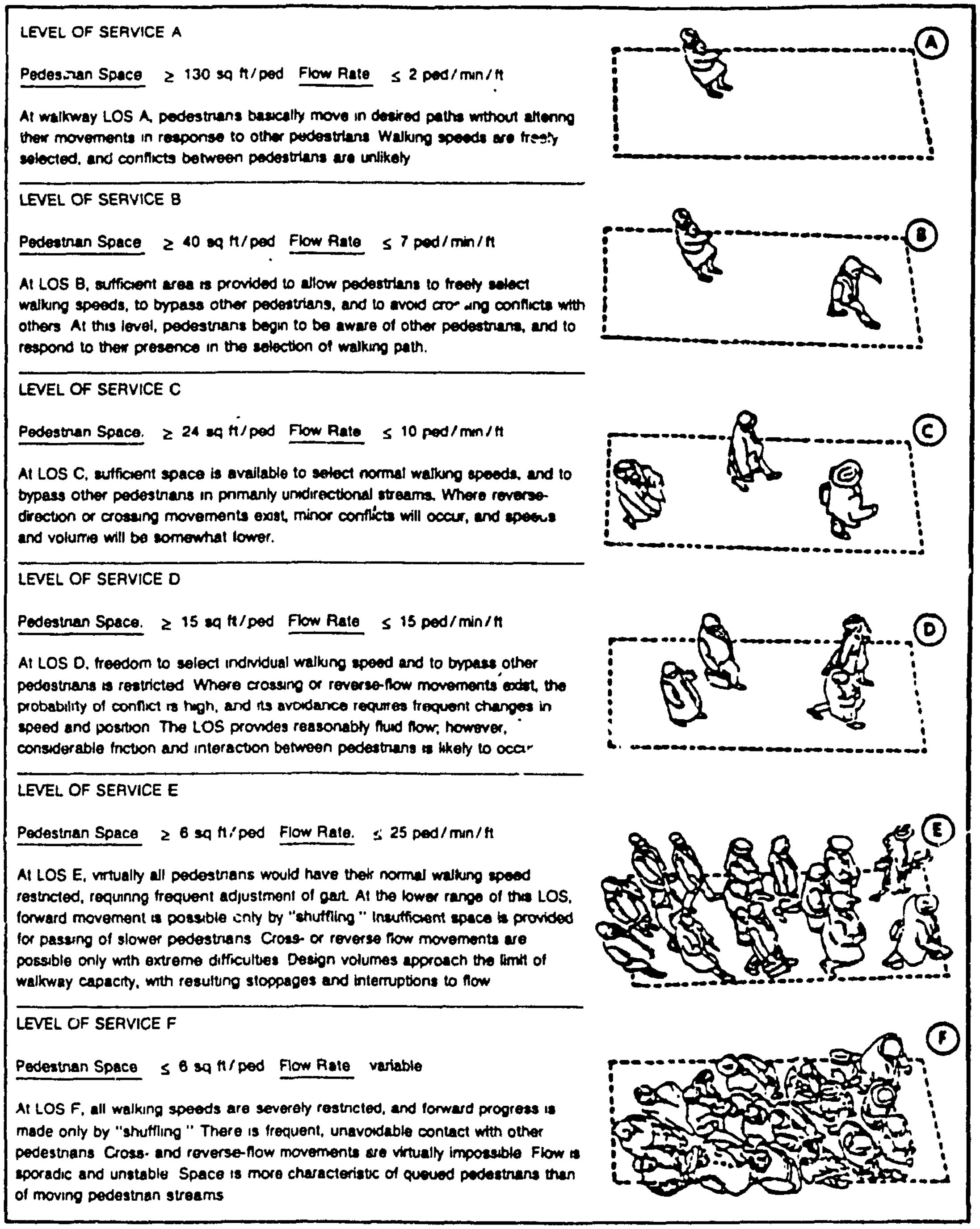
The software for simulation of the above design was written in Turbo PASCAL, and has been tried on an IBM PC/AT 386 computer. In order the verify the algorithm, real traffic and pedestrian situations were recorded at different intersections in downtown Montreal, Quebec, under cloudy weather conditions. These videotaped data included pedestrian flow conditions at every Level Of Service (LOS): A, B, C, D and E, as specified in the *Highway Capacity Manual* (HCM) (1985). In the HCM, average pedestrian space for A, B, C, D and E arc greater than 130, between 40 to 130, between 24 to 40, between 15 to 24, and between 6 to 15 ft^2 per pedestrian, respectively. This is illustrated in Fig. 6.11.

A total of 120 observations have been analyzed in order to estimate the accuracy of the algorithm. These 120 observations were collected from two videotape sessions, each spanning a period of approximate two hours. Each observation was taken during a period of about 45 seconds considering with the green traffic signal light. The actual count of the number of pedestrians in each observation was compared with that predicted by the algorithm and the difference as well as percentage of error were calculated accordingly. The data is presented in Appendix A. A statistical analysis was carried out and the results are presented Table 6.1.

TABLE 6.1 Mean and standard deviation of all samples

		Calculated Errors	
	Sample Size	Mean	Standard Deviation
Session 1	60	7.2%	4.8%
Session 2	60	9.4%	5.5%
Both Sessions	120	8.3%	5.3%

Error analysis indicated that the major reason behind detection false alarms was overlapped pedestrians under high density situation (equivalent to LOS E). In order to estimate errors under different density conditions, the data was divided into three groups



LEVEL OF SERVICE A

Pedestrian Space ≥ 130 sq ft/ped Flow Rate ≤ 2 ped/min/ft

At walkway LOS A, pedestrians basically move in desired paths without altering their movements in response to other pedestrians Walking speeds are freely selected, and conflicts between pedestrians are unlikely

LEVEL OF SERVICE B

Pedestrian Space ≥ 40 sq ft/ped Flow Rate ≤ 7 ped/min/ft

At LOS B, sufficient area is provided to allow pedestrians to freely select walking speeds, to bypass other pedestrians, and to avoid crossing conflicts with others At this level, pedestrians begin to be aware of other pedestrians, and to respond to their presence in the selection of walking path.

LEVEL OF SERVICE C

Pedestrian Space. ≥ 24 sq ft/ped Flow Rate ≤ 10 ped/min/ft

At LOS C, sufficient space is available to select normal walking speeds, and to bypass other pedestrians in primarily unidirectional streams. Where reverse-direction or crossing movements exist, minor conflicts will occur, and speeds and volume will be somewhat lower.

LEVEL OF SERVICE D

Pedestrian Space. ≥ 15 sq ft/ped Flow Rate ≤ 15 ped/min/ft

At LOS D, freedom to select individual walking speed and to bypass other pedestrians is restricted Where crossing or reverse-flow movements exist, the probability of conflict is high, and its avoidance requires frequent changes in speed and position The LOS provides reasonably fluid flow; however, considerable friction and interaction between pedestrians is likely to occur

LEVEL OF SERVICE E

Pedestrian Space ≥ 6 sq ft/ped Flow Rate. ≤ 25 ped/min/ft

At LOS E, virtually all pedestrians would have their normal walking speed restricted, requiring frequent adjustment of gait. At the lower range of this LOS, forward movement is possible only by "shuffling" Insufficient space is provided for passing of slower pedestrians Cross- or reverse flow movements are possible only with extreme difficulties Design volumes approach the limit of walkway capacity, with resulting stoppages and interruptions to flow

LEVEL OF SERVICE F

Pedestrian Space ≤ 6 sq ft/ped Flow Rate variable

At LOS F, all walking speeds are severely restricted, and forward progress is made only by "shuffling" There is frequent, unavoidable contact with other pedestrians Cross- and reverse-flow movements are virtually impossible Flow is sporadic and unstable Space is more characteristic of queued pedestrians than of moving pedestrian streams

Fig. 6.11 Pedestrian flow conditions
(From Highway Capacity Manual--1985)

according to the number of pedestrians observed in each observation. These three groups are for pedestrian number less than 20, between 20 to 29, and more than 29 representing low, medium, and high density, respectively. The average error and standard deviation associated with each group were calculated and shown in tables 6.2, 6.3, and 6.4, respectively.

TABLE 6.2 Means and Standard deviations (Pedestrians less than 20)

		Calculated Errors	
	Sample Size	Mean	Standard Deviation
Session 1	27	3.7%	4.4%
Session 2	17	4.6%	5.2%
Both Sessions	44	3.9%	4.7%

TABLE 6.3 Means and Standard deviations (Pedestrian between 20 to 29)

		Calculated Errors	
	Sample Size	Mean	Standard Deviation
Session 1	12	8.6%	2.8%
Session 2	11	8.0%	3.1%
Both Sessions	23	8.3%	3.0%

TABLE 6.4 Means and Standard deviations (Pedestrians more than 29)

	Sample Size	Calculated Errors	
		Mean	Standard Deviation
Session 1	21	11.0%	2.6%
Session 2	32	12.7%	3.9%
Both Sessions	53	12.0%	3.6%

The above results show that errors in detection increase with the increase pedestrian density. To verify this finding, "Before-and-After tests" (Refer to Appendix B) were applied according the data shown in Table 6.5 which summarizes that given in tables 6.2 to 6.4. Test results indicated that the hypothesis of no change in mean between every two groups is rejected at a 5% level of significance.

TABLE 6.5 Means and standard deviations under different density

Pedestrian Numbers	Sample Size	Mean	Standard Deviation
0 - 19	44	3.9%	4.7%
20 - 29	23	8.3%	3.0%
30 and up	53	12.0%	3.6%

In conclusion, the experimental results fully support the theoretical prediction of the developed pedestrian algorithm. The overall accuracy was above 91% when examining low, medium and high density pedestrian flow situations. Error analysis indicated that detection errors were usually caused by overlapped pedestrians seen on images. Overlapping occurs when pedestrian are walking abreast, when they are closely following one another, or when they are closely passing one another. When the number of overlapped pedestrians and the degree of overlapping increases, the accuracy of the algorithm decreases. In order to overcome this drawback, the angle of the camera mount tilt, θ should be reduced as much as possibly practical to near $-\pi/2$. In other words, if the camera can be placed directly above pedestrians, the occurrence of overlapping can be significantly reduced.

CHAPTER 7

CONCLUSIONS

7.1 SUMMARIES AND CONCLUDING REMARKS

Eight newly developed models (NR model, PP model, DOS model, DOB model, LM model, WHE model, PCE model, and TTC model) and four algorithms (vehicle algorithm, signal algorithm, classifier algorithm and pedestrian algorithm) are presented in this thesis for collection and analysis of real-time traffic data on vehicle count, speed, volume and classification as well as pedestrian count and volume measurement. The NR model was developed to remove noises in digitized video images. The PP model is an extension of the model documented in Duda (1973) and Bow (1992), and is used to measure objects in images. Instead of projecting a three-dimensional point onto an image plane, the newly developed PP model enables the user to convert the projection in order to convert the measures in the image plane into their real sizes. The models of DOS and DOB were developed for the detection of vehicle turning signal lights. In order to determine the overall dimension of a vehicle from video images, the LM and WHE models were developed. The LM model is able to measure the length of a vehicle while the WHE model is able to estimate its width and height. The PCE model was developed to provide characteristics of the type of vehicle shape. The TTC model provides a new method for vehicle classification.

The four algorithms, each dedicated for a specific function of traffic data collection and analysis, were developed by integrating the eight newly developed models.

The vehicle algorithm was designed to analyze video images in order to measure traffic volume and vehicle speed. In contrast to other algorithms which were limited to processing monochrome images, this algorithm is able to process color images in order to reduce detection errors caused by shadows. Although its accuracy is estimated with limited number of data as a proof of concert, more data is needed to precisely determine its accuracy. The algorithm is incapable of reducing the errors caused by vehicle image occlusion. This algorithm has verified that color image processing has a great potential to increase detection accuracy, by solving shadow related problems, and to perform real-time vehicle detection from color images.

The signal algorithm provides a new method for detecting vehicles intending to turn at intersections based on signal light detection. Signal light detection is not reported in literature pertaining to video traffic detections. Unlike current detection algorithms, the developed signal algorithm offers extra features based on processing and detection of signal lights. Two models (DOS and DOB) were used in this algorithm. Because DOS and DOB introduced color and blink of signal lights as criteria for the detection, the signal algorithm no longer suffers from the difficulties of accurately recognizing signal lights of various sizes or differentiating them from the reflections of small objects whose sizes and shapes are similar. Also, it is able to distinguish turning signal light from other illuminated vehicle lights. These functions are not available in the algorithm developed earlier by Lu et al (1988). Although the algorithm may not provide an accurate practical estimate of the number of turning vehicles, simply due to a fact that some drivers do not use their turning indicators, it provides a method for detecting emergency vehicles such as ambulances and police cars. The limitation of the new algorithm is that it cannot detect moving lights and lights in a distance of more than 30 meters away from a video sensor.

The classifier algorithm was developed for vehicle classification. In order to increase its capability, it considered more geometrical parameters ir the classification process than those used in current systems. For instance, the classifier algorithm is able to differentiate buses from trucks, and vans from cars. Three models (LM, WHE and PCE) were utilized in the classifier algorithm to provide information about overall length, width, height and type of vehicle shape. Also, the TTC model was used to classify vehicles according to the overall dimension and shape. Although the classifier algorithm does not detect the number of axles directly, as it is the case with existing video detection systems (Autoscope 1995, CCATS 1995), it is capable of estimating the number of axles approximately based on lengths and number of units of each vehicle. This feature represents an extension to the current developments.

The pedestrian algorithm provides a new method for automatic measurement of pedestrian flow and overcomes a common disadvantage associated with the previous algorithms (Hwang 1983; Lu 1990). The latter cannot output satisfactory results when measuring pedestrian flows at intersections. This is because they were incapable of differentiating vehicles from pedestrians and detecting pedestrian movements in various directions. The newly developed algorithm was designed to measure the multidirectional flow on a crosswalk without this function being affected by passing vehicles. The overall accuracy of the pedestrian algorithm was satisfactory when examining actual low- to average-density pedestrian flow situations. The accuracy was reduced under the situations of heavy-density pedestrians, lighting transition periods and shadows.

The computer software for each algorithm has been implemented and tested in the laboratory. The experimental results showed that the algorithms can generate and predict traffic data with satisfactory accuracy. However, the software is still in the prototype stage.

The contributions of this thesis can be summarized as follows:

- Developed a color image processing method which can perform real-time detection and reduce errors caused by shadow. This method has a potential for improving accuracy of the existing systems;
- Provide methods to increase accuracy in some ways for collection and analysis of (1) vehicle count, speed and volume, (2) vehicle classification, and (3) pedestrian count and volume;
- Provide methods to solve problems of existing systems, such as (1) shadow related problems, (2) inability to detect vehicles intending to turn at intersections, (3) inability to count pedestrian at intersections, and (4) limited number of categories available in vehicle classification;
- Developed new functions: (1) detection of blink lights, and (2) estimating 3-D size of object from 2-D images.

In conclusion, computer vision techniques have a great potential in traffic data collection and analysis. They can supplement other available techniques and be used to provide more accurate real-time data. With the help of computer vision systems, capabilities of traffic control and management systems can be improved. Currently, the existing video traffic detection systems still suffer from a number of problems. Considerable work is still needed to modify these systems in order to increase their accuracy and expand their capability.

7.2 RECOMMENDATIONS FOR FUTURE RESEARCH

The current work could be extended in the future from the following aspects.

Interface development and enhancement. Although the interfaces for the software of two algorithms (vehicle and pedestrian) were fully developed to facilitate operation, such interfaces were not fully developed for the other two algorithms. Future work is needed in interface development and enhancement.

Integration of algorithms. The modules of the four algorithms could be integrated to form a computer vision system. This system should be able to provide all the required analysis of traffic data collected from a common input of video images.

Enhancement of the vehicle algorithm. The vehicle algorithm could be expanded to derive more traffic flow data from image sequences. These data include headway, density and spacing. Also, the vehicle algorithm could be advanced to make incident detection possible.

Advancement of detection method. The method developed for moving object detection could be modified to detect and recognize stopped vehicles instead of simply detecting moving vehicles. Such advancement will enable the vehicle algorithm to determine the vehicle queue length.

Enhancement of classifier algorithm. The classifiei algorithm could be expanded to determine the number of axles of a vehicle. Although the number of axles could be estimated by applying LM and PCE models to find out the number of units and oveiall length of the vehicle, it is not accurate enough for classifying vehicles by their number of axles. In this case, a neural network (Vaillant 1993) or Hough transformation (Bow 1992) should be considered to detect and locate the axles in images.

New application of the classifier algorithm. The classifier algorithm could be modified to meet the requirements of vehicle classification for an electronic-toll-collection system.

Advancement of the DOS and DOB models. The models of DOS and DOB were developed for the detection of turning-signal lights of vehicles. Their capabilities could be extended to permit the detection of emergency vehicles when emergency lights are turned on. Also, after slight modification, the models will be able to detect any kind of blinking signal lights.

Enhancement of signal algorithm. Although the developed signal algorithm can detect stationary vehicle signal lights, it is incapable of dealing with those in motion. A neural network with recurrent architecture (Elman 1990; Meng 1993) could be considered to enable the signal algorithm to predict the location of a moving signal light at a subsequent moment and match the same signal light in different image frames.

Advancement of pedestrian algorithm. The pedestrian algorithm could be advanced to generate satisfactory results under heavy pedestrian flow situations. Researchers in the U.K have been working on measuring heavy pedestrian flows (Hamer 1994). Although they did not consider the situation of pedestrians at an intersection, their prospective methods for dealing with heavy pedestrian flow will be helpful in measuring the same at intersections.

REFERENCES

AASHTO, (1984) "A policy on geometric design of highways and streets," American Association of State Highway Transportation Officials, Washington, D.C.

Ashworth, R., & Kentros, A. (1976). "Performance characteristics of an ultrasonic detector for measurements of vehicle occupancy," Traffic engineering and control. v17 n12 p502-504.

Ashworth, R. (1976). "A videotape-recording system for traffic data collection and analysis," Traffic engineering and control. v17 n11 p468-470.

Autoscope (1995). "The vehicle classification capability of the AUTOSCOPE", Technical fax received from Mr. Jeff P. Spinazze and Gary Crow, Econolite Control Products, Inc.

Bell, M.G.H., Cowell, M.P.H., & Heydecker, B.G. (1989). "Traffic-responsive signal control at isolated junctions," Traffic Control Methods, Edited by Sam Yagar and Edwin Rowe. Proceedings of the engineering foundation conference on traffic control methods. p273-294.

Behnam, J., and Patel, B. G., (1977) "A method for estimating pedestrian volume in a central business district." Transportation Research Record 629, TRB, National Research Council, Washington, D. C., p22-26.

Berinzon, M. (1993) "Development of a transportation data processing system for metropolitan Toronto." Proceedings of the IEEE-IEE vehicle navigation and information system conference, Ottawa, Canada, October 12-15 1993. Publ. by

IEEE. IEEE service center. Piscataway. NJ. USA. p186-190.

Blosseville, J.M., Krafft, C., Lenoir, F., Motyka, V., and Beucher, S. (1989). "TITAN: A traffic measurement system using image processing techniques." IEE Second International Conference on Road Traffic Monitoring, IEE Publication No. 299. p84-88.

Blosseville, J.M., Krafft, C., Lenoir, F., Motyka, V., & Beucher, S., (1990). "TITAN: New traffic measurements by image processing," Control, Computers, Communications in transportation, Ed. by Perrin, J. P., Pub. by Pergamon press. Oxford, U.K. p35-42.

Bonsall, P., and Bell, M. (1987). "Information Technology Applications in Transport." VNU Science Press, Utrecht, The Netherlands.

Bow, S. (1992) "Pattern recognition and image processing." Marcel Dekker, Inc. New York. p323-325

Braston, D.M. (1975). "A method of collecting data on speeds and headway on a motorway." Traffic engineering and control. v16 n10 p430-432.

Burkhard, M., and Rehfeld, N. (1987). "Real-time road surveillance by an optical sensor." SPIE Proceedings on Real Time Image Processing: Concepts and Technologies, Vol.860.

Cameron, R. M., (1977) "Pedestrian volume characteristics." Traffic Engineering, v47 n1 p36-37.

Cammarata, G.; Cavalieri, S.; Fichera, A. and Marletta, L. (1993) " Noise prediction in urban traffic by a neural approach." New Trends in Neural Computation.

Proceeding.s of the international workshop on artificial neural networks, IWANN '93, Sitges, Spain, June 9-11, 1993. Springer-Verlag Berlin Heidelberg. p611-619.

CCATS (1995). "The vehicle classification capability of the CCATS VIP2", Technical fax received from Mr. Frans Lemaire, Traficon NV, Belgium.

Chamberlin, G.J. and Chamberlin, D.G. (1980). "Colour: Its Measurement, Computation and Application." Heyden & Son Ltd., London.

Chatziioanou, A.E., Hockaday, S.L.M., McCarley, C.A. and Sullivan, E.C., (1991). "Testing and feasibility of VIPS for traffic detection," Application of Advanced Technologies in Transportation Engineering. Edited by Stephanedes, Y.J., and Sinha, K.C., Proceeding of the second international conference on application of advanced technologies in transportation engineering. Minneapolis, Minnesota, August 18-21, 1991.

Dickinson, K.W., & Waterfall, R.C. (1984a). "Image processing applied to traffic: 1. A general review," Traffic engineering and control. v25 n1 p6-13.

Dickinson, K.W., & Waterfall, R.C. (1984b). "Image processing applied to traffic: 2. Practical experience," Traffic engineering and control. v26 n2 p60-67.

Dickinson, K.W., & Wan, C.L. (1989). "Road traffic monitoring using the TRIP II system," Proceeding of the second International conference on road traffic monitoring. Feb. 7-9, 1989. p56-60.

Driver's Handbook. (1983) Gouvernement du Quebec, Ministere des Communication. p122.

Duda, R. O., and Hart, P. E. (1973) Pattern Classification and Scene Analysis. John

Wiley & Sons, Inc. U.S.A.

Duvieubourg, L., Postaire, J.G., & Depaaris, J.P. (1990). "3-D motion from line image sequences for intrusion detection on L. R. T. tracks," Control, Computers, Communications in Transportation. Edited by J. P. Perrin. Published by Pergamon Press. Oxford, U.K. p51-56

Elman, J.L. (1990) Cognitive Science, 14, p179-211.

Enkelmann, W. (1989). "Interpretation of traffic scenes by evaluation of optical flow fields from image sequences." IFAC Control, Computer, Communications in Transportation. Edited by J. P. Perrin. Published by Pergamon Press. Oxford, U.K. p43-50

Frence, A., and Solomon, D., (1986) "Traffic data collection and analysis: methods and procedures." National Cooperative Highway Research Program Synthesis of Highway Practice 130. Transportation Research Board.

Funabiki, N.; Takefuji,Y.; Lee, K.C. (1993) "Comparisons of seven neural network models on traffic control problems in multistage interconnection networks." IEEE Transactions on Computers. v42 n4 p497-501.

Hamer, M. (1994) "Smart video keep track of pedestrians." New Scientist, v142 May 28 '94 p21.

Harvey, B.A.; Champion, G.H. and Deaver, R. (1993) "Accuracy of traffic monitoring equipment field tests." Proceedings of the IEEE-IEE vehicle navigation and information system conference, Ottawa, Canada, October 12-15 1993. Publ. by IEEE, IEEE service center, Piscataway, NJ. USA. p141-144.

Hilbert, E.E., et al. (1978). "Wide-area detection system conceptual design study," Report FHWA-RD-77-86. FHWA, U.S. Department of Transportation. Feb. 1978.

Hockaday, S. (1991). "Evaluation of image processing technology for applications in highway operations." Final Report TR 91-2 (Contract 51K287) to Caltrans by California Polytechnic State University, San Luis Obispo, CA, June.

Hoose, N., and Willumsen, L.G., (1987). "Automatically extracting traffic data from video-tape using the CLIP4 parallel image processor," Pattern Recognition Letters. 6 (1987) p199-213.

Hoose, N. (1989). "Queue detection using computer image processing." IEE Second International Conference on Road Traffic Monitoring, Publication No.299. p94-98.

Hoose, N. (1990). "Computer vision as a traffic surveillance tool," Control, Computers, Communications in Transportation. Edited by J. P. Perrin. Published by Pergamon Press. Oxford, U.K. p57-64.

Hoose, N. (1992) "IMPACTS: am image analysis tool for motorway surveillance." Traffic Engineering and Control. v33 n3 p140-147

Houghton, A.D., Hobson, G.S., Seed, N.L., and Tozer, R.C. (1989). "Automatic vehicle recognition." IEE Second International Conference on Road Traffic Monitoring, Publication No.299. p71-78.

Hsiao, C., Lin, C., and Cassidy, M. (1994) "Application of fuzzy logic and neural network to automatically detect freeway traffic incidents," Journal of Transportation Engineering. v120 n5 p753-772.

Huang, T.S. (1981) Image Sequence Analysis. Published by Berlin; New York: Springer-

Verlag.

Huang, T.S. (1983) Image Sequence Processing and Dynamic Scene Analysis. Published by Berlin: Springer.

Hwang B.W. and Takaba S. (1983). "Real-time measurement of pedestrian flow using processing of ITV images," Systems, Computers, Controls. v14 n4 p46-55.

Inigo, R.M. (1985). "Traffic monitoring and control using machine: A survey," IEEE transactions on industrial electronics, Vol. IE-32.

Inigo, R.M., (1989). "Application of machine vision to traffic monitoring and control," IEEE Transactions on Vehicular technology, v38 n3 p112-122.

ITE, (1982) "Transportation and traffic engineering handbook," 2nd Edition, Institute of Traffic Engineers, Prentice Hall, New York.

Jain, Anil K. (1989) "Fundamentals of digital image processing," Prentice-Hall, Inc., New Jersey, U.S., p69.

Kaseko, M.S., Lo, Z. and Ritchio, S.G. (1994) "Comparison of traditional and neural classifiers for pavement-cra^k detection." Journal of Transportation Engineering. v120 n4 p552-569.

Kell, J. H., Fullerton, I. J. (1982). "Manual of traffic signal design," Institute of Transportation Engineers. Prentice-Hall Inc., New Jersey, USA. p78-90.

Khattak, A.J., Koppelman, F.S., and Schofer, J.L. (1993) "Automobile Commuters' Response to adverse weather: Effect of weather and traffic information and implications for information systems." Paper presented at 1993 Annual Meeting, Transportation Research Board.

Kikuchi, S., (1991). "Application of fuzzy set theory to the analysis of transportation problem," Application of Advanced Technologies in Transportation Engineering. Edited by Stephanedes, Y.J., and Sinha, K.C., Proceeding of the second international conference on application of advanced technologies in transportation engineering. Minneapolis, Minnesota, August 18-21, 1991. p41-45.

Klein, L.A. (1993) "Parameter measurement technology evaluation." Proceedings of the IEEE-IEE vehicle navigation and information system conference, Ottawa, Canada, October 12-15 1993. Publ. by IEEE, IEEE service center, Piscataway, NJ. USA. p529-533.

Koller, D.; Daniilidis, K and Nagel, H.H. (1993) "Model-based object tracking in monocular image sequences of road traffic scenes." International Journal of Computer Vision, v10 n6 p257-281.

Krose, B.J.A. et al. (1991) "An introduction to neural networks." Fourth Edition, University of Amsterdam, Amsterdam, Holland.

Labell, L.N., Spencer, M., Skabardonis, A., and May, A.D. (1989). "Detectors for freeway surveillance and control." Working Paper UCB-ITS-WP-89-1. Institute of Transportation Studies, University of California at Berkeley, May.

Lisa, F.; Carrabina, J.; Perez-Vicente, C.; Avellana, N. and Valderrama E. (1993) "Feed forward network for vehicle license character recognition." New Trends in Neural Computation. Proceedings of the international workshop on artificial neural networks, IWANN '93, Sitges, Spain, June 9-11, 1993. Springer-Verlag Berlin Heidelberg. p638-644.

List, G., Pond. J., Raess, R., Knitowski, D. and Krishnamurthy, S. (1989). "Video image processing / Pattern recognition to perform traffic counts." Presented at Application of Advanced Technology in Transportation conference. San Diego, Calif. Feb. 1989. p133-138.

Lu, Y., Hsu, Y.H. and Tan, G. G. (1988) "An application of the image analysis technique to detect left-turning vehicles at intersections," Transportation Research Record 1194, p120-128.

Lu, Y., Tang, Y., Pirard, P., Hsu, Y. and Cheng, H. (1990). "Measurement of Pedestrian Flow Data Using Image Analysis Technique." Transportation Research Board, **1281**: p87-96.

Lu, Y., Yuan, X. and Yan, C. (1992). "Image Sequence Analysis for Measurement of Pedestrian Flows." The Journal of the Transportation Research Forum. v32 n2 p399-408.

Lu, Y., Hsu, Y., and Maldague, X. (1992). "Vehicle classification using infrared image analysis." ASCE Journal of Transportation Engineering. v118 n2 p223-240.

Lu, Y., Yuan, X. and Yan, C. (1992) "Application of Image Sequence Analysis to Pedestrian Volume Measurement at Intersections." Presented at the 71st TRB Annual Meeting, Jan. 12-16, 1992. Washington, DC. Paper code 920187.

Lu, Y. and Yuan, X. (1993) "Color Image Analysis for Vehicle Speed Measurement." The Canadian Journal of Civil Engineering. Vol. 20, No.2, April 1993, p228-235.

Lyles, R.W., & Wyman, J.H., (1983). "An operational review of traffic data collection system," ITE Journal. v53 n12 p18-24.

Mackeown, W.P.J. and Thomas, B.T. (1993) "Labelling images with a neural network" Proceedings of the third international conference on artificial neural networks, Bright Conference Center, UK. May 25-27, 1993. The Institi.tion of Electrical Engineers, London p16-20.

Malenstein, J. (1993) "Operating traffic information and traffic monit ring and control of Dutch national police force TIC, using advanced telematics, and results from the DRIVE II programm." Proceedings of the IEEE-IEE vehicle navigation and information system conference, Ottawa, Canada, October 12-15 1993. Publ. by IEEE, IEEE service center, Piscataway, NJ. USA. p195-198.

Maren, A.J., Harston, C.T., and Pap, R.M. (1990) Handbook of neural computing applications. Academic Press, Inc. San Diego, California.

Maxwell, J. C. (1890). On the Theory of Three Primary Colors. Lectures delivered in 1861. W.D.Nevin (ed.), Scientific Paper 1, Cambridge University Press, London, p445-450.

McShane, W. R. and Roess, R. P. (1990) Traffic Engineering. Prentice Hall, Englewood Cliffs.

Meng, H. and Picton, P.D. (1993) "A neural network motion predictor." Proceedings of the third international conference on artificial neural networks, Bright Conference Center, UK. May 25-27, 1993. The Institution of Electrical Engineers, London p177-181.

Michalopoulos, P.G., Fitch, R., & Geokezas, M. (1986). "Development of a visible / infrared vehicle detection system: Feasibility study," Final report, Department of civil and mineral engineering, University of Minnesota, Minneapolis, Minn.

Michalopoulos, P.G., & Wolf, B. (1990). "Machine-vision system for multispot vehicle detection," Journal of Transportation Engineering. v116 n3 p299-309.

Michalopoulos, P.G., (1991). "Incident detection through video image processing." Application of Advanced Technologies in Transportation Engineering. Edited by Stephanedes, Y.J., and Sinha, K.C., Proceeding of the second international conference on application of advanced technologies in transportation engineering. Minneapolis, Minnesota, August 18-21, 1991. p41-45.

Miyasako, T., et al. (1989). "The ultrasonic vehicle profile classifiers," Proceeding of the second International conference on road traffic monitoring, IEE, U.K. p110-113.

Moore, R. C. (1981). "Road sensors for traffic data collection," Sensors in Highway and Civil Engineering, Proceedings of the conference organized by the Institution of Civil Engineers, London, Feb. 1981. p73-83.

National Safety Council. (1986). **Accident Facts**. Chicago, p55.

Okamoto, H. (1993) "Recent developments in Japanese traffic information supply system." Proceedings of the IEEE-IEE vehicle navigation and information system conference, Ottawa, Canada, October 12-15 1993. Publ. by IEEE, IEEE service center, Piscataway, NJ. USA. p206-207.

O'Neill, W.A., Ullah, K., and Wang, M. (1993) "IVHS and rural road safety: A prototype ATIS." ITE Journal, Vol. 63, No. 11, November 1993, p30-37.

Onoe, M., Nobuo, H., and Ohba, K. (1973) "Computer analysis of traffic flow observed by subtractive television." Computer Graphics and Image Processing. 1973(2), p377-392.

Onoe, M., and Ohba, K. (1976) "Digital image analysis of traffic flow." Proceeding of International Joint Conference on Pattern Recognition, Coranado, November, 1976, p803 804

Fan, W., Kuah, G. K., Su, S., and Wang, A. (1991). "Automatic vehicle classification system." Appl. of Adv. Tech. in Transp. Engrg., Proc. of the second Int. Conf., Minneapolis, Minnesota, August 18-21, 1991. Ed. by Stephanedes, Y. J., and Sinha, K. C., p11-15.

Pant, P.D., Balabrishman, P. (1994) "Neural network for gap acceptance at stop-controlled intersections," Journal of Transportation Engineering. v120 May/June '94 p432-436.

Patel, A.K.; Wright, W.A. and Collins, P.R. (1993) "Target classification using neural and classical techniques." Proceedings of the third international conference on artificial neural networks, Bright Conference Center, UK. May 25-27, 1993. The Institution of Electrical Engineers, London p238-242.

Pignataro, L.J., (1973). "Traffic Engineering: Theory and Practice." Published by Prentice-Hall, Inc. New Jersey, USA.

Pfannerstill, E., (1991). "Measuring section-related traffic data by correlation methods -- A new approach to traffic monitoring and control," Application of Advanced Technologies in Transportation Engineering. Edited by Stephanedes, Y.J., and Sinha, K.C., Proceeding of the second international conference on application of advanced technologies in transportation engineering. Minneapolis, Minnesota.

August 18-21, 1991. p1-5

Polus, A., Livneh, M., & Borovsky, S. (1978). "A multi-purpose portable data collection system," Traffic engineering and control. v19 n3 p123-125.

Pursula, M. & Kosonen, I. (1989). "Microprocessor and PC-based vehicle classification equipments using induction loops," Proceedings of the second International conference on road traffic monitoring, IEE, U.K. p24-28.

Rizenbergs, R.L., et al., (1976) "State survey of skid resistance of pavements," Kentucky Department of Transportation, Lexington. December.

Rourke, A. (1989). "University of Leeds: Progress report 1988." Traffic Engineering and Control, v30, n? p200-201.

Rourke, A. & Bell, M.G.H., (1991). "Traffic queue detection using image-processing," Application of Advanced Technologies in Transportation Engineering. Edited by Stephanedes, Y.J., and Sinha, K.C., Proceeding of the second international conference on application of advanced technologies in transportation engineering. Minneapolis, Minnesota, August 18-21, 1991. p106-110.

Rourke, A. & Bell, M.G.H., (1991). "Queue detection and congestion monitoring using image processing," Traffic Engineering and Control. Vol.32, No.5, p412-421.

Rowe, E. (1993) "IVHS--Making it work, pulling it all together." ITE Journal, v63 n11 p45-48.

Schlutsmeyer, A.P., et al. (1982). "Wide-area detection system (WADS)," Report FHWA-RD-82-144. FHWA, U.S. Department of Transportation. Sept. 1982.

Shimizu, K., and Shigehara, N. (1989). "Image processing system using cameras for vehicle surveillance." IEE Second International Conference on Road Traffic Monitoring. Publication No.299. U.K. p61-65.

Special Report 209: Highway Capacity Manual. Chapter 13: Pedestrians. TRB. National Research Council, Washington. D.C., 1985, p.9

Spencer. M., Labell. L.N., and May, A.D. (1989). "Detectors for freeway surveillance and control: An update." Working Paper UCB-ITS-WP-89-9. Institute of Transportation Studies. University of California at Berkeley, December.

Takaba, S., Sakauchi, M., and Kaneko, T. (1982). "Measurement of traffic flow using real time processing of moving pictures." Proceedings of 32nd IEEE Conference on Vehicle Technology, p488-494.

Tang, Y.Y., and Yuan, X. et al. (1993) "An Approximation of Perspective Image Using Bilinear Model." Proceedings of the International Conference on Signal Processing '93 / Beijing. October 26-30, 1993.

Taylor, M.A.P., & Young, W. (1988). "Traffic analysis: New technology and new solutions," Hargreen Publishing Co., Melbourne, Australia.

Traffic monitoring guide. (1985). Federal Highway Admin., Office of Highway Planning Highway Statistics Div., U.S. Dept. of Transp.

Tsao, S., Kehtarnavaz, N., Chan, P. and Lytton, R. (1994) "Image-based expert system approach to distress detection on CRC pavement," Journal of Transportation Engineering. v120 n1 p52-64.

Vaillant, R.; Monrocq, C. and Cun, Y.L. (1993) "An original approach for the localization of objects in images." Proceedings of the third international conference on artificial neural networks, Bright Conference Center. UK. May 25-27, 1993. The Institution of Electrical Engineers, London p26-30.

Versavel. J., Lemaire, F., and Van der Stede, D. (1989). "Camera and computer-aided traffic sensor." IEE Second International Conference on Road Traffic Monitoring, Publication No.299. U.K. p66-70.

Wan, C.L. and Dickinson, K. W. (1989). Road Traffic Monitoring Using Image Processing - A Survey of Systems, Techniques and Applications. Proc., IFAC Control, Computers, Communications in Transportation, Paris, France, p27-34.

Young, T. (1802). On the Theory of Light and Colors. Philosophical Transactions of the Royal Society of London, **92**: p20-71.

Young, W. (1989). Application of VADAS to complex traffic environments. Journal of Transportation Engineering. v115 n5 p521-536.

Yuan, X., Lu, Y. and Sarraf, S. (1993) "A Computer Vision System for Measurement of Pedestrian Volume." Proceedings of the 1993 IEEE Region 10 International Conference on "Computer, Communication and Automation". October 19-21, 1993, Vol.2 p1046-1049

Yuan, X., Lu, Y., and Sarraf, S. (1994) "A Computer Vision System for Automatic Vehicle Classification." The ASCE Journal of Transportation Engineering. v120 n6 p861-876.

Zornetzer, S. F., Davis, J.L, and Lau, C. (1990) An introduction to neural and electronic networks. Academic Press, Inc. San Diego, California.

APPENDIX A

TABLE A-1 Experimental data on pedestrian counting

Session 1.				Session 2.			
Actual	Output	Diff.	Error %	Actual	Output	Diff.	Error %
29	32	3	10	47	55	8	17
6	6	0	0	42	48	6	14
2	2	0	0	38	32	-6	16
9	9	0	0	52	44	-8	15
10	10	0	0	35	38	3	9
13	14	1	8	38	34	-4	11
31	35	4	13	53	60	7	13
34	31	-3	9	30	28	-2	7
15	16	1	7	14	12	-2	14
30	27	-3	10	48	56	6	17
45	40	-5	11	33	37	4	12
38	34	-4	11	21	23	2	10
13	13	0	0	32	30	-2	6
49	54	5	10	17	19	2	12
21	20	-1	5	5	5	0	0
16	15	-1	6	17	16	-1	6
3	3	0	0	24	23	-1	4
9	10	1	11	16	17	1	6
34	36	2	6	34	33	-1	3
30	32	2	7	27	30	3	11
30	27	-3	0	38	43	5	13

36	32	-4	11	11	12	1	9
12	13	1	8	32	29	-3	9
21	23	2	10	4	4	0	0
27	29	2	7	46	55	9	20
34	37	3	9	43	48	5	12
10	11	1	10	20	21	1	5
10	10	0	0	36	31	-5	14
36	41	5	14	31	28	-3	10
39	44	5	13	32	29	-3	9
23	27	4	4	3	3	0	0
5	5	0	0	33	31	-2	6
8	8	0	0	21	22	1	5
29	26	-3	10	16	16	0	0
46	53	7	15	39	33	-6	15
25	27	2	8	48	40	-8	17
20	21	1	5	32	28	-4	13
39	33	-6	15	49	42	-7	14
16	14	-2	13	2	2	0	0
3	3	0	0	18	18	0	0
32	36	4	13	33	39	6	18
32	30	-2	6	26	24	-2	8
26	24	-2	8	47	38	-9	19
15	15	0	0	11	10	-1	9
5	5	0	0	8	9	1	13
13	14	1	8	45	50	5	11
19	21	2	10	45	51	6	13
38	34	-4	11	50	43	-7	14
26	29	3	12	40	36	-4	10

6	6	0	0	11	10	-1	9
13	14	1	8	25	23	-2	8
18	19	1	6	23	25	2	9
32	35	3	9	6	6	0	0
9	9	0	0	12	12	0	0
38	42	4	11	27	23	-4	15
27	24	-3	11	52	60	8	15
17	16	-1	6	24	22	-2	8
4	4	0	0	35	30	-5	14
24	21	-3	13	7	7	0	0
37	42	5	12	20	19	-1	5

Note: Actual = The actual number of pedestrians who pass though the detection line during the time of observation (about 45 seconds);

Output = The number of pedestrians detected by the pedestrian algorithm;

Diff. = Output - Actual;

Error = (Diff. / Actual)*100 %

The equations used for determining the computed means and standard deviations of above sample are

$$Mean = \frac{1}{N} \sum_{i=1}^{i=N} (error_i)$$

$$Std.\ Dev. = \sqrt{\frac{1}{N-1} \sum_{i=1}^{i=N} (error_i - computed\ mean)^2}$$

where N denotes the number of samples.

APPENDIX B

The equations of Before-and-After tests (McShane 1990) are presented below.

$$\mu_\Delta = \mu_A - \mu_B \tag{B-1}$$

$$\sigma_\Delta = \sqrt{\frac{\sigma_A^2}{N_A} + \frac{\sigma_B^2}{N_B}} \tag{B-2}$$

where μ_A and μ_B indicates two means; σ_A and σ_B indicates two standard deviations; and N_A and N_B are the number of samples; μ_Δ and σ_Δ are the mean and standard deviation of a new distribution that is the difference between the two means, μ_A and μ_B.

The hypothesis of interest is "H_0: This new distribution has zero mean," which would indicate no change in the mean as a result of pedestrian density change. Using an $\alpha = 0.05$ confidence, a table showing the probability to the left of a point Z in a standard normal distribution indicates that a Z of -1.645 or 1.645 is appropriate. Thus if

$$\frac{(\textit{observed mean}) - (\textit{hypothesized mean})}{\textit{standard deviation}} < -1.645$$

or

$$\frac{(\textit{observed mean}) - (\textit{hypothesized mean})}{\textit{standard deviation}} > 1.645 \tag{B-3}$$

then the hypothesis is *rejected* at a 5% level of significance.

Using the data summarized in Table 6.5, the calculations necessary for the tests is presented below.

1. Test between groups with low and medium pedestrian densities:

From the equations indicated in Equations (B-1) and (B-2):

$$\mu_\Delta = \mu_A - \mu_B = 3.9 - 8.3 = -4.4\ \%$$

$$\sigma_\Delta = \sqrt{\frac{\sigma_A^2}{N_A} + \frac{\sigma_B^2}{N_B}} = \sqrt{\frac{4.7^2}{44} + \frac{3.0^2}{23}} = 0.945\ \%$$

and then the quantity indicated by Equation (B-3):

$$\frac{(-4.4) - (0)}{0.945} = -4.7 < -1.645$$

so that the hypothesis of no change in the mean is *rejected* at a 5% level of significance.

2. Test between groups with medium and high pedestrian densities:

Similarly, μ_Δ and σ_Δ are obtained as -3.7% and 0.797%, respectively. Since

$$\frac{(-3.7) - (0)}{0.797} = -4.4 < -1.645$$

the hypothesis of no change in the mean is also *rejected* at a 5% level of significance.

As a result of above tests, it can be concluded that the errors in pedestrian detection will increase with the increase in pedestrian density.