2015

Lane detection and classification for assistive navigation of the visually impaired

Manh Cuong Le
University of Wollongong

Lane Detection and Classification for Assistive Navigation of the Visually Impaired

A thesis submitted in partial fulfilment of the requirements
for the award of the degree

Doctor of Philosophy

from

UNIVERSITY OF WOLLONGONG

by

Manh Cuong Le

School of Electrical, Computer and Telecommunications Engineering

January 2015

Statement of Originality

I, Manh Cuong Le, declare that this thesis, submitted in partial fulfillment of the requirements for the award of Doctor of Philosophy, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

Manh Cuong Le
January 28, 2015

Contents

List of Figures

List of Tables

Acronyms

GIS	Geographic information systems
GPS	Global positioning system
HoG	Histogram of oriented gradients
HSI	Hue-saturation-intensity
HT	Hough transform
ICM	Iterated conditional modes
IIS	Illumination invariant space
MIL	Multiple instance learning
MRF	Markov random field
NCC	Normalized cross correlation
Pdf	Probability density function
PGS	Personal Guide System
POI	Patch of interest
RGB	Red-Green-Blue
RANSAC	RANdom SAmple Consensus
SVM	Support vector machine
VPE	Vanishing point estimation

Abstract

Visually impaired people face many difficulties in daily life. Among their essential activities, traveling safely and independently is one of the most challenging tasks. This thesis focuses on assisting vision-disabled people to deal with this problem. We propose a non-invasive system that locates various types of walking lanes for visually impaired travelers, using a novel image processing architecture and machine learning algorithms. In this system, a camera is employed to capture the scene image in front of the traveler. Then, the lane type is identified as marked-lanes, unmarked-lanes or non-lanes. Finally, a suitable lane detector for the lane type is applied to locate the walking region.

To recognize the lane type in each image, we propose a new method using multiple instance learning. The proposed method represents each image as a bag of instances. Each instance is an image region and described by a feature vector. A vocabulary-based framework of multiple instance learning is then employed to categorize bags.

To detect marked-lanes that are located by lane markers, we propose a region-based method. The proposed method for marked-lane detection extracts first local image regions locating on the borders of lane markers. The lane markers are then found using a Markov random field model. The walking region is finally determined using the geometric cues of lane marker pairs.

We also propose a new method to find pedestrian lanes in unstructured environments where lanes have no painted markers, vary significantly in appearance and have different shapes. The proposed method for unmarked-lane detection locates the walking lane using both appearance and shape features. The lane ap-

pearance model is learned on-the-fly from a sample region, which is automatically determined employing the vanishing point and the properties of lane surfaces and boundaries. Shape contexts are employed to model the shape of pedestrian lanes.

All the proposed methods for classifying and detecting pedestrian lanes are evaluated on a large and new data set of images, collected from different scenes under challenging illumination conditions. The experimental results prove the robustness and efficiency of the proposed methods.

Acknowledgments

The making of this doctoral dissertation has been a challenging and exciting experience. In addition to the efforts of myself, it would not have been possible to write this thesis without the help and support of many people who encouraged me and provided valuable assistance for preparing and completing this study.

First and foremost, I would like to gratefully acknowledge the enthusiastic supervision of Dr. Son Lam Phung and Prof. Abdesselam Bouzerdoum during this study. They have provided me invaluable guidance and shared valuable insights in the relevance of the study.

I also wish to give special thanks to the staff of the School of Electrical, Computer and Telecommunications Engineering, and my colleagues and friends. They have provided me personal and professional support during my studies at the University of Wollongong.

Finally, I would like to express my gratitude to my wife and parents, who have encouraged and supported me during my research studies.

Chapter 1

Introduction

Chapter contents

1.1 Research objectives

A significant percentage of the world population suffers from visual impairment and blindness. The World Health Organization estimates that the global population of vision-disabled people is about 285 million, of which 39 million are blind [10]. For the visually impaired, traveling safely and independently in different environments is a challenging task, for which an assistive navigation system is essential. The system should assist in several vital micro-navigation tasks such as finding pedestrian lanes, detecting and recognizing traffic obstacles, and sensing dangerous traffic situations.

Traveling aids for visually impaired people include conventional tools and technology systems. For traditional aids, white canes and guide dogs are widely used. A white cane assists the visually impaired in detecting obstacles and walking paths only at a close range. Compared with the white cane, a guide dog is more flexible in assisting the blind to evade obstacles and hazards, and follow a familiar route. However, the visually impaired with these tools cannot travel safety and independently in unknown environments.

To improve the mobility of the vision impaired, many technology aids have been developed for detecting obstacles (e.g. Sonicguide [11], GuideCane [12] and Miniguide [13]), finding routes and locations (e.g. Personal Guidance System [14], MoBIC [15], and Drishti [16]). However, little work has been done on pedestrian crossing lane detection for the visually impaired [1, 17], despite fact that straying outside of the lane region is very dangerous for blind travelers.

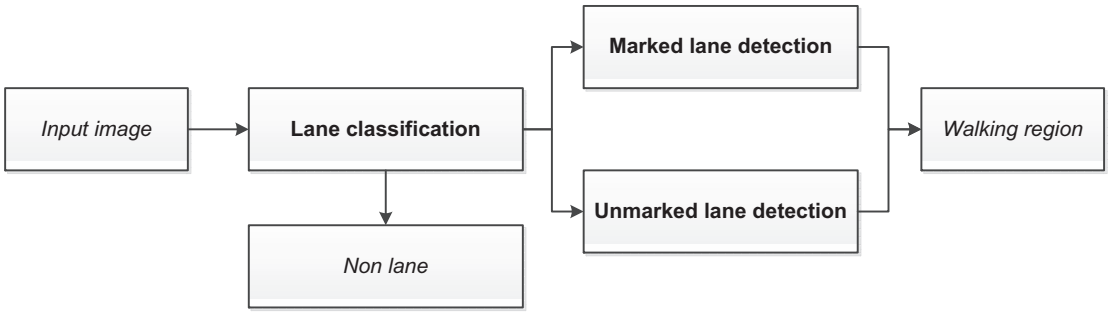


Figure 1.1: Block diagram of the proposed system for detecting automatically pedestrian lanes.

The objective of this research is to develop an assistive system of detecting automatically pedestrian lanes in various environments as shown in Fig. 1.1. The system is designed to detect robustly different pedestrian lanes from images captured by a camera. While pedestrian lanes at traffic junctions have painted markers, many other pedestrian lanes do not have painted markers. The proposed system includes three main components: 1) *lane classifier* to recognize the lane type in the input image, 2) *marked-lane detector* to locate the lane region that is indicated by painted markers, 3) *unmarked-lane detector* to locate the lane region that have no painted markers. From an input image, the lane classifier is employed first to identify the lane type as *marked-lane*, *unmarked-lane* or *non-lane*. Then, a detector corresponding to the type lane (*marked-lane* or *unmarked-lane*) is selected to locate the walking lane or a notification of *non-lane* is generated.

In general, this research project aims to address the following questions:

- What are the characteristics of pedestrian lanes? How to recognize different types of pedestrian lanes (e.g. marked lanes and unmarked lanes)?
- What are the characteristics of lane markers? How to detect pedestrian lanes

with painted markers?

- What are the characteristics of pedestrian lanes without lane markers? How to detect pedestrian lanes without lane markers?

1.2 Research contributions

The main contributions of this research are briefly presented as follows:

- We propose a vision-based approach of automatic pedestrian lane detection to assist vision-disabled people navigating in various environments. The proposed approach first identifies the lane type in the input image and then selects a suitable detector to locate the walking region.

- We propose a robust method to recognize automatically different types of pedestrian lanes under various illumination conditions. The proposed method for lane classification is based on local regions and multiple instance learning to categorize the input image, which belongs to one of three classes: marked-lanes, unmarked-lanes and non-lanes.

- We propose a robust method to detect automatically pedestrian lanes that are identified by painted markers under different illumination conditions. The proposed method are based on local regions and Markov random fields to find lane markers in the input image. The walking lane is then determined by optimizing a criterion that integrates multiple geometric cues of lane markers.

- We propose a robust method to detect pedestrian lanes in unstructured environments where lanes vary significantly in appearance, have different shapes and are not indicated by painted markers. The proposed method detects the walking lane using shape and appearance information. In our method, the lane appearance model is learned on-the-fly from a sample region, which is extracted directly in the input image using the vanishing point and the properties of lane surfaces and borders. Shape contexts are employed to model the shape of pedestrian lanes.

- We create a large and new image data set of different pedestrian lanes with manually annotated detection ground-truth and non-lane scenes for evaluating pedestrian lane detection methods and the entire system of detecting various pedestrian lanes. The data set was collected from various indoor and outdoor environments, under different illumination conditions. Although several data sets exist for road/lane detection for vehicles, our pedestrian-lane detection data set, to the best of our knowledge, is the first for assistive navigation of blind people, and is expected to facilitate research in assistive computer vision.

1.3 Publications

The publications for this PhD research project, which took place from late July 2010 to January 2015, include:

- M. C. Le, S. L. Phung, A. Bouzerdom, "'Pedestrian lane detection for assistive navigation of blind people"', in *The* 21*st International Conference on Pattern Recognition*, Tsukuba,Japan, 2012, pp 2594 - 2597.

- M. C. Le, S. L. Phung, A. Bouzerdom, "'Pedestrian lane detection for the visually impaired"', in *International Conference on Digital Image Computing: Techniques and Applications*, Fremantle WA, Australia, 2012, pp 1-6.

- M. C. Le, S. L. Phung, A. Bouzerdom, "'Pedestrian lane detection in unstructured environments for assistive navigation"', in *International Conference on Digital Image Computing: Techniques and Applications*, Wollongong, NSW, Australia, the **Best Recognition Paper** prize, 2014.

- M. C. Le, S. L. Phung, A. Bouzerdom, "'Lane detection in unstructured environments for autonomous navigation systems"', in *Asian Conference on Computer Vision*, National University of Singapore, Singapore, 2014.

1.4 Thesis organization

This thesis consists of six chapters:

- **Chapter 1** introduces the research project and its objectives, and summarizes related publications.

- **Chapter 2** gives a comprehensive review on assistive navigation systems for the visually impaired. This chapter presents conventional mobility tools and different technology aids for travels of vision-disabled people.

- **Chapter 3** presents a robust method to find automatically pedestrian lanes that are located by painted markers. Pedestrian lane detection relies on finding lane markers in the input image. Lane marker detection is based on local image regions and Markov random fields. Lane marker verification is done using multiple geometric cues.

- **Chapter 4** proposes a pedestrian lane detection method in unstructured environments where pedestrian lanes have no painted markers, vary significantly in appearance and have different shapes. The walking lane is detected using shape and appearance features. This chapter also presents an improved vanishing point estimation method using local orientations of edge pixels.

- **Chapter 5** describes a new method to recognize the lane type in the input image, using multiple instance leaning. The proposed method focuses on classifying input images into three classes: *marked-lanes*, *umarked-lanes* and *non-lanes*. This chapter also analyzes the effectiveness of using the lane classification in the pedestrian detection system.

- **Chapter 6** summarizes the research findings, and provides concluding remarks and future directions.

Chapter 2

Literature Review

Chapter contents

This chapter reviews existing navigational assistances for the visually impaired. First, mobility of vision-disabled people is presented in Section 2.1. Then, conventional mobility tools are described in Section 2.2. Next, assistive technologies are reviewed in Section 2.3. Finally, the content of the chapter is summarized in Section 2.4.

2.1 Mobility of the visually impaired

Mobility is the ability to travel safely and independently from one place to other places and is an important aspect of daily life. Vision plays a very significant role in mobility as it allows sighted people to collect most of the information required for perceiving the surrounding environments. People with vision loss must rely on other senses (e.g. hearing and touch) to gather information of the surrounding objects, and therefore face great difficulties in traveling.

The mobility of blind people includes two aspects: perception and orientation

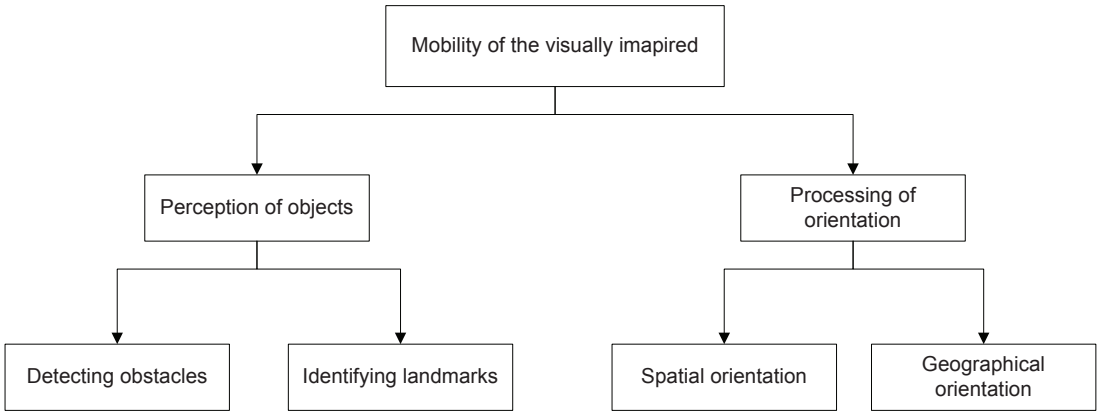


Figure 2.1: Brambing's model for mobility of blind people.

as shown in Fig 2.1 [18]. Perception refers to a blind traveler obtaining information about the environment via the non-vision senses in order to detect obstacles and identify landmarks. Orientation refers to the knowledge to recognize the position in relation to surrounding objects and the location in the routes of the entire journey. To understand the surrounding environments, vision disabled people employ touching and hearing as primary modalities [19]. In near space, both touching and hearing are used to perceive objects. In far space, the people reply mainly on hearing to find objects. Orientation is performed based on identification of landmarks. Blind people often employ many different types of landmarks to determine their position in the environments. Example landmarks include rises and falls in the walking path, changes in texture of the walking path, the presence of walls and hedges, traffic sounds and temperature changes.

People with vision impairments have three major barriers in traveling [19, 20] as follows:

1) Relying mainly on touching and hearing, visually impaired people are limited in detecting and avoiding obstacles, finding the travel path, and identifying early hazards ahead.

2) They have great difficulties in determining routes for a journey, understanding the scene layout, and identifying their position in the scene.

3) They cannot to obtain visual or textural information such as road signs and

bus numbers.

These barriers make visually impaired people unable to travel safely and independently in unknown environments.

2.2 Traditional aids

There are two popular traditional aids for navigation of the visually impaired: the white cane and the guide dog.

The white cane is a hand-held tool and is considered as an extension of the user's arms. This tool is a lightweight and cylindrical cane, and is often made of aluminium. It is adjustable to the height of the user and includes a purposely designed handle-grip and tip. To detect objects on the walking path, the user swings the white cane from side to side while the tip of the cane directly contacts with the ground. Based on vibrations and force feedback from the cane, the user can recognize objects ahead and characteristics of the ground. To use effectively a white cane, blind people are often required about 100 hours of training [19]. This is because an incorrect use of the cane can lead to dangers for both the user and others.

The white cane is a cheap, reliable and robust tool to detect obstacles and drop-offs at ground levels, identify the characteristics (e.g. texture and hardness) and conditions of the ground, and find the walking path at close range [20]. However, this tool is unable to detect obstacles at torso and face levels, which are dangerous for blind travelers [19,20]. Furthermore, using the white cane over a long distance causes arm fatigue for travelers [20].

The guide dog is a specially trained dog to assist the blind in mobility. A blind user can travel safely with a trained dog. The dog is trained to have the following major skills [21]:

- Walking in a straight line and on the left-hand side slightly ahead of the user.
- Stopping at all kerbs, the top and bottom of stairs.
- Waiting for user's commands before crossing roads.

- Avoiding obstacles at head level.
- Avoiding too narrow spaces where a person and a dog cannot walk through side by side.
- Boarding and traveling on all different types of public transportation.
- Taking the user to a lift.
- Refusing the user's commands that may cause dangers for the user.
- Recognizing the familiar routes.

The user controls the guide dog and receives information about its activities via a handle encircled the dog's chest.

With the above trained skills, the guide dog provides assistance of safe travel for vision-disabled people in the familiar environments. However, a guide dog typically requires about two to three years and 40,000 USD to train, and has a working life of only eight to ten years [19]. Due to the high cost of training and maintenance, only three percent of the visually impaired population are reported to use a guide dog [22]. Furthermore, the guide dog cannot detect obstacles at head levels.

Both the white cane and guide dog are useful aids for blind travelers in detecting and avoiding obstacles at ground levels in the near space. However, these tools cannot assist blind travelers in determining spatial and geographical orientations, recognizing locations and finding routes for their journey in unfamiliar environments.

2.3 Technology aids

Many assistive technology systems have been developed to improve the mobility of vision disabled people. These systems can be categorized into three classes: electronic obstacle detectors, computer-vision based systems, and GPS based systems. In the following, these assitive system classes are presented.

2.3.1 Electronic obstacle detectors

Obstacle detectors are small electronic devices that can be attached on white canes or glasses. These devices apply the echolocation principle of emitted signals for detecting obstacles. Based on the type of employed signals, the devices are classified into laser detectors and ultrasound detectors.

- Laser detectors: These devices are based on Cranberg's principle of optical triangulation [20]. The devices emit infrared light pluses from the transmitters, and these pulses will be reflected back when they meet obstacles. The receivers measure the angles of the reflected pluses in order to compute the distances to obstacles.

Examples of laser detectors are the C-5 laser cane [23] and the talking laser cane [24]. The C-5 laser cane employs three beams of infrared lights to detect obstacles in upward, forward and downward directions and a proximal range of 4 m as shown in Fig. 2.2. Therefore, this device can detect not only obstacles in front of the user, but also drop-offs and overhead obstructions. The information of detected obstacles is conveyed to the user by auditory tones or vibration levels of the cane handle.

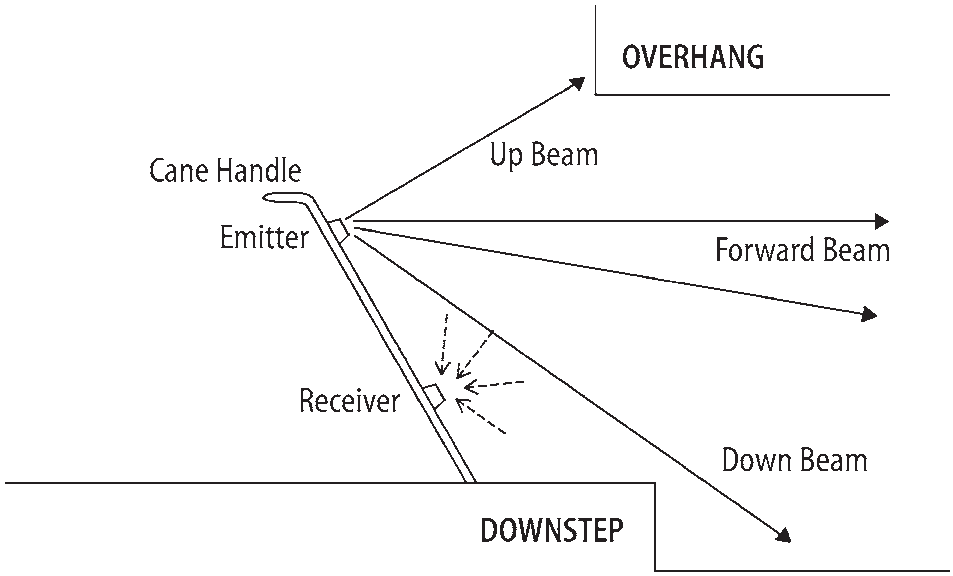


Figure 2.2: Beam geometry of laser cane [20].

In comparison with the C-5 laser cane, the talking laser cane is more complex.

In addition to obstacle detection in the forward travel path within a 20^0 spread angle, the talking laser cane is able to detect and identify signs of landmarks at distances about 10 m. The signs are designed to label landmarks. Each sign has an individual bar-code, and is attached to a landmark. When the laser light meets a sign, the user will hear a distinct beep to determine the direction and location of the sign, and the microprocessor of the device will try to identify the bar-code of the sign. If the sign is recognized, the device will indicate the landmark name by a spoken message. Thus, the talking laser cane provides obstacle avoidance assistance and orientation assistance.

Laser devices have several limitations. First, these devices cannot detect non-reflective obstacles, e.g. objects made of glass or transparent plastic. Second, since laser devices employ short laser pulses, the device only detect objects at proximal distances and provide information about the nearest object.

- Ultrasound devices: These devices are based on ultrasonic waves to detect obstacles. When a ultrasonic beam emitted from the device meets an obstacle, a portion of the beam will be reflected back and received at the device. The time period between emitting and receiving the ultrasound beam is measured to compute the distance to the obstacle.

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

Figure 2.3: Examples of ultrasound devices: (a) Sonicguide [11], (b) Miniguide [13], (c) UltraCane [25], (d) NavBelt [12] .

Example ultrasound devices include Sonicguide [11], Miniguide [13], UltraCane [25], NavBelt [26] and GuideCane [12]. Table 2.1 shows the capabilities of these ultrasound detectors. The Sonicguide consists of a ultrasound transmitter and two ultrasound receivers embedded into eyeglasses as demonstrated in Fig. 2.3(a). This device is designed to capture a sonic image of the environment.

The sonic image represents the distances to the obstacles by sound tones of different frequencies, and it also describes the direction of objects via delivering the sounds in the binaural headphones. The Sonicguide can detect multiple obstacles at a maximum distance of 6m from the traveler in a 60^0 forward field of view.

Table 2.1: Comparison of ultrasound detectors.

Systems	Number of sensors	Maximum distance (m)	Orientation aid
Sonicguide [11]	1	6	No
Miniguide [13]	1	8	No
UltraCane [25]	2	4	No
NavBelt [26]	8	5	Yes
GuideCane [12]	10-16	5	Yes

The Miniguide is a hand-held device as shown in Fig. 2.3(b), and includes a ultrasound transmitter and receiver. The Miniguide has four options for detecting objects in different distance ranges: 0.5, 1, 2, 4 and 8 m. The information of detected objects is conveyed to the user by auditory tones or vibration levels. The Miniguide cannot detect drop-offs and therefore it is often combined with a cane or guide dog.

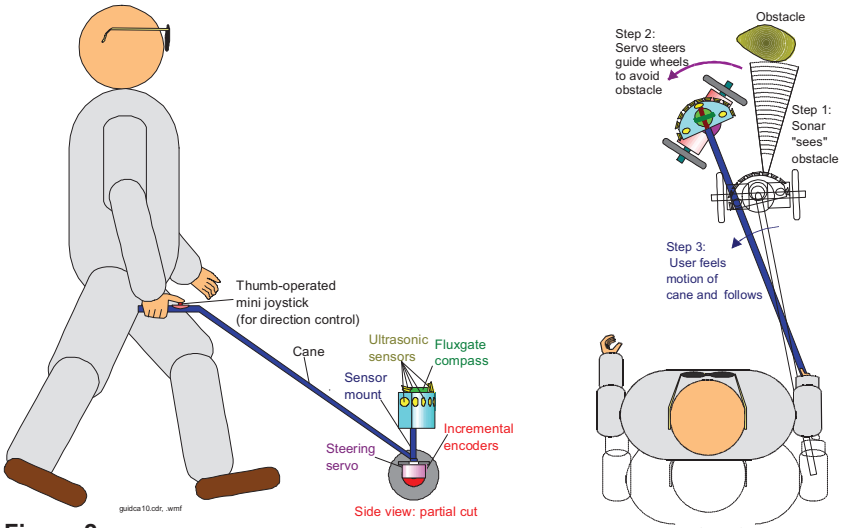
The UltraCane is a combination of ultrasonic sensors and a cane as shown in Fig. 2.3(c). The ultrasonic sensors are employed to detect objects with a maximum distance of 4 m in upward and forward directions. The forward and rear vibrators of this device indicate the information of head-level objects and ground-to-chest level objects, respectively.

Compared with Sonicguide, Miniguide and UtraCane, the NavBelt and GuideCane are more complex. Both NavBelt and GuideCane use multiple ultrasound sensors to detect simultaneously obstacles in different directions. Furthermore, the devices employ a computer to estimate the travel direction for the user in avoiding obstacles. These devices are considered as robotic guiders.

The NavBelt employs 8 ultrasound sensors integrated into a belt as shown in Fig. 2.3(d), and each sensor detects objects in an angle range. The NavBelt has two modes: the guidance mode and the image mode. For the guidance mode, the device provides the information of the travel direction and speed for the user via auditory signals. For the image mode, the device creates an acoustic panoramic image of the surroundings using stereophonic effects. The direction of objects is

represented by delivering sound signals from the right ear to the left ear. The volume of sound signals denotes the distance to objects.

The GuideCane is designed to hold like a white cane as shown in Fig. 2.4. When an object is found in the forward path, the GuideCane guides the user by changing its moving direction as a guide dog. The device includes a handle (cane) and a main unit. The main unit contains from 10 to 16 ultrasound sensors, a computer, wheels and a steering mechanism. The GuideCane can detect small objects at ground levels and sideways objects such as walls.



(a) (b)

Figure 2.4: The GuideCane prototype [12]: (a) schematic, (b) operation.

Ultrasound devices effectively detect hard objects that have a good reflection of ultrasonic waves. However, ultrasound devices have low detection accuracy for soft obstacles, which absorb ultrasonic waves. Furthermore, ultrasound devices are sensitive to ultrasound from environmental sources such as the air brakes on buses and lorries [20]. Another disadvantage is that ultrasound devices are unable to detect drop-offs, e.g. steps going down and the edges of platforms, which cause significant hazards for blind travelers.

In general, both laser and ultrasound devices detect successfully objects where the signals emitted from the devices are reflected back well. However, these devices are ineffective in finding obstacles that absorb the emitted signals. Furthermore, the devices employ a line of sight propagation, and therefore for crowded

environments, obstacle detection is affected significantly.

2.3.2 GPS-based systems

Many assistive navigation systems for blind people have been developed by combining the global positioning system (GPS) with the geographic information systems (GIS) [14–16]. These navigation systems assist the blind travelers in determining routes, locations and landmarks during their journey.

Examples of GPS-based systems for the visually impaired include Personal Guide System (PGS) [14], MoBIC system [15], Trekker Breeze [27], BrailleNote GPS [28] and StreetTalk [29] as shown in Table 2.2. The Personal Guide System employs a GPS receiver and a compass for determining the position and orientation of the traveler, and a GIS database for finding routes, landmarks and locations. The output of this system includes two options: a conventional speech display and a virtual acoustic display. The virtual acoustic display represents the positions of the surrounding objects in the 3D sound space. The conventional speech display provides spoken instructions for traveling and spoken descriptions of the surrounding objects.

In addition to providing travel guides, the MoBIC system developed by Petrie *et al.* allows users to explore the journey before traveling [15]. The system consists of two major components: MoBIC pre-journey and the MoBIC outdoor. The MoBIC pre-journey component employs GIS databases to assist users in planning journeys and exploring the environments where the journeys will take place, e.g. routes, landmarks, public transportation systems and facilities. The MoBIC outdoor component is considered as a travel guider. Based on the planned journey and the user's position provided by a GPS receiver, this component determines the travel direction, the surrounding landmarks and locations using GIS databases. The output of the system are spoken messages.

Trekker Breeze [27], BrailleNote GPS [28], and StreetTalk [29] are portable GPS devices for visually impaired people. These devices are a combination of a PDA computer and a GPS receiver. The devices provide travel instructions and information about the surrounding locations and landmarks. The Trekker Breeze is a hand-held device and combines all its components into a package. This device

Table 2.2: Examples of GPS-based systems for visually impaired people.

Systems	**Interface and components**	**Advantages**	**Disadvantages**
Personal Guide System (PGS) [14]	A GPS receiver, compass, computer and headphones	Portable; determining routes, landmarks and locations; providing travel instructions	Separated components and not convenient for users; blocking the user's hearing.
MoBIC system [15]	A computer, a GPS receiver, a compass and mobile telecommunications facilities	Allowing users to plan a journey; providing the information of surrounding environments and travel instructions	Separated components and not convenient for users; blocking the user's hearing.
Trekker Breeze [27]	A PDA integrated with a internal GPS receiver and a internal speaker	Small and light-weight package; determining the surrounding objects and locations; providing travel instructions; recording routes, locations and landmarks for using later.	Routes limited to the map of the device.
BrailleNote GPS [28]	A PDA combined with an external GPS receiver; two options for output: speech and Braille.	Providing travel instructions and information about the surrounding environments; allowing users to access the maps, internet and email.	Software is only compatible with specific Windows CE devices.
StreetTalk [29]	A PacMate PDA combined with a external GPS receiver; two options for output: speech and Braille	Allowing users to plan a journey; providing travel instructions and information about the surrounding environments.	Software is developed from a GPS program for sighted people and not all functions are accessible; not allowing for input of personalized points of interest or personalized routes [20].

supports recording the traveled routes and landmarks for using later. The output of the Trekker Breeze is spoken messages. The BrailleNote GPS consists of a PDA computer and an external GPS receiver. The BrailleNote GPS has two options (speech and Braille) for output and two options (Braille or QWERTY keyboards) for input. The StreetTalk employs a PacMate PDA and a external GPS receiver, and is developed from a navigation program for sighted people. The StreetTalk allows users to plan a journey and supports two output options: speech and Braille.

There are also several GPS-based systems called tele-assistance systems that allow a remote sighted person to guide blind pedestrians [30, 31]. These systems employ the combination of the GPS technology, GIS and 3G mobile networks.

The systems includes two major units: a mobile unit taken by the traveler and a stationary unit at the place of the guider. The mobile unit includes a camera to record the scene image in front of the traveler and a GPS receiver to determine the traveler's position. The stationary unit used by a sighted guider consists of a personal computer with a GIS database and a display to present the scene image and the traveler's location on the map. Based on the received information, the guider gives instructions to the user. The communication between these units is implemented over the 3G mobile network.

GPS-based systems have several limitations. They are often expensive due to the cost of the hardware and maps. Furthermore, the GPS signals are often disrupted when the user travels between tall buildings or under dense foliage [32]. Another disadvantage is that the GPS-based systems cannot detect immediate changes of the surroundings (e.g. moving objects).

2.3.3 Computer-vision based systems

Computer-vision based systems use images captured from cameras to obtain information about the environment. Many studies have been conducted on assistive navigation of blind people [33–35]. Table 2.3 shows examples of computer-vision based systems for assistive navigation.

Several assistive systems are designed to convert the captured image into other modalities for representing the 2D structure of scenes [33,34]. For example, the vOICe system is designed to transform the image into a sound-scape [33]. Each pixel is represented by a sinusoidal tone, where each audible frequency corresponds to a vertical position, each time corresponds to a horizontal position, and amplitude levels denote bright levels. The vOICe system consists of a digital camera attached to eyeglasses, headphones and a portable computer installed the software as shown in Fig. 2.5. Capelle *et al.* also proposed an assistive system similar to the vOICe, but the resolution and refresh rates of this system are higher than the vOICe [40].

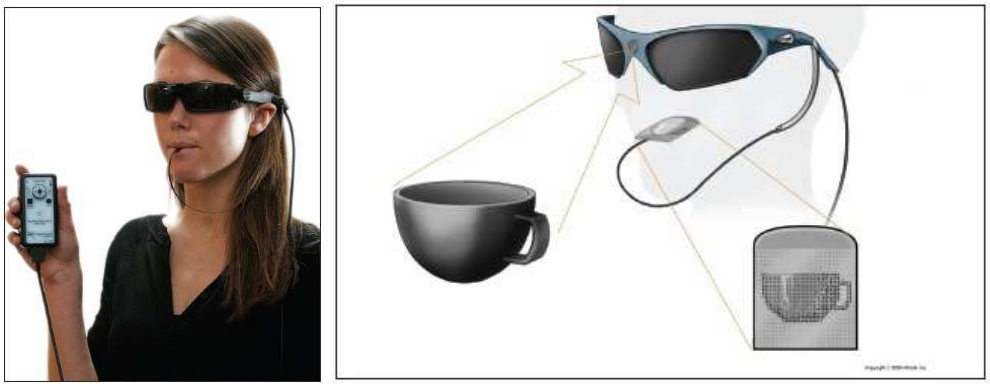
Instead of using the sound-scape, the BrainPort vision device converts the image into a pattern of gentle electrical stimulation [34]. The device includes a postage-stamp-size electrode array to put on the top surface of the tongue

Table 2.3: Examples of computer-vision based systems.

Systems	**Functionality**	**Interface and components**	**Advantages**	**Disadvantages**
vOICe [33]	Representing acoustically the environment	A digital camera embedded in eyeglasses; headphones; portable computer	Portable.	Require extensive training; blocking the user's hearing; only represent the 2D structure of scenes.
BrainPort [34]	Representing the environment by gentle electrical stimulation	A camera mounted on sunglasses; a postage-stamp-size electrode; a hand-held controller	Portable; does not block user's hearing.	Require extensive training; only represent the 2D structure of scenes.
Virtual Acoustic Space [36]	Representing acoustically the environment	Two cameras embedded in eyeglasses; headphones; portable computer	Portable; small size; reconstructing the 3D space of the environment.	Require extensive training; blocking the user's hearing.
Electron-neural vision system [37]	Detecting obstacles; providing information by electric stimulation in both hands; each finger represents a zone in forward field of view.	Two cameras; a compass; a laptop with GPS; gloves with stimulators in each finger.	Real-time performance, does not block user's hearing.	blocks using hands; does not detect objects at head and ground levels.
Tactile Vision System [38]	Detecting obstacles; representing obstacles by vibrations	A belt with 14 vibrators; two cameras; laptop	Does not block hands and hearing; operating in real time	Does not detect objects at head and ground levels.
Tyflos [35]	Detecting obstacles; representing obstacles by vibrations across chest	Vest with a 4×4 vibrator array; two cameras; laptop; range sensors	Does not block hands and hearing; detecting obstacles at different height levels.	No tests on real users .
Virtual White Cane [39]	Detecting obstacles; representing obstacles by vibrations of a smart phone	Android smart phone integrated with a laser pointer	Easy to use; does not block hands and hearing; detecting obstacles at head level.	Experiments done only on indoor paths.

Figure 2.5: The vOICe system [33].

as an output, a camera mounted on sunglasses and a hand-held controller for settings and processing as shown in Fig.2.6(a). The captured image is sent to the controller to translate into a stimulation pattern for displaying on the tongue as demonstrated in Fig.2.6(b). The bright levels of pixels are represented by stimulation levels.



(a) (b)

Figure 2.6: The BrainPort vision device [34]: (a) implementation of the BrainPort, (b) illustration of object detection.

The vOICe and BrainPort systems are simple, lightweight and cheap. However, these systems require months of training for users because of the complicated patterns for representing the environments. Furthermore, the systems do not provide depth information for the user, and therefore obstacle detection is limited.

To represent the 3D space of the surrounding environments, many assistive systems apply the stereo-vision technique [36–38, 41]. In these systems, the depth map of the surroundings is obtained from the images, and then is conveyed to the user by different methods. For instance, Gonzalez-Mora *et al.* proposed a virtual acoustic space to represent the depth map [36]. The virtual acoustic space is a 3D

sound environment and is formed using head-related transfer functions (HRTF).

The electron-neural vision system designed by Meers and Ward employs electric stimulation on the user's fingers for representing the detected obstacles and landmarks [37]. Each finger indicates objects in a forward zone of view. Stimulation levels are proportional to the distances. Figure 2.7 demonstrates the electron-neural vision system.

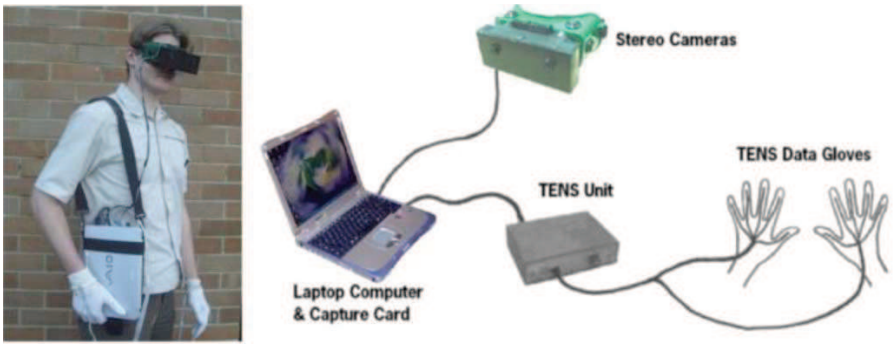


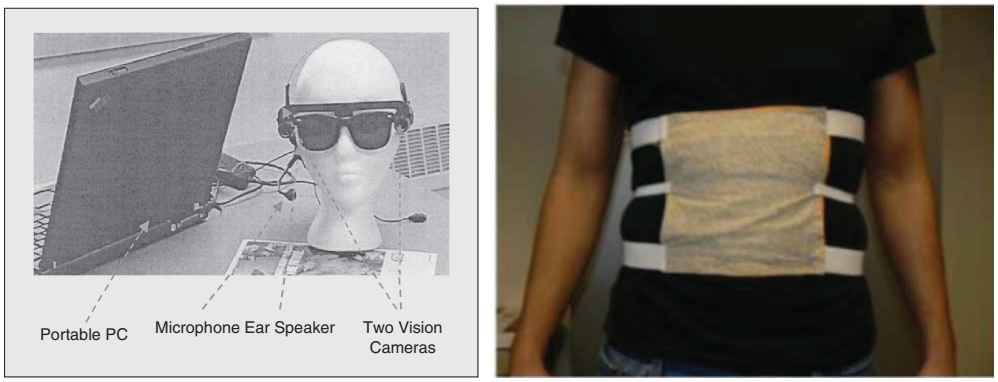
Figure 2.7: The electron-neural vision system and its components [37].

The tactile vision system of Johnson and Higgins conveys the information of obstacles to the user via a belt of 14 vibrators [38]. In the tactile vision system, the depth map is divided into 14 vertical regions. Each vibrator indicates the closest object in one vertical region, and each distance is represented by one vibration frequency. Figure 2.8 shows the tactile vision system and its components.

In another approach, several systems use a combination of cameras and laser sensors to find obstacles [35, 39]. For example, the Tyflos system employs combining two cameras with a range sensor to find dynamic changes and represents the traveler's surrounding environment in a 3D space [35]. The 3D space is reconstructed by fusing image data and range data, and converted then into a two-dimensional vibration array for users. Figure 2.9 shows the components of the Tyflos system.

The Virtual White Cane system detects objects using a smart phone and a laser pointer as shown in Fig. 2.10 [39]. The object's distance from the user is estimated from the image of the laser's reflection off the planar surface, using active triangulation. The image is captured by the camera of the smart phone.

Figure 2.8: The tactile vision system [38]: (a) the tactor belt, (b) the camera belt, (c) the tactile vision system.



(a) (b)

Figure 2.9: The Tyflos system [35,42]: (a) Tyflos prototype, (b) the 2D-dimensional vibration array vest.

The detected objects are conveyed to the user through vibration of the smart phone.

In addition to the above obstacle detection systems, several vision-based systems focus on detecting crossings of zebra patterns at traffic junctions for the visually impaired [1,17,43]. These systems apply the image processing and pattern recognition techniques to detect lane markers in the image captured by a

Figure 2.10: The Virtual White Cane system [39].

camera, and determine then the walking region for users. For example, Se proposes an assistive system to find the zebra-crossing lane in the image [1]. The walking lane is determined by finding lane markers using the Hough transform.

In [17], Ivanchenko *et al.* design the Crosswatch system that employs a mobile phone to find and locate zebra-crossings as shown in Fig 2.11. The authors proposed using a graphical model with geometric and color cues to find the borders of lane markers in the image captured from the phone camera.

Figure 2.11: The Crosswatch system [17].

In [43], Uddin and Shioyama employ the bipolar feature of zebra patterns

to segment the crossing region, and verify the detected crossing using the constant width periodic feature of white bands in a zebra-crossing. The system by Radvanyi *et al.* segments first the crossing region using color features, and then verifies the detected region by finding the lane markers with an adaptive thresholding technique [44].

In general, most existing assistive systems based on computer-vision focus on detecting obstacles. These systems do not provide assistance for detecting the travel path, despite the fact that this is a major task for traveling safely and independently in unknown environments.

2.4 Chapter summary

This chapter presents the mobility of vision-disabled people and reviews navigation aids for them. By relying mainly on touching and hearing, traveling safely and independently in various environments is a challenging task for visually impaired people. They can detect only objects and hazards in near spaces. Furthermore, they cannot determine locations, landmarks, travel orientations and routes for a journey in unknown environments.

Traditional aids are insufficient to support vision-disabled people for traveling safely and independently in different environments. Many technology systems have been developed to assist the visually impaired travelers in avoiding obstacles, determining locations and landmarks, and finding routes. The successes of these assistive technology systems have enhanced significantly the mobility of visually impaired people in outdoor environments. GPS-based systems assist vision-disabled users in determining locations, landmarks and routes, while vision-based systems and electronic detectors support the people finding obstacles. However, with most existing assistive systems, the users must find the travel path themselves in each scene.

Detecting pedestrian paths is a crucial and challenging task for travels of visually impaired people. Straying outside the walking region is dangerous for the blind travelers. However, there has been little work on detecting pedestrian lanes for assistive navigation of visually impaired people. Furthermore, existing

assistive systems are only designed to locate the crossing lanes of zebra patterns at traffic intersections. Therefore, an assistive system of detecting pedestrian lanes in different environments is essential for vision-disabled people.

Chapter 3

Marked-lane detection

Chapter contents

Work in this chapter has been published in

M. C. Le *et al.*, "'Pedestrian lane detection for assistive navigation of blind people"', in *International Conference on Pattern Recognition*, Tsukuba,Japan, 2012, pp 2594 - 2597.

M. C. Le *et al.*, "'Pedestrian lane detection for the visually impaired"', in *International Conference on Digital Image Computing: Techniques and Applications*, Fremantle WA, Australia, 2012, pp 1-6.

Automatically finding pedestrian crossing lanes at traffic intersections is a crucial and challenging task in assistive navigation for vision-disabled people. This chapter presents a robust method to detect pedestrian crossing lanes, which are

indicated by painted markers, in various scenes and under different illumination conditions. The proposed method is based on local image regions and Markov Random Fields to find the lane markers in the input image. The lane region is then determined by optimizing a criterion, which combines multiple geometric cues of lane markers. The proposed method is evaluated on a large data set collected from various scenes under challenging imaging conditions. The experimental results demonstrate the efficiency and robustness of the proposed method compared to existing techniques.

3.1 Introduction

Traffic intersections are one of the most dangerous places for blind travelers. Almost blind pedestrians cannot align themselves precisely with the crossing lane. An assistive system of detecting pedestrian crossing lanes is essential to help the travelers enter the lane in the right direction and avoid the dangers of straying outside the lane region. The system must cope with various scenes where painted markers are solid or dash stripes, eroded partially and affected by shadows and bright areas. Many different assistive devices have been developed for blind people to detect obstacles and find routes, landmarks and location as presented in Chapter 2. However, little work has been done on pedestrian crossing lane detection [1,17]. Furthermore, existing methods for detecting pedestrian crossing lanes are mostly designed for zebra-painted patterns [1,17,43]. To address this gap, this chapter focuses on automatic detection of pedestrian crossing lanes that are marked by two white stripes (markers).

Existing methods of detecting marked lanes for assistive navigation of autonomous vehicles and vision-disbaled people rely mainly on edge, intensity or color information of lane markers. Edge-based methods do not work well for scenes containing many extraneous edges [1,17,45], whereas intensity or color based methods are sensitive to illumination conditions [43,44]. In this chapter, we propose a new region-based method for detecting lane markers, which is motivated by the distinctive features of the lane-marker borders, and the robustness of Markov random fields in image analysis. The proposed method uses image

regions, called patches of interest or POIs, to represent the local appearance of lane markers, and a Markov random field (MRF) to model the structure of lane markers. We also propose verifying the detected lane markers based on lane scores, which represent the geometric characteristics of a walking region located by two lane markers.

The proposed method has several advantages in comparison with the existing methods. First, POIs are more informative and discriminative than edge and intensity or color information of individual pixels. The image patches capture the local appearance of both lane markers and road surface, and include also edge information. Second, MRF is more powerful than the traditional line-detection techniques in modeling the local and global structure of lane markers. In addition, applying the MRF on only POIs (instead of image pixels as in conventional image segmentation [46]) enhances the computational efficiency. Third, unlike several lane detection algorithms for autonomous vehicles [47, 48], our method does not require the prior knowledge such as camera parameters for verifying lane markers.

The remainder of the chapter is organized as follows. Existing work on detecting pedestrian lanes at traffic junctions and vehicle lane detection is presented in Section 3.2. Then, the proposed method for pedestrian crossing lane detection is described in Section 3.3. Next, experimental methods and results are discussed in Section 3.4. Finally, the chapter summary is given in Section 3.5.

3.2 Related work

This section presents a brief review of existing techniques for pedestrian crossing detection at traffic junctions. It also describes lane detection methods for autonomous vehicles in relation to pedestrian lane detection.

There are two major approaches for pedestrian lane detection. The first approach uses the edge map to find the borders of lane markers [1, 17, 45]. For example, Se *et al.* detected the borders by finding a group of convergent lines using the Hough transform [1, 45]. However, the Hough transform often fails to detect the desired lines in scenes that contain extraneous edges [17]. Instead of

using the Hough transform, Ivanchenko *et al.* extracted line segments from the linked edges, and then employed a factor graph model to find the borders [17].

The second approach exploits intensity and color information to segment the crosswalk region [43, 44]. For example, Uddin and Shioyama proposed finding image regions that have bipolar intensity patterns [43]. The candidate crossings were then verified noting that the white and black stripes have equal widths. Radvanyi *et al.* employed color information to segment the road surface region, and used intensity information to find the white stripes [44].

Similarly to pedestrian lane detection, methods for vehicles follow either an edge-based approach [6, 47, 49] or a color/intensity-based approach [48, 50, 51]. Edge-based methods first detect straight lines via the Hough transform, and then determine the lane boundaries according to a predefined lane model, such as straight lines [2, 47], splines [6], parabola [49, 52], or hyperbola [53]. Intensity-based methods use the contrast between lane markers and road surfaces. Examples in this category include intensity-bump (dark-bright-dark) filters [54, 55], steerable filters [56], ridge detectors [57], top-hat filters [58], and symmetrical local thresholding [50]. Other methods consider color cues in classifying marker pixels from road surface pixels. For example, Cheng *et al.* modeled the color probability density functions of two classes - *road surface* and *lane marker* [51]. They used three multivariate Gaussian distributions for the three dominant lane-marker colors (white, yellow, red). In [48], Kim formed a color feature vector from all pixels in an image block. The input image was decomposed into image blocks, and each block was classified as marker or non-maker by a neural network.

The edge-based methods are sensitive to background clutter, which is present in most real-world images. The intensity/color-based methods are less sensitive to clutter, but they are affected by weather and illumination conditions. In severe outdoor conditions (e.g. too bright or too dark scenes), the difference in color between lane markers and road surface is significantly reduced. In addition, unwanted regions, such as white cloud or bright cars, may have colors that are similar to the lane markers. Therefore, employing color or intensity information of only individual pixels is insufficient for robust detection of lane markers.

3.3 Proposed method

The proposed method for detecting pedestrian lanes at traffic junctions is based on finding lane markers in the input image. Figure 3.1 shows its block diagram, which comprises three main stages: POI extraction, lane marker detection, and lane detection. Each stage is described in more detail in the following subsections.

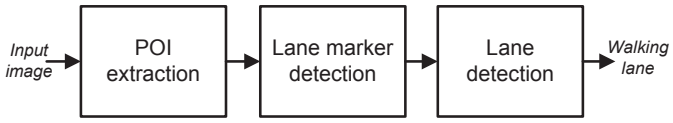


Figure 3.1: Block diagram of the proposed method.

3.3.1 Patch of interest extraction

Lane markers differ from road surfaces in appearance. Generally, lane markers have whitish color while the road surface has the greyish color of asphalt concrete. Furthermore, small regions located on the marker borders contain pixels from both the markers and road surface, and include edge orientation of the lane markers. Our method considers these local regions as *patches of interest* (see Fig. 3.2). Compared to edge pixels and inner regions of lane markers considered in [48], POIs are more distinct from the background scene.

When a POI is divided into 4 sub-regions P_1, P_2, P_3 and P_4, as shown in Fig. 3.3(a), there exist two diagonally opposite sub-regions having uniform brightness, e.g. P_2 (bright) and P_3 (dark). These sub-regions, which have the maximum contrast, belong to the lane marker and the road surface, respectively. This is a useful cue for finding POIs in the input image. In our method, the contrast between two sub-regions P_i and P_j is defined by $D_{ij} = \bar{I}_i - \bar{I}_j$, where $\bar{I}_i$ is the mean pixel intensity of sub-region P_i.

Consider an image patch X and let (P_u, P_v) denote the pair of diagonally opposite subregions having the maximum contrast $|D_{uv}|$, $(u, v) \in \{(1, 4), (3, 2)\}$. In our approach, an image patch X is represented by a feature vector combining

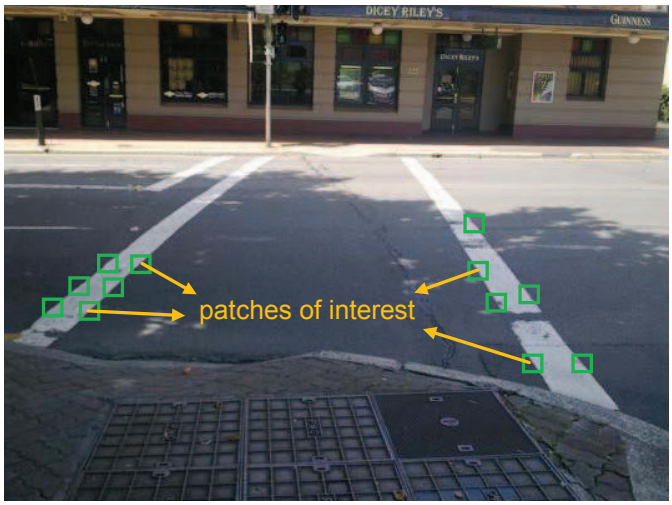


Figure 3.2: Example POIs used in the proposed method.

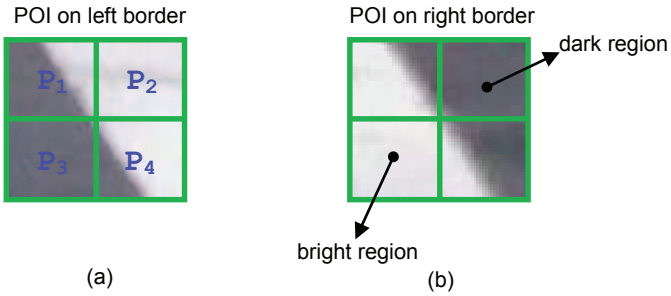


Figure 3.3: POIs divided into 4 sub-regions: (a) a POI located on left border, (b) a POI located on right border.

color appearance and edge orientation: $\mathbf{f} = (d_1, d_2, d_3, \theta)$. Here, d_1, d_2, and d_3 are the three color differences between sub-regions P_u and P_v:

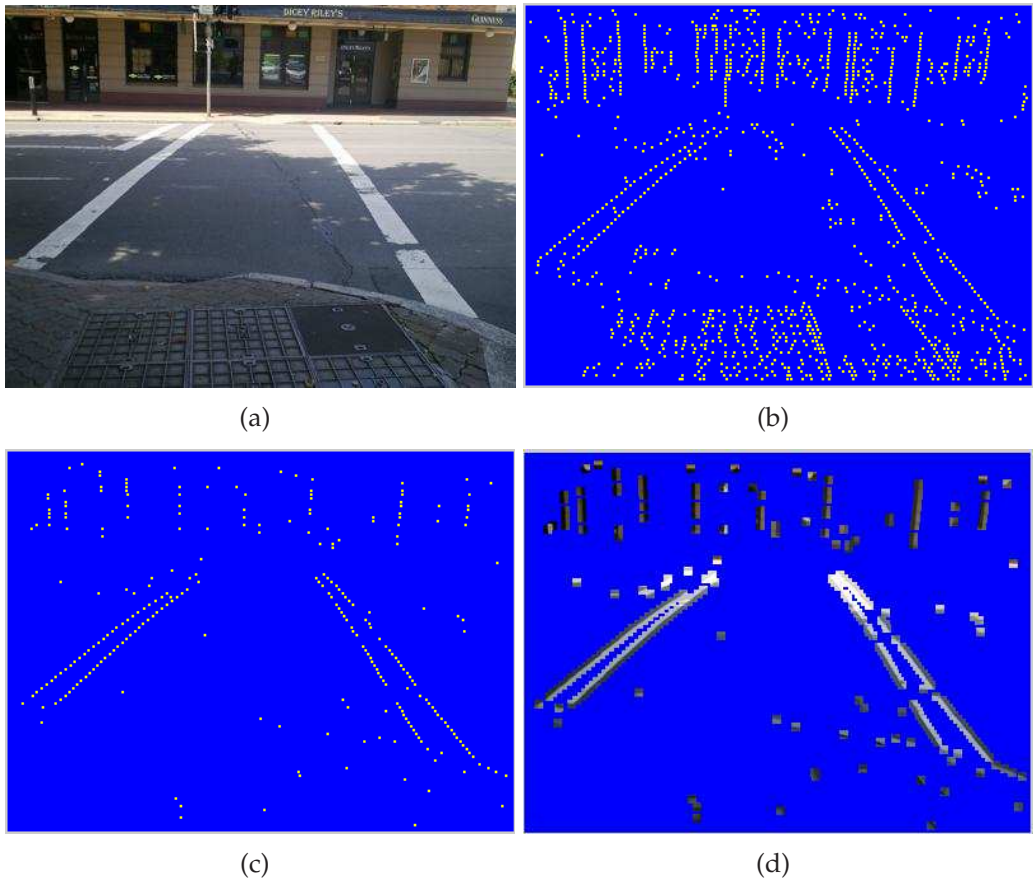
$$d_k = |\bar{c}_{uk} - \bar{c}_{vk}|, \text{ for } k = 1, 2, 3, \tag{3.1}$$

where $\bar{c}_{ik}$ denotes the average of the color component k of all pixels in sub-region P_i . The edge orientation angle θ of image patch X is estimated as the weighted average of the edge orientations of all pixels in X, where weights are the edge magnitudes.

Let $p(\mathbf{f}|\omega_1)$ and $p(\mathbf{f}|\omega_2)$ denote the probability density functions (*pdfs*) for two classes: *lane maker* ω_1 and *non-marker* ω_2. An image region X is considered as a POI if

$$\frac{p(\mathbf{f}|\omega_1)}{p(\mathbf{f}|\omega_2)} \geq \tau_c, \tag{3.2}$$

where τ_c is a predefined threshold and estimated from training.



(a) (b)

(c) (d)

Figure 3.4: Illustration of POI extraction: (a) input image, (b) anchor point map, (c) the locations of extracted POIs, (d) extracted POIs. See electronic color image.

To reduce the search space of POIs, we consider only image regions centered at *anchor* points; an *anchor* point is a pixel location where the gradient magnitude is a local maximum [59]. Figure 3.4(b) shows the anchor points extracted from the image shown in Fig. 3.4(a); clearly the search space of POIs is significantly reduced. Figures 3.4(c) illustrates the locations of the extracted POIs, which are shown in Fig. 3.4(d).

3.3.2 Lane marker detection

This section presents a method for detecting lane markers from the extracted POIs. A Markov random field is employed to model the structure of lane markers and label the POIs as *marker* or *non-marker*. The image patches of the marker class are then clustered into groups of connected components, and lane markers are finally verified from these POI groups.

We design a MRF model to find the lane markers as follows. Let $\mathcal{X}$ denote a set of the POIs extracted using the method presented in Section 3.3.1, $\mathcal{X} = \{X_1, X_2, ..., X_N\}$. Let $l_i \in \{\omega_1, \omega_2\}$ be the label of X_i; $l_i = \omega_1$ if X_i belongs to a lane maker, $l_i = \omega_2$ otherwise. The lane markers are determined by finding an optimal assignment of labels to POIs that minimizes the energy function of the MRF model. For an assignment of labels $L = \{l_1, l_2, ..., l_N\}$, the energy function is defined as

$$E(L) = \sum_{l_i \in L} \Big(\psi_1(l_i) + \gamma \sum_{X_j \in \mathcal{N}_i} \psi_2(l_i, l_j) \Big), \tag{3.3}$$

where $\mathcal{N}_i$ is the set of neighbors of X_i, $\psi_1(l_i)$ is the log likelihood function of POI X_i being assigned the label l_i, $\psi_2(l_i, l_j)$ is the co-occurrence prior of labels l_i and l_j, and γ is a positive constant.

Next, we describe how to determine the neighbors $\mathcal{N}_i$ of a POI X_i, the log likelihood $\psi_1(l_i)$, and the prior $\psi_2(l_i, l_j)$.

-Neighborhood: To determine if X_i and X_j are neighboring POIs, we use the following four parameters:

- $d(X_i, X_j)$ is the Euclidean distance between the centers of X_i and X_j.
- $\varphi(X_i, X_j)$ is the orientation difference between X_i and X_j,

$$\varphi(X_i, X_j) = \left| \frac{\theta_i + \theta_j}{2} - \theta_{ij} \right|, \tag{3.4}$$

  where θ_{ij} is the orientation angle of the line L_{ij} connecting the centers of X_i and X_j, and θ_i is the edge orientation angle of X_i.

- $s(X_i, X_j)$ is the correlation coefficient between X_i and X_j,

$$s(X_i, X_j) = \frac{\sum_k \left(X_i^k - \overline{X_i}\right)\left(X_j^k - \overline{X_j}\right)}{\sqrt{\sum_k \left(X_i^k - \overline{X_i}\right)^2 \sum_k \left(X_j^k - \overline{X_j}\right)^2}}, \tag{3.5}$$

  where X_i^k is the intensity of the k-th pixel of X_i and $\overline{X_i}$ is the mean intensity of X_i.

- $\delta(X_i, X_j)$ is the percentage of the lane marker pixels on the line L_{ij}. To determine lane marker pixels on the line, we employ the symmetrical local

thresholding technique proposed in [50]. A pixel $\mathbf{x}(x, y)$ on L_{ij} is considered as a marker pixel if its intensity is higher than an adaptive threshold λ,

$$\lambda = \max(\mu_-, \mu_+) + \tau_\lambda, \tag{3.6}$$

where μ_- and μ_+ are the mean intensity values of pixels (x_-, y) and (x_+, y), respectively. Here, $x_- \in [x - 2d_x, x]$ and $x_+ \in [x, x + 2d_x]$, d_x is the distance in the x-direction between the centers of X_i and X_j, and τ_λ is a positive parameter estimated from the training data set.

A POI X_j is considered as a neighbor of X_i ($X_j \in \mathcal{N}_i$) if the following conditions are satisfied:

(i) X_j is spatially close to X_i, i.e.

$$d(X_i, X_j) \leq \rho, \tag{3.7}$$

where ρ is a positive parameter and it is determined from the training data in relative to the image size.

(ii) X_j and X_i are located on the same lane marker, i.e.

$$\begin{cases} \varphi(X_i, X_j) \leq \tau_\varphi, \\ s(X_i, X_j) \geq \tau_s, \end{cases} \tag{3.8}$$

or

$$\begin{cases} \delta(X_i, X_j) \geq \tau_\delta, \\ s(X_i, X_j) \leq -\tau_s, \end{cases} \tag{3.9}$$

where τ_s, τ_φ and τ_δ are predefined thresholds that are determined from the training data set. When two POIs belong to the same lane marker, they either reside on the same border (see Fig. 3.5(a)), or on opposite borders (see Fig. 3.5(b)). The conditions in (3.8) are employed to verify when two POIs X_i and X_j are located on the same border. In this case, the two patches have a high correlation coefficient, i.e. $s(X_i, X_j)$ is high. Furthermore, the orientation of the line L_{ij} connecting X_i and X_j is similar to the edge orientations of these patches, and hence $\varphi(X_i, X_j)$ is low. The conditions in (3.9) are applied to verify when X_i and X_j are located on the two opposite borders. In this case, $s(X_i, X_j)$ is negative and $|s(X_i, X_j)|$ is high. In addition, $\delta(X_i, X_j)$ is high because most pixels on the line L_{ij} are lane marker pixels.

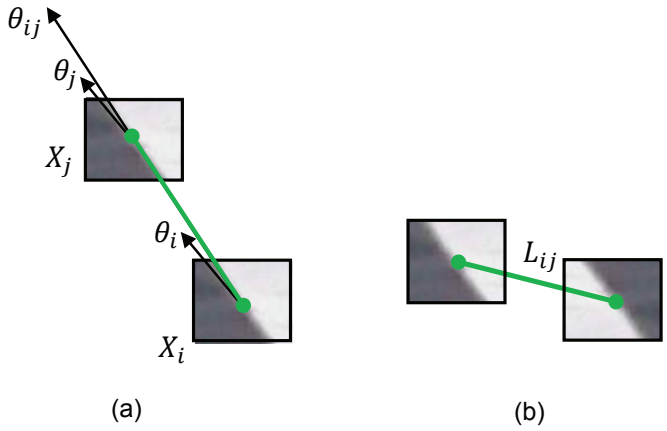


Figure 3.5: The spatial and appearance relations between two POIs of a marker: (a) two image patches located on the same border of a marker; (b) two image patches located on two different borders of a marker. Note that θ_{ij} is the orientation of the line L_{ij} joining the centers of X_i and X_j, θ_i and θ_j are the edge orientations of X_i and X_j, respectively.

-Log likelihood: The log likelihood function $\psi_1(l_i)$ is defined as

$$\psi_1(l_i) = -\log p(\mathbf{f}_i|l_i), \tag{3.10}$$

where $p(\mathbf{f}_i|l_i)$ is the class conditional probability density function, $\mathbf{f}_i$ is the feature vector of X_i and l_i is the label of X_i.

-Prior: The prior $\psi_2(l_i, l_j)$ is computed using the Potts model [60] as

$$\psi_2(l_i, l_j) = \begin{cases} 0, & \text{if } l_i = l_j, \\ 1, & \text{if } l_i \neq l_j. \end{cases} \tag{3.11}$$

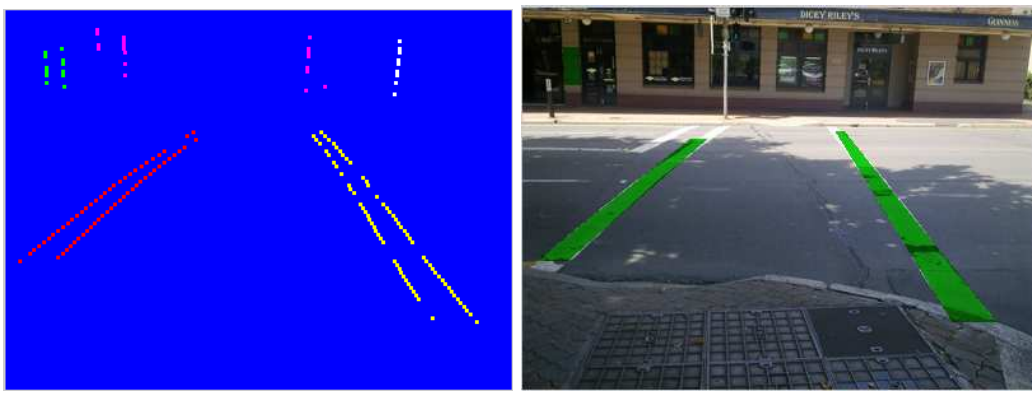
There are several approaches to finding the optimum assignment of labels to POIs, including simulated annealing [61,62], belief propagation [63], and iterated conditional modes (ICM) [64]. The simulated annealing method has a slow convergence rate, and therefore is not suitable for high-speed applications. The belief propagation algorithm is more suitable for tree structures, whereas our model involves cycles because an image patch X_i is linked to every node $X_j \in \mathcal{N}_i$. The ICM algorithm has a high convergence rate, and hence it is adopted in our implementation. The ICM algorithm for labeling POIs is described in Algorithm 1. This algorithm makes local optimum decisions, but its accuracy is sufficient for our task.

Algorithm 1 The ICM algorithm for labeling POIs.

{Initializing labels for POIs}
for each POI $X_i \in \mathcal{X}$ **do**
  if $p(\mathbf{f_i}|\omega_1) \geq p(\mathbf{f_i}|\omega_2)$ **then**
    $l_i \leftarrow \omega_1$
  else
    $l_i \leftarrow \omega_2$
  end if
end for
{finding the optimal labels for POIs}
$E_1 = \sum_{X_i \in \mathcal{X}} \Big(-\log(p(\mathbf{f_i}|l_i) + \sum_{X_j \in \mathcal{N}_i} \psi_2(l_i, l_j) \Big)$
continue $\leftarrow$ TRUE
Iternum $\leftarrow 0$
while (*continue*) **do**
  $E_2 \leftarrow 0$
  for each POI $X_i \in \mathcal{X}$ **do**
    $e_1 = -\log(p(\mathbf{f_i}|\omega_1) + \sum_{X_j \in \mathcal{N}_i} \psi_2(\omega_1, l_j)$
    $e_2 = -\log(p(\mathbf{f_i}|\omega_2) + \sum_{X_j \in \mathcal{N}_i} \psi_2(\omega_2, l_j)$
    if $e_1 \leq e_2$ **then**
      $l_i \leftarrow \omega_1$
      $E_2 \leftarrow E_2 + e_1$
    else
      $l_i \leftarrow \omega_2$
      $E_2 \leftarrow E_2 + e_2$
    end if
  end for
  Iternum $\leftarrow$ *Iternum* $+ 1$
  if *Iternum* $\geq$ *ITERMAX* **or** $|E2 - E1| \leq \epsilon$ **then**
    continue $\leftarrow$ FALSE
  else
    $E_1 \leftarrow E_2$
  end if
end while

After the POIs are labeled, non-marker patches are removed. The remaining POIs are considered as a graph, in which node X_i is linked to node X_j if $X_j \in \mathcal{N}_i$ or $X_i \in \mathcal{N}_j$. The graph is then clustered into several groups of POIs via connected component labeling. Figure 3.6(a) shows the POI groups found from the POIs in Fig.3.4(d). It can be seen that the proposed method is able to identify lane markers even when not all POIs are detected, e.g. markers highlighted in yellow in Fig. 3.6(a).

Each group of POIs is a potential lane marker, and is verified as follows. A



(a) (b)

Figure 3.6: Illustration of lane marker detection: (a) POI groups extracted, (b) lane markers detected (green regions). See electronic color image.

lane marker has a left and right border, and these borders are nearly straight. Therefore, a group of POIs is considered as a lane marker if it includes a pair of left and right borders. For each group of POIs, the borders are found using the RANdom SAmple Consensus (RANSAC) fitting technique [65]. When a POI is on the left border of a lane marker (see Fig. 3.3(a)), the contrast D_{uv} of sub-regions P_u and P_v is negative. On the other hand, if a POI is on the right border of a lane marker (see Fig. 3.3(b)), D_{uv} is positive. The left border is constructed from the patches where D_{uv} is negative. The right border is constructed from the patches where D_{uv} is positive. The groups without a pair of left and right borders are removed, and the remaining groups are considered as lane markers. Figure 3.6(b) shows the lane markers detected from the POI groups in Fig. 3.6(a).

3.3.3 Lane detection

This sub-section presents a lane detection method from a set of lane markers $\mathcal{M} = \{m_1, m_2, ...\}$ obtained in the previous stage. The walking region is identified by finding a pair of lane markers (m_i, m_j) that has the highest lane score.

For each pair of lane markers (m_i, m_j), three features are defined:1)the angle ϕ_{ij} between m_i and m_j ; 2) the orientation angle φ_{ij} of the bisector between m_i and m_j; 3) the vertical overlap o_{ij} between m_i and m_j. Figure 3.7 demonstrates the features of two lane markers.

Let m^t and m^b denote, respectively, the top and bottom coordinates of the

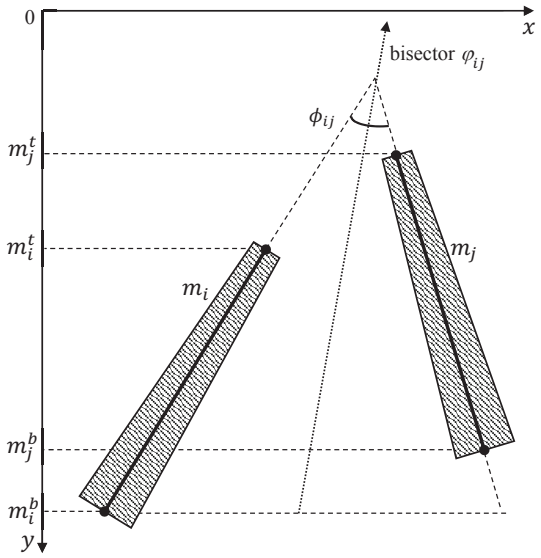


Figure 3.7: Geometric characteristics of two lane markers (shaded areas).

central line of a lane marker m, see Fig. 3.7. Note that the image coordinate is employed here, i.e. $(0, 0)$ is the left-top corner of the image. The vertical overlap o_{ij} between m_i and m_j is defined as

$$o_{ij} = \frac{\min(m_i^b, m_j^b) - \max(m_i^t, m_j^t)}{\max(m_i^b, m_j^b) - \min(m_i^t, m_j^t)}. \tag{3.12}$$

Given three features, the lane score of (m_i, m_j) is computed as

$$f(m_i, m_j) = f_1(\phi_{ij}) f_2(\varphi_{ij}) f_3(o_{ij}), \tag{3.13}$$

where the individual score functions are given by:

$$f_1(\phi_{ij}) = \frac{1}{\sigma_\phi \sqrt{2\pi}} \exp\left[\frac{-(\phi_{ij} - \mu_\phi)^2}{2\sigma_\phi^2}\right], \tag{3.14}$$

$$f_2(\varphi_{ij}) = \frac{1}{\sigma_\varphi \sqrt{2\pi}} \exp\left[\frac{-(\varphi_{ij} - \mu_\varphi)^2}{2\sigma_\varphi^2}\right], \tag{3.15}$$

$$f_3(o_{ij}) = \frac{1}{1 + \alpha e^{-\beta o_{ij}}}. \tag{3.16}$$

In Equations (3.14) to (3.16), μ_ϕ, σ_ϕ, μ_φ, σ_φ, α and β are estimated from the training data. The individual score functions are chosen to model the relationship between a feature and the lane score. Equations (3.14) and (3.15) are based on analyzing the distributions of ϕ_{ij} and φ_{ij} on the training set that these angles form

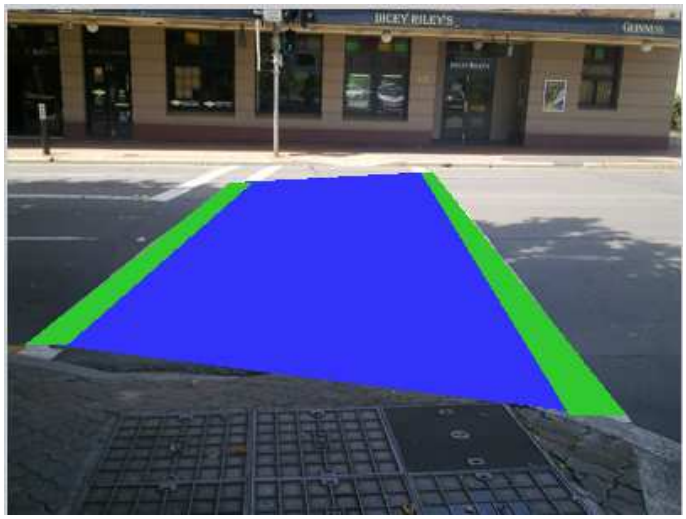


Figure 3.8: Illustration of lane detection. The lane markers are shown in green. The lane region are shown in blue.

approximately normal distributions. Equation (3.16) means that the higher is the vertical overlap o_{ij}, the higher is $f_3(o_{ij})$.

The optimal lane markers (m_i^*, m_j^*) are found by maximizing the lane score:

$$(m_i^*, m_j^*) = \underset{(m_i, m_j) \in \mathcal{M}^2}{\arg\max} f(m_i, m_j). \tag{3.17}$$

Finally, the optimal lane markers (m_i^*, m_j^*) obtained by (3.17) are considered as the lane markers of a pedestrian lane if

$$f(m_i^*, m_j^*) \geq \tau_m, \tag{3.18}$$

where τ_m is a predefined threshold and estimated from the training data.

Figure 3.8 shows the two lane markers determined from the three candidate markers in Fig 3.6(b). As demonstrated in Fig. 3.8, the Bayesian verification method selects correctly the true markers.

3.4 Experiments and Results

This section presents the experimental methods and results of the proposed pedestrian lane detection algorithm. First, the image data and the evaluation measures of lane and marker detection methods are described in Section 3.4.1. Then, the selection of the parameters is discussed in Section 3.4.2. The effectiveness of the

steps in the proposed method is evaluated in Section 3.4.3. Finally, the comparison between the proposed method and several existing methods is presented in Section 3.4.4.

3.4.1 Experimental methods

To evaluate pedestrian lane detection methods, we collected a data set of 2000 images of traffic crossings, with different backgrounds, times, and weather conditions. The data set includes pedestrian lanes affected by extreme illumination conditions (e.g. very low or very high illumination). In many cases, the markers are eroded partially or covered by shadows and lighting areas. Figure 3.9 shows several example images in this data set.

We manually annotated the lane markers for every image in the data set. Each lane marker was represented by a polygon. Lane regions were then obtained from the annotated markers. Figure 3.10 shows an example input image with annotated markers and annotated lane region. In the experiments, we used 600 images for training, and 1400 images for testing.

To evaluate the detection performance, the detected regions are compared with the annotated ground-truth regions. Let R_d denote a detected region and R_g denote a ground-truth region. The matching score between R_d and R_g is computed as

$$\chi(R_g, R_d) = \frac{|R_g \cap R_d|}{|R_g \cup R_d|}, \tag{3.19}$$

where $\cap$ denotes the intersection of R_d and R_g, $\cup$ denotes the union of R_d and R_g, and $|R|$ is the area of region R.

A detected region R_d is considered as a *correct detection* if there exists a ground-truth region R_g that satisfies

$$\chi(R_g, R_d) \geq \tau_e, \tag{3.20}$$

where τ_e is an evaluation threshold. Similarly to the evaluation of most object detection systems [66, 67], τ_e is set to 0.5 in our experiments.

The detection performance of a method is evaluated by three measures: recall, precision, and F-measure. *Recall* is the percentage of the ground-truth regions that are detected correctly. *Precision* is the percentage of the detected regions that are

Figure 3.9: Example images of pedestrian crossings in different conditions. Rows 1-2: pedestrian lanes with shadows or in bright lighting conditions. Row 3: pedestrian lanes with eroded markers. Row 4: pedestrian lanes with markers of dash patterns. Row 5: pedestrian lanes in dark lighting conditions.

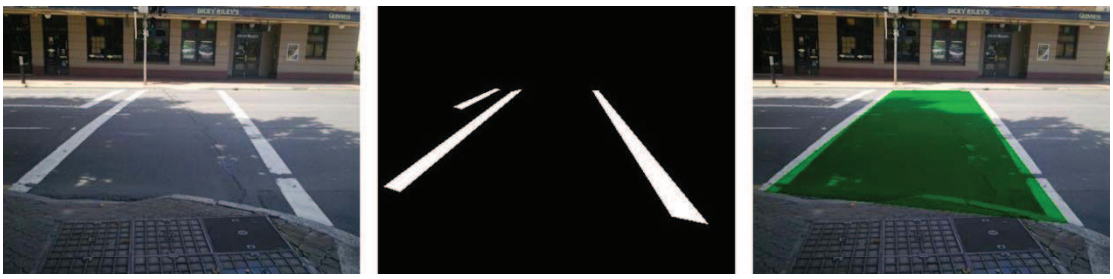
considered to be correct. *F-measure* is the harmonic mean of precision and recall:

$$F\text{-}measure = 2 \times \frac{Recall \times Precision}{Recall + Precision}. \tag{3.21}$$

Note that the above matching score and performance measures are applied to evaluate lane marker detection as well as lane detection.

3.4.2 Selection of parameters

This section presents the parameters used in the proposed method. These parameters were estimated from a training set comprising 600 images.



(a) (b) (c)

Figure 3.10: Example of ground-truth data for pedestrian lane detection: (a) input image, (b) annotated markers, (c) ground-truth lane region (green area).

The parameters τ_λ in (3.6), τ_s and τ_φ in (3.8), and τ_δ in (3.9) were determined by analyzing their distributions on the training set. Our experiments show that the distributions of these parameters are unimodal (see Fig. 3.11), and therefore these parameters were calculated using the thresholding technique proposed in [68]. For a unimodal distribution h, the threshold is selected as the bin k, which maximizes the perpendicular distance d_τ between the point (k, h_k) on the histogram and the line connecting the highest peak to the lowest non-zero point of the histogram, see Fig. 3.11(a).

The parameter ρ in (3.7) was estimated by using the width of lane markers. This parameter is used to verify the spatial constraint of two POIs in a lane marker; these POIs can be located on the same border or two opposite borders. On the training set, we have found that the width of lane markers is smaller than $0.15W$, where W is the width of the images. In our work, this parameter was chosen as $\rho = 0.15W$.

The paramters μ_ϕ and σ_ϕ in (3.14), μ_φ and σ_φ in (3.15), were estimated as the means and standard deviations of the angles ϕ_{ij} and φ_{ij} in the training set, as mentioned in Section 3.3.3.

The size $L \times L$ for POIs was found by evaluating the lane marker detection for different values of L, ranging from 3 to 11, relative to the image size of 400×600 pixels. Table 3.1 shows the performance of lane marker detection with various sizes of POIs on the training set. These results indicate that too small or too large L (e.g. $L = 3$ or $L = 11$) reduces the precision and recall rates. This is because

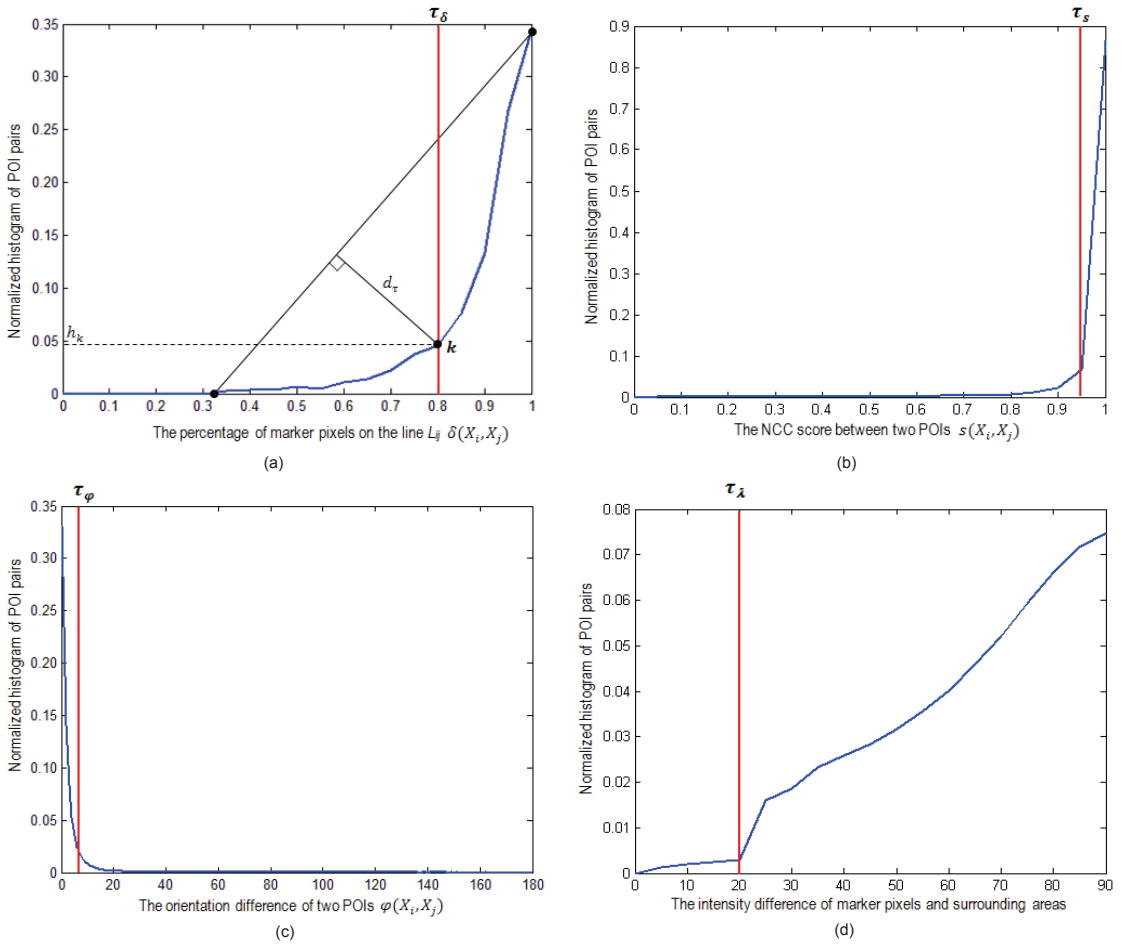


Figure 3.11: Determining the parameters using the training set: (a) distribution of marker pixel ratio $\delta(X_i, X_j)$ on the line L_{ij} connecting two POIs, (b) distribution of NCC score $s(X_i, X_j)$, (c) distribution of orientation difference $\varphi(X_i, X_j)$, (d) distribution of the intensity contrast between the marker pixels and surrounding areas. The red vertical lines represent the selected values.

too small POIs will contain insufficient information of the lane marker and road surface patterns, whereas too large POIs will include background information. Furthermore, the computational speed is slow when L is high. The best performance of lane marker detection was obtained when $L = 7$; this value was chosen in our experiments.

The remaining parameters, i.e. τ_c in (3.2), γ in (3.3), α and β in (3.16), were also selected by analyzing the performance of the proposed method on the training set. Our experiments indicate that these parameters do not affect the detection performance significantly. Table 3.2 summarizes the algorithm parameters with their selected values.

Table 3.1: The performance of the lane marker detection for different sizes of POIs on the training set.

POI size L	3	5	7	9	11
Recall (%)	86.3	94.4	97.6	95.2	94.4
Precision (%)	79.3	95.1	93.8	89.4	93.6
F-measure(%)	82.6	94.7	95.7	92.2	94.0
Processing time(s)	0.63	0.54	0.81	1.22	1.87

Table 3.2: Algorithm parameters and corresponding values.

Parameter	*Equation or Condition*	*Value*
τ_c	(3.2)	0.6
τ_λ	(3.6)	20
τ_s	(3.8)	0.95
τ_φ	(3.8)	5
τ_δ	(3.9)	0.8
ρ	(3.7)	0.18W
γ	(3.3)	3.5
μ_ϕ	(3.14)	78.9^o
σ_ϕ	(3.14)	16.9^o
μ_φ	(3.15)	91.5^o
σ_φ	(3.15)	19.7^o
α	(3.16)	0.9
β	(3.16)	2.3
τ_m	(3.18)	0.002

The class conditional probability density functions $p(\mathbf{f}|\omega_1)$ and $p(\mathbf{f}|\omega_2)$ in (3.2) and (3.10) were estimated by using the histogram approach on the training set. The color difference d_i $(i = 1, 2, 3)$ and the orientation θ were quantized into 32 bins and 9 bins, respectively.

3.4.3 Analysis of the proposed processing steps

Experiments were conducted on the test set to evaluate the effects of individual stages in the proposed method. The test set includes 700 images of pedestrian lanes with solid markers and 700 images of pedestrian lanes with dash markers. Our method has three major stages: POI extraction, lane marker detection, and lane detection.

To evaluate the impact of POI extraction, we implemented the lane marker detection *with* POI extraction and *without* POI extraction. Table 3.3 presents the lane marker detection performances in these two cases. Without POI extraction, all $L \times L$ image regions centered on *anchor* points were considered. The lane marker detection achieved a recall rate of 90.6% and a precision rate of 80.2% (F-measure

= 85.1%). With POI extraction using (3.2), the recall rate increased to 92.7% and the precision rate increased to 81.2% (F-measure = 86.6%). Furthermore, using POI extraction, the average computational time of lane marker detection was reduced from 1.37 second to 0.66 second. Clearly, using POI extraction removed a large number of background image patches, and enhanced both the accuracy and computational efficiency of lane marker detection.

Table 3.3: The performances of lane marker detection with POI extraction and without POI extraction.

Algorithms	*Recall (%)*	*Precision (%)*	*F-measure (%)*	*Average time (s)*
Without POI extraction	90.6	80.2	85.1	1.37
With POI extraction	92.7	81.2	86.6	0.66

For lane marker detection, the effectiveness of using the MRF was assessed as follows. We also implemented an approach that does not use MRFs to detect lane markers, based on the algorithm proposed in [4]. In this approach, line segments are first found from the extracted POIs using the RANSAC algorithm. Then, lane markers are determined from pairs of detected lines employing shape and intensity information. This approach achieved a recall rate of 87.8% and a precision rate of 74.8% (F-measure = 80.8%). In our proposed approach using the MRF, the recall rate was 92.7%, and precision rate was 81.2% (F-measure = 86.6%). These results show the robustness and effectiveness of using MRFs for detecting lane markers. Note that our MRF model could successfully detect lane markers even when several patches of interest are missing.

For lane detection, the proposed method achieved a recall rate of 94.4% and a precision rate of 97.2% (F-measure = 95.8%). Compared with lane marker detection, both the recall and precision rates of lane detection are improved. This is because not all the lane markers contribute to the lane regions. For example, Fig. 3.10 indicates that only two of the three markers belong to the pedestrian lane region. Furthermore, the matching score in (3.19) is different when used for markers and lane regions. Figures 3.12 and 3.13 present several sample results of lane detection. These results show the robustness of the proposed method in detecting pedestrian lanes, under challenging conditions such as background

clutter, low image contrast, and eroded markers.

Table 3.4: The processing time of individual steps in the proposed method.

Steps	*Processing time (s)*
POI extraction	0.38
Lane marker detection	0.28
Lane detection	0.01
Total time	0.67

We also evaluated the computational speed of the proposed method. Table 3.4 shows the average processing time of each stage in the proposed method. Our MATLAB implementation of the proposed method took an average time of 0.67 second for an image of 400×600 pixels on a PC with 3.7 GHz CPU. We have found that the processing time of the proposed method is sufficient for assistive navigation of blind people, but it could be further optimized.

3.4.4 Comparison with existing methods

The proposed method was compared with several existing methods:

- *Hough transform -based method* [1,2]: This approach uses the Hough transform (HT) to detect straight lines on the edge map of the input image, and then finds the vanishing point. Based on the vanishing point, the borders of lane markers are detected. This technique is referred to as "Edge + HT". In the experiments, the distance and orientation resolutions in the Hough transform, and the parameters of the method were tuned using the training data.

- *Segmentation-based method* [3]: This is our previous work. In this method, lane markers are extracted by using color and intensity information of pixels. The lane region is then verified by a probabilistic framework using geometric cues extracted from the lane markers. This method is referred to as "Segmentation". In the experiment, the parameters of this method was adjusted to suit the best performance using the training set.

- *POI-based method* [4]: This is also previous work. This algorithm extracts first POIs from the input image, using the normalized cross correlation template

matching. The RANSAC algorithm is applied then on the centers of POIs to locate the borders of markers. Next, lane markers are determined from the detected borders by using shape and intensity information. Spatial layout of markers is finally used to find the lane region. This algorithm is referred to as "NCC POI + RANSAC". In the experiment, the parameters of the algorithm was chosen to obtain the best performance using the training set.

We evaluated the algorithms on two different test sets: 700 images of pedestrian lanes with solid markers (see Fig. 3.12) and 700 images of pedestrian lanes with dash markers (see Fig. 3.13). Table 3.5 summarizes the detection performance of the different methods on the test set of pedestrian lanes with solid markers. Table 3.6 shows the detection performance of the methods on the test set of pedestrian lanes with dash markers.

Table 3.5: Comparison of algorithms for detecting pedestrian lanes with solid markers.

Methods	*Recall (%)*	*Precision (%)*	*F-measure (%)*	*Average time (s)*
Edge + HT [1,2]	76.6	81.4	78.9	0.20
Segmentation [3]	88.2	90.1	89.1	1.32
NCC POI + RANSAC [4]	89.0	92.9	90.9	1.78
Proposed method	**94.3**	**97.4**	**95.8**	**0.71**

Table 3.6: Comparison of algorithms for detecting pedestrian lanes with dash markers.

Methods	*Recall (%)*	*Precision (%)*	*F-measure (%)*	*Average time (s)*
Edge + HT [1,2]	59.7	65.1	62.3	0.20
Segmentation [3]	62.2	81.3	70.5	0.96
NCC POI + RANSAC [4]	81.7	92.4	86.7	1.56
Proposed method	**94.5**	**97.0**	**95.7**	**0.64**

For detecting pedestrian lanes with solid markers, the proposed method with a recall rate of 94.3%, a precision rate of 97.4% and F-measure of 95.8%, significantly outperformed the "edge + HT" method, which had a recall rate of 76.6%, a precision rate of 81.4% and F-measure of 78.9%. The proposed method also achieved better recall and precision rates compared with the "segmentation" method (recall rate of 88.2%, precision rate of 90.1% and F-measure of 89.1%) and "NCC POI +

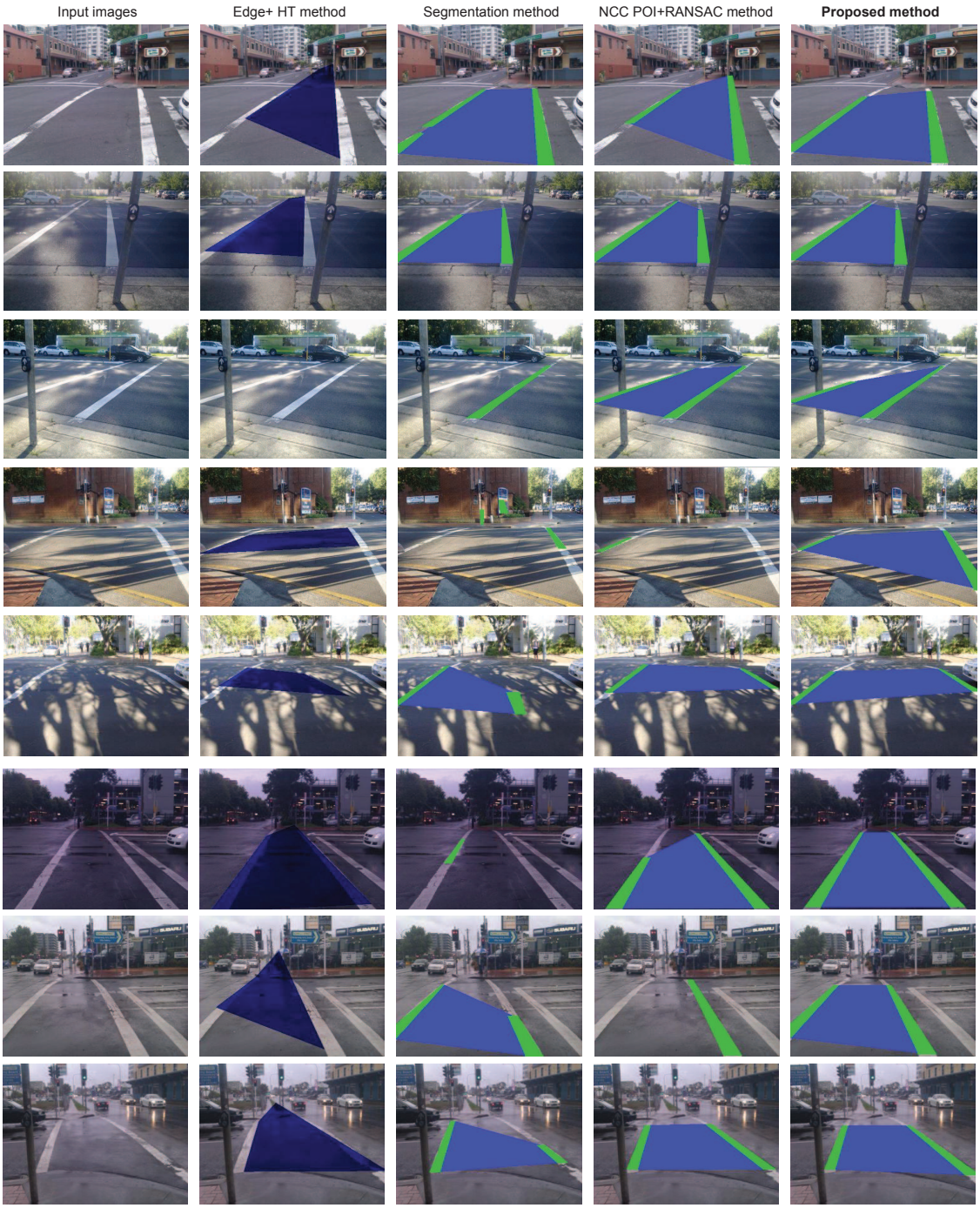


Figure 3.12: Visual results of detecting pedestrian lanes with solid markers. Column 1: input images. Column 2: lane regions detected by the edge + HT method [1, 2]. Column 3: lane regions detected by the segmentation method [3]. Column 4: lane regions detected by the NCC POI + RANSAC method [4]. Column 5: lane regions detected by the proposed method. Detected lane regions are blue areas.

RANSAC" method (recall rate of 89.0%, precision rate of 92.9% and F-measure of 90.9%).

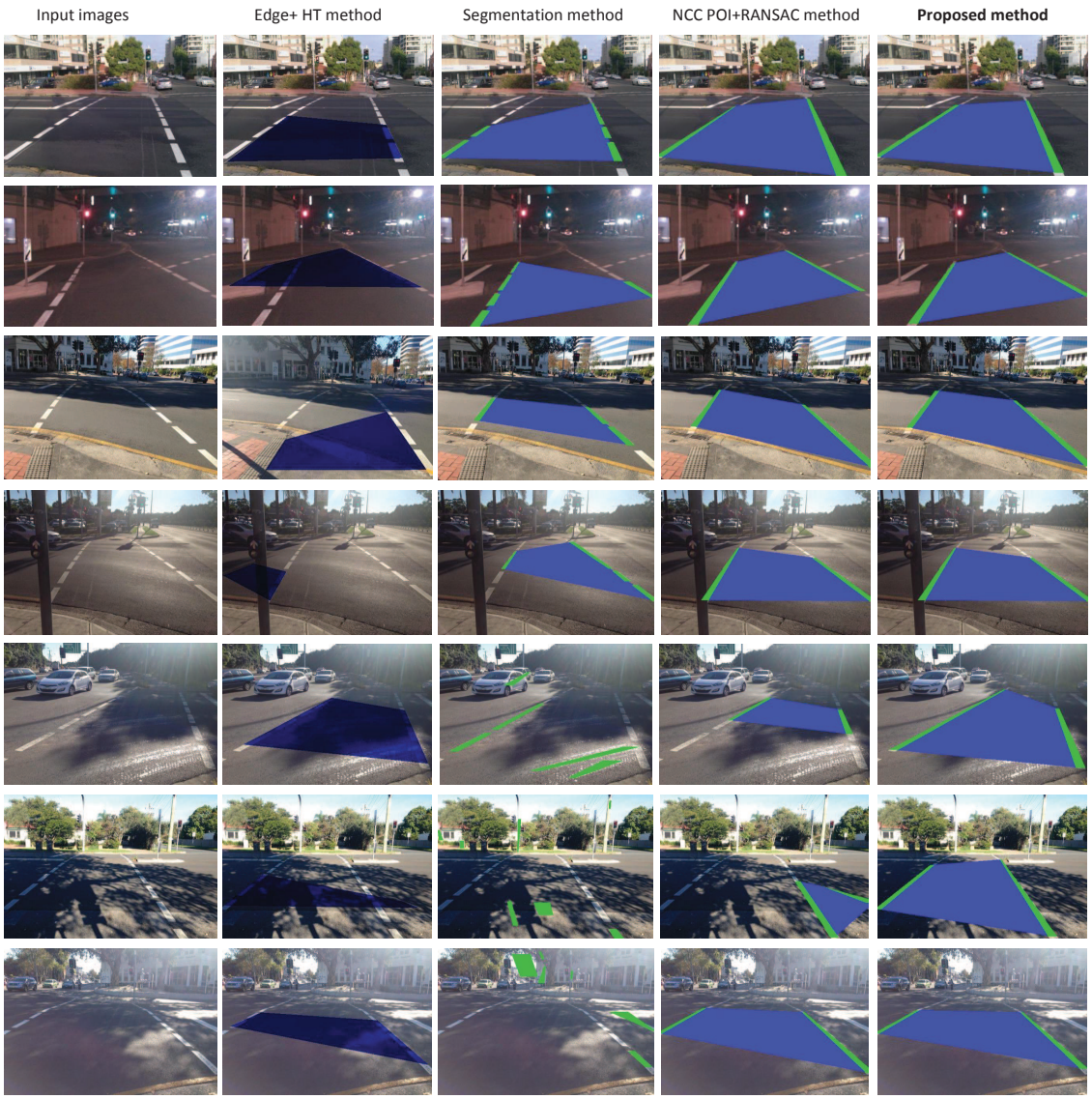


Figure 3.13: Visual results of detecting pedestrian lanes with dash markers. Column 1: input images. Column 2: lane regions detected by the edge + HT method [1, 2]. Column 3: lane regions detected by the segmentation method [3]. Column 4: lane regions detected by the NCC POI + RANSAC method [4]. Column 5: lane regions detected by the proposed method. Detected lane regions are blue areas.

For detecting pedestrian lanes with dash markers, the recall and precision rates and F-measure of the proposed method (94.5%, 97.0% and 95.7%) are clearly higher than those of the "edge + HT" method (59.7%, 65.1% and 62.3%) and the "segmentation" method (62.2%, 81.3% and 70.5%). These results indicates that the POIs have more discriminative power in representing the lane markers compared with the individual edge pixels used in the "edge + HT" method and color information of individual pixels employed in the the " segmentation" method.

The proposed method also outperformed the "NCC POI + RANSAC" method, which had a recall rate of 81.7%, a precision rate of 92.4% and F-measure of 86.7%).

In addition, the average computational time of the proposed method (0.67 second) for an image of 400×600 pixels was significantly shorter than that of the "NCC POI + RANSAC" method (1.67 second) and "segmentation" method (1.16 second). The "NCC POI + RANSAC" method had a high computational time because it requires the NCC template matching for POI extraction. Sample output images presented in Fig. 3.12 and Fig. 3.13 also demonstrate the robustness of the proposed method in comparison with the other methods.

3.5 Chapter summary

This chapter presents a new method for detecting pedestrian crossing lanes at traffic junctions. The proposed method detects lane markers using POIs and Markov random fields. Lane marker verification are based on lane scores that combines geometric features of lane markers. Evaluation results on a large data set with detection ground-truth have shown that the proposed method is able to detect robustly pedestrian crossing lanes under challenging environmental conditions. It also significantly outperforms other existing methods.

Chapter 4

Unmarked-lane detection

Chapter contents

Work this chapter has been published in

M. C. Le *et al.*, "'Lane detection in unstructured environments for autonomous navigation systems'", in *Asian Conference on Computer Vision*, Singapore, 2014.

M. C. Le *et al.*, "'Pedestrian lane detection in unstructured environments for assistive navigation'", in *International Conference on Digital Image Computing: Techniques and Applications*, Wollongong NSW, Australia, 2014.

The content of this chapter is currently under review in

M. C. Le *et al.*, "Pedestrian lane detection in unstructured environments", *CVIU Special Issue on Assistive Computer Vision and Robotics*, 2015.

Automatic lane detection becomes more difficult in unstructured environments where lanes vary significantly in appearance, have different shapes and

are not indicated by painted markers. This chapter proposes a new method to find pedestrian lanes in such unstructured environments. The proposed method detects the walking lane in the input image using both appearance and shape information. In our method, the appearance model of the lane is learned on-the-fly from a sample region, which is determined automatically using the vanishing point and the properties of lane borders and lane surfaces. The shape of pedestrian lanes is modeled by shape contexts. This chapter also introduces a fast and robust vanishing point estimation method using local orientations of color edge pixels. The proposed pedestrian unmarked-lane detection method is evaluated on a new data set, collected from various indoor and outdoor scenes with different types of unmarked lanes. Experimental results and comparisons with other existing methods on the new data set have shown the efficiency and robustness of the proposed method.

4.1 Introduction

Lane detection has a crucial role in assistive navigation for blind people, autonomous vehicles and mobile robots. Automatically finding lanes in unstructured environments is a challenging task, because the system must cope with variations in the scene, the illumination condition, and the lane type. A large number of studies have been carried out on vehicle lane detection for assistive navigation of autonomous cars [6, 7, 48, 69–71]. However, there has been little work on pedestrian lane detection for assistive navigation of visually impaired people [3, 17, 43]. Furthermore, most existing pedestrian lane detection methods are designed to find pedestrian crossing lanes that are identified by painted markers [3, 4, 17, 43, 45]. To address this gap, this chapter focuses on vision-based detection of pedestrian lanes that have no painted markers in different indoor and outdoor scenes, under varying illumination conditions and lane surfaces.

Most existing algorithms for unstructured (i.e. unmarked) lane detection employ the appearance features (e.g. color and texture) of lane surfaces to classify the lane pixels from the background [69, 72–74]. These algorithms require off-line training, and hence the classification performance decreases when the lane

appearance differs from the training data. In practice, the lane appearance varies significantly due to different lane surfaces or illumination conditions. Other existing algorithms find the lane boundaries from edges directing to the vanishing point, using edge features (e.g. color and orientation) [7] or the color difference between the lane region and non-lane regions [75]. However, these algorithms are sensitive to background clutter.

In this chapter, we propose a new method to detect unmarked pedestrian lanes using both color, edge, and shape features. In contrast to the existing methods, the proposed method constructs a lane model dynamically using only the input image, and is therefore more adaptive to different illumination conditions and lane surfaces. The main contributions of the chapter can be briefly described as follows:

- Firstly, we propose an improved vanishing point estimation method using local orientations of color edge pixels. Estimating the vanishing point from edge pixels is more efficient than from all pixels as in the existing methods [7, 75]. In addition, to estimate local orientations and edge pixels more robustly, we apply the color tensor on multiple color channels, instead of relying on only the intensity channel.

- Secondly, we present a method to define automatically a sample region, from which the lane appearance is learnt on-the-fly. This sample region is determined using the vanishing point and the geometric/color features of lane borders and surfaces. The lane model is therefore adaptive to various types of lane surfaces. To make the lane model more robust to the lighting conditions, the proposed method employs the illumination-invariant color space (IIS). In addition, we propose novel lane scores that combines color, edge, and shape features for detecting unmarked pedestrian lanes.

- Lastly, we created a new data set with manually annotated detection ground-truth for objective evaluation of pedestrian unmarked-lane detection methods. This data set is collected from realistic indoor/outdoor scenes, with various shapes, textures, and surface colors.

The remainder of the chapter is organized as follows. Existing methods for lane detection in unstructured environments are reviewed in Section 4.2. The proposed method is described in Section 4.3. Experimental results are presented in Section 4.4. Finally, conclusions are given in Section 4.5.

4.2 Related work

Current vision-based approaches for detecting pedestrian lanes in unstructured scenes can be divided into two categories: (i) lane segmentation; and (ii) lane-border detection. In the *lane segmentation* approach, off-line color models are used to differentiate the lane pixels from the background [69,72,76,77]. Different color spaces and classifiers have been used. For example, Crisman and Thorpe use Gaussian models in the red-green-blue (RGB) color space to represent the on-road and off-road classes [69]. Also using the RGB space, Tan *et al.* capture the the variability of the road surface with multiple color histograms, and the background with a single color histogram [72]. Instead of using the RGB space, Ramstrom and Christensen employ UV, normalized red and green, and luminance components and construct Gaussian mixture models for the road-surface and background classes [77]. Sotelo *et al.* employ the hue-saturation-intensity (HSI) color space [76]. In their method, achromatic pixels (i.e. with extreme intensities or low saturations) are classified using intensity only, whereas other pixels are classified by thresholding their chromatic distance to the training colors. Because the color models are trained off-line, these methods do not cope well with the appearance variations in lane surfaces.

To address this problem, several methods model the lane pixels directly from sample regions in the input image [70,78–80]. These methods determine the sample lane regions in different ways. For example, Alvarez *et al.* select small random areas at the bottom and in the middle of the input image [70,71]. Miksik *et al.* initialize the sample lane region as a trapezoid at the bottom and center of the image, and then refine the sample region using the vanishing point [80]. He *et al.* form a sample lane region from the candidate lane boundaries, which are detected using the vanishing point and an assumption about the lane width [78]. The per-

formance of these methods depends on the quality of the sample regions, which in turn relies on prior knowledge about the walking lane.

In the *lane-border detection* approach, the lane boundaries are determined using the vanishing point [7, 75] or templates of the lane boundaries [81]. In [75], the lane borders are detected among the edges pointing to the vanishing point. The optimal left and right edges are judged using an objective function that measures the color and texture differences between lane and non-lane regions. This method is effective only when the lane region is homogeneous and differs significantly from non-lane regions in terms of color and texture. Kong *et al.* also find the lane borders from the edges directing to the vanishing point, except that their method ranks edges using texture orientation and color features [7]. Because this method relies only on edges for lane-border detection, it is sensitive to background edges. In another method, the lane boundaries are found from the edges of homogeneous color regions by matching with lane templates [81]. Recently, Chang *et al.* propose combining lane-border detection and road segmentation for detecting lanes [82]. Similarly to [7], their method detects lane borders using the vanishing point. The lane region is segmented using the color model learned from a homogeneous region at the bottom and middle of the input image.

4.3 Proposed method

In this section, we present the new method for detecting unstructured pedestrian lanes, which comprises three main stages: (i) vanishing point estimation; (ii) sample region selection; and (iii) lane detection.

4.3.1 Vanishing point estimation

The vanishing point in an image is often located using either line segments [6, 83, 84] or local orientations [7, 75, 85]. For unstructured scenes with non-straight edges, using local orientations is more suitable than using line segments for vanishing point estimation. However, most existing methods based on local orientations have high computational complexity and are sensitive to background clutter. Furthermore, they rely on only the intensity channel, even though color

channels provide photometric information that can lead to more robust detection of edges and local orientations. In this chapter, we propose to improve the accuracy and efficiency of vanishing point detection, by employing color tensor to capture image structure and focusing on edge pixels only.

The color tensor is a tool for analyzing the local differential structure of a color image [86]. Consider an image with three color channels: $F = \{F_k; k = 1, 2, 3\}$. Let $D_{k,x}$ and $D_{k,y}$ denote the derivatives of F_k along the horizontal and vertical direction, respectively. Let $\mathbf{w}$ be the convolution kernel of a smoothing filter. The color tensor of the image is represented as

$$\begin{pmatrix} G_{xx} & G_{xy} \\ G_{yx} & G_{yy} \end{pmatrix} \quad \text{where} \quad \begin{cases} G_{xx} &= \mathbf{w} * \left[\sum_{k=1}^{3} D_{k,x} \circ D_{k,x} \right] \\ G_{yy} &= \mathbf{w} * \left[\sum_{k=1}^{3} D_{k,y} \circ D_{k,y} \right] \\ G_{xy} &= \mathbf{w} * \left[\sum_{k=1}^{3} D_{k,x} \circ D_{k,y} \right] \end{cases}. \tag{4.1}$$

Here, $*$ denotes the 2-D convolution, and $\circ$ denotes the element-wise multiplication (Hadamard product). Based on eigenvalue analysis of the color tensor [86], we estimate the dominant local orientation θ and the edge strength λ for all image pixels as

$$\theta = \frac{1}{2} \arctan\left(\frac{2G_{xy}}{G_{xx} - G_{yy}} \right) + \frac{\pi}{2}, \tag{4.2}$$

$$\lambda = \frac{1}{2}\left(G_{xx} + G_{yy} + \sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2} \right), \tag{4.3}$$

where the arithmetic operations are performed element-wise. Next, the edge pixels in the image are identified via non-maximum suppression and hysteresis thresholding, as done in the intensity-based Canny edge detector. The main difference in our method is that the dominant local orientation θ and the edge strength λ are estimated more reliably using photometric information obtained from (4.2) and (4.3). For the example input image in Fig. 4.1(a), Fig. 4.1(b) shows the computed local orientations, and Fig. 4.1(c) shows the estimated edge map.

To determine the vanishing point (VP), each pixel location $v = (x_v, y_v)$ is considered as a candidate, for which a VP score is computed. Let P be the set of edge pixels $\{p = (x_p, y_p)\}$ where $y_p > y_v$. Let Δ_{vp} be the difference between the

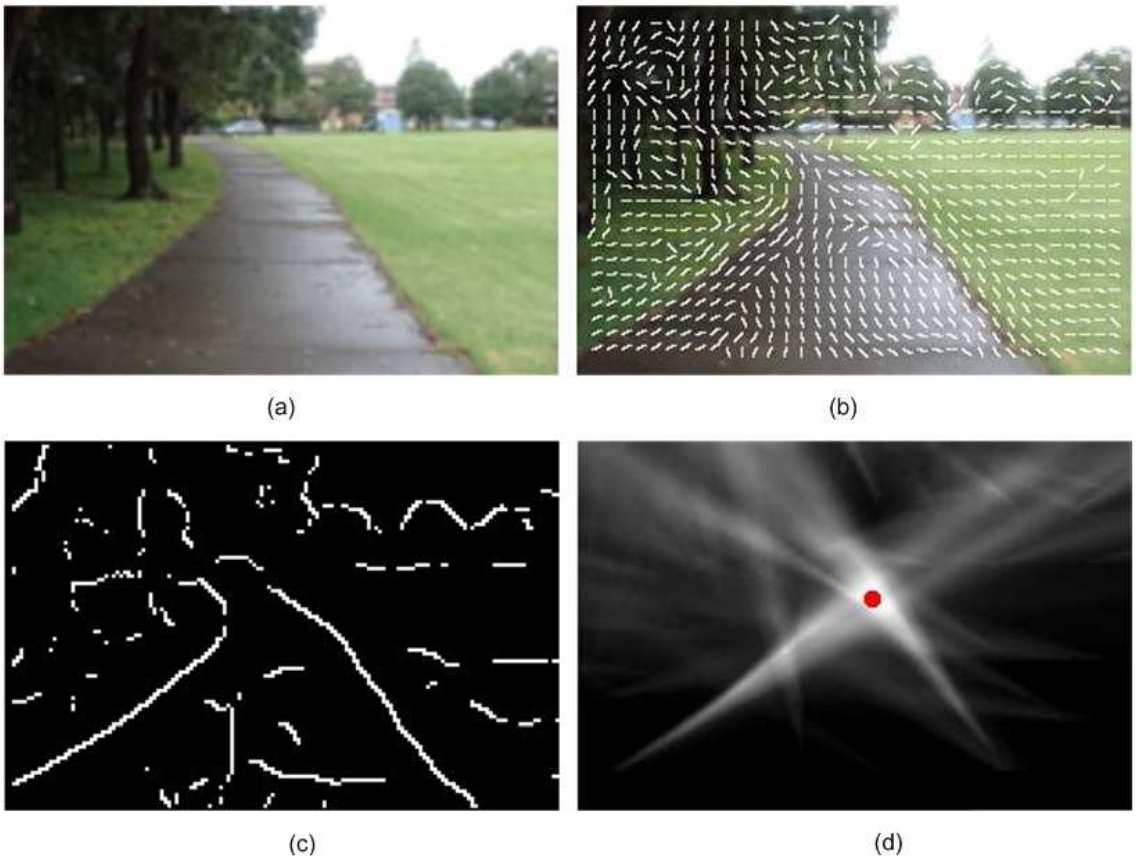


Figure 4.1: Illustration of the proposed vanishing point estimation: (a) color input image; (b) local orientations estimated by the color tensor for sampled pixels; (c) edge map obtained by the *color* Canny edge detector; (d) VP map and the vanishing point (in red).

dominant local orientation at pixel p and the angle of vector $\boldsymbol{\ell}_{vp}$ connecting v to p: $\Delta_{vp} = |\theta_p - \angle\boldsymbol{\ell}_{vp}|$. Let μ_{vp} be the ratio between the length of $\boldsymbol{\ell}_{vp}$ and the diagonal length L of the image: $\mu_{vp} = |\ell_{vp}|/L$. After investigating several choices, we propose to define the voting score contributed by pixel p to candidate v as

$$s(v,p) = \begin{cases} \exp\{-\Delta_{vp}\,\mu_{vp}\}, & \text{if } \Delta_{vp} \leq \tau_{\text{o}}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

Here, τ_{o} is a positive constant to verify the orientation similarity between p and $\boldsymbol{\ell}_{vp}$. Equation (4.4) means that $s(v,p)$ is high if: (i) pixel p has a similar orientation to vector $\boldsymbol{\ell}_{vp}$; and (ii) pixel p is spatially close to v. The VP score of candidate v is the sum of all voting scores:

$$S(v) = \sum_{p \in P} s(v,p). \tag{4.5}$$

The vanishing point is finally found as the pixel with the highest VP score. Figure 4.1(d) demonstrates the VP map and the vanishing point computed for the image in Fig. 4.1(a). More results of the proposed method for vanishing point estimation are given in Section 4.4.2.

4.3.2 Sample region selection

Because the appearance (e.g. color, edge, shape, texture) of pedestrian lanes in unstructured scenes varies significantly and is strongly affected by illumination conditions, it is difficult to obtain a robust appearance model with off-line training. Hence, it is more plausible to construct an appearance model adaptively and directly from the input image. To this end, existing methods (e.g. [71, 79, 80]) usually select the sample region as a small region at the bottom or center of the input image. However, the sample region selected in such a manner tends to include non-lane regions. In our method, the sample region is automatically defined using the vanishing point (estimated in the previous stage), and then verified using color and orientation features of both lane borders and lane regions.

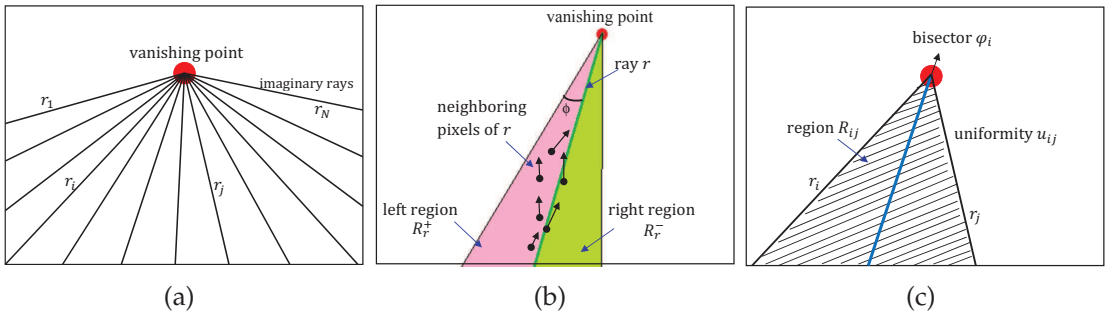


Figure 4.2: Selecting the sample lane region: (a) rays created from the vanishing point; (b) properties of a single ray; (c) properties of a pair of rays.

Although the lane may have various shapes, its main part can be approximated with straight borders. Hence, it is possible to represent the border of the sample region using imaginary rays. To this end, a set of rays $\mathcal{B} = \{r_1, r_2, ..., r_N\}$ emanating from the vanishing point is created as shown in Fig. 4.2(a). These rays are uniformly distributed in an angle range $[\phi_{\min}, \phi_{\max}]$ relative to the horizontal direction. The sample region is identified by finding a ray pair (r_i, r_j) that best captures the main part of the pedestrian lane.

For a given ray r, two features are defined: 1) the orientation difference d_{o} between ray r and its neighboring pixels; 2) the color difference d_{c} between two regions adjacent to r, see Fig. 4.2(b). Let θ_r denote the angle of ray r. Let $\mathcal{N}_r$ be the set of pixels whose Euclidean distance to r is smaller than $L\tau_{\mathrm{d}}$. Here, L is the diagonal length of the image, and τ_{d} is a threshold. The orientation difference d_{o} between r and its neighboring pixels is calculated as

$$d_{\mathrm{o}} = \frac{1}{|\mathcal{N}_r|} \sum_{p \in \mathcal{N}_r} |\theta_r - \theta_p|, \tag{4.6}$$

where θ_p is the orientation of pixel p computed in (4.2).

Let R_r^+ and R_r^- be two neighboring regions on the left and right of ray r as shown in Fig. 4.2(b). These regions are formed from ray r by an angular spacing of ϕ. Suppose that $\mathbf{c}^+$ and $\mathbf{c}^-$ are the mean color of all pixels in R_r^+ and R_r^-, respectively. The color difference d_{c} between adjacent regions of ray r is computed as

$$d_{\mathrm{c}} = \frac{\|\mathbf{c}^+ - \mathbf{c}^-\|_2}{\max(\|\mathbf{c}^+\|_2, \|\mathbf{c}^-\|_2)}, \tag{4.7}$$

where $\|\cdot\|_2$ denotes the L_2-norm.

Next, for a given ray pair (r_i, r_j), two features are defined: 1) the color uniformity u_{ij} of pixels between r_i and r_j ; 2) the angle ϕ_{ij} of the bisector between r_i and r_j. Let R_{ij} denote an image region formed by a ray pair (r_i, r_j) as shown in Fig. 4.2(c). The uniformity u_{ij} of R_{ij} is computed as

$$u_{ij} = \sum_{m=1}^{M} \sum_{n=1}^{M} \sum_{k=1}^{M} h(m,n,k)^2, \tag{4.8}$$

where h is the normalized 3-D color histogram of pixels in R_{ij}, and M is the number of bins for each color channel.

In summary, for a given ray pair (r_i, r_j), six features are extracted: 1) the orientation difference $d_{\mathrm{o},i}$ of ray r_i and its neighboring pixels; 2) the orientation difference $d_{\mathrm{o},j}$ of ray r_j and its neighboring pixels; 3) the color difference $d_{\mathrm{c},i}$ between adjacent regions of ray r_i; 4) the color difference $d_{\mathrm{c},j}$ between adjacent regions of ray r_j; 5) the color uniformity of u_{ij} of region R_{ij}; 6) the bisector angle ϕ_{ij} of ray r_i and r_j.

Given these six features, we propose the following lane score for the ray pair (r_i, r_j):

$$f(r_i, r_j) = f_1(d_{\mathrm{o},i})\ f_1(d_{\mathrm{o},j})\ f_2(d_{\mathrm{c},i})\ f_2(d_{\mathrm{c},j})\ f_3(u_{ij})\ f_4(\phi_{ij}), \tag{4.9}$$

where the individual score functions are given by:

$$f_1(d_o) = \exp\{-d_o/\pi\}, \tag{4.10}$$

$$f_2(d_c) = \frac{1}{1 + a\, e^{-b\, d_c}}, \tag{4.11}$$

$$f_3(u) = \frac{1}{1 + \alpha\, e^{-\beta\, u}}, \tag{4.12}$$

$$f_4(\phi) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(\phi - \overline{\phi})^2}{2\sigma^2}\right\}. \tag{4.13}$$

In Equations (4.10) to (4.13), a, b, α, β, σ and $\overline{\phi}$ are fixed parameters that are learnt from the training data. The individual score functions are chosen to model the relationship between a feature and the lane score. For example, Equation (4.10) means that the smaller is the orientation difference d_o (i.e. when neighboring pixels have similar orientations as ray r), the higher is the score $f_1(d_o)$, and vice versa. Equation (4.11) indicates that the higher is the color difference d_c (i.e. when ray r is at the lane border), the higher is the score $f_2(d_c)$. Equation (4.12) means that the higher is the color uniformity u, the higher is the score $f_3(u)$. Lastly, Equation (4.13) is based on the observation that the bisector angle on training data approximates a normal distribution.

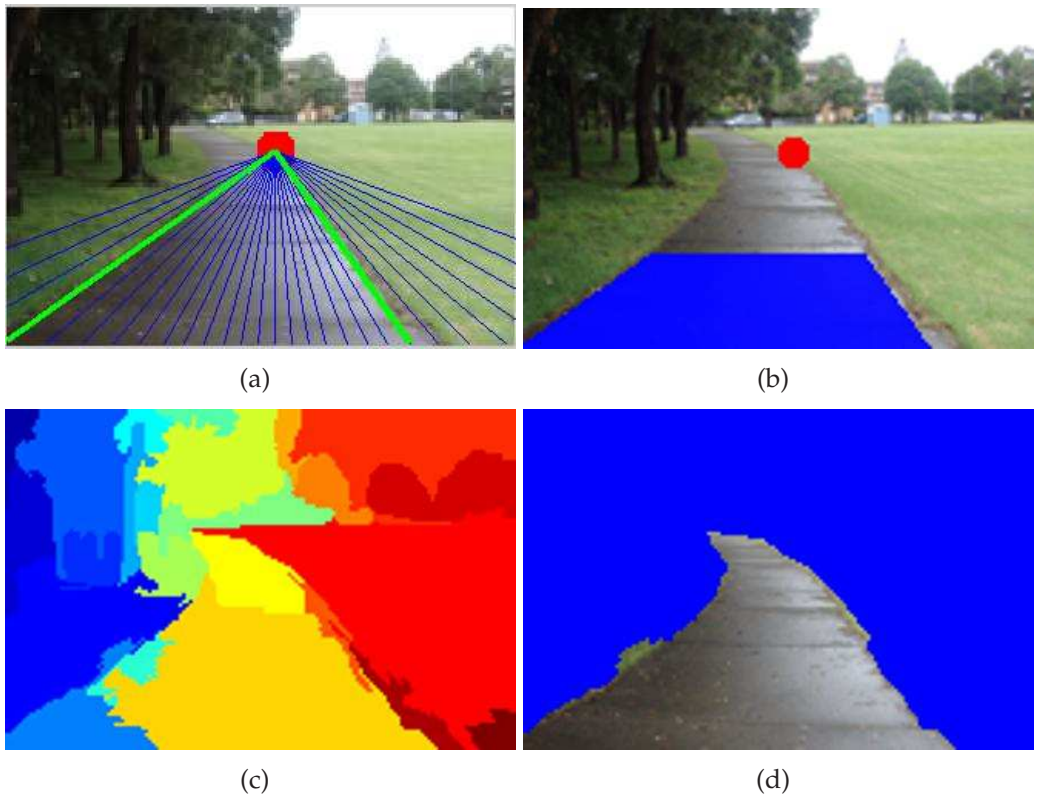
The optimal pair (r_i^*, r_j^*) for the sample region is obtained by maximizing the lane score:

$$(r_i^*, r_j^*) = \underset{(r_i, r_j) \in \mathcal{B}^2}{\arg\max}\, f(r_i, r_j). \tag{4.14}$$

Figure 4.3(a) shows an example of detecting the borders of the sample region. To account for the case where the vanishing point is located outside the image or the pedestrian lane region, we use only the lower half of the region formed by (r_i^*, r_j^*) as a lane sample region, see Fig. 4.3(b).

4.3.3 Lane detection

In this stage, the input image is segmented initially into color homogeneous sub-regions. Numerous image segmentation algorithms can be applied. We use the graph-based segmentation algorithm presented in [5], because it is fast and suitable for our task. This algorithm initializes sub-regions as single pixels. Adjacent sub-regions are then merged iteratively, according to the color difference



(a) (b)

(c) (d)

Figure 4.3: Illustration of the proposed method for pedestrian lane detection: (a) the imaginary rays (blue lines) and the detected borders (green lines) of the sample region; (b) lane sample region (blue region); (c) color homogeneous sub-regions segmented using the graph-based method [5]; (d) detected walking lane. See electronic color image.

between the sub-regions. Figure 4.3(c) illustrates the segmented regions for the input image of Fig. 4.1(a).

Next, the pedestrian lane is detected. Let $\mathcal{R} = \{R_1, R_2, ...\}$ be the set of color homogeneous sub-regions. The pedestrian lane is treated as a set Z of connected sub-regions of $\mathcal{R}$. Two sub-regions R_i and R_j are considered to be connected if there exist two pixels $p_i \in R_i$ and $p_j \in R_j$ that are connected (e.g. 4-connected pixels).

A connected region $Z \subset \mathcal{R}$ is represented by a color feature and a shape feature. The color feature $\mathbf{c}$ is the mean of all color pixels in Z. The lane score for a given color feature $\mathbf{c}$ is defined as

$$g_1(\mathbf{c}) = p(\mathbf{c}|\mathcal{L}), \tag{4.15}$$

where $p(\mathbf{c}|\mathcal{L})$ is the class-conditional probability density function for the lane class.

It is estimated from the color histogram of all pixels in the sample lane region, which is found as in Section 4.3.2. In our work, two color spaces are considered: red-green-blue (RGB) and the illumination invariant space (IIS). Compared to the RGB, the IIS is less sensitive to illumination conditions and shading. Conversion from the RGB to the IIS is as follows [87]:

$$\begin{cases} C_1 &= \arctan\{R/\max(G,B)\}, \\ C_2 &= \arctan\{G/\max(R,B)\}, \\ C_3 &= \arctan\{B/\max(R,G)\}. \end{cases} \tag{4.16}$$

The shape feature $\mathbf{s}$ is extracted using the shape contexts proposed in [88]. The shape contexts are known for their robustness to local deformation and partial occlusion, and their invariance to scale and rotation. Consider a shape with sampling points on its contour. The shape context of a sampling point p is the histogram h_p of the angles and distances from the remaining sampling points to p.

The dissimilarity between the shape contexts of two points p and q is represented as

$$C(p,q) = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_p(k)-h_q(k)]^2}{h_p(k)+h_q(k)}, \tag{4.17}$$

where K is the number of bins of each shape context. On a single shape, the shape contexts of the points p and q are different, i.e. $C(p,q)$ is high. However, on two similar shapes, the shape contexts of two corresponding points p and q are similar, i.e. $C(p,q)$ is low.

Let $\mathcal{T} = \{T_1, T_2, ...\}$ be a set of shape templates for pedestrian lanes. Examples of the shape templates obtained from the training data are shown in Fig. 4.4. To obtain shape feature $\mathbf{s}$, the outer contour of region Z is sampled in a similar way as the templates. The matching cost $D(\mathbf{s}, T)$ between $\mathbf{s}$ and a template T is modeled as

$$D(\mathbf{s},T) = \frac{1}{|\mathbf{s}|}\sum_{p\in\mathbf{s}}\min_{q\in T} C(p,q), \tag{4.18}$$

where $|\mathbf{s}|$ denotes the number of sampling points on $\mathbf{s}$. The smaller is the matching cost $D(\mathbf{s}, T)$, the higher is the similarity between $\mathbf{s}$ and T. Consequently, the lane

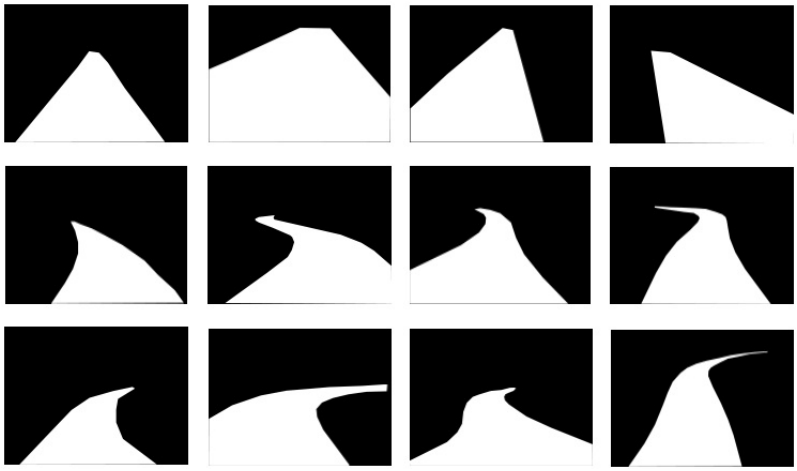


Figure 4.4: Example shape templates for pedestrian lanes. Row 1: straight lanes. Row 2: left-curved lanes. Row 3: right-curved lanes.

score for shape feature $\mathbf{s}$ is defined as

$$g_2(\mathbf{s}) = \exp\left[-\lambda \min_{T \in \mathcal{T}} D(\mathbf{s}, T)\right], \tag{4.19}$$

where λ is a positive scalar determined through training data.

Collectively, the lane score for region Z with color feature $\mathbf{c}$ and shape feature $\mathbf{s}$ is calculated as

$$g(Z) = g_1(\mathbf{c})\, g_2(\mathbf{s}). \tag{4.20}$$

The optimal region Z^* of $\mathcal{R}$ is found by maximizing the lane score:

$$Z^* = \arg\max_{Z \subset \mathcal{R}} g(Z). \tag{4.21}$$

The optimal region Z^* can be obtained with a computational complexity of $O(2^{|\mathcal{R}|})$ via an exhaustive search among the subsets of $\mathcal{R}$. To reduce the computational load, we adopt a greedy-search algorithm [89], which generates Z^* by iteratively adding and removing sub-regions (see Algorithm 2). At each iteration, a sub-region is added to or removed from Z^*, only if the connectivity of the new Z^* is satisfied and the lane score $g(Z^*)$ is increased. In addition, for faster search we consider only sub-regions $R_i \in \mathcal{R}$ with $p(\mathbf{c}|\mathcal{L})$ greater than or equal to a predefined threshold τ_c. Because the number of sub-regions is finite and the operators in Algorithm 2 are deterministic, the algorithm will converge.

Algorithm 2 Adding and removing regions for pedestrian lane detection.

$\mathcal{R}' \leftarrow \{R_i \in \mathcal{R} \mid p(\mathbf{c}_i|\mathcal{L}) \geq \tau_c\}$
$Z^* \leftarrow \arg\max_{R_i \in \mathcal{R}'} p(\mathbf{c}_i|\mathcal{L})$
continue $\leftarrow$ TRUE
while continue **do**
  $\mathcal{R}_{\text{add}} \leftarrow \{R_i \in \{\mathcal{R}' - Z^*\}$ so that $Z^* \cup R_i$ is a connected set$\}$
  $R^+ \leftarrow \arg\max_{R_i \in \mathcal{R}_{\text{add}}} g(Z^* \cup R_i)$
  $\mathcal{R}_{\text{rmv}} \leftarrow \{R_i \in Z^*$ so that $\{Z^* - R_i\}$ is a connected set$\}$
  $R^- \leftarrow \arg\max_{R_i \in \mathcal{R}_{\text{rmv}}} g(Z^* - R_i)$
  if $g(Z^* \cup R^+) > g(Z^*)$ **and** $g(Z^* \cup R^+) \geq g(Z^* - R^-)$ **then**
    $Z^* \leftarrow Z^* \cup R^+$
  else if $g(Z^* - R^-) > g(Z^*)$ **then**
    $Z^* \leftarrow Z^* - R^-$
  else
    continue $\leftarrow$ FALSE
  end if
end while

Finally, the optimal region Z^* obtained using Algorithm 2 is considered as a pedestrian lane region if

$$g(Z^*) \geq \tau_v, \tag{4.22}$$

where τ_v is a verification threshold learnt using training data. This step is necessary because the scene may contain no pedestrian lane. Figure 4.3(d) illustrates the result of lane detection for the input image shown in Fig. 4.1(a).

4.4 Experimental Results

In this section, we first describe the image data, evaluation measures, and parameters used in the proposed method. We then present the experimental results on vanishing point estimation and pedestrian lane detection.

4.4.1 Experimental methods

For experimental evaluation, we collected a data set of 2000 images in different indoor and outdoor scenes. Some of the images were taken with a video camera by subjects walking while blindfolded. The data set includes unmarked pedestrian lanes with various surface structures (pavement, brick, concrete, soil) and shapes (straight or curved). In many cases, lane regions are affected by extreme

lighting conditions (e.g. very low or high illumination). To enable quantitative performance evaluation, we manually annotated lane regions and the vanishing point in each image. In the experiments, 500 images were used for training, and 1500 images were used for testing.

To evaluate pedestrian lane detection, the detected regions are compared with the annotated regions. Let R_d denote a detected region and R_g a ground-truth region. The matching score between R_d and R_g is computed as

$$\chi(R_g, R_d) = \frac{|R_g \cap R_d|}{|R_g \cup R_d|}, \tag{4.23}$$

where $|R|$ denotes the area of region R, $\cap$ denotes the intersection, and $\cup$ denotes the union of R_d and R_g. Detected region R_d is considered as *correct* if there exists a ground-truth region R_g where $\chi(R_g, R_d)$ is greater than or equal to an evaluation threshold τ_e. Similar to the evaluation of other object-detection systems [66,67], τ_e is set to 0.5. Next, three evaluation measures are computed: recall, precision and F-measure. *Recall* is the percentage of the ground-truth lanes that are detected correctly. *Precision* is the percentage of the detected lanes that are considered to be correct. *F-measure* is the harmonic mean of precision and recall:

$$F\text{-}measure = 2 \times \frac{Recall \times Precision}{Recall + Precision}. \tag{4.24}$$

To evaluate vanishing point estimation, suppose that P_d be a detected vanishing point, and P_g is the ground-truth vanishing point. Consistently with [85], the estimation error is measured as the ratio of the Euclidean distance from P_d to P_g versus the diagonal length L of the image:

$$Error_{\text{vp}} = \frac{|P_d - P_g|}{L}. \tag{4.25}$$

In our experiments, the parameters of the proposed method were selected by analyzing the performance of the pedestrian lane detection on the training set. For the steps described in Section 4.3.1, the window size of the Gaussian filter $\mathbf{w}$ was chosen as $H = 11$. The constant τ_o in (4.4) was set as $\tau_o = \pi/36$. For the steps described in Section 4.3.2, the number of imaginary rays was selected as $N = 29$, and the angle range of imaginary rays was $[\phi_{\min}, \phi_{\max}] = [\pi/9, 8\pi/9]$. The angular spacing was $\phi = \pi/12$. The parameters a and b in (4.11), α and β in (4.12),

σ and $\overline{\phi}$ in (4.13) were set as $a = 0.9$, $b = 2.3$, $\alpha = 0.9$, $\beta = 2.3$, and $\sigma = 0.52\pi$ and $\overline{\phi} = 0.09\pi$. For the steps described in Section 4.3.3, the parameter λ in (4.19) and the thresholds τ_c and τ_v were selected as $\lambda = 5$, $\tau_c = 0.02$, and $\tau_v = 0.01$.

To select a suitable number of color quantization bins M, we analyzed the proposed method for different values: $M = 16, 32, 64, 128$, and 256. Table 4.1 shows the lane detection performance on the training set. The best lane detection performance was obtained with $M = 128$ bins; therefore, this value was chosen in our experiments.

Table 4.1: Lane detection performance of the proposed method for different color bin numbers on the training set.

Number of color bins M	16	32	64	128	256
Recall (%)	90.3	94.1	95.7	97.5	95.2
Precision (%)	88.2	91.8	96.5	98.3	96.1
F-measure (%)	89.2	92.9	96.1	97.9	95.6

4.4.2 Analysis of vanishing point estimation

The proposed method for vanishing point estimation was compared with two existing methods.

- *Hough-based method* [6]: This method first applies the Hough transform on the edge map to find line segments. It then computes the vanishing point by voting the intersections of line pairs in another Hough transform. In the experiments, we used the same edge map as in the proposed method. The distance and orientation resolutions in the Hough transforms were tuned using the training set.

- *Gabor-based method* [7]: This method applies Gabor filters on the intensity image to compute local orientations, and then estimates the vanishing point using these orientations. Each pixel location v in the top 90% region of the image is considered as a VP candidate. It is voted by pixels p in the half-disk region, which is centered on v and below v. Our experiments used the MATLAB code provided by the authors of [7]. However, the parameters of the Gabor-based method were tuned using the training set.

Figure 4.5: Visual results of vanishing point estimation. Ground-truth VP: *red* dot. VP detected by the proposed method: *green* dot. VP detected by Hough-based method [6]: *yellow* dot. VP detected by Gabor-based method [7]: *blue* dot. See electronic color image.

Table 4.2: Accuracy and speed of vanishing point estimation algorithms.

Method	*Average error*	*Computational time (s)*
Hough-based method [6]	0.108	0.07 ± 0.01
Gabor-based method [7]	0.085	3.81 ± 0.64
Proposed VPE method	**0.057**	**0.52 ± 0.10**

Table 4.2 shows the performance of different VPE algorithms. The average error of the proposed method (0.057) was significantly lower than that of the Hough-based method (0.108) and the Gabor-based method (0.085). The Hough-based method employs straight lines for finding the vanishing point. It does not work well for natural scenes that contain many non-straight edges. The Gabor-based method calculates the voting score for each vanishing point candidate from all pixels in a local region, and therefore it is affected significantly by the clutter pixels. Furthermore, the Gabor-based method uses only intensity for computing

the edge orientations and magnitudes. In comparison, the proposed method employs multiple color channels for finding edge pixels and their orientations (via color tensor). Hence, it can distinguish color pixels even if they have similar intensity. Moreover, the proposed method uses only edge pixels for voting the vanishing point, and therefore reduces significantly the computation load and the influence of background pixels.

For images of size 100×140 pixel, the average processing time per image of the proposed method (0.52 s) was significantly shorter than that of the Gabor-based method (3.81 s). That is, the proposed method was about 7.3 times faster than the Gabor-based method. Although the Hough-based method had the shortest processing time per image (0.07 s), it also had the lowest accuracy among the three tested methods. Figure 4.5 shows several visual results of different VPE methods. As can be seen, the proposed method estimates the vanishing point more accurately, compared to both the Hough-based and Gabor-based methods.

4.4.3 Analysis of pedestrian lane detection

For pedestrian lane detection, we evaluated the proposed method with several related methods:

- *Edge-based method* [7]: This approach detects the lane boundaries from edges directing to the vanishing point, using the color and orientation features of lane borders. In the experiments, we used the MATLAB code provided by the authors of [7], and adjusted it using the training data to suit better this application.

- *Lane-border detection method* [8]: This method is our previous work, and it finds two lane borders among the edges directing to the vanishing point. Each edge is represented by two features: i) the color difference between two regions adjacent to the edge; and ii) the orientation difference of neighboring pixels to the edge. Each region formed by a pair of edges is described by two features: i) the color uniformity of the region; (ii) the direction of the bisector of the edges. A pair of edges is considered as the lane borders if the likelihood of their edge and region features is the highest among all

pairs. This method does not use region segmentation technique described in Section 4.3.3.

- *Lane segmentation method* [9]: The algorithm is also our previous work, and it detects the walking lane from color homogeneous regions using appearance and shape information. The appearance model of the lane region is constructed on-the-fly from a sample region, and the lane shape is modeled by shape contexts. The sample region is located by finding a pair of edges directing to the vanishing point that best capture the characteristics of a walking region. The characteristics include: i) the color uniformity of the region formed by the edges; ii) the direction of the bisector of the edges. This algorithm does not use the features of lane borders presented in Section 4.3.2 for detecting the sample region.

Table 4.3 shows the performance of different methods for pedestrian lane detection on the test set. Using the RGB color space, the proposed method had a recall rate of 91.3%, a precision rate of 95.4%, and a F-measure of 93.3%. Using the IIS color space, it achieved a recall rate of 95.7%, a precision rate of 96.6%, and a F-measure of 96.1%.

The proposed method outperformed the edge-based method [7], which had a recall rate of 63.9%, a precision rate of 66.1% and a F-measure of 65.0%. The edge-based method uses only the color and orientation properties of lane borders, and it is therefore susceptible to background edges. In contrast, the proposed method employs the properties of not only lane borders but also lane regions (appearance and shape).

Table 4.3: Comparison of algorithms for pedestrian lane detection.

Methods	*Recall (%)*	*Precision (%)*	*F-measure (%)*	*Processing time (s)*
Edge-based method [7]	63.9	66.1	65.0	3.2
Lane-border detection method [8]	88.2	90.8	89.5	1.2
Lane segmentation method [9]	90.5	95.8	93.1	1.5
Proposed method using RGB	91.3	95.4	93.3	1.5
Proposed method using IIS	**95.7**	**96.6**	**96.1**	**1.5**

The proposed method also had better recall and precision rates than the lane-border detection method [8] (recall rate of 88.2%, precision rate of 90.8% and F-

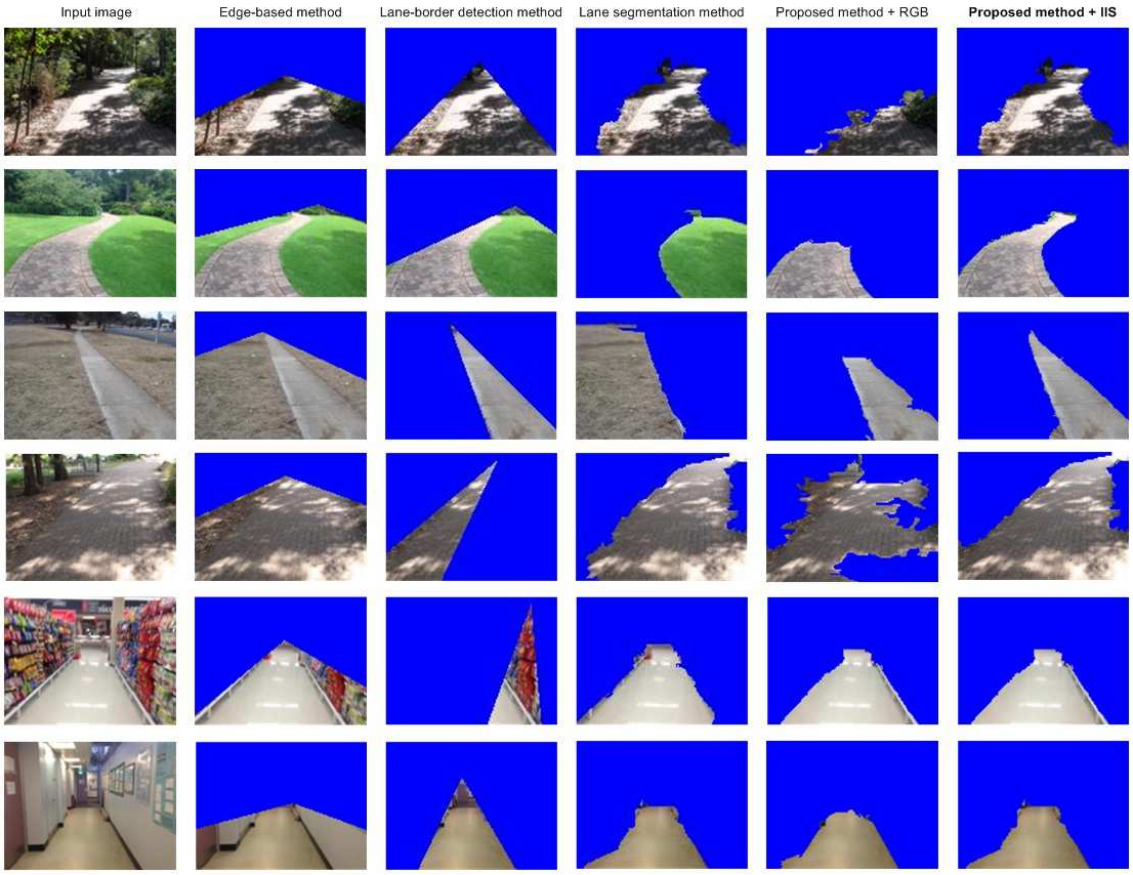


Figure 4.6: Visual comparative results of different methods for pedestrian lane detection. Column 1: input images. Column 2: output of the edge-based method [7]. Column 3: output of the lane-border detection method [8]. Column 4: output of the lane-border detection method [9]. Column 5: output of the proposed method using the RGB color space. Column 6: output of the proposed method using the IIS color space. See electronic color image.

measure of 89.5%) and the lane segmentation method [9] (recall rate of 90.5%, precision rate of 95.8% and F-measure of 93.1%). The lane-border detection method finds the lane borders from edges directing to the vanishing point, and hence it only detects straight lanes or the straight part of curved lanes. The lane segmentation method determines the sample region for training the appearance model of the lane, using only the uniformity and orientation properties of lane surfaces. Therefore, the appearance model only represents the lane part that has high uniformity. In contrast, by combining the features of both the lane surface and lane border, the proposed method detects the sample region as the straight part of the lane.

Figure 4.7: Visual sample results of the proposed method for detecting pedestrian lanes in indoor and outdoor environments. Column 1, 3, 5 and 7: input images. Column 2, 4, 6 and 8: detected lanes. See electronic color images.

Figure 4.6 shows visual results of different methods for pedestrian lane detection. The results show the robustness and effectiveness of the proposed method compared with the previous methods [7–9]. These results also demonstrate that the proposed method using the IIS color space is more robust than using the RGB color space.

Table 4.3 shows that the average processing time of the proposed method (1.5 s) is significantly shorter than that of the edge-based method [7] (3.2 s). These processing times were recorded for MATLAB implementation and an image size of 100×140 pixel on a PC with 3.7 GHz CPU. The run-time speed of the proposed method can still be optimized further. Several outputs of the proposed method are shown in Fig. 4.7. In summary, the experimental results presented in this section show that the proposed method detects robustly pedestrian lanes with

various surfaces, under different imaging conditions.

4.5 Chapter summary

This chapter presents a method for pedestrian lane detection in unstructured environments, by combining color, edge, and shape features. The proposed method uses the vanishing point to automatically determine a sample lane region, from which a lane model is adaptively constructed. Evaluation results on a large data set with detection ground-truth have shown that the proposed method is able to detect robustly various types of unstructured pedestrian lanes, in outdoor and indoor scenes under challenging environmental conditions. It also significantly outperforms other existing methods. The chapter also presents an efficient and accurate method based on the color tensor for vanishing point estimation. Besides assistive navigation for vision-impaired people, the proposed methods for vanishing point detection and pedestrian lane detection can be applied to autonomous vehicles or robots operating on open roads.

Chapter 5

Lane classification

Chapter contents

Classification of pedestrian lane types plays an important role in assistive navigation systems. This chapter proposes a method for pedestrian lane classification based on multiple instance learning. The proposed method encodes each image into a bag of instances. Each instance is an image region and represented by a feature vector. A vocabulary-based framework of multiple instance learning is then employed to categorize bags. The proposed lane classification method is evaluated on a large and new data set collected from various environments.

Experimental results on the data set have shown the efficiency and robustness of the proposed method.

5.1 Introduction

Automatic lane detection is an essential component in assistive navigation systems. However, existing lane detection approaches are limited to a particular type of pedestrian lanes. For example, methods in [3, 4, 17, 43] focus on detecting marked-lanes that are indicated by painted markers, while methods in [8, 9] are designed for detecting unmarked-lanes that have no painted markers. In addition, these methods assume that a lane region is always found in an input image. In fact, this assumption may not be held, because the scene in front of blind travelers may not have a walking lane.

To make a lane detection system general and adaptive to various lane types, a straight forward approach is to use all different lane detectors for each input image in which each detector is designed for one lane type. The lane region is determined as the output of a detector that has the maximum detection score. However, this approach has two drawbacks. First, more false alarms are generated due to invoking inappropriate lane detectors. Second, the system will have a high computational complexity since all lane detectors are involved in the detection process. These issues inspire us to develop a lane classification method which is able to verify the presence of a pedestrian lane in the input image and to identify the type of the pedestrian lane. Based on the output of the lane classification method, a suitable detector will be selected to locate the lane region, or a notification of "no-lane" will be generated. In this chapter, we concentrate on three different types: *marked-lane*, *unmarked-lane* and *non-lane* images. However, the proposed lane classification method is extendable to more pedestrian lane types.

Pedestrian lane classification is somewhat similar to scene classification. However, pedestrian lane classification has its own challenges due to the variation of the shape and appearance of the lane region. In addition, the difference between marked-lanes and unmarked-lanes is small that is only the presence of the lane

markers. Moreover, the location of the lane region in each image is unknown. To overcome these difficulties, we propose a lane classification method using the multiple instance learning approach. In our method, images are considered as bags and instances are local image regions. To describe instances, we exploit the color- and edge-based features from image regions which capture some parts of a lane region (e.g. painted markers and lane boundaries). A vocabulary is then constructed from the instances of training images, using the k-mean algorithm. Bags are finally converted into global feature vectors based on the vocabulary, and classified into different categories using the support vector machine (SVM) technique.

The remainder of the chapter is organized as follows. Existing methods for scene classification and the background of multiple instance learning are reviewed in Section 5.2. The proposed method is described in Section 5.3. Experimental results are presented in Section 5.4. Finally, conclusions are given in Section 5.5.

5.2 Related work

This section reviews existing approaches for scene classification and also presents the background of multiple instance learning, which is applied in the proposed method for lane classification.

5.2.1 Scene classification

The existing methods can be categorized into two major approaches: *local* and *global*. These approaches are presented as follows.

In the *local* approach, scene classification is based on finding semantic objects in each scene category [90–93]. The image is first partitioned into different local regions. Then, local regions are classified independently into different object classes. Finally, the global scene is identified based on the classification results of individual local regions. Many different techniques for partitioning and classifying local regions have used in this approach. For example, Szmummer and Picard divide an image into 4×4 sub-blocks, and employ color and texture features to represent each sub-block [90]. The color feature is estimated as a color histogram

and the texture features are computed as the parameters of the multi-resolution simultaneous autoregressive (MSAR) model [94]. The sub-blocks are then classified as *indoor* and *outdoor* classes, using K-nearest neighbor (K-NN) algorithm. The label of the image is finally assigned as the most common label among the regions. In [95], Paek and Chang divide the image into 8×8 sub-blocks, and use color and edge orientation distributions to describe each sub-block. They employ multiple classifiers to categorize sub-blocks as *indoor* or *outdoor*, *sky* or *no sky*, and *vegetation* or *no vegetation*. The output of the classifiers are then fed to a belief network for recognizing the scene. In [91], Serano *et al.* partition each image into 4×4 sub-blocks, and categorize the sub-blocks into *indoor* and *outdoor*, using SVM classifiers with color and texture features. The scene is recognized by a Bayesian network that integrates the classification results of the regions with the semantic scene attributes (e.g. blue-sky, grass and cloud) of the image. The semantic attributes are extracted using blue-sky, cloud and grass detectors.

Several methods rely on image segmentation to obtain local regions, and then find semantic objects for recognizing the global scene [93, 96–98]. For example, Fan *et al.* propose an image classification method that is based on semantic object detection [93]. In their method, the image is segmented into color and texture homogeneous regions using the mean shift technique [99], and each region is represented by a feature vector that includes density ratio, locations, dominant colors, color variances, Tamura and wavelet texture features. The semantic objects are then determined from the regions, using SVM classifiers and label-based aggregation. The scene is finally identified by maximizing the posterior probability estimated from the detected objects. Aksoy *et al.* propose a Bayesian framework to model the distinct spatial relationships (e.g. distance and orientation) between local regions in each scene for image classification [98]. The local regions are obtained by a split-and-merge algorithm, and representative region groups for scenes are then found using Bayesian classifiers. The scene is determined based on the posterior probability of the detected groups. In [100], Fredembach *et al.* propose eigenregions to represent area, location, and shape properties of an image region. The eigenregions are computed based on the principal components of region locations. The eigenregions features are then integrated with local infor-

mation of semantic objects for scene classification. In general, the *local* approach is only suitable for categorizing the small number of scenes. Furthermore, the performance of segmentation-based methods rely significantly on image segmentation.

In the *global* approach, the scene is described by features extracted from the entire image [101–104]. There are different methods for image representation. Traditional methods use directly low-level features (e.g. color and texture) to represent the entire image [101, 102, 105–107]. For example, Vailaya *et al.* propose classifying vacation images into a hierarchy of high-level classes by Bayesian classifiers with low-level features [102]. In their method, images are first categorized as *indoor* or *outdoor* using spatial color moments and MSAR texture features. Then, outdoor images are classified as *city* or *landscape* using edge orientation coherence vectors. The subset of landscape images is further classified into *sunset*, *forest* and *mountain* using global color distributions. In [107], Chang *et al.* also employ global color and texture features to categorize images using Bayes point machines, which approximate the Bayesian inference for linear classifiers in a kernel space. The color feature includes color histograms, color means, color variances and shape characteristics (e.g. elongation and spread). The texture feature are wavelet features of three scales and three orientations. The color and texture features are then combined to form a 144-dimension vector for representing each image.

In addition to using low-level features, several methods exploit the global properties (e.g. naturalness, openness, roughness, expansion and ruggedness) of scenes for image classification [103, 104, 108–110]. For instance, Oliva and Torralba propose building the gist of each scene by the Spatial Envelope that combines global features into a feature vector [103, 108, 109]. The global features represent the naturalness, openness, roughness, expansion and ruggedness of a scene, and are estimated based on the local responses of Gabor filters with different scales and orientations. Grossberg and Huang propose an image classification system using an evidence accumulation of the gist and texture types of each image [110]. The gist of an image is recognized employing a 304-dimensional feature vector that integrates the color means and orientations at different scales of the local image regions. The texture types in an image are determined from

the three largest homogeneous regions. Each homogeneous region is represented by a texture feature vector that includes the mean colors, the orientations at different scales, the centroid location and the area of the region. Recently, Ali *et al.* propose a global feature for image classification that integrates the openness, roughness, dominant color and dominant orientation features of an image into a 128-dimensional vector [104].

Considering another direction, some methods apply the census transform (CT) to capture the essence of a scene image [111–114]. The census transform encodes the intensity differences of a pixel with its eight neighboring pixels into binary bits, and replaces the intensity value of the pixel by the value of combining the binary bits. The census transform is equivalent to the local binary pattern (LBP) code of each local 3×3 window in [115]. In [111], Wu and Rehg propose a holistic image descriptor using the census transform histograms, which are computed from the image sub-blocks. Song and Li employ combining the wavelet and LBP features at multiple different levels (the pixel-level, patch-level and image-level) for image description [114]. In [112], Gazolli and Salles propose the contextual mean census transform (CMCT) for image representation. The CMCT is similar to the CT, but uses the differences between the intensities of the eight neighboring pixels and the mean intensity of all pixels in the local window.

Instead of using directly features, many methods employ the bag-of-words approach proposed for text document analysis to represent each scene image [116–121]. Constructing the bag-of-words consists of four steps [122]: 1) detect local regions or points of interest, 2) extract local features (e.g. SIFT features [123]) from the regions or points, 3) create the visual vocabulary by quantizing the features of training images into discrete visual words, 4) represent each image by the histogram of the visual words in the image. Different methods based on the bag of words have been used for image representation. For example, Lazebnik *et al.* propose using spatial pyramid matching for representing natural scene images. In their method, the visual vocabulary is formed by SIFT features that are extracted from 16×16 pixel patches over a grid with a spacing of 8 pixels. The spatial histograms of each word in the vocabulary are then estimated at three levels. The global feature of the image is computed finally from the

spatial histograms, using the spatial pyramid matching algorithm. In [116, 121], the probabilistic Latent Semantic Analysis (pLSA) model [124] is employed to represent image categories. Here, images are considered as *documents*, semantic object categories (e.g. grass, houses and blue sky) are considered as *topics*, and an image containing several semantic objects is modeled as a mixture of topics. The pLSA model represents topics as latent variables associated with observed objects (words). In [117, 120], image categories are modeled using the latent Dirichlet Allocation (LDA) model [119]. The LDA model is similar to the pLSA model, except that the LDA model represents each topic as a multinomial distribution with weights sampled from a Dirichlet distribution.

5.2.2 Multiple-instance learning

Multiple-instance learning (MIL) follows the supervised learning paradigm [125]. In supervised learning, the classifier is constructed from a training set and each training sample is associated with a class label. One difficulty of the conventional supervised learning is that a class label is required for every training sample. This is not always possible due to many reasons, e.g. incomplete training data is provided, there is an ambiguity in data annotation, or manual annotation is impossible for large-scale data.

Instead of requiring a complete annotation of the training data, multiple instance learning (MIL) allows a weaker level of data labeling. The training set is represented as a set of bags of instances, in which only a label for each bag is required. For the sake of simplicity, binary classification problem is considered, i.e. bags are labeled positive or negative. A bag is considered as positive if it has at least one positive instance. On the other hand, a bag is classified as negative if all of its instances are classified as negative.

Existing methods for MIL also follows either *local* approach or *global* approach [126]. In the *local* approach, bag classification is based on instance-level information [125, 127–133]. For each bag, instances are first classified into different categories, and the bag is then identified by aggregating the classification scores of instances.

A group of methods in the *local* approach is based only on the characteristics

of certain instances in each bag for categorizing bags [125,127–129]. For example, Dietterich *et al.* employ the axis parallel rectangle (APR) algorithm to find the positive instances in each bag [125]. The APR algorithm is designed to determine axis parallel rectangles, which contain the maximum number of training positive instances and have no negative instances. In [127], Maron and Perez propose a framework, called Diverse Density (DD) for classifying instances. Similar to the APR, the Diverse Density algorithm is designed to determine an area that has the high density of positive instances and the low density of negative instances. Andrews *et al.* apply the SVM method to categorize the instances in each bag and employ an iterative procedure for estimating the SVM parameters [129]. In another method, Bunescu and Mooney propose a sparse algorithm that is also based on SVM to classify instances [131]. Their algorithm is to train a SVM classifier with a relaxed constraint for categorizing positive instances. The classifier aims to avoid providing positive values for all the instances of a positive bag, but only to at least one of the instances.

Another group of methods in the *local* approach exploits information of all instances in each bag for categorizing bags [130,132,133]. These methods consider all the instances of a bag for finding the bag's label, whereas the methods in [125,127–129] only consider few instances per positive bag. In this group, methods are designed to find weights of instances from training data, and then identify the bag based on a weighted sum of classification scores of instances. Xu determines the instance weights by a wrapper algorithm from the training instances with their bags' labels [130]. The algorithm weights all the instances so that each bag has equal total weight. In [132], Foulds proposes an iterative procedure for estimating the instance weights that is based on the wrapper algorithm. The procedure gives higher weights to the instances that contain most information of each bag. In another method, Mangasarian *et al.* propose a linear SVM classifier to find positive bags from negative bags [133]. The SVM classifier is learned from training data to represent a convex combination of the positive instances in each bag.

In contrast to the *local* approach, the *global* approach employs bag-level information for categorizing bags [134–139]. Many methods in this approach treat bags

as a whole, and learn the classifiers directly in the space of bags [134, 135, 140]. In these methods, each bag is a non-vector entity, and the similarity of two bags is determined using a distance measure (such as the Hausdorff distance, Earth Movers distance and Chamfer distance) or kernel functions [140]. The classifiers are constructed based on the similarity scores of bags using the K-NN or SVM method.

Several methods of the *global* approach transform the space of bags into a space of feature vectors and then categorize the bags using a standard classifier such as AdaBoost or SVM [136–139]. To convert bags into feature vectors, these methods construct a vocabulary from the training instances. The words of the vocabulary are determined as the local maximums of the training instances using the Diverse Density algorithm. Each bag is then represented by a feature vector whose elements are the minimum distances from the words in the vocabulary to the instances in the bag. In another method, Zhou and Zhang construct a bag-level vocabulary for converting bags [141]. The words in the vocabulary is found as the centroids of groups of the similar training bags, and each bag is converted into a feature vector where its elements are the Hausdorff distances between the words and the bag.

In general, the *global* approach is more robust and accurate than the *local* approach as proved by the experimental results in [126]. This is because the local approach uses only instance-level information and does not exploit the global characteristics of bags, whereas the global approach employs both the local information of individual instances and the global information of bags. Furthermore, the global approach can cope with different data, while the local approach only deals with data where the difference between the instance groups in each bag is clear.

5.3 Proposed method

This section presents a method for classification of pedestrian lanes based on multiple instance learning. We adopt the vocabulary-based framework in [136–139] and focus on three different lane types: *marked lane*, *unmarked lane* and *non-*

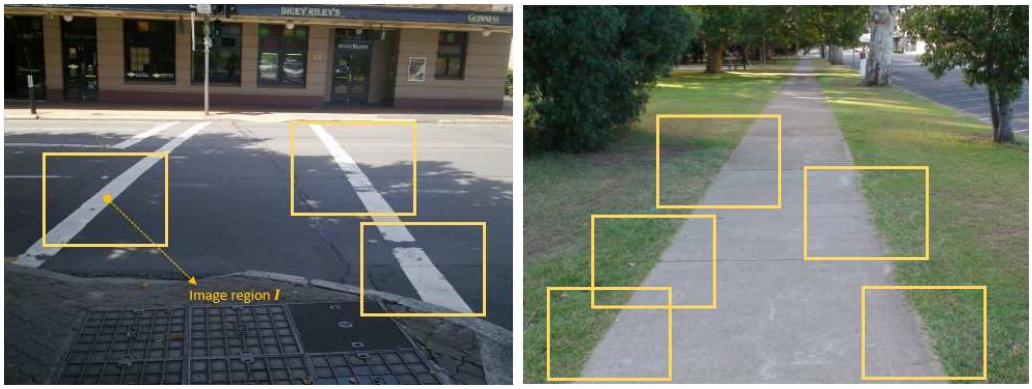
lane images. In the proposed method, each image is considered as a bag of instances, which are image regions in the image. Each instance is represented by an instance-level feature vector, and hence an image is a bag of instance-level feature vectors. A vocabulary is generated from the training instances using the k-mean algorithm. Then, bags are converted into global feature vectors. Finally, SVM classifiers are trained from a training data set in which only bag-level labels are required, i.e. only the lane-type of the entire training image. Note that since the location of the lane region or markers in an image is arbitrary and unknown and only the lane-type is annotated, multiple instance learning fits well our problem. The following subsections describe the bags, instances, vocabulary and learning the classifiers.

5.3.1 Bags and Instances

We consider each image as a bag and image regions as instances. Analyzing images of pedestrian lanes, we have found that for marked-lane images, image regions partially containing a lane marker are informative and representative for the existence of the marked-lane region. These regions contain the local appearance information of both the lane marker and road surface, and the shape information of the marker. Similarly, for unmarked-lane images, image regions partially containing a lane boundary also possess important information of both the lane surface and the surrounding background. Figure 5.1 (a) shows examples of image regions partially covering a lane marker in a marked lane image. Figure 5.1(b) shows examples of image regions partially covering a lane boundary in an unmarked-lane image.

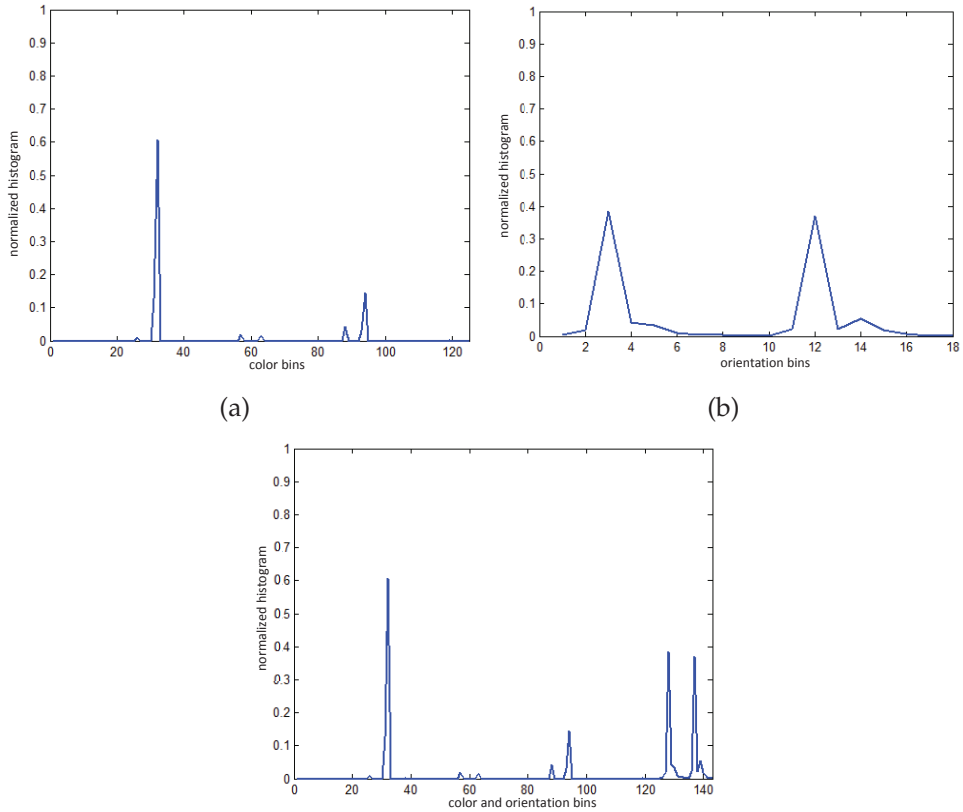
For a given image, its instances are determined as follows. We scan the image by a local window of size $w \times w$ pixels, with the horizontal and vertical stride steps of (d_x, d_y), i.e. $d_x = 0.5w$ and $d_y = 0.5w$. Each image region located on such local windows is considered as an instance.

To describe an instance I in the image, we employ combining color and texture features. The color feature h_c is first estimated as the 3D normalized color histogram of all pixels in I. Then, h_c is converted into a 1D vector. In our experiments, normalized r-g-b color components are employed to compute h_c.



(a) (b)

Figure 5.1: Sample image regions of pedestrian lane images: (a) image regions partially covering a lane marker, (b) image regions partially covering a lane boundary.



(a) (b)

(c)

Figure 5.2: Feature extraction of an example instance I in Fig. 5.1: (a) color feature of I, (b) texture feature of I, (c) feature vector of I.

Figure 5.2(a) shows the color feature of a sample instance I in Fig. 5.1(a).

The texture feature is computed as the normalized histogram h_o of edge orientations of all pixels in I. The edge orientation of a pixel is weighted by the

gradient magnitude of the pixel. Let G_x and G_y denote the derivatives of the input grayscale image in the horizontal and vertical directions. These derivatives are obtained by using the horizontal and vertical filters with kernels $[-1, 0, 1]$. For each pixel location (x, y), the gradient magnitude $G(x, y)$ and orientation $\theta(x, y)$ are estimated as

$$\begin{cases} G(x,y) = \sqrt{(G_x(x,y)^2 + G_y(x,y)^2}, \\ \theta(x,y) = \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right). \end{cases} \tag{5.1}$$

Figure 5.2(b) shows the texture feature of a sample instance I in Fig. 5.1(a).

Finally, a feature vector describing instance I is generated by aggregating h_c and h_o as

$$\mathbf{f} = h_c \oplus h_o. \tag{5.2}$$

Here, $\oplus$ denotes the concatenation operation. Equation (5.2) shows that, for each instance I, instance-level feature vector $\mathbf{f}$ has $N_c^3 + N_o$ elements, where N_c is the bin number of each color component and N_o is the number of orientation bins. Figure 5.2 (c) shows a sample instance-level feature vector that is a combination of the color feature in Fig. 5.2(a) and the texture feature in Fig. 5.2(b).

Since each instance I is described by a feature vector, the input image is represented as a bag of feature vectors $X = \{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, ...\}$.

5.3.2 Lane classification

Given a training set, we have a set of bags $\mathcal{X}_T = \{X_1, X_2, ..., X_M\}$ and a set of corresponding labels $\mathcal{Y} = \{y_1, y_2, ..., y_M\}$. Here, $X_i = \{\mathbf{f}_{1i}, \mathbf{f}_{2i}, ...\}$ is a set of instance-level feature vectors estimated as in Section 5.3.1, y_i is the label of bag X_i, and $y_i \in \{\omega_1, \omega_2, \omega_3\}$ (ω_k is a label value for class k). Let $\mathcal{I}_T$ denote a set of instance-level feature vectors of all bags in $\mathcal{X}_T$, i.e $\mathcal{I}_T = \{\mathbf{f}_{11}, \mathbf{f}_{21}, ..., \mathbf{f}_{jk}, ...\}$ ($\mathbf{f}_{jk}$ denotes the j-th instance-level feature vector of the bag X_k). The multiple instance learning for categorizing images of pedestrian lanes is determined as follows.

In the first stage, we employ the k-means algorithm to cluster $\mathcal{I}_T$ into N groups $(C_1, C_2, ..., C_N)$. The k-means algorithm includes the following steps:

1. Initialize group centroids $V = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N\}$ as N random elements in $\mathcal{I}_T$.

2. Assign each element $\mathbf{f}_{ij}$ in $\mathcal{I}_T$ to a group C_k if

$$d(\mathbf{f}_{ij}, \mathbf{v}_k) = \min_{\mathbf{v} \in V} d(\mathbf{f}_{ij}, \mathbf{v}), \tag{5.3}$$

   where $d(\mathbf{f}_{ij}, \mathbf{v}_k)$ is the Euclidean distance between two vectors $\mathbf{f}_{ij}$ and $\mathbf{v}_k$.

3. Recalculate the group centroids. The centroid of a group is calculated as the average vector of all element vectors in the group.

4. Repeat step 2 and 3 until the centroids converge.

After clustering, for each group C_i, we compute the standard deviation σ_i of the Euclidean distances from the centroid $\mathbf{v}_i$ to all elements in C_i. The vocabulary is finally constructed as a set $\mathcal{V} = \{(\mathbf{v}_1, \sigma_1), (\mathbf{v}_2, \sigma_2), ..., (\mathbf{v}_N, \sigma_N)\}$.

In the second stage, each bag $X_i \in \mathcal{X}_T$ is transformed into a N-dimension vector $\mathbf{z}_i = (z_{1i}, z_{2i}, ..., z_{Ni})$. The elements z_{ki} of $\mathbf{z}_i$ is computed as

$$z_{ki} = \max_{\mathbf{f} \in X_i} \exp\left(-\frac{d(\mathbf{f}, \mathbf{v}_k)^2}{2\sigma_k^2}\right). \tag{5.4}$$

Equation (5.4) means that z_{ki} is high if the k-th word in $\mathcal{V}$ is similar to an instance in bag X_i, and z_{ki} is low if the k-th word is dissimilar to instances in X_i. Consequently, the samples (X_i, y_i) are mapped to samples $(\mathbf{z}_i, y_i)$ $(i = 1, 2, ..., M)$.

In the final stage, the SVM classifiers are learned from the set of samples $(\mathbf{z}_i, y_i)$, using the *LIBSVM* tool proposed in [142]. Here, we choose the SVM classifiers because their high accuracy have been found in many different practical applications [143–145]. The Gaussian radial basis kernel function is employed for the SVM classifiers. Compared with the several other kernels (linear and polynomial), the radial basis kernel function is designed to solve complex non-linear problems. Support vector machines are originally formulated for two-class classification problems. To handle categorizing images into three classes (*marked-lane*, *unmarked-lane* and *non-lane*), we employ the one-versus-all approach. In this approach, a k-class problem is decomposed into k two-class problems [145]. Each SVM classifier is trained with all training samples. For the i-th classifier $(i \leq k)$, samples in the i-th class are labeled as positive and samples in all other classes are labeled as negative.

For the testing set, bags are also transformed first into N-dimension vectors as in the second stage of training. Then, testing bags are categorized by the trained classifiers. Finally, the labels of the bags are assigned to the class label that has the highest classification score.

5.4 Experiments and Results

This section presents image data, parameters in the proposed method, and the experimental results of pedestrian lane classification.

5.4.1 Image data

The proposed method for pedestrian lane classification is evaluated on a new data set of 6000 images that is collected in different environments and under various weather and illumination conditions. The data set includes 2000 images of pedestrian lanes with markers, 2000 images of pedestrian lanes without markers, and 2000 images without pedestrian lanes. We used 600 images of marked-lanes, 500 images of unmarked-lanes and 600 images of non-lanes for training. The remaining images were employed for testing. Figure 5.3 shows examples of image types.

5.4.2 Selection of parameters

This subsection presents the parameters employed in the proposed method. These parameters were estimated based on the training set as follows.

The size w of instances in Section 5.3.1 was found based on the width of lane markers in the training set. We have been expected to capture the properties of image regions that contain partially a lane marker or lane boundary. These image regions are discriminative and representative for images of pedestrian lanes. Our experiments show that the maximum width of lane markers in an image is smaller than $0.15W$, where W is the width of the image. Therefore, the value of w was chosen as $w = 0.2W$.

To determine a suitable size for the vocabulary in Section 5.3.2, we implemented the proposed method with different values N, ranging from 300 to 1300,

(a) marked-lane images

(b) unmarked-lane images

(c) non-lane images

Figure 5.3: Examples of image types for lane classification.

Table 5.1: The accuracy of pedestrian lane classification for different sizes of the vocabulary.

Vocabulary size N	300	500	700	900	1100	1300
Overall accuracy (%)	98.3	99.6	99.8	100.0	99.9	99.7

on the training set. Table 5.1 shows the accuracy of pedestrian lane classification with the various sizes of the vocabulary. The highest accuracy of pedestrian lane classification was obtained when $N = 900$; this value was chosen in our experiments.

Similarly, analyzing the accuracy of pedestrian lane classification on the training set, we also found the optimal bin number N_c of each color component and the optimal bin number N_o of gradient orientations for estimating the color and orientation histograms as $N_c = 5$ and $N_o = 18$. Table 5.2 shows the accuracy of lane classification with different bin numbers for each color component and the orientation bin number $N_o = 18$. Table 5.3 shows the accuracy of lane classification with different orientation bin numbers and the color bin number $N_c = 5$.

Table 5.2: The accuracy of pedestrian lane classification with different bin numbers for each color component and the orientation bin number $N_o = 18$.

Color bin number N_c	3	4	5	6	7	8
Overall accuracy (%)	93.3	95.9	97.2	96.9	96.3	95.3

Table 5.3: The accuracy of pedestrian lane classification with different orientation bin numbers and the color bin number $N_c = 5$.

Color bin number N_o	9	18	36	72
Overall accuracy (%)	94.1	97.2	96.9	95.9

5.4.3 Analysis of the feature extraction in the proposed method

To analyze the effectiveness of the feature extraction for instances in the proposed method, we evaluated the proposed method on the testing set in three scenarios:

1) Using only the histogram of gradient orientations (HoG) for representing instances.

2) Using only the histogram of colors (HoC) for representing instances.

3) Combining the color and orientation histograms (HoG+HoC) for representing instances.

Table 5.4 shows the accuracy of using different features for classifying images. Using the HoG+HoC feature, the proposed method obtained accuracy of 95.0%. Using only the HoG feature, the accuracy decreased to 68.3%. Furthermore, the proposed method using the HoG+HoC feature also achieved better accuracy than using only the HoC (only 86.3%).

Table 5.4: Accuracy comparison of using different features for pedestrian lane classification.

Features	*HoG* (%)	*HoC* (%)	***HoG + HoC*** (%)
Marked-lane images	62.4	86.8	**96.4**
Unmarked-lane images	71.5	85.9	**95.2**
Non-lane images	65.4	87.2	**91.7**
Overall	68.3	86.3	**95.0**

5.4.4 Analysis of the multiple instance learning model in the proposed method

To evaluate the effectiveness of the multiple instance learning model in the proposed method, we implemented the following two methods on the testing set:

- The bag-based method: This algorithm adopts the multiple instance learning framework in [141]. It uses a vocabulary to transform bags into feature vectors and then employs the SVM classifiers for categorizing bags. However, the words of the vocabulary are determined as the centroids of groups of similar bags. The similarity between two bags is measured by the Hausdorff distance. Each bag is converted into a feature vector where its elements are the Hausdorff distances from the bag to the words of the vocabulary.

- The bag-of-words method: This technique employs the bag-of-words model to categorize images. In this technique, we employ the same vocabulary as in the proposed method, and detect the visual words in each image by matching image regions (instances in the proposed method) with the words in the vocabulary. An image is then represented by a histogram of the detected words in the image. The image classification is similar to the proposed method.

Table 5.5: The lane classification accuracy of different methods.

Methods	*BoW method* (%)	*Bag-based method* (%)	**Proposed method** (%)
Marked-lane images	89.8	95.2	**96.4**
Unmarked-lane images	85.7	94.9	**95.2**
Non-lane images	96.2	84.1	**91.7**
Overall	88.7	92.7	**95.0**

Table 5.5 shows the lane classification accuracy for the different methods. The proposed method with accuracy of 95.0% outperformed the BoW method that had accuracy of only 88.7%. Furthermore, the accuracy of the proposed method was better than that of the bag-based method (92.7%). These results demonstrate the robustness and effectiveness of the proposed method for lane classification.

5.4.5 Analysis of using the lane classification in detecting pedestrian lanes

To evaluate the effectiveness of using the lane classification in detecting pedestrian lanes, we implemented lane detection *with* and *without* lane classification on the testing set, which includes 1400 images of marked-lanes, 1500 images of unmarked-lanes and 1400 images of non-lanes as follows:

- For lane detection *with* lane classification, the lane type of each input image is first identified and a suitable lane detector is then selected to find the walking lane in the input image. If the input image is an image of marked-lanes, the marked-lane detector proposed in Chapter 3 is applied. If the input image is an image of unmarked-lanes, the unmarked-lane detector proposed in Chapter 4 is employed.

- For lane detection *without* lane classification, we employ both the marked-lane and unmarked-lane detectors to find the walking lane in each input image. The walking lane is determined as the output of the detector that has the highest detection score.

We evaluated the performance of pedestrian lane detection *with* and *without* lane classification as in Chapter 3 and Chapter 4.

Table 5.6: The performance of lane detection *with* and *without* lane classification on the testing set.

Methods	*Recall (%)*	*Precision (%)*	*F-measure (%)*	*Average processing time (s)*
With lane classification	**94.1**	**97.6**	**95.8**	**1.1**
Without lane classification	83.1	83.9	83.5	2.1

Table 5.6 summarizes the performance of pedestrian lane detection *with* and *without* lane classification on the testing set. *Without* using lane classification, the lane detection had a recall rate of 83.1%, a precision rate of 83.9% and F-measure of 83.5%. Using lane classification, the recall and precision rates and F-measure significantly increased to 94.1%, 97.6% and 95.8%, respectively. Furthermore, using lane classification, the average processing time of lane detection (1.1 s) was significantly shorter than *without* using lane classification (2.1 s).

5.5 Chapter summary

This chapter presents a method for lane classification using multiple instance learning. The proposed method employs edge- and color-based features to represent the properties of both lane borders and lane markers. Each image is considered as a bag of instances, which are image regions and represented by feature vectors. Image classification is performed based on the multiple instance learning model. The proposed method is evaluated on a large and new data set collected from various environments, under varying illumination conditions. The experimental results show that the proposed method classifies robustly pedestrian lane images. The experimental results also prove that the lane detection performance is significantly improved by using lane classification.

Chapter 6

Conclusion

Chapter contents

Among essential activities in daily life, traveling safely and independently in different environments is a challenging task of vision-disabled people, and hence an assistive navigation system is necessary. This thesis investigates vision-based techniques to design an assistive system of pedestrian lane detection for the visually impaired. The system aims to locate automatically the walking region in front of the traveler in each scene from images captured by a camera In this chapter, Section 6.1 summarizes the major research activities undertaken during this project. Section 6.2 presents future research directions. Section 6.3 gives concluding remarks.

6.1 Research summary

This research focuses on automatic detection of pedestrian lanes in different environments. The activities have been documented in several chapters of the thesis, and are summarized as follows:

- **Chapter 2**: Literature review. In this chapter, we investigate the travels of visually impaired people, and comprehensively review on the literature of traveling aids for the people as

  - Vision-disabled people reply mainly on hearing and touching to perceive the surrounding environment, and therefore have difficulties in traveling from one place to other places.
  - Traditional aids includes white canes and guide dogs. These tools have been used widely in travels of vision-disabled people. While a white cane only assists the blind traveler in detecting obstacles and finding the walking path at a close range, a guide dog assists the traveler in avoiding obstacles and hazards, and following the familiar routes. However, these tools requires a lot of time for training and cannot assist the users in traveling safely and independently in various environments.
  - Electronic obstacle detection devices are based on the reflection principle of ultrasound or optical signals to represent the distance map of obstacles in front of the traveler.
  - GPS-based devices employ a GPS receiver and GIS database to provide the information of the traveling orientation, routes and locations for users.
  - Computer-vision based systems use images captured from cameras to represent the surrounding environment of the traveler. However, automatically finding paths still absent in the existing computer-vision based systems.

- **Chapter 3**: Marked-lane detection. This chapter focuses on detecting pedestrian lanes at traffic junctions. These lanes are located by painted markers. The contents of the chapter is briefly described as
  - We propose a method for detecting pedestrian crossing lanes at traffic junctions. The proposed method extracts first patches of interest that are local image regions located on the lane marker borders in the input image using color and orientation features. Next, lane markers are found from the detected POIs, using a Markov random field. Finally, the lane markers are verified using multiple geometric cues.
  - To evaluate methods for detecting pedestrian crossing lanes, we created

a new and large data set of 2000 images with manually annotated detection ground-truth. The data set is collected from different scenes, under challenging illumination conditions, and includes pedestrian crossing lanes with various marker types: solid or dash. In many cases, the lane markers are eroded partially or covered by shadows and lighting areas.

  - The experimental results demonstrate that proposed method detects robustly pedestrian crossing lanes in various scenes under challenging illumination conditions. The results also show that the proposed method outperforms existing methods that are based on edge features or color features.

- **Chapter 4**: Unmarked-lane detection. The chapter concentrates on pedestrian lane detection in unstructured environments, where lanes have no painted markers, vary significantly in appearance, are affected by shadows and lighting areas, and have different shapes (e.g. straight or curved). The contents of the chapter is highlighted as

  - A new method is proposed for detecting pedestrian lanes in unstructured environments. The walking lane in the input image is determined from color homogeneous regions, using color and shape features. The color model of the lane surface is learned on-the-fly from a sample region, which is extracted directly from the input image based on the vanishing point and the characteristics of lane borders and lane surfaces. The shape of pedestrian lane is modeled using shape contexts.

  - A fast and robust method is proposed for vanishing point detection. Vanishing point detection is based on voting local orientations of edge pixels. To estimate robustly local orientations and edge pixels under severe illumination conditions, multiple color channels are employed, instead of only the intensity channel.

  - We created a new and large data set of 2000 images with manually annotated detection ground-truth for evaluating pedestrian unmarked-

lane detection methods. The data set is collected from various indoor and outdoor scenes of unmarked-lanes under different illumination conditions.

  - The experimental results of vanishing point detection show that the proposed method has higher accuracy than state-of-arts methods. Furthermore, the processing time of the proposed method is significantly shorter than existing methods, and sufficient for assistive navigation of visually impaired people.

  - The experimental results of unmarked-lane detection demonstrate that the proposed method detects robustly unmarked-lanes with different shapes and surface patterns under challenging illumination conditions.

- **Chapter 5**: Lane classification. This chapter addresses to identifying the pedestrian lane type in images captured from a camera. Identifying the lane type enables to select a suitable method for detecting the pedestrian lane in each scene. The major contents of the chapter are:

  - Exploiting image regions that contain partially a lane marker or a lane boundary to extract the discriminative features for each lane type. The features of an image region include the color and orientation histograms of all pixels in the region.

  - Investigating a multiple instance learning model for categorizing images into three classes: marked-lane, unmarked-lane and non-lane. Each image is a bag of multiple instances, and each instance is an image region. Bags are then transformed into global feature vectors using a vocabulary. Finally, the SVM classifiers are trained to identify unknown bags.

  - The proposed method for lane classification was evaluated on a large and new data set of 6000 images. The experimental results have shown that the proposed method classifies robustly input images. Furthermore, these results have proved the effectiveness of using lane classification for detecting different pedestrian lanes.

6.2 Future work

Following the investigations presented in this thesis, improvements to the proposed approaches that could be made in the future includes:

- Applying parallel computing techniques to enhance the processing speed of unmarked-lane detection. In the proposed method, vanishing point estimation has significant processing time because computing the voting scores is carried out sequentially at each image pixel. By using parallel computing techniques, the voting scores are estimated simultaneously for multiple pixels. In another way, we can divide the lane detection process in three phases: *sample region selection*, *image segmentation* and *lane detection*. The sample region selection and image segmentation phases are possible to be performed in parallel because these phases are independent of each other.

- Applying pedestrian detection to cope with occlusion. Inspired by the fact that the walking region, lane markers or lane borders are not fully visible due to the presence of pedestrians on the lane, we can consider occlusion as an effect of pedestrians appearing on the lane region. Considering recent advances in pedestrian detection [66], the results of pedestrian detection are used to infer image regions on the pedestrian lane where occlusion may occur.

- Extending the proposed method for lane marker detection in Chapter 3 to detect pedestrian lanes of zebra patterns at traffic intersection. The proposed method is possible to apply for detecting lane markers of the zebra crossing lanes. Since the layout of lane markers in the zebra lanes differs from the lanes identified by two white stripes, we need to design an algorithm for verifying the detected lane markers of the zebra lane.

- Exploiting the relations between several adjacent image frames in subsequent image frames for pedestrian lane detection. In adjacent frames, the lane varies little in appearance, shape and location. This cue is useful to detect the lane in next frames and verify the lane region in each frame.

- Using 3D cameras for detecting pedestrian lanes. The 3D cameras provide the depth map of objects in each scene, and hence we can obtain the 3D structure of the scene. Understanding the 3D structure of the scene is more powerful to extract the lane region from the background.

6.3 Conclusion

This thesis presents a vision-based system to detect robustly pedestrian lanes in different environments for assistive navigation of visually impaired people. The system aims to locate the walking lane in front of the traveler in each scene. The major tasks of the system include:

(i) Identifying the lane type in the image captured from a camera.

(ii) Detecting pedestrian lanes with painted markers.

(iii) Detecting pedestrian lanes without painted markers.

Three different approaches are proposed and each approach is designed to handle a task of the system. A large and new data set of images is created to evaluate the proposed approaches. The experimental results have shown the effectiveness and robustness of the proposed approaches in comparison with existing algorithms.

References

[1] S. Se, "Zebra-crossing detection for the partially sighted," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 211–217.

[2] D. Schreiber, B. Alefs, and M. Clabian, "Single camera lane detection and tracking," in *IEEE Conference on Intelligent Transportation Systems*, 2005, pp. 302–307.

[3] M. C. Le, S. L. Phung, and A. Bouzerdoum, "Pedestrian lane detection for assistive navigation of blind people," in *International Conference on Pattern Recognition*, 2012, pp. 2594–2597.

[4] ——, "Pedestrian lane detection for the visually impaired," in *International Conference on Digital Image Computing Techniques and Applications*, 2012, pp. 1–6.

[5] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[6] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-snake," *Image and Vision Computing*, vol. 22, no. 4, pp. 269–280, 2004.

[7] H. Kong, J. Y. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Transactions on mage Processing*, vol. 19, no. 8, pp. 2211–2220, 2010.

[8] M. C. Le, S. L. Phung, and A. Bouzerdoum, "Pedestrian lane detection in unstructured environments for assistive navigation," in *International Conference on Digital Image Computing Techniques and Applications*, 2014, pp. 1–8.

[9] ——, "Lane detection in unstructured environments for autonomous navigation systems," in *Asian Conference on Computer Vision*, 2014, pp. 1–16.

[10] "Visual impairment and blindness," WHO, Tech. Rep. Fact Sheet No. 282, 2013.

[11] L. Kay, "A sonar aid to enhance spatial perception of the blind: engineering design and evaluation," *Radio and Electronic Engineer*, vol. 44, no. 11, pp. 605–627, 1974.

[12] J. Borenstein and I. Ulrich, "The guidecane - A computerized travel aid for the active guidance of blind pedestrians," in *IEEE International Conference on Robotics and Automation*, 1997, pp. 1283–1288.

[13] GDP Research, "The miniguide ultrasonic mobility aid," Tech. Rep., 2011. [Online]. Available: http://www.gdp-research.com.au

[14] J. M. Loomis, R. G. Golledge, R. L. Klatzky, J. M. Speigle, and J. Tietz, "Personal guidance system for the visually impaired," in *ACM conference on Assistive technologies*, 1994, pp. 85–91.

[15] H. Petrie, V. Johnson, T. Strothotte, A. Raab, R. Michel, L. Reichert, and A. Schalt, "MoBIC: An aid to increase the independent mobility of blind travellers," *British Journal of Visual Impairment*, vol. 15, no. 2, pp. 63–66, 1997.

[16] A. Helal, S. E. Moore, and B. Ramachandran, "Drishti: An integrated navigation system for visually impaired and disabled," in *International Symposium on Wearable Computers*, 2001, pp. 149–156.

[17] V. Ivanchenko, J. Coughlan, and S. Huiying, "Detecting and locating crosswalks using a camera phone," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1–8.

[18] M. Brambring, "Mobility and orientation processes of the blind," in *Electronic spatial sensing for blind*, D. Warren and E. Strelow, Eds., Dordrecht, the Netherlands, Nijhoff, 1985, pp. 493–508.

[19] P. Strumillo, "Electronic interfaces aiding the visually impaired in environmental access, mobility and navigation," in *Conference on Human System Interactions*, 2010, pp. 17–24.

[20] M. A. Hersh and M. A. Johnson, *Assistive technology for visually impaired and blind people*. Springer, 2008.

[21] Guide Dogs NSW ACT, "Guide dog training," Tech. Rep., 2014. [Online]. Available: http://www.guidedogs.com.au/guide-dogs/guide-dog-training

[22] A. J. Jackson and J. S. Wolffsohn, "Low vision manual," 2007.

[23] J. M. Benjamine, N. A. Ali, and A. F. Schepis, "A laser cane for the blind," in *San Diego Biomedical Symposium*, 1973, p. 5357.

[24] Sten Lofving, "The talking cane," Tech. Rep., 2014. [Online]. Available: http://www.opticalsensors.se/talkingcane.html

[25] Sound Foresight Technology, "Ultracane," Tech. Rep., 2014. [Online]. Available: http://www.ultracane.com

[26] S. Shoval, J. Borenstein, and Y. Koren, "Auditory guidance with the Navbelt - A computerized travel aid for the blind," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 28, no. 3, pp. 459–467, 1998.

[27] Humanware, "Trekker Breeze," Tech. Rep., 2014. [Online]. Available: http://www.humanware.com/en-usa/products/blindness/talking_gps/trekker_breeze/_details/id_101/trekker_breeze.html

[28] Sendero Group, "BrailleNote GPS," Tech. Rep., 2014. [Online]. Available: http://www.humanware.com/en-usa/products/blindness/talking_gps/braillenote_gps_software_and_receiver_package/_details/id_325/braillenote_gps_software_and_receiver_package.html

[29] Freedom Scientific, "Streettalk gps solution," Tech. Rep., 2014. [Online]. Available: http://www.freedomscientific.com/fs_products/StreetTalk_info.asp

[30] V. Garaj, R. Jirawimut, P. Ptasinski, F. Cecelja, and W. Balachandran, "A system for remote sighted guidance of visually impaired pedestrians," *The British Journal of Visually Impaired*, vol. 21, no. 2, pp. 55–63, 2003.

[31] M. Bujacz, P. Baranski, M. Moranski, P. Strumillo, and A. Materka, "Remote guidance for the blind and visually impaired: a proposed teleassistance system and navigation trials," in *Conference on Human System Interactions*, 2008, pp. 888–892.

[32] N. A. Giudice and G. E. Legge, *Blind navigation and the role of technology*, ser. The Engineering Handbook of Smart Technology for Aging, Disability and Independence. John Wiley, 2008.

[33] P. B. L. Meijer, "An experimental system for auditory image representations," *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 2, pp. 112–121, 1992.

[34] A. Arnoldussen and D. C. Fletcher, "Visual perception for the blind: The brainport vision device," vol. 9, 2012. [Online]. Available: http://www.retinalphysician.com/articleviewer.aspx?articleid=106585

[35] N. Bourbakis, "Sensing surrounding 3-d space for navigation of the blind," *IEEE Engineering in Medicine and Biology Magazine*, vol. 27, no. 1, pp. 49–55, 2008.

[36] J. L. Gonzlez-Mora, A. Rodrguez-Hernndez, L. F. Rodrguez-Ramos, L. Daz-Saco, and N. Sosa, "Development of a new space perception system for blind people, based on the creation of a virtual acoustic space," *Engineering Applications of Bio-Inspired Artificial Neural Networks*, vol. 1607, pp. 321–330, 1999.

[37] S. Meers and K. Ward, "A substitute vision system for providing 3D perception and GPS navigation via electro-tactile stimulation," in *International Conference on Sensing Technologies*, 2005, p. 2123.

[38] L. A. Johnson and C. M. Higgins, "A navigation aid for the blind using tactile-visual sensory substitution," in *IEEE International Conference on Engineering in Medicine and Biology Society*, 2006, pp. 6289–6292.

[39] P. Vera, D. Zenteno, and J. Salas, "A smartphone-based virtual white cane," *Pattern Analysis and Applications*, pp. 1–10, 2013.

[40] C. Capelle, C. Trullemans, P. Arno, and C. Veraart, "A real-time experimental prototype for enhancement of vision rehabilitation using auditory substitution," *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 10, pp. 1279–1293, 1998.

[41] N. Molton, S. Se, J. M. Brady, D. Lee, and P. Probert, "A stereo vision-based aid for the visually impaired," *Image and Vision Computing*, vol. 16, no. 4, pp. 251–263, 1998.

[42] D. Dakopoulos and N. G. Bourbakis, "Wearable obstacle avoidance electronic travel aids for blind: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 1, pp. 25–35, 2010.

[43] M. S. Uddin and T. Shioyama, "Bipolarity and projective invariant-based zebra-crossing detection for the visually impaired," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2005, pp. 22–30.

[44] M. Radvanyi, B. Varga, and K. Karacs, "Advanced crosswalk detection for the bionic eyeglass," in *International Workshop on Cellular Nanoscale Networks and Their Applications*, 2010, pp. 1–5.

[45] S. Se and M. Brady, "Road feature detection and estimation," *Machine Vision and Applications*, vol. 14, no. 3, pp. 157–165, 2003.

[46] H. Deng and D. A. Clausi, "Unsupervised image segmentation using a simple MRF model with a new implementation scheme," *Pattern Recognition*, vol. 37, no. 12, pp. 2323–2335, 2004.

[47] A. H. S. Lai and N. H. C. Yung, "Lane detection by orientation and length discrimination," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 30, no. 4, pp. 539–548, 2000.

[48] Z. W. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 16–26, 2008.

[49] C. R. Jung and C. R. Kelber, "Lane following and lane departure using a linear-parabolic model," *Image and Vision Computing*, vol. 23, no. 13, pp. 1192–1202, 2005.

[50] P. Charbonnier, F. Diebolt, Y. Guillard, and F. Peyret, "Road markings recognition using image processing," in *IEEE Conference on Intelligent Transportation System*, 1997, pp. 912–917.

[51] H. Cheng, B. Jeng, P. Tseng, and K. C. Fan, "Lane detection with moving vehicles in the traffic scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 571–582, 2006.

[52] Q. Li, N. Zheng, and H. Cheng, "Springrobot: a prototype autonomous vehicle and its algorithms for lane detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 300–308, 2004.

[53] A. A. M. Assidiq, O. O. Khalifa, R. Islam, and S. Khan, "Real time lane detection for autonomous vehicles," in *International Conference on Computer and Communication Engineering*, 2008, pp. 82–88.

[54] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62–81, 1998.

[55] S.-S. Ieng, J.-P. Tarel, and R. Labayrade, "On the design of a single lane-markings detectors regardless the on-board camera's position," in *IEEE Intelligent Vehicles Symposium*, 2003, pp. 564–569.

[56] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, 2006.

[57] A. Lopez, J. Serrat, C. Canero, F. Lumbreras, and T. Graf, "Robust lane markings detection and road geometry computation," *International Journal of Automotive Technology*, vol. 11, no. 3, pp. 395–407, 2010.

[58] T. Veit, J. P. Tarel, P. Nicolle, and P. Charbonnier, "Evaluation of road marking feature extraction," in *International IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 174–181.

[59] C. Topal, C. Akinlar, and Y. Genc, "Edge drawing: A heuristic approach to robust real-time edge detection," in *International Conference on Pattern Recognition*, 2010, pp. 2424–2427.

[60] R. B. Potts, "Some generalized order-disorder transformations," *Proceedings of the Cambridge Philosophical Society*, vol. 48, no. 1, pp. 106–109, 1952.

[61] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[62] N. Metropolis, A. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, p. 10871091, 1953.

[63] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.

[64] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 48, no. 3, pp. 259–302, 1986.

[65] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[66] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[67] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[68] P. L. Rosin, "Unimodal thresholding," *Pattern Recognition*, vol. 34, no. 11, pp. 2083–2096, 2001.

[69] J. D. Crisman and C. E. Thorpe, "Scarf: A color vision system that tracks roads and intersections," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, pp. 49–58, 1993.

[70] J. M. Alvarez and A. M. Lopez, "Road detection based on illuminant invariance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 184–193, 2011.

[71] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *European Conference on Computer Vision*, 2012, pp. 376–389.

[72] C. Tan, H. Tsai, T. Chang, and M. Shneier, "Color model-based real-time learning for road following," in *IEEE Conference on Intelligent Transportation Systems*, 2006, pp. 939–944.

[73] Y. Sha, G.-y. Zhang, and Y. Yang, "A road detection algorithm by boosting using feature combination," in *IEEE Intelligent Vehicles Symposium*, 2007, pp. 364–368.

[74] J. M. Alvarez, T. Gevers, and A. M. Lopez, "Vision-based road detection using road models," in *IEEE International Conference on Image Processing*, 2009, pp. 2073–2076.

[75] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 470–477.

[76] M. Sotelo, F. Rodriguez, L. Magdalena, L. Bergasa, and L. Boquete, "A color vision-based lane tracking system for autonomous driving on unmarked roads," *Autonomous Robots*, vol. 16, no. 1, pp. 95–116, 2004.

[77] O. Ramstrom and H. Christensen, "A method for following unmarked roads," in *IEEE Intelligent Vehicles Symposium*, 2005, pp. 650–655.

[78] Y. He, H. Wang, and B. Zhang, "Color-based road detection in urban traffic scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 309–318, 2004.

[79] C. Oh, J. Son, and K. Sohn, "Illumination robust road detection using geometric information," in *International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 1566–1571.

[80] O. Miksik, P. Petyovsky, L. Zalud, and P. Jura, "Robust detection of shady and highlighted roads for monocular camera based navigation of ugv," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 64–71.

[81] J. Crisman and C. Thorpe, "Unscarf, a color vision system for the detection of unstructured roads," in *IEEE International Conference on Robotics and Automation*, 1991, pp. 2496 – 2501.

[82] C.-K. Chang, C. Siagian, and L. Itti, "Mobile robot monocular vision navigation based on road region and boundary estimation," in *IEEE/RSJ International Conference on on Intelligent Robots and Systems*, 2012, pp. 1043–1050.

[83] J. P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection," in *International Conference on Computer Vision*, 2009, pp. 1250–1257.

[84] F. A. Andal, G. Taubin, and S. Goldenstein, "Vanishing point detection by segment clustering on the projective space," in *European Conference on Computer Vision*, 2012, pp. 324–337.

[85] P. Moghadam, J. A. Starzyk, and W. S. Wijesoma, "Fast vanishing-point detection in unstructured environments," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 425–430, 2012.

[86] J. v. d. Weijer, T. Gevers, and A. W. M. Smeulders, "Robust photometric invariant features from the color tensor," *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 118–127, 2006.

[87] T. Gevers, S. A.W.M., and H. Stokman, "Photometric invariant region detection," in *British Machine Vision Conference*, 1998, pp. 659–669.

[88] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 4, pp. 509–522, 2002.

[89] H. Chang and R. Nevatia, "High performance object detection by collaborative learning of joint ranking of granules features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 41–48.

[90] M. Szummer and R. W. Picard, "Indoor-outdoor image classification," in *IEEE International Workshop on Content-Based Access of Image and Video Database*, 1998, pp. 42–51.

[91] N. Serrano, A. E. Savakis, and J. Luo, "Improved scene classification using efficient low-level features and semantic cues," *Pattern Recognition*, vol. 37, no. 9, pp. 1773–1784, 2004.

[92] J. Vogel and B. Schiele, "Semantic modeling of natural scenes for content-based image retrieval," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 133–157, 2007.

[93] J. Fan, Y. Gao, H. Luo, and G. Xu, "Statistical modeling and conceptualization of natural images," *Pattern Recognition*, vol. 38, no. 6, pp. 865–885, 2005.

[94] J. Mao and A. K. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models," *Pattern Recognition*, vol. 25, no. 2, pp. 173–188, 1992.

[95] S. Paek and S. fu Chang, "A knowledge engineering approach for image classification based on probabilistic reasoning systems," in *IEEE International Conference on Multimedia and Expo*, 2000, pp. 1133–1136.

[96] A. Mojsilovic, J. Gomes, and B. E. Rogowitz, "Isee: Perceptual features for image library navigation," in *SPIE Human vision and electronic imaging*, 2002, pp. 266–277.

[97] J. Luo, A. E. Savakis, and A. Singhal, "A bayesian network-based framework for semantic image understanding," *Pattern Recognition*, vol. 38, no. 6, pp. 919–934, 2005.

[98] S. Aksoy, K. Koperski, C. Tusk, G. Marchisio, and J. C. Tilton, "Learning bayesian classifiers for scene classification with a visual grammar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 581–589, 2005.

[99] D. Comaniciu, P. Meer, and S. Member, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603–619, 2002.

[100] C. Fredembach, M. Schrder, and S. Ssstrunk, "Eigenregions for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1645–1649, 2003.

[101] A. Vailaya, M. Figueiredo, A. Jain, and Z. Hongjiang, "Content-based hierarchical classification of vacation images," in *IEEE International Conference on Multimedia Computing and Systems*, 1999, pp. 518–523 vol.1.

[102] A. Vailaya, M. A. Figueiredo, A. K. Jain, and Z. Hong-Jiang, "Image classification for content-based indexing," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 117–130, 2001.

[103] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[104] M. M. Ali, M. B. Fayek, and E. E. Hemayed, "Human-inspired features for natural scene classification," *Pattern Recognition Letters*, vol. 34, no. 13, pp. 1525–1530, 2013.

[105] M. M. Gorkani and R. W. Picard, "Texture orientation for sorting photos "at a glance"," in *International Conference on Pattern Recognition*, 1994, pp. 459–464.

[106] A. Vailaya, A. Jain, and H. J. Zhang, "On image classification: City images vs. landscapes," *Patter Recognition*, vol. 31, pp. 1921–1935, 1998.

[107] E. Chang, G. Kingshy, G. Sychay, and G. Wu, "Cbsa: Content-based soft annotation for multimodal image retrieval using bayes point machines," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 1, pp. 26–38, 2003.

[108] A. B. Torralba and A. Oliva, "Semantic organization of scenes using discriminant structural templates," in *IEEE International Conference on Computer Vision*, 1999, pp. 1253–1258.

[109] A. Oliva and A. Torralba, "Building the gist of a scene: The role of global image features in recognition," in *Progress in Brain Research*. Elsevier, 2006, pp. 23–36.

[110] S. Grossberg and T. ren Huang, "ARTSCENE: a neural system for natural scene classification," *Journal of Vision*, vol. 9, pp. 493–497, 2003.

[111] W. Jianxin and J. M. Rehg, "Centrist: A visual descriptor for scene categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1489–1501, 2011.

[112] K. Gazolli and E. Salles, "A contextual image descriptor for scene classification," in *Trends in Innovative Computing*, 2012.

[113] ——, "Exploring neighborhood and spatial information for improving scene classification," *Pattern Recognition Letters*, vol. 46, no. 0, pp. 83–88, 2014.

[114] T. Song and H. Li, "Wavelbp based hierarchical features for image classification," *Pattern Recognition Letters*, vol. 34, no. 12, pp. 1323–1328, 2013.

[115] T. Ojala, M. Pietikinen, and T. Menp, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[116] A. Bosch, A. Zisserman, and X. Muoz, "Scene classification using a hybrid generative/discriminative approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 712–727, 2008.

[117] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 524–531.

[118] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2169–2178.

[119] D. Blei and M. Jordan, "Latent dirichlet allocation," *Machine Learning Research*, vol. 3, pp. 177–196, 2001.

[120] G. Gu, Y. Zhao, and Z. Zhu, "Integrated image representation based natural scene classification," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11 273–11 279, 2011.

[121] P. Quelhas, F. Monay, J. M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool, "Modeling scenes with local descriptors and latent aspects," in *International Conference on Computer Vision*, vol. 1, 2005, pp. 883–890 Vol. 1.

[122] A. Bosch, A. Zisserman, and X. Mu?oz, "Scene classification via plsa," in *European Conference on Computer Vision*, 2006, pp. 517–530.

[123] D. G. Lowe, "Object recognition from local scale-invariant features," in *IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[124] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, no. 1-2, pp. 177–196, 2001.

[125] T. G. Dietterich, R. H. Lathrop, and T. Lozano-P?rez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 12, pp. 31–71, 1997.

[126] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artificial Intelligence*, vol. 201, no. 0, pp. 81–105, 2013.

[127] O. Maron and T. Lozano-Prez, "A framework for multiple-instance learning," in *Advances in Neural Information Processing Systems*, 1998, pp. 570–576.

[128] Q. Zhang and S. A. Goldman, "EM-DD: An improved multiple-instance learning technique," in *In Advances in Neural Information Processing Systems*, 2001, pp. 1073–1080.

[129] I. T. Stuart Andrews and T. Hofmann, "Support vector machines for multiple-instance learning," in *In Advances in Neural Information Processing Systems*, 2003, pp. 561–568.

[130] X. Xu, "Statistical learning in multiple instance problems," Master thesis, 2003.

[131] R. C. Bunescu and R. J. Mooney, "Multiple instance learning for sparse positive bags," in *International Conference on Machine Learning*, 2007.

[132] J. Foulds, "Learning instance weights in multi-instance learning," Master thesis, 2008.

[133] O. L. Mangasarian and E. W. Wild, "Multiple instance classification viasuccessivelinearprogramming," *Journal of Optimization Theory and Applications*, vol. 137, pp. 555–568, 2008.

[134] J. Wang and J.-D. Zucker, "Solving the multiple-instance problem: A lazy learning approach," in *International Conference on Machine Learning*, 2000.

[135] Z. hua Zhou, Y. yin Sun, and Y. feng Li, "Multi-instance learning by treating instances as non-I.I.D. samples," in *International Conference on Machine Learning*, 2009.

[136] Y. Chen, J. Z. Wang, and D. Geman, "Image categorization by learning and reasoning with regions," *Journal of Machine Learning Research*, vol. 5, pp. 913–939, 2004.

[137] Y. Chen, J. Bi, and J. Z. Wang, "MILES: Multiple-instance learning via embedded instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1931–1947, 2006.

[138] P. Auer and R. Ortner, "A boosting approach to multiple instance learning," in *European Conference on Computer Vision*, 2004, pp. 63–74.

[139] P. Viola, J. C. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *Advances in Neural Information Processing Systems*, 2006, pp. 1419–1426.

[140] T. Grtner, P. A. Flach, A. Kowalczyk, and A. J. Smola, "Multi-instance kernels," in *In Proc. 19th International Conf. on Machine Learning*, 2002, pp. 179–186.

[141] Z. hua Zhou and M. ling Zhang, "Multi-instance multilabel learning with application to scene classification," in *In Advances in Neural Information Processing Systems*, 2007.

[142] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[143] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 4, pp. 18–28, 1998.

[144] N. Cristianini and J. Shawe-Taylor, in *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2001.

[145] S. Abe, in *Support Vector Machines for Pattern Classification*. Springer, 2005.