Biologically inspired visual intelligence for unmanned ground vehicles

Peyman Moghadam

2012

Peyman, M. (2012). Biologically inspired visual intelligence for unmanned ground vehicles.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/48017>

<https://doi.org/10.32657/10356/48017>

Biologically Inspired Visual Intelligence for Unmanned Ground Vehicles

Peyman Moghadam

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2012

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere appreciation to my advisor, Associate Professor Wijerupage Sardha Wijesoma for his inspiring role in this work as well as his support, professional guidance, and valuable suggestions. This study could not be completed without his enthusiastic help. He is not among us anymore, but I am sure he is now rest in peace and love.

I would like to express my deep gratitude to Professor J. A. Starzky for introducing me to the joy of computational cognitive neuroscience. His visionary thinking and unforgettable discussions about human brain research had a great influence on me. A significant amount of the work reported in this thesis was influenced by discussions with him.

Besides, I would like to express my deep gratitude to my advisor, Professor Danwei Wang Head of Department of Control and Instrumentation, School of Electronics and Electrical Engineering (EEE) for all his help and support throughout preparation of this thesis and my special thanks to Nanyang Technological University for their financial and spiritual help.

I thankfully acknowledge Dr. Bharath Kalyan for his valuable comments and constructive criticisms on the draft of this thesis. I also like to thank JunFeng Dong for working with me in various projects. My thanks also go to all my friends, staff members and postgraduates in Intelligent Robotics Lab for their help and friendship during the course of my study.

Furthermore, my special thanks go to my dear, beloved Saba who accompanied me through the ups and downs of this journey and given me the most wonderful time in my life. She has been such a relief in stressful moments and I could not have done this without her.

Last but not least, I wish to thank my parents and my sister, for their endless love, encouragement, support, and care and would like to dedicate this thesis to my dearest parents.

SUMMARY

Human drivers effectively navigate through most outdoor unstructured environments purely based on their visual inputs. The human visual system has a capability to learn and adapt to its surrounding environment robustly despite the complexity possessed by ground cover variations, uncontrolled lighting, weather conditions, and shadows.

However, none of the artificial perceptual systems in the current state of autonomous Unmanned Ground Vehicles (UGVs) have this level of intelligent perception that can continuously cope with the various unexpected situations and perform robustly in dynamic environments. Therefore, understanding the mechanisms underlying the intelligent abilities of a human driver can be the key to design fully autonomous UGVs.

The main philosophy underlying this thesis is to understand how humans use their vision systems (eyes) and processing units (brains) in driving scenarios in order to develop humanlike visual intelligence for UGVs that can navigate autonomously in an unstructured outdoor environment. To achieve this goal two important aspects need to be investigated.

First aspect of this research addresses the case where a UGV is expected to navigate in a familiar outdoor unstructured environment. Based on empirical evidences, it is known that when human drivers become familiar with the road situations, they prefer to look at the far field where the road edges converge (called the vanishing point) to anticipate the upcoming road trajectory and the car steering with maximal lead time. Based on these findings, vanishing point is considered as a salient and consistent feature during most of the open driving behavior tasks regardless of type of the environments. Hence, in this work, a computational model for the vanishing point estimation based on the visual gaze behavior of drivers is developed for UGVs. The proposed model uses Local Dominant Orientation (LDO) patterns such as road edges, ruts and tire tracks left by previous vehicles to vote for the global convergence point of the road in the visual field.

For robust processing of Local Dominant Orientation (LDO), a novel biologically inspired mechanism is proposed utilizing distributed population coding and the opponency mechanism, found in the activities of ensemble of correlated neurons in the primary visual cortex of the mammalian brain. Apart from being biologically plausible, the proposed LDO model significantly outperforms the state of the art LDO methods in terms of both accuracy and robustness in natural and synthetic images.

This LDO model is next employed to efficiently extract the local oriented patterns and a new voting scheme, which comprises of a distance based voting algorithm and an adaptive weighting function, is presented for vanishing point location estimation for UGVs. This model is further examined by comprehensive stochastic analyses and numerical evaluations on natural images taken during the DARPA Grand Challenge.

The second aspect of this research addresses the case where a UGV navigates in an unknown, unseen outdoor environment. It is believed that humans spend more time in exploring their near field while driving along an unknown route compared to a familiar environment. Humans adapt to changing environments concurrently by incorporating the knowledge of near field into far field, using their ability to continuously learn the association between terrain geometry characteristics and appearances in the near field and then, use such information to make predictions about the upcoming road to the end of its visual field.

Accordingly, in this study, a novel and robust learning approach is developed for understanding the traversability of the outdoor unstructured terrains by integrating learning mechanism into the visual perception system. A new Self-Supervised Learning (SSL) framework is proposed to employ stereo vision system to learn the correlation between traversability of nearby terrain regions with their visual appearance attributes. Based on this learned knowledge in the near field, SSL framework utilizes a monocular vision to predict the traversability of the environment in the far field without human intervention. The proposed SSL framework is composed of a robust ground plane and obstacles estimation, confidence-based auto labeling, biologically inspired top-down attention model, and an adaptive online classifier. Finally, a series of extensive qualitative and quantitative experimental results are presented to demonstrate the effectiveness, robustness, and accuracy of the proposed learning framework using the natural datasets logged by vehicles competing in the DARPA-LAGR program.

TABLE OF CONTENTS

Acknowledgement.....	i
Summary	iii
Table of Contents	v
List of Figures	x
List of Tables.....	xviii
Acronyms	xx
1 Introduction.....	1
1.1 Motivations.....	1
1.2 Objectives and Approach	3
1.3 Major Contributions	3
1.4 Thesis Outline	5
2 State of The Art.....	7
2.1 An Overview of Unmanned Ground Vehicles	7
2.1.1 The Demo Series Program	10
2.1.2 The Grand Challenge	11

2.1.3	The DARPA-LAGR.....	12
2.2	Machine Vision and Robotics	13
2.2.1	Monocular Vision	14
2.2.2	Stereo Vision.....	15
2.3	Machine Learning and Robotics	16
2.3.1	Supervised Learning	17
2.3.2	Active Learning	17
2.3.3	Reinforcement Learning	18
2.3.4	Unsupervised Learning	18
2.3.5	Self-Supervised Learning, Online Learning, Near-to-Far Learning.....	19
3 Local Dominant Orientation Estimation Using Population Coding		21
3.1	Introduction	21
3.2	Related Work.....	23
3.2.1	Gradient-based Method.....	23
3.2.2	Steerable Filter Banks	25
3.3	Orientation Selectivity in Primary Visual Cortex	27
3.3.1	Computational Model of V1 Simple Cells.....	28
3.3.2	Computational Model of V1 Complex Cells	30
3.3.3	Neural Coding of Orientation	30
3.4	Models of Orientation Opponency Mechanism	33
3.4.1	Multiscale Estimation	35

3.5	Experimental Evaluation	38
3.5.1	Experimental Results on Synthetic Images.....	38
3.5.2	Experimental Results on Natural Images.....	55
3.6	Chapter Summary.....	58
4 Vanishing Point Estimation for Unmanned Ground Vehicles		60
4.1	Introduction	60
4.2	Related Work.....	65
4.3	Vanishing Point Voting.....	68
4.3.1	Weighting Method	69
4.3.2	Voting Scheme.....	70
4.3.3	Skyline Detection.....	74
4.4	Results and Analyses.....	76
4.4.1	Dataset.....	76
4.4.2	Manual Ground-truth Vanishing Point Detection.....	77
4.4.3	Performance Metric	78
4.4.4	Experimental Parameters	79
4.4.5	Experimental Results	81
4.4.6	Comparison with State of The Art and Representative Outputs.....	83
4.5	Chapter Summary.....	89
5 Self-Supervised Learning Applied to Unmanned Ground Vehicles		91
.....		91
5.1	Introduction	91

5.2	Related Work.....	93
5.2.1	Vision-Based Obstacles Detection and Ground Plane Estimation	94
5.2.2	Feature Extraction	100
5.2.3	Classifier Training	105
5.3	Proposed Self-Supervised Learning Framework.....	111
5.3.1	Image Pre-Processing.....	115
5.3.2	Robust Ground Plane Estimation.....	116
5.3.3	Auto Labeling	120
5.3.4	Confidence-Based Auto Labeling.....	122
5.3.5	Optimal Feature Extraction.....	127
5.3.6	Computational Model of Top-Down Attention	129
5.3.7	Online Learning	135
5.3.8	Cost Sensitive SVMs	138
5.3.9	Novelty Detection	140
5.4	Results and Analyses.....	142
5.4.1	Datasets	142
5.4.2	Evaluation Measurements.....	145
5.4.3	Experimental results.....	148
5.5	Chapter Summary.....	182
6	Conclusion and Recommendation	184
6.1	Summary	185
6.2	Recommendations for Further Research	187

A LDO Estimation on Natural Images	189
B Manual VP Detector	194
C Vanishing Point Estimation.....	196
D Author's Related Publications	197
D.1 Journal Papers	197
D.2 Conference Papers.....	197
Bibliography.....	199

LIST OF FIGURES

Figure 2.1 Examples of tele-operated UGVs	8
Figure 2.2 Examples of Autonomous UGVs.	9
Figure 2.3 (a) On Oct 8, 2005, Stanley was the first robot to complete the DARPA Grand Challenge. (b) The robot is being honored by DARPA Director Dr. Tony Tether [20]....	11
Figure 2.4 LAGR platform.	13
Figure 3.1 An overview of classical hierarchical visual pathway in the human brain.	28
Figure 3.2 Examples of Gabor filter kernels (a) symmetric and (b) anti-symmetric.....	30
Figure 3.3 A synthetic image with the ground truth angle of 30 degrees (left), a synthetic image with the ground truth angle of 120 degrees (right).	38
Figure 3.4 Four Gabor energy orientation tuning curves for all the angles of synthetic images (0 to 3.14 radians).	40
Figure 3.5 Four Gabor energy responses of a synthetic image containing bars with angle of 30 degrees.	41
Figure 3.6 The angular estimation error from 0-3.14 radians of the proposed method based on the true orientations.....	42
Figure 3.7 The angular estimation error of the proposed method after cubic polynomial fit from 0-3.14 radians.	43

Figure 3.8 The angular estimation error of two-stage LDO estimation method from 0-3.14 radians.	44
Figure 3.9 An example of a synthetic image with an angle of 150 degrees (left), noisy synthetic image with an angle of 150 degrees (right).	45
Figure 3.10 The results of opponent LDO and classical LDO as a function of noise variance.	46
Figure 3.11 The effect of increasing the number of Gabor filters in the population.	47
Figure 3.12 Examples of synthetic input images with three different contrast levels (10, 50, 80 percent of stimulus full intensity).	48
Figure 3.13 Average and standard deviation orientation estimation errors of the opponent LDO model as a function of contrast level.	48
Figure 3.14 Real part Gabor filter kernels	50
Figure 3.15 Multiscale LDO methods vs. single-scale LDO as a function of noise variance.	51
Figure 3.16 Multiscale LDO methods vs. single-scale LDO as a function of bar thickness.	52
Figure 3.17 Sinusoidal grating image ($\pi/6$ radians), noisy image with noise variance of 0.6.	54
Figure 3.18 Average estimation error as a function of noise variance across different methods.	55
Figure 3.19 Sample images from Brodatz textures.	56
Figure 3.20 Sample images from OuTex textures.	57

Figure 3.21 LDO estimations are projected back to the sample image from Brodatz textures.....	58
Figure 4.1 Drivers' eye fixation patterns on straight roads, Adapted from [87].	62
Figure 4.2 Drivers' eye fixation patterns on 13 degrees curved roads, Adapted from [87].	63
Figure 4.3 Gaze distribution in the driving phase as a function of speed a) 30 km/hr, b) 110 km/hr c) 190 km/hr, Adapted from [90].	64
Figure 4.4 (a) Vanishing point in the straight road (b) Vanishing point in the curved road	66
Figure 4.5 The upward voting, the candidate vanishing point vp_{max} is receiving supports from possible texture orientations voters.	69
Figure 4.6 (a) Input image, (b)-(e) The Gabor energy responses.	70
Figure 4.7 The Bi-directional voting, the candidate vanishing point vp_{max} is receiving supports from possible texture orientations voters from top and bottom of its current location in the image.	71
Figure 4.8 An example where Bi-directional voting with $\sin\theta_{popp}$ weighting method performs well (a) Input image with black dot showing the ground truth and white dot shows the estimated vanishing point, (b) Accumulator space.	72
Figure 4.9 An example where Bi-directional voting with $\sin\theta_{popp}$ weighting method fails (a) Input image with black dot showing the ground truth and white dot shows the estimated vanishing point, (b) Accumulator space.	72

Figure 4.10 An example where distance-based voting with $\sin\theta\text{popp}$ weighting method performs well (a) Input image with black dot showing the ground truth and white dot shows the estimated vanishing point, (b) Accumulator space.	73
Figure 4.11 Examples of skyline detection—skylines are drawn as a straight line.....	76
Figure 4.12 Representative images from our dataset.....	77
Figure 4.13 Representative images with Gaussian fit of a set of human responses marked with ellipses.	78
Figure 4.14 Comparison of vanishing point detection methods in a 11-bin histogram.	86
Figure 4.15 Comparison of vanishing point estimation methods. (a) Edge-based method: first row is the estimated vanishing point, second row is the Canny edge map. (b) Rasmussen method [105]: first row is the estimated vanishing point, second row is the accumulator space. (c) Kong et al. method [104]: first row is the estimated vanishing point, second row is the accumulator space (d) the best proposed method: first row is the estimated vanishing point (red line showing the estimated skyline), second row is the accumulator space.	87
Figure 4.16 Examples of vanishing point detection for structured road. First row and third row: outputs of the proposed algorithm (black dot is the ground truth and white dot is the estimated vanishing point location and red line is the estimated skyline); second row and fourth row: accumulator space.....	88
Figure 4.17 Examples of failed vanishing point detection from [104]. First row: output of the proposed algorithm (black dot is the ground truth and white dot is the estimated vanishing point location and red line is the estimated skyline); second row: accumulator	

space of the proposed algorithm; third row: output of method in [104] (red dot is the estimated vanishing point); fourth row: accumulator space of method in [104].	89
Figure 5.1 Block diagram of the algorithm structure for baseline self-supervised, near-to-far learning.	94
Figure 5.2 MLP Neural Network.	106
Figure 5.3 Training points that are nearest to the separating planes are called support vectors (SV).	108
Figure 5.4 Example of KNN with $K=3$ and $K=5$ (outer circle).	109
Figure 5.5 Block diagram of data flow in the proposed Self-Supervised Learning framework.	114
Figure 5.6 Original RGB image	115
Figure 5.7 Colored stereo disparity map	116
Figure 5.8 A plane is fitted to disparity map	117
Figure 5.9 An example of an input image which obstacles occupy most of the near field of the vehicle.	119
Figure 5.10 Ground plane and obstacles overlaid on RGB image in the near field.	121
Figure 5.11 Overview of the proposed top-down attention model.	135
Figure 5.12 Only 7% of near field occupies with positive examples.	137
Figure 5.13 Near field training data contains only negative examples (green colored projected on original RGB image).	140
Figure 5.14 Representative images from six datasets.	144
Figure 5.15 (a) Example RGB images (b) Human labeling RGB images overlaid.	145

Figure 5.16 Plots by dataset of ground plane and obstacle estimation accuracy (mean and std) in the near field of the vehicle (less than 7 meters). The results verify the existence of noise in the ground plane and obstacle estimation provided by the stereo supervision. .	151
Figure 5.17 Representative output of the ground plane and obstacle estimation algorithm in the near field of the vehicle (less than 7 meters), (DS1A Frame 30).	152
Figure 5.18 Representative output of the ground plane and obstacle estimation algorithm in the near field of the vehicle (less than 7 meters), (DS1B Frame 80).....	153
Figure 5.19 Representative output of the ground plane and obstacle estimation algorithm in the near field of the vehicle (less than 7 meters), (DS2A Frame 39).	154
Figure 5.20 Representative output of the ground plane and obstacle estimation algorithm in the near field of the vehicle (less than 7 meters), (DS2B Frame 71).....	155
Figure 5.21 Representative output of the ground plane and obstacle estimation algorithm in the near field of the vehicle (less than 7 meters), (DS3A Frame 17).	156
Figure 5.22 Representative output of the ground plane and obstacle estimation algorithm in the near field of the vehicle (less than 7 meters), (DS3B Frame 49).....	157
Figure 5.23 Representative output of the Basic auto labeling method and the proposed confidence-based auto labeling (elimination method), (Frame 40 DS1B). The mislabeled regions belong to footlines of hay bales are removed from the training labeled data in our proposed confidence based auto-labeling method.	160
Figure 5.24 Plots by dataset of SSL algorithm with three different auto labeling algorithms. The results depict that the proposed confidence based auto labeling methods outperform the baseline (auto labeling method with no error handling).	161

Figure 5.25 The overall mean results for the SSL with three auto labeling algorithms across all datasets.	162
Figure 5.26 Representative outputs of the SSL using three different auto labeling methods (frame 49 DS1B).	163
Figure 5.27 The hand-labeled ground truth image provides supervisory labels for training data (left image), output of SSL using manual labeling (right image) (Frame 74 DS1B).	165
Figure 5.28 The stereo supervision provides traversability labels for training data (left image), output of SSL using Auto labeling (right image) (Frame 74 DS1B).	165
Figure 5.29 Plots by dataset of manual labeling vs. auto labeling. G-means values are reported on a frame-by-frame basis.	166
Figure 5.30 The overall mean number of Support Vectors (SVs) utilized in SSL method with or without Top-Down (TD) attention across all datasets.	169
Figure 5.31 Positive class labels ratio frame-by-frame across all datasets.	171
Figure 5.32 An example of high unbalanced level in the training data (Frame 12 DS2A).	172
Figure 5.33 An example of near field training data that contains only negative examples (Frame 1 DS2A).	173
Figure 5.34 Proposed SSL Framework versus baseline SSL Method frame-by-frame across all datasets.	175
Figure 5.35 Representative outputs of the baseline SSL Method and the proposed SSL Framework, (Frame 50 DS1A).	177

Figure 5.36 Representative outputs of the baseline SSL Method and the proposed SSL Framework, (Frame 50 DS1B).	178
Figure 5.37 Representative outputs of the baseline SSL Method and the proposed SSL Framework, (Frame 50 DS2A).	179
Figure 5.38 Representative outputs of the baseline SSL Method and the proposed SSL Framework, (Frame 50 DS2B).	180
Figure 5.39 Representative outputs of the baseline SSL Method and the proposed SSL Framework, (Frame 50 DS3A).	181
Figure 5.40 Representative outputs of the baseline SSL Method and the proposed SSL Framework, (Frame 50 DS3B).	182
Figure A.1 LDO estimations are projected back to the sample image from Brodatz textures.....	191
Figure A.2 LDO estimations are projected back to the sample image from OuTex textures.....	193
Figure B.1 Screenshot of the GUI for Manual VP detector.....	195
Figure B.2 The output of the Manual VP detector.....	195
Figure C.1 The outputs of the vanishing point estimation algorithm.....	196

LIST OF TABLES

Table 3.1 Fixed parameters used in the model	39
Table 3.2 Fixed parameters used in the multiscale model	49
Table 3.3 Average orientation estimation errors (radians)	57
Table 4.1 Two main groups of experimental parameters in this study	79
Table 4.2 Overall algorithms performance (NormDist) of all methods.....	82
Table 4.3 ANOVA results.....	82
Table 4.4 The accuracy comparison	84
Table 5.1 The ground plane and obstacle estimation modules from LAGR teams.	99
Table 5.2 Appearance features used by LAGR teams	102
Table 5.3 Learning method used by LAGR teams	111
Table 5.4 Confusion matrix	146
Table 5.5 Ground plane and obstacle detection accuracy (G-means) in each dataset.	150
Table 5.6 Mean and standard deviation percentage of noisy labels in each dataset.	150
Table 5.7 The performance (G-means) of the Self-Supervised Learning (SSL) with three different auto labeling algorithms in each dataset.	159
Table 5.8 Numerical results for Self-Supervised Learning (SSL) with Auto Labeling vs. Manual Labeling in each dataset (G-means).	164

Table 5.9 Numerical results for the SSL with Navalpakkam and Itti [138] Top-Down model vs. the proposed Top-Down model in each dataset (G-means).	167
Table 5.10 Overall percentage of positive class labels across all datasets.	170
Table 5.11 Numerical results for proposed Self-Supervised Learning (SSL) framework versus baseline SSL method in each dataset (G-means).....	174

ACRONYMS

ACRONYMS	Description
ANS	Autonomous Navigation System
DARPA	Defense Advanced Research Projects Agency
GMM	Gaussian Mixture Models
HVS	Human Visual System
HSV	Hue, Saturation, Value
KNN	K-Nearest Neighbors
LAGR	Learning Applied to Ground Robots
LDO	Local Dominant Orientation
MOG	Mixture of Gaussians
MLE	Maximum Likelihood Estimation
RANSAC	RANdom SAmple Consensus
RGB	Red, Green, Blue
SSL	Self-Supervised Learning
SVMs	Support Vector Machines
UGVs	Unmanned Ground Vehicles

Chapter 1

INTRODUCTION

1.1 Motivations

“The scientist is not a person who gives the right answers, he's one who asks the right questions.”

~Claude Lévi-Strauss, 1964

The past three decades of mobile robotics research showcases tremendous achievements in the field of autonomous Unmanned Ground Vehicles (UGVs). These autonomous UGVs have been extensively deployed in hazardous environments such as disaster struck areas, military conflict zones to significantly reduce the number of injuries and loss of human life. Despite this, there are still many open research issues and challenges which need to be addressed to operate these UGVs with minimal human intervention particularly in outdoor unstructured environments.

UGVs generally are equipped with an Autonomous Navigation System (ANS), which consists of a series of fundamental components to estimate the vehicle's position and orientation, perceive its environment, and plan a path towards a goal location while avoiding obstacles. The focus of this thesis is on the perception component and more

specifically on the essential part of road detection. Road detection system should be capable of discriminating between the traversable paths (roads) and non-traversable path (off roads). Furthermore, the system should predict the direction of the road with respect to the vehicle. It is important to note that the direction of the road is a necessary prerequisite for effective vehicle path planning.

Like human drivers, current road detection systems typically use visual information to sense and detect road features. In structured, urban environments (e.g. well constructed, paved and maintained roads) a vision-based road detection system usually searches for consistent features of the road such as painted lane markings, pavement curbs, or distinct appearance characteristics of road regions. However, in unstructured environments there are no specific roadway features like lane markings or well-paved roads surfaces that can separate road from its surroundings. Further, there are various types of terrain covers (soil, grass, rock, etc) with complex geometric characteristics, uncontrolled lighting, weather conditions, and shadows, which make the distinction of road region from surroundings extremely difficult. Thus, conventional vision-based autonomous navigation systems of unmanned ground vehicles need be revised for unknown outdoor unstructured environments.

Although, numerous efforts have been made in this field, despite all these accomplishments, we are still a long way from developing a fully autonomous vision-based ground vehicle system for unknown unstructured environments. Most of the current machine intelligent systems have trouble understanding the environments, as well as not being robust enough to handle unexpected situations.

In contrast, humans have inherently a strong ability to handle a wide range of complex visual tasks such as navigation in outdoor environments. Human drivers can locate pathways, identify and avoid potential obstacles, and naturally plan a path towards a goal in an unknown scene solely based on visual information. It is clear that understanding such visually guided behaviors plays a pivotal role in the development of artificial intelligent perceptual systems that currently exists only in the biological vision systems.

Therefore, the main inspiration and motivation of this study is to exploit human visual processing mechanisms to develop visual intelligence for unmanned ground vehicles.

1.2 Objectives and Approach

The objective of this thesis is to mimic the visually guided behavior of human drivers in outdoor unstructured environments and build computational models to develop artificial visual intelligence for Unmanned Ground Vehicles (UGVs) navigation. A key question is therefore how such human-like behavior mechanisms can be emulated by machine intelligence. This thesis attempts to address this question with a combination of computational modeling, machine vision and machine learning based on the current understanding of Human Visual System (HSV). To achieve this goal, the computational principles behind such highly complex and robust visual processing methodologies are investigated and subsequently several models of human visual intelligence for autonomous UGVs are presented. This would hopefully bring a new perspective in the robotics research community by incorporating knowledge of biological vision into autonomous unmanned ground vehicles.

1.3 Major Contributions

Three fundamental contributions of this thesis are as follows:

- Development of a novel biologically inspired mechanism for Local Dominant Orientation (LDO) estimation. The proposed model is inspired by the orientation selectivity characteristic of the ensemble of neurons at earlier stages of primates' visual processing systems. 2D Gabor filter functions are used to model the orientation selective receptive field of individual simple cells and complex cells in primary visual cortex (V1) of the mammalian brain. Next, correlation between the responses of individual Gabor filters in the ensemble is modeled by introducing the opponency mechanism to the distributed population coding in

accordance with the activities of the population of neurons in V1. The proposed model is further improved by a hierarchical structure to provide better estimation precision. In addition, the model is extended by introducing two new multiscale filtering methods to take into consideration of multiple scales in an input stimulus. Finally, the noise robustness and accuracy of proposed model is assessed with respect to state of the art LDO estimation methods on both synthetic and natural datasets. The proposed model is general and can replace the conventional gradient-based methods or oriented filter banks applied to varieties of applications such as fingerprint identification, defect detection, and industrial inspection.

- A new methodology for road direction detection for autonomous UGVs based on computational modeling of visual gaze behaviors of drivers in open driving scenarios. The algorithm employs the proposed LDO estimation model to detect local oriented patterns robustly and then make them vote for the global convergence point of the road (called the vanishing point) in outdoor unstructured environments. This model utilizes only monocular vision information similar to human perception to detect road directions with respect to the vehicle. The fundamental features of the proposed method are an adaptive weighting function that inhibits or excites detected LDOs based on their orientations and a distance based voting scheme for efficient estimation of the vanishing point location. Robustness and speed are further enhanced through exclusion of sky pixels by the method of skyline detection in the processing. This model is further examined by comprehensive stochastic analyses and numerical evaluations on natural images taken during the DARPA Grand Challenge. The competencies of the model are also evaluated qualitatively using several challenging representative images.
- Development of a new, robust Self-Supervised Learning (SSL) framework for autonomous UGVs to understand the traversability of the terrain, based on the stereo vision systems. This is inspired based on human ability to navigate in a new, unknown environment by integrating learning mechanism into perception system. The proposed framework can adapt to changing environment concurrently

based on its own experiences. It identifies the appearance characteristics of nearby traversable and non-traversable terrains and then based on this learned knowledge tries to predict the traversability of far field terrains. The new SSL framework is comprised of several contributions such as robust ground plane and obstacles estimation, confidence-based auto labeling, top-down attention modeling, and an adaptive online learning method that make the framework more robust and accurate compared to baseline algorithms. Finally, a series of extensive qualitative and quantitative experimental results are presented to demonstrate the effectiveness, robustness, and accuracy of the proposed learning framework using the natural datasets logged by vehicles competing in the DARPA-LAGR program.

1.4 Thesis Outline

The rest of the thesis is organized in the following manner. Chapter 2 presents a brief review of state of the art in Unmanned Ground Vehicles research. The major focus of this chapter is on the literature that utilizes vision systems as the primary way of perception for Unmanned Ground Vehicles that operate in outdoor, unstructured environments. In Chapter 3, a biologically inspired mechanism for local dominant orientation estimation is described. First, the state of the art computer vision local dominant orientation estimation techniques are introduced briefly. Then, a proper background of the orientation selectivity characteristics in primary visual cortex of primates' brain is given. Next, the novel model for LDO estimation, which combines the population coding with the opponency mechanism, found in the activities of neurons in primary visual cortex is presented. Finally, the proposed model is evaluated with regards to different criteria in a series of synthetic and natural datasets. This LDO estimation model is employed as a basis for the next chapter. In Chapter 4, first the psychological evidences regarding a driver gaze behaviors is explained. Next, a new computational model for visual gaze behaviors of drivers in open driving scenarios using only monocular vision system is proposed based on the LDO model presented in Chapter 3. Further, the results of the proposed model are compared with the state of the art methods. In Chapter 5, a new Self-Supervised Learning

(SSL) framework is described which extends and enhances the current existing methods in several ways. First, each step of state of the art of self-supervised, near-to-far learning techniques is reviewed in detail. Then, the SSL framework based on human ability to navigate in a new, unknown environment is explained. The proposed SSL framework comprises of several novel sections that provides better accuracy and robustness compared to existing the state of the art methods. Finally, series of quantitative and qualitative experiments are conducted to demonstrate the effectiveness of the proposed SSL framework. Chapter 6 summarizes and concludes the thesis and provides some possible directions and recommendations for further research.

Chapter 2

STATE OF THE ART

In this chapter, a brief overview of the current state of the art of Unmanned Ground Vehicles (UGVs) is given. The major focus of this chapter is on the vision-based autonomous UGVs in outdoor unstructured environments. The rest of this chapter is organized as follows. In section 2.1, an overview of existing Unmanned Ground Vehicles (UGVs) is given followed by a review of recent research projects for developing autonomous UGVs in outdoor unstructured environments. Finally, section 2.2 and 2.3 briefly summarize machine vision and machine learning techniques applied to Unmanned Ground Vehicles preparing a proper background for Chapter 4 and Chapter 5.

2.1 An Overview of Unmanned Ground Vehicles

Over the last few decades, the demand for developing Unmanned Ground Vehicles (UGVs) has been strongly increased in either structured, urban environments or unstructured conditions. Existing UGVs can be categorized into two general classes: tele-operated, and autonomous.

Tele-operated UGVs are currently the most widely used vehicles serving in both civilian emergency situations and military operations applications. A tele-operated UGV is

unmanned and controlled directly by a human operator from a distance based on onboard sensory information. Figure 2.1 illustrates some of the example of tele-operated UGVs. The main limitation of these UGVs is the need for individual human expert operators for individual tele-operated UGVs to attend to streaming videos from onboard cameras and control all the actions of these vehicles.

(a) iRobot PackBot [1]

(b) Frontline Robotics Teleoperated UGV [2]

Figure 2.1 Examples of tele-operated UGVs

In order to address this limitation, many advanced UGVs now have some level of autonomy. An autonomous UGV generally is equipped with an Autonomous Navigation System (ANS), which consists of a series of fundamental components to estimate the vehicle's position and orientation like wheel encoders, IMU (Inertial Measurement Unit), and/or INS (Inertial Navigation System), another suite to provide good perception of its local environment like monocular color cameras, range sensors such as LIDAR (Light Detection and Ranging) or stereo vision systems, radar, and finally path planning component to plan a path towards a goal location while avoiding obstacles without human intervention. The focus of this thesis is on this type of UGVs. Figure 2.2 shows VisLab's autonomous car and Google's autonomous car project, two recent driverless car projects that have driven autonomously thousands of kilometers in structured, urban environments.

(a) VisLab's autonomous car [3]

(b) Google driverless car [4]

Figure 2.2 Examples of Autonomous UGVs.

Autonomous Navigation Systems are generally associated with the road formation. There are two types of roads defined in the robotics community: structured and unstructured. The structured road is usually associated with consistent and specific features such as painted lane markings, well constructed road boundaries (e.g. curbs), and well paved and maintained man-made road surfaces. An autonomous navigation system typically uses a set of sensors like a stereo vision system [5], LIDAR [6], radar [7], monocular camera [8] or a combination of them [9] to detect painted lane markings or road boundaries (e.g. curbs) as the reliable features of the road regions. Some of the pioneering navigation systems developed for outdoor structured environment are Navlab [8], RALPH (Rapidly adapting lateral position handler) [10], VITS (Vision Task Sequencer) [11], and GOLD (Generic obstacle and lane detection system) [5].

However, in unstructured roads, no permanent assumptions can be made about the appearance or structure of the road. There are no specific roadway features like lane markings or well-paved roads surfaces that can separate road from its surroundings. Further, there are many various types of terrain covers (soil, grass, rock, etc) with complex geometric characteristics, uncontrolled lighting, weather conditions and shadows, which make distinguishing road region from surroundings extremely difficult. Due to these key challenges, less research groups have tackled the problem of

autonomous navigation system in unstructured environments. SCARF (Supervised Classification Applied to Road Following) [12], UNSCARF (UNSupervised Classification Applied to Road Following) [13], and the ALVIN (Autonomous Land Vehicle In a Neural Network) [14] are among those few but significant initiatives that have addressed developing autonomous navigation systems for unstructured environments.

To draw more attention to this field of research, the Defense Advanced Research Projects Agency (DARPA) has promoted the development of autonomous navigation systems in the unstructured or ill-structured environments in a series of research projects such as the DEMO I, II and III [15], the DARPA Grand Challenge [16-17] and the DARPA Learning Applied to Ground Robots (LAGR) [18]. Next, each of these projects which have had a tremendous impact on autonomous navigation research and development are concisely described and reviewed.

2.1.1 *The Demo Series Program*

In 1992, the Defense Advanced Research Projects Agency (DARPA) and Joint Robotics Program (JRP) jointly sponsored a series of programs to develop technology for the UGVs. The DEMO III program was specially designed to address the operational limitations through previous programs [19]. The US Army Demo III program follows about ten years of earlier work sponsored by its incarnation as Demo I and Demo II [15]. This program had primary focus on the development of autonomous mobility technology for a small ground vehicle over cross-country, rugged terrain operating during the day or night with an emphasis on perception, navigation, task planning, and intelligent system architecture. The Demo III *Experimental Unmanned Vehicle* (XUV) employs a wide variety of sensors for mobility. These include day/night stereo vision, a scanning laser rangefinder, and two radar systems, a 4 GHz radar to penetrate vegetation, and a 77 GHz radar for imaging obstacles at longer ranges [15]. The XUV had to be able to drive autonomously at speed of up to 40 mph on roads, 20 mph off road by day and 10 mph off

road by night or in foul weather conditions while avoiding both negative and positive obstacles.

2.1.2 *The Grand Challenge*

The Grand Challenge was the first long distance competition for robot cars in the world till now. The DARPA launched the Grand Challenge in 2004 to speed up the development of autonomous vehicles to move quickly over rough terrain is an open challenge [16-17]. The goal of the Challenge was to develop an autonomous robot capable of traversing unstructured terrains. The first competition, which carried a prize of \$1M, took place on March 13, 2004. It required robots to navigate a 142-mile long course through the Mojave Desert in no more than 10 h. 107 teams registered and 15 raced, yet none of the participating robots navigated more than 5% of the entire course. The challenge was repeated on October 8, 2005, with an increased prize of \$2M. This time, 195 teams registered and 23 raced. Of those, five teams finished. Stanford's robot "Stanley" finished the course ahead of all other vehicles in 6 h, 53 min, and 58 s, and was declared the winner of the DARPA Grand Challenge; see Figure 2.3 [20].

Figure 2.3 (a) On Oct 8, 2005, Stanley was the first robot to complete the DARPA Grand Challenge. (b) The robot is being honored by DARPA Director Dr. Tony Tether [20].

Although the DARPA Grand Challenge was a great victory for developing autonomous UGVs in unstructured environments, it requires extensive GPS waypoint (few meters apart), where the terrain was almost obstacle free.

2.1.3 *The DARPA-LAGR*

The LAGR - Learning Applied to Ground Robots- was sponsored by DARPA from December 2004 through January 2008. The goal was to develop autonomous vision-based ground navigation in unstructured natural environments using Machine Vision and Machine Learning techniques [18]. Despite the DARPA Grand Challenge where the robot that won the task used very fine-grained GPS waypoints to navigate in a desert road with sparse obstacles, the DARPA LAGR program primarily addresses issues of vision-based navigation in rough, complex terrains with widely spaced waypoints (hundred meters). Unlike the Grand Challenges or Mars rover, LAGR is an attempt to develop adaptive methods for Unmanned Ground Vehicles (UGVs) that can navigate without the aid of active sensors such as radar or scanning lasers. Except for information sensed through a stereoscopic vision system, and a GPS, it has no clue about the outside world at all.

LAGR Phase I completed in June 2006, and Phase II began in August 2006. The project began with eight research teams: Jet Propulsion Laboratory, Applied Perception, Stanford Research Institute, National Institute of Standards and Technology, Net-Scale Technology (with New York University), the University of Pennsylvania, and the Stanford team. There are several university teams that also contributed to this program such as: the University of Colorado, the University of Idaho, the University of Missouri, and the University of Central Florida. All of the teams used a standard ground vehicle platform with identical sensors; see Figure 2.4.

LAGR addressed the shortcomings of current robotic ground vehicle autonomous navigation systems through an emphasis on learned autonomous navigation. It was clear that significant improvements were required in perception, planning, and control to master climbing [18]. Progress was measured quantitatively through a series of competitions

conducted at the LAGR Test Facilities (LTF). These monthly competitions measured the ability of the performer systems to learn from experience (online learning) or ability of the performer systems to learn from examples.

In October 2007 test (which was Test 24 of this program); six teams met the goal to double the performance of the baseline controller using Machine Learning algorithms. This was a significant achievement which demonstrated the ability of Machine Learning to address the unsolved problems in unstructured environments like nearsighted behavior, erroneous classification, and inability to learn from experience.

Figure 2.4 LAGR platform.

2.2 Machine Vision and Robotics

Like human drivers, most of the current autonomous navigation systems typically use visual information to sense their environments. In structured, urban environments (e.g. well constructed, paved and maintained roads) a vision-based system usually searches for

consistent features of the road such as painted lane markings, pavement curbs, or distinct appearance characteristics of road regions. However, in unstructured environments there are no specific roadway features like lane markings or well-paved roads surfaces that can separate road from its surroundings. A recent survey paper also highlights the challenges dominating the vision-based autonomous navigation systems in outdoor unstructured environments [21]. Existing autonomous vision-based navigation systems can be divided into two groups based on the type of vision sensors they utilized. The first one is monocular vision systems, and second one is the stereo vision systems.

2.2.1 *Monocular Vision*

Monocular vision systems typically are used as the primary ways of perception in UGVs. It can provide the visual appearance features of environments with no geometry information.

Many researchers have considered using monocular vision for road detection in UGVs [11, 13]. Majority of monocular road detection methods in the literature are grouped into two main categories: edge and region-based methods. Edge-based methods reduce the road detection to the extraction of road boundaries using gradient operators (e.g. Canny operator). However, these approaches are suitable for structured roads where the lane markings or edge boundaries (i.e. curbs) are distinct features of the road [5, 8] while they have difficulties in unstructured environments [22]. Region-based methods look for distinct appearance characteristics (i.e. color and/or texture) over local neighborhood in order to segment the scene to path and non-path regions [12-13, 23-24]. Blas et al. [23] use a compact texture/color descriptor (also known as texton) in a two-stage k-means clustering algorithm. One drawback of this method is that an appropriate value of k (number of classes) is unknown in advance in unstructured environments. Alon et al. [25] used two independent region-based and edge-based components to segment image to path and non-path regions and the final system output is selected from the component with the highest confidence. The first component uses an Adaboost classifier to train based on the texture of the road and non-road regions. The second component uses geometric

constraints of the road and the cameras parameters (roll and pitch). It assumes the existence of a planar surface bounded by parallel edges immediately in front of the vehicle.

These methods are suitable for cross-country environments where there is at least one unique surface road feature to distinguish from surroundings. They also assume that road surfaces belong to relatively homogenous regions [26]. However, challenging problems such as ground cover variation, uncontrolled lighting, weather conditions and shadows make this segmentation difficult.

2.2.2 *Stereo Vision*

Recently, most of the autonomous navigation systems are facilitated with color stereo vision systems to sense their environments in order to predict the traversability of the terrain (i.e. to discriminate traversable path and non-traversable path regions). Color stereo vision cameras can provide not only 3D perception of terrain geometry but also color and texture information. However, the effective maximum range of typical off the shelf stereo vision systems is limited up to about 10-15 meters due to their short baseline (10-12cm). As a result, they can only provide accurate geometry information from their close range. Beyond this, the accuracy of the estimates degrades as the distance from the camera pair increases. In order to drive at higher speeds while being able to avoid obstacles the path planning system needs information about road directions well beyond the range of a typical stereo vision system. This near-sightedness with stereo vision often causes the vehicles to get caught in cul-de-sacs that could possibly be avoided if the vehicle had access to information or could make inferences about the terrain well beyond the range of the vision system [18].

This problem can be alleviated by associating the geometry of terrain segments close to the vehicle (near field) with their visual appearance attributes (e.g. color and/or texture) and then, using such information to segment terrain and obstacles in the far field of the camera images [27-28]. Navlab CMU [8] is one of the most prominent project on outdoor

environments which uses pixel classification for road following and 3D vision for obstacle detection. Each pixel is assigned two classifiers; one is a class depending on the color distributions and the other is based on texture. The probability of a pixel belonging to a class is computed using Mahalanobis distance. The normalized gradient index for each pixel is computed as texture descriptor based on prior knowledge that a road region area is much smoother than non-road. The final classification is a weighted sum of the two probabilities.

However, finding a global correlation between terrain geometry characteristics and appearances that can be broadly applied is formidable due to the complex variability of appearance with terrain geometry, lighting, shadows, and weather conditions, especially in outdoor environments. Therefore, most hand-designed deterministic and rule-based system has proved to be futile as they are not robust to changing environments due to their inability to adapt to unforeseen situations. To address this problem, one promising approach is to use Machine-Learning (ML) techniques to replace hand-designed deterministic vision-based autonomous navigation systems for UGVs. These methods are explained in section 2.3.

2.3 Machine Learning and Robotics

To understand far field terrain for accurate path planning or fast driving, deterministic approaches are insufficient. Machine-Learning is a promising paradigm in order to replace hand-designed deterministic autonomous navigation systems for UGVs. Such learning techniques may be based on their own experience on-the-fly or based on the experiences during the previous runs (learning from experiences). They can also rely on human expert actions over similar terrain in the past (learning from examples). In this section, the literature review of the related research is organized by its learning strategy.

2.3.1 *Supervised Learning*

Supervised learning is a machine learning technique which is trained offline by a number of training examples which are labeled by human experts. The ALVIN (Autonomous Land Vehicle In a Neural Network) [14] by Pomerleau was the first learning approach to robot navigation. A neural network is trained using image data associated with steering angle for road following by watching a human driver's actions when driving on roads of varying properties. Obtaining sufficient training data for the neural network was identified as a major problem, and the author proposed to artificially alter input output pairs in order to increase the number of training patterns available. Manduchi et al. [29] have successfully implemented learning color distributions of different terrains by training over a large number of images taken under widely different illumination conditions. They used Maximum Likelihood classifier (ML), based on Mixture of Gaussians (MOG) modeling. Since the outdoor environments are highly unconstrained, collecting and hand-labeling of a large amount of data may be difficult, time consuming and impractical in many real world applications and finally this type of learning will be limited to the certain environment types seen during training phase.

2.3.2 *Active Learning*

To make supervised learning practical in outdoor environments, active learning technique is introduced. Active learning reduces the amount of labeled data required for training supervised learning while obtaining as good a classifier performance as possible given the whole input labeled dataset. Dima et al. [30] adapted the active learning for robotics application and demonstrated that active learning techniques can be applied to large amount of unlabeled data in order to select only informative images to present to the human expert for labeling in an autonomous navigation system. They showed by labeling only 5 images selected by their method they got a surprisingly good classification performance comparable to the one used hundreds of labeled images were obtained; however this technique still needs the human supervision in order to label the training data.

2.3.3 *Reinforcement Learning*

There are many situations when we do not know the correct answers that supervised learning requires, but if a robot could learn to solve the problems through trial and error, that would be great progress through fully autonomous navigation systems. Reinforcement Learning (RL) is an approach of ML that achieves the goal using trial-and-error interactions with its environment. In reinforcement learning, the robot interacts with its environment by producing some sort of actions. These actions affect the state of the environment, which in turn provides the machine with some scalar rewards (or punishments). The goal of the machine is to learn to act in a way that maximizes the future rewards it receives (or minimizes the punishments) over its lifetime [31]. Behavior improves based on knowledge of which actions led to rewards and punishments. In this way, both good and bad experiences are a valuable part of the learning process.

Michels et al. [32] implemented a system to navigate at high speeds through unstructured environments using reinforcement learning. First, supervised learning is applied to estimate depth from monocular images. Then RL search is used to select a steering direction as a function of the vision system output. The resulting control policy helps the robot to select a steering direction as a function of the vision system's output. However, the main drawbacks of reinforcement learning techniques are complicated training algorithms and necessity of prior information about the train type.

2.3.4 *Unsupervised Learning*

Unsupervised learning is a class of machine learning techniques that use only unlabeled data. There are no target outputs or rewards from its environments. It is often known as estimating a density of input data that can be used for predicting future inputs. The classical examples of unsupervised learning are clustering (k _means, ART) and dimensionality reduction (PCA, ICA). Howard et al. [33] use unsupervised k _means clustering to segment a frame into four appearance classes ($k=4$) and then use the traversability labels produced by stereo disparity data to infer traversability of the entire

class. The training error, which is an average number of incorrect classified proxies per cluster, is a key measure in their algorithm to allow fast evaluation of k _means clustering whether it should be discarded or used. Despite the advantages of unsupervised learning, it requires the prior knowledge of environments regarding the number of terrain classes and it does not perform well in noisy environments.

2.3.5 *Self-Supervised Learning, Online Learning, Near-to-Far Learning*

The performance of most offline learning methods (supervised classifications) is limited to the specific environments seen during training, thus it cannot adapt to changing environments. Moreover, these methods need an expert to hand label the training data which most of the time is too difficult, time consuming and impractical for large amounts of data. Recently Self-Supervised Learning (SSL) has been introduced which becomes essential for many real-time navigation systems. Self-Supervised Learning is an approach for designing a system that can train on the incoming data stream, adapting to unknown environments without using any hand-labeled training data. The terms online learning or near-to-far learning are often used to refer to Self-Supervised Learning. In detail, Self-Supervised Learning (SSL) consists of two modules: an auto labeling module which, continuously receives a new input data, generates online auto-labeled training examples for training a classifier, and a classifier module in which novel input data are evaluated based on a learned model. Therefore, one full training and classification cycle are completed on every batch of incoming data stream. The near-to-far learning specially refers to an approach of using information found in the near field of a scene to train a classifier and use that learned model to classify the far field of the same scene.

The winner of 2005 DARPA Grand Challenge used a self-supervised, near-to-far learning technique for road detection. Laser is used to scan for flat, drivable surface area near the robot, then road appearance (color histograms) is extracted from this area and passed to computer vision classification algorithm to classify entire image. Although this online classifier can be utilized for autonomous navigation system, it was applied only for speed control system during 2005 DARPA Grand Challenge [20, 34].

In [35] on-line traversability prediction is used to train a ladar-based navigation system to estimate the ground surface that lies under vegetation. The system learns the surface height and terrain features by using an incremental locally weighted learning method. In order to overcome the complexity of the vegetation environments, the system stores a set of features from laser points in the cells from different distances. When the vehicle drives over those cells the true surface height are measured and the stored features associated with this true height are used as an input to train the classifier. Online learner uses the interaction between the robot and its environment to learn and predict the traversability properties of the terrain. This process happens continuously, so the more the vehicle interacts with the environment, the more training data the learning system receives.

Thrun et al. [36] use a combination of self-supervised learning and reverse optical flow technique in the case of the adaptive road following. The self-supervised learning takes information from the near-range local sensors such as physical bumpers and infrared range sensors about objects and terrain types and labels input data without human assistance. When an object is registered in the near field of local sensors, the optical flow procedure is called to trace the view of that object in the current image back to where it appeared first in the field of view of the vehicle in order to extract visual appearance of the object at a greater distance. This information is then used to train the Mixture Of Gaussian classifier. Therefore, the vehicle learns the visual characteristics of objects at different distances for making early navigation decisions.

Recently, different mechanisms for self-supervised, near-to-far learning have been investigated by several research groups participating in the DARPA-LAGR program [18]. Since, these methods are closely related to one of the primary contribution of this work, they are addressed in detail in Chapter 5.

Chapter 3

LOCAL DOMINANT ORIENTATION

ESTIMATION USING POPULATION CODING

3.1 Introduction

Local Dominant Orientation (LDO) estimation has proven to be an essential part of both machine and human visual processing. LDO estimation has been addressed previously in many machine vision applications including texture segmentation, and discrimination [37-38], scene recognition [39], edge detection [40], industrial inspection [41], defect detection [42] and fingerprints identification [43]. The term “Local Dominant Orientation (LDO)” refers to a local directionality characteristic of parallel structure or coherent texture flow in the neighboring pixels of the image. Local Dominant Orientation is calculated for every pixel and unlike other visual features such as color or intensity, it defines as spatial property inside a given neighborhood.

On the other hand, Hubel and Wiesel in 1962 [44] discovered that the majority of neurons in the primary visual cortex (V1) of the mammalian brain have the orientation selectivity

characteristic that indicates the importance of local orientation detection in biological vision systems. The mechanism of orientation selectivity provides the ability to primate's vision systems to detect orientation of local bars and edges at the early stages of visual cortex hierarchy.

Hence, in this chapter, the orientation tuning properties of the ensemble of neurons are adopted to develop a novel biologically inspired mechanism for local dominant orientation estimation. The proposed model is inspired by psychophysical evidences, which reveal that at earlier stages in the primary visual cortex, basic visual feature such as line orientations are encoded using a population of correlated neurons and is implemented on both synthetic and natural datasets to evaluate its robustness and accuracy. The performance of the proposed algorithm is also compared with the state of the art machine vision LDO estimation methods. Apart from being biologically plausible, the results demonstrate several advantages including higher accuracy, more robustness to input noise and scale variations, and contrast invariance.

The rest of this chapter is organized as follows. Section 3.2 outlines the state of the art machine vision Local Dominant Orientation (LDO) detection techniques. Section 3.3 is commenced by discussing the orientation selectivity characteristics in the primary visual cortex (V1) followed by the computational models of simple cells and complex cells found in this area of the brain. Next, encoding and decoding of the visual information in primary visual cortex are described. Then, distributed population coding is elaborated based on psychophysical evidences that at earlier stages in the V1, basic visual features such as line orientations are represented in this way. In section 3.4, the proposed opponent model for LDO estimation is presented, which combines the population coding with the opponency mechanism, found in the activities of ensemble of neurons in primary visual cortex. In section 3.5, experiments are presented that show the effectiveness of the proposed method for local dominant orientation estimation. To this end, a series of experiments on both synthetic and natural datasets is ran. Finally, section 3.6 will draw conclusions.

3.2 Related Work

The existing machine vision Local Dominant Orientation (LDO) detection techniques can be classified into two major categories: gradient-based methods, and filtering based methods. Gradient-based methods estimate orientation as the orthogonal direction to the gradient of gray intensity at each location in the image. On the other hand, filtering based methods, apply a bank of oriented filters of different orientations in all possible directions to an input image to find the direction of local dominant orientation. Filtering can be applied in either spatial domain or frequency domain. A bank of Gabor filters and steerable filters are the most well known filtering based methods. In this section, we will give brief overview of gradient-based methods and steerable filters. Gabor filters will be explained in section 3.3.

3.2.1 *Gradient-based Method*

The most popular LDO estimation approach is to estimate local orientation based on the gradient of gray intensity. Consider an input image I , the local orientation in pixel location (x, y) can be obtained by taking the first derivatives in two orthogonal directions at (x, y) as follows:

$$\nabla I(x, y) = \left[\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right] = [I_x, I_y], \quad (3.1)$$

$$\hat{\theta}_{xy} = \tan^{-1} \frac{I_y}{I_x}, \quad (3.2)$$

where ∇ is a gradient operator and $\hat{\theta}_{xy}$ is the estimated dominant orientation at pixel location (x, y) . However, the gradient based methods are very sensitive to image noise since they are computed at pixel levels. In order to overcome this limitation, several enhanced gradient-based algorithms have been presented. The most common approach is to use smoothing and averaging techniques to reduce the noise effects [45-48]. Consider the smoothing filter is denoted as F . The gradient image can be computed after convolution of the smoothing filter with input image I . This smoothing step reduces the

high frequency noise prior to the gradient step. After smoothing, the gradient operator is applied to the resulting smoothed image $\hat{I}$.

$$\hat{I} = F \otimes I \quad (3.3)$$

$$\nabla \hat{I}(x, y) = \left[\frac{\partial \hat{I}(x, y)}{\partial x}, \frac{\partial \hat{I}(x, y)}{\partial y} \right] = [\hat{I}_x, \hat{I}_y], \quad (3.4)$$

where $\otimes$ represents the spatial convolution operator. Since the convolution is linear operation, an alternative approach is to first apply gradient operator to the smoothing filter and then convolve the resulting filter to the image. The main advantage of this method is that the derivatives of the smoothing filters can be calculated in advance so only the convolution needs to be performed on each image.

$$\frac{\partial F}{\partial x} = F_x, \quad \frac{\partial F}{\partial y} = F_y \quad (3.5)$$

$$\hat{I}_x = F_x \otimes I, \quad \hat{I}_y = F_y \otimes I. \quad (3.6)$$

Then, the local dominant orientation $\hat{\theta}_{xy}$ is determined by equation (3.2). Once the local dominant orientations are estimated an averaging techniques is applied to combine them over a small neighborhood to get a better estimation of orientation in each local window.

In addition, there is a body of research based on more sophisticated gradient-based methods to find the dominant orientation. Bigun et al. [49] showed that eigenvector analysis of the covariance matrix of the gradient vectors in a local neighborhood represents the dominant orientation in each block. Feng and Milanfar [50] extended this idea using Principal Component Analysis (PCA). In their approach, initially the gradient vectors for each pixel are calculated. Then, these gradient vectors are grouped within a local neighborhood block and Singular Value Decomposition (SVD) is used to find the local dominant orientation at each block. Although, these methods showed better robustness against noise they are very computationally expensive.

3.2.2 Steerable Filter Banks

Filtering based methods apply a set of oriented filters of different orientations to an input image to find the direction of local dominant orientation. In order to achieve a precise angular resolution, one approach is to design a bank of filters, which are oriented in all possible directions and study the outputs of the filters as a function of their orientations. However, designing a bank of differently rotated filters and applying them to the image one by one is computationally expensive.

Hence, a more efficient approach was first proposed by Freeman and Adelson [51] called “steerable filters”. Steerable filters refer to the class of filters in which each arbitrary orientation can be formed by a linear combination of a fixed set of basis oriented filters. In this section, we give a brief overview of the concept of steerable filters, and its drawback in estimating local dominant orientation.

A function $f(x, y)$ is steerable if it can be written at an arbitrary angle θ , as a linear combination of rotated copies of itself $f^{\theta_j}(x, y)$.

$$f^{\theta}(x, y) = \sum_{j=1}^M K_j(\theta) f^{\theta_j}(x, y), \quad (3.7)$$

where $K_j(\theta)$ is the interpolation function, M is the number of bases required. Therefore, one needs to know what kind of functions are steerable, how many bases are necessary and how to interpolate between them. Freeman and Adelson [51] argued that generally all functions that are angularly bandlimited in frequency domain are steerable, given enough basis filters. Thus, they used circularly symmetric Gaussian function as the steerable function $G(x, y) = e^{-(x^2+y^2)}$ and used the second directional derivatives of a Gaussian (G_2) as basis functions. The reason is that the second directional derivative of a Gaussian can be expressed in x and y direction separately, which can decrease the computational cost, and require a small number of basis filters to form an oriented filter in an arbitrary orientation. For example, the second directional derivative of a Gaussian function in zero direction can be written as:

$$f^0(x, y) = G_2^0 = (4x^2 - 2)e^{-(x^2+y^2)}. \quad (3.8)$$

They used three basis functions using $\theta_1 = 0^\circ, \theta_2 = 60^\circ, \theta_3 = 120^\circ$. The corresponding interpolation functions are given by:

$$K_j(\theta) = \frac{1}{3} [1 + 2\cos(2(\theta - \theta_j))]. \quad (3.9)$$

Therefore, one can find the filter for any arbitrary orientation θ as follows:

$$G_2^\theta = K_1(\theta)G_2^{0^\circ} + K_2(\theta)G_2^{60^\circ} + K_3(\theta)G_2^{120^\circ} \quad (3.10)$$

Given the I is an input image, the response of input image to arbitrary orientation θ is computed from the convolution of input image with the oriented filter G_2^θ by:

$$Y_\theta(x, y) = I(x, y) \otimes G_2^\theta. \quad (3.11)$$

However, the response $Y_\theta(x, y)$ not only depends on the orientation but also on the local phase. In order to obtain responses controlled only by the local orientations, the steerable quadrature pair is usually used. The quadrature pair summation is a sum of squared of two filters with same frequency responses but with a phase difference of $\frac{\pi}{2}$ radians (i.e. are Hilbert transforms of each other) [51]. Given the second derivative of a Gaussian G_2 , and its Hilbert transform as its bandpass filters H_2 , one can measure the local oriented energy with quadrature pairs of steerable filters as follows:

$$E(\theta) = [G_2^\theta]^2 + [H_2^\theta]^2, \quad (3.12)$$

where H_2 is the 3rd order, odd parity polynomial approximation of the Hilbert transform of G_2 , which is steerable by four basis functions. The local dominant orientation θ_d will be the angle of maximum response of local oriented energies applied to the input image.

$$\theta_d(x, y) = \operatorname{argmax}_\theta (I(x, y) \otimes E(\theta)), \quad \theta \in \{1, \dots, \pi\} \quad (3.13)$$

In order to find the local dominant orientation of each pixel location in an input image one requires to convolve the input image to a set of oriented energy filters spanning all orientations. However, since the convolution is a linear operation, it is more efficient to

first pre calculate the image filtered with each of basis functions (G_2, H_2) and then calculate the response $Y_\theta(x, y)$ for each arbitrary orientation θ by combining pre-computed filter results with their corresponding interpolation functions.

Although the steerable filter is more efficient approach compared to the bank of oriented filters, still it requires steering the basis oriented filters in all orientations with precise angle step size $\theta \in \{1, \dots, \pi\}$ (e.g. 1 degree interval) and studying the outputs of the filters as a function of their orientations to find the maximum response as the local dominant orientation (3.13). To overcome the limitations of current LDO estimation methods, a biologically inspired mechanism for LDO estimation, based on the orientation tuning properties of the ensemble of neurons in primary visual cortex (V1) of the mammalian brain, is proposed in the following sections.

3.3 Orientation Selectivity in Primary Visual Cortex

Orientation selectivity in primary visual cortex (V1) of the mammalian brain, which was first discovered by Nobel prize winners Hubel and Wiesel [44], attracted many researchers to describe the properties of these cells [52-54] and to propose computational models to approximate their functional behaviors [55-59]. According to Hubel and Wiesel [44], there are two main types of cells in the visual cortex: simple and complex cells. Contrary to separate ON and OFF-center cells in Lateral Geniculate Nucleus (LGN), a simple cell receptive field (RF) comprises of elongated excitatory and inhibitory regions located next to each other [53]. Simple cells are optimally tuned to specific orientation, spatial frequency, and spatial phase (position). On the other hand, complex cells' receptive fields are phase invariant while similar to simple cells they respond to preferred orientations and frequencies. It is well known that the responses of several simple cells are combined to form the respective field of complex cells in a hierarchically fashion. Figure 3.1 shows an overview of classical hierarchical visual pathway in the human brain proposed by Hubel and Wiesel [44].

3.3.1 Computational Model of V1 Simple Cells

Computational modeling of simple cells has received considerable attention in the literature since this type of cells is the fundamental element of the visual cortex. In those studies, most of the models are based on a linear filtering followed by nonlinear normalization and thresholding [60]. Among them, characteristics of two-dimensional Gabor filters show more similarities to the respective field of simple cells [61-62]. 2D Gabor filters are the product of a Gaussian kernel with a complex sinusoid that are tuned to preferred orientation θ_n , frequency $\frac{1}{\lambda}$, and phase φ . A two-dimensional Gabor kernel g can be written as:

$$g_{\theta_n, \lambda, \varphi}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{a^2 + \gamma b^2}{2\sigma^2}\right) \cos\left(\frac{2\pi}{\lambda} + \varphi\right), \quad (3.14)$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \cos\theta_n & \sin\theta_n \\ -\sin\theta_n & \cos\theta_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (3.15)$$

Figure 3.1 An overview of classical hierarchical visual pathway in the human brain.

The standard deviation σ represents the size of receptive field and the parameter γ is spatial aspect ratio. The phase parameter φ determines the symmetry of the Gabor kernel. For $\varphi = 0$ and $\varphi = \pi$ it is symmetric, or even. For $\varphi = \frac{\pi}{2}$ and $\varphi = -\frac{\pi}{2}$, it is antisymmetric

or odd [57]. The preferred frequency $\frac{1}{\lambda}$, and width of the filter σ are inversely related. The ratio of $\frac{\sigma}{\lambda}$ is controlled by the half-response spatial frequency bandwidth B (in octaves) and it corresponds to the number of excitatory and inhibitory regions:

$$B = \log_2 \frac{\frac{\sigma}{\lambda} + \frac{1}{\pi} \sqrt{\frac{\ln 2}{2}}}{\frac{\sigma}{\lambda} - \frac{1}{\pi} \sqrt{\frac{\ln 2}{2}}}, \quad (3.16)$$

or

$$\frac{\sigma}{\lambda} = \frac{1}{\pi} \sqrt{\frac{\ln 2}{2}} \cdot \frac{2^B + 1}{2^B - 1}. \quad (3.17)$$

According to neurophysiological findings, the spatial-frequency bandwidth B of mammals' simple cell ranges from 0.4 to 2.6 octaves [63]. Finally, the response of a simple cell is modeled by convolution of the input stimulus $I(x, y)$ with Gabor filter operator $g_{\theta_n, \lambda, \varphi}$ as follows:

$$R_{\theta_n, \lambda, \varphi}(x, y) = I(x, y) \otimes g_{\theta_n, \lambda, \varphi}(x, y), \quad (3.18)$$

where $\otimes$ denotes the spatial convolution operator. An input stimulus is usually convolved with a bank of Gabor filters with evenly spaced preferred orientations given by

$$\theta_n = \frac{(n-1)\pi}{N_\theta}, \quad n = 1, 2, \dots, N_\theta, \quad (3.19)$$

where N_θ is the total number of orientations. Figure 3.2 illustrates sample receptive fields of the symmetric and anti-symmetric Gabor filters.

Figure 3.2 Examples of Gabor filter kernels (a) symmetric and (b) anti-symmetric.

3.3.2 Computational Model of V1 Complex Cells

Biological studies of visual cortex reveal that complex cells receive input from the responses of a number of simple cells while they have larger and more complex RFs [64]. Complex cells have response preferences to orientation and frequency while they are insensitive to spatial phase. On the other hand, psychophysical observations suggest that neighboring pairs of simple cells in V1 have opposite phase preferences (i.e. $\nabla\varphi = \frac{\pi}{2}$) [65]. Hence, their responses can be grouped together to represent the complex cells RF. This clustering can be interpreted as quadrature pair summation and it can be approximated by local Gabor energy operator, which is a sum of squared responses of pair of simple cells with similar frequency and preferred orientation but with a phase difference of $\frac{\pi}{2}$, as follows:

$$E_{\theta_n, \lambda}(x, y) = \sqrt{R_{\theta_n, \lambda, 0}^2(x, y) + R_{\theta_n, \lambda, \frac{\pi}{2}}^2(x, y)} \quad (3.20)$$

3.3.3 Neural Coding of Orientation

Over the last several decades, the question of how the visual information is processed in the brain remains largely unknown. What is well understood is that an external stimulus is encoded in the neural representation in the brain and subsequently this information is

decoded in the hierarchical organization along the visual cortex. Neuroscientists considered two major hypotheses about the encoding and decoding of the visual information in the primary visual cortex. A number of studies proposed the hypothesis of population coding [66-67]. In the population coding, a specific stimulus is represented by the unique joint activity of an ensemble of neurons. Each neuron carries only a part of information. This is based on the psychophysical evidences that the activation of single neuron is noisy and contains less accurate information than a population of neurons.

An alternative hypothesis known as sparseness stated that there are highly specialized neurons (a so-called “grandmother cell”) in the brain that each is coded to represent a specific stimulus [68]. For instance, there is a unique cell that would respond when a person looks at a photo of their grandmother.

Although, there is considerable debate among neuroscientists about how information is coded in the brain, there is evidence that neurons at earlier stages of visual cortex hierarchy (V1) represent more basic visual features like line orientation, color, or direction of motion in a distributed population coding while in the upper stages of the hierarchy this information is encoded in a more sparse representation. This work will focus on the representation of basic visual features, (i.e. orientation) at earlier stages in the V1; hence, distributed population coding is addressed.

In population coding, information from external stimulus is distributed throughout the activities of large neuronal ensembles. Each neuron carries only a part of information. This stage is referred to as encoding. Later, to extract this distributed information brain needs to interpret the activities of the population of neurons. This is known as decoding stage. Several decoding techniques have been proposed to recover all the information encoded in the population of neurons. Here, those methods for recovering the stimulus orientation from population of neurons are only reviewed. For more general decoding methods, refer to [69].

Winner-take-all Decoder

It is the simplest decoding method to estimate the orientation from the neuronal ensembles. Winner-take-all decoder selects the preferred orientation of the most active unit as the estimation of the stimulus orientation. In this work, in order to recover the local dominant orientation of a stimulus from population of neurons by Winner-take-all decoder, Gabor filters spanning in all orientations from 0 to 180 degrees are applied to input image, and the preferred orientation of the most active unit is chosen as the estimation of the stimulus dominant orientation. Hence, total number of orientations (N_θ) in the Gabor filter bank plays a vital role for precisely estimating the local dominant orientation. For instance, to achieve a 1° degree level of precision we have to apply 180 oriented filters and choose the orientation θ_j corresponding to the maximum magnitude Gabor energy response as the local dominant orientation:

$$\hat{\theta}_{\text{WTA}}(x, y) = \theta_j, \quad j = \operatorname{argmax}_i \|E_{\theta_i, \lambda}(x, y)\| \quad (3.21)$$

The drawback of this decoder comes from the fact that the final estimation depends only on single unit activation and information from the rest of the population is discarded. The activation of single neuron is noisy and contains less accurate information than a population of neurons. In addition, the memory capacity in a single neuron coding is limited by the number of neurons.

Population Vector Decoder

In the 1980s, Georgopoulos et al. [67] introduced the concept of population vector coding, in which each unit represents by a vector whose magnitude is proportional to the activity and its direction corresponds to preferred orientation of respective unit. Therefore, in this work, each Gabor energy response is considered as a vector whose magnitude is proportional to the Gabor energy response value and its direction corresponds to orientation θ_n . The population vector is computed by summing all the vectors together. By vector summation of individual Gabor filter responses we can estimate the LDO more

accurately while it is less sensitive to input noises. The orientation of the population of units corresponds to the angle of the resultant vector:

$$\mathbf{z}(x, y) = \sum_{i=1}^{N_\theta} \|E_{\theta_i, \lambda}(x, y)\| \mathbf{e}_i, \quad \mathbf{e}_i = (\cos\theta_i, \sin\theta_i), \quad (3.22)$$

$$\hat{\theta}_{\text{pop}}(x, y) = \tan^{-1} \frac{z_y}{z_x} \quad (3.23)$$

where θ_i is the preferred direction of each unit and $\hat{\theta}_{\text{pop}}$ the angle of estimated population vector corresponds to the local dominant orientation. Although, population vector coding is more robust decoder, it ignores the correlation between the different units in the population and assumes that they are independent from one another. This method is referred as classical Gabor energy LDO estimation using population vector coding or in abbreviation the classical LDO [70]. To obtain better accuracy and more robustness, while considering the correlation within the population, a novel opponent LDO model is proposed in the following section.

3.4 Models of Orientation Opponency Mechanism

Several psychophysical studies suggest that activity of a neuron in the primary visual cortex not only depends on the input stimulus but also is correlated to the responses of other neurons in population [70-71]. One example of this type of neural correlation is observed in the middle temporal area (MT) called motion opponency [72-74]. Motion opponency refers to the mechanism in visual motion perception that the activity of a neuron to a moving stimulus is suppressed by the activity of a neuron in the opposite direction. For instance, it is impossible to observe both leftward and rightward motion in a same location of visual field. Adelson and Bergen [72] suggested an opponent-motion energy unit by taking arithmetic difference between the leftward and rightward responses. This unit estimates the direction of motion as a rightward if the sign of it is positive and a leftward motion if it has a negative sign [72].

Since the perception of motion and orientation are very similar in visual cortex [72], an idea of opponency mechanism in motion perception is adopted and employed as part of the proposed orientation estimation model. One difference between direction of motion and orientation is that the opponent offset in orientation comes from the orthogonal direction ($\theta + \frac{\pi}{2}$) not from the opposite direction ($\theta + \pi$) in motion estimation. In fact, a vector with direction θ and a vector with opposite direction ($\theta + \pi$) defines the same orientation. In our model, each Gabor energy response is suppressed by its orthogonal oriented Gabor energy response. The resulting responses are then combined through population vector coding to estimate more precisely the LDO.

The magnitude of the resulting vectors $\|S_{\theta_{S_i}, \lambda}(x, y)\|$ are calculated based on the absolute difference between two orthogonal Gabor energy responses. The sign of difference between two orthogonal Gabor energy responses will reveal the direction θ_{S_i} of resulting vectors as follows:

$$\|S_{\theta_{S_i}, \lambda}(x, y)\| = \left| \|E_{\theta_i, \lambda}(x, y)\| - \|E_{\theta_i + \frac{\pi}{2}, \lambda}(x, y)\| \right|, \quad (3.24)$$

$$i=1, 2, \dots, \frac{N_\theta}{2}$$

$$\begin{cases} \theta_{S_i} = \theta_i & \text{if } \|E_{\theta_i, \lambda}(x, y)\| - \|E_{\theta_i + \frac{\pi}{2}, \lambda}(x, y)\| > 0 \\ \theta_{S_i} = \theta_i + \frac{\pi}{2} & \text{if } \|E_{\theta_i, \lambda}(x, y)\| - \|E_{\theta_i + \frac{\pi}{2}, \lambda}(x, y)\| < 0 \end{cases} \quad (3.25)$$

Note that the N_θ Gabor energy responses are reduced to $\frac{N_\theta}{2}$ resulting vectors ($S_{\theta_{S_i}, \lambda}(x, y)$). In this model, the total number of Gabor filters N_θ should be even. Then, $\|E_{\theta_i, \lambda}(x, y)\|$ in equation (3.22) can be replaced by new vectors $S_{\theta_{S_i}, \lambda}(x, y)$ as follows:

$$\mathbf{z}(x, y) = \sum_{i=1}^{\frac{N_\theta}{2}} \|S_{i, \lambda}(x, y)\| \mathbf{e}_i \quad \mathbf{e}_i = (\cos \theta_{S_i}, \sin \theta_{S_i}) \quad (3.26)$$

Next, the local dominant orientation at pixel location (x, y) can be defined by equation (3.23). This method is referred as opponent LDO model in this work. Opponent LDO is

more robust and gives higher accuracy compared to the classical LDO method. The proposed approach is evaluated in terms of both accuracy and robustness with classical method and several state of the art computer vision LDO algorithms in section 3.5.

The opponent LDO model can be further improved by hierarchical estimation procedure. Once the initial dominant orientation $\hat{\theta}_{pop}(x,y)$ is estimated, the second stage is performed as follows. Five Gabor filters in the range of $\{\hat{\theta}_{pop}(x,y) - \theta_n : \frac{\theta_n}{2} : \hat{\theta}_{pop}(x,y) + \theta_n\}$ are applied to each pixel location (x,y) with the initial estimated dominant orientation $\hat{\theta}_{pop}(x,y)$ and the resulting vector is calculated based on the vector summation of these five Gabor filter responses. The resulting orientation estimation will provide better precision than the first stage. This procedure will be elaborated more on the section 3.5.

3.4.1 *Multiscale Estimation*

So far, the opponent LDO model based on a single scale Gabor filter is introduced. Thus, one has to select a single scale carefully to capture most of the information in an input image. In this case, Bovik et al. [75] proposed a supervised method for Gabor filter parameter selection based on the local power spectrum characteristics of the input image. They considered three general types of textures and tried to select appropriate frequency peak for each of them. One drawback of the supervised method is that it depends on sophisticated power spectrum analysis, which may not always be applicable to all types of texture patterns. On the other hand, a single scale Gabor filter will detect only a specific pattern of a certain frequency, while in most of the natural imagery, there is wide range of frequencies from low, medium to high.

Psychophysical observations in visual cortex reveal that receptive field of simple cells are tuned jointly to both different frequencies and orientations. Thus, one possible solution is to use multiscale, multi-orientation Gabor filtering approaches (also known as filter banks). Gabor filter banks approaches has been used for variety of applications such as document analysis, object detection, and texture classification/ segmentation. Generally,

an input image is convolved with a set of Gabor filters, which are fine tuned to detect different frequencies and orientations. Then, these filter bank responses are added to a feature vector for each pixel location of input image. Later, these feature vectors constructed by multichannel Gabor features are passed into a classifier or segmentation algorithm to decompose an image accordingly. However, in the case of LDO estimation, the aim is to estimate the dominant orientation at each pixel location, rather than segmenting the whole image. Therefore, two methods for combining the multichannel Gabor filters is proposed in the following.

First of all, in order to take into account different scales (i.e. frequencies) in Gabor filter formula, λ in equations (3.14) is replaced by λ_m , where $m \in \{1, \dots, N_F\}$. N_F is the total number of frequencies. The two major issues that play important roles in multiscale Gabor filtering are the total number of frequencies (N_F), and center frequency spacing S_F (i.e. frequency ratio F_r). The number of frequencies used in the literatures ranges from 3 to 7 [76]. Spatial frequencies normally range from 0.04 to 0.5 cycles per pixel (cpp) (equivalently, wavelengths of 2.0 to 25.0 pixels per cycle (ppc)). Center frequency spacing S_F (or frequency ratio) are normally set to one-octave or half-octave apart (2 or $\sqrt{2}$ frequency ratio respectively). The frequency bandwidths B is also set to center frequency spacing S_F in order to minimize the amount of overlaps between filters [77].

Finally, a total number of $N_F * N_\theta$ Gabor filters are applied to input images using (3.18) at each pixel location (x, y) and Gabor energy $E_{\theta_n, \lambda_m}(x, y)$ (3.20) is calculated for a specific frequency $\frac{1}{\lambda_m}$ and orientation θ_n . Once the Gabor energy responses are computed, a method should be proposed to combine them together. In the first proposed method, multiscale Gabor energy responses are initially combined in the frequency domain. In other words, at each orientation θ_n , the filter with the maximum magnitude response is selected at each pixel location:

$$T_{\theta_n}(x, y) = \max\{E_{\theta_n, \lambda_m}(x, y) | m = 1, \dots, N_F\}. \quad (3.27)$$

Therefore, the total number of filters is reduced to N_θ responses corresponding to N_θ filters' orientations. $E_{\theta_n, \lambda_m}(x, y)$ is then replaced by $T_{\theta_n}(x, y)$ in equations (3.24) and (3.25) as follows:

$$\begin{aligned} \|S_{\theta_{S_i}}(x, y)\| &= \left| \|T_{\theta_i}(x, y)\| - \left\| T_{\theta_i + \frac{\pi}{2}}(x, y) \right\| \right|, \\ i &= 1, 2, \dots, \frac{N_\theta}{2}. \end{aligned} \quad (3.28)$$

$$\begin{cases} \theta_{S_i} = \theta_i & \text{if } \|T_{\theta_i}(x, y)\| - \left\| T_{\theta_i + \frac{\pi}{2}}(x, y) \right\| > 0 \\ \theta_{S_i} = \theta_i + \frac{\pi}{2} & \text{if } \|T_{\theta_i}(x, y)\| - \left\| T_{\theta_i + \frac{\pi}{2}}(x, y) \right\| < 0 \end{cases} \quad (3.29)$$

Finally, the local dominant orientation at pixel location (x, y) can be achieved through equation (3.26). This type of combination is referred to the “*Maxfrequency*” method in this thesis.

In the second approach, instead of first finding the local maxima at each orientation in frequency domain and then combining the resulting Gabor energy responses with opponency mechanism, we first perform the opponent LDO estimation at each scale (frequency) separately and then select an orientation as the local dominant orientation for which the maximum response is achieved across different scales. Suppose that we have N_F number of frequencies, at each frequency “ m ” the vector summation of the resulting vectors $\|S_{i, \lambda_m}(x, y)\|$ can be found by:

$$\mathbf{z}_m(x, y) = \sum_{i=1}^{\frac{N_\theta}{2}} \|S_{\theta_{S_i}, \lambda_m}(x, y)\| \mathbf{e}_i, \quad (3.30)$$

$$\mathbf{e}_i = (\cos \theta_{S_i}, \sin \theta_{S_i}), \quad m=1, \dots, N_F$$

The dominant orientation $\hat{\theta}_{\text{pop}}$ is the orientation for which the maximum response $\|\mathbf{z}_m(x, y)\|$ is achieved. The second proposed approach is denoted as “*Multipopulation*” method.

$$\hat{\theta}_{\text{pop}}(x, y) = \theta_j, \quad j = \operatorname{argmax}_m \|\mathbf{z}_m(x, y)\| \quad (3.31)$$

3.5 Experimental Evaluation

In this section, several series of experiments on both synthetic and natural stimuli are conducted to show the effectiveness of the proposed methods for local dominant orientation estimation and the results are compared with the several state of the art LDO methods in terms of accuracy, and noise robustness.

3.5.1 *Experimental Results on Synthetic Images*

A set of synthetic images with associated ground truth are generated in order to assess the accuracy of the proposed methods. Each image contains nine elongated bars scattered evenly in the visual field with full contrast in a black background with an image resolution of 640x480. Each bar is rotated counter-clockwise around its origin by a set angle ranging from 0 degrees to 179 degrees with a step size of 1 degree. Figure 3.3 shows two examples of synthetic images with the ground truth angle of 30 degrees and 120 degrees.

Figure 3.3 A synthetic image with the ground truth angle of 30 degrees (left), a synthetic image with the ground truth angle of 120 degrees (right).

For computational efficiency and simplicity, all the input images are scaled down three times by the image Gaussian pyramid to 80x60 pixels. Gaussian pyramid is computed by applying low-pass filtering followed by sub-sampling of the original image. The result in each step is an image with half the width and height of the original image. The images at

the level three of the Gaussian pyramid are utilized for all the experimental results reported in this section.

The first step of the opponent LDO model is to select appropriate filter parameters. In the first set of experiments, single-scale LDO model is used. In this work, the value of the central frequency is set to $\frac{1}{8\sqrt{2}}$ cycles/pixels. Next, four orientations with angular spacing of $\pi/4$ radians are employed. In addition, the spatial aspect ratio γ is set to 1 for simplicity. The size of Gabor filter depends on the size of the input image. The size of Gabor filter window is set to 15×15 . First set of experiments are run with a number of fixed parameters summarized in Table 3.1.

TABLE 3.1 FIXED PARAMETERS USED IN THE MODEL

Parameters	Values
No. of filter orientations	4
No. of filter scales	1
Gabor filter orientations	$\{0, \pi/4, \pi/2 \text{ and } 3\pi/4\}$
Filter size	15×15
Image resolution	80×60
Spatial frequency	$\frac{1}{8\sqrt{2}}$

Four oriented Gabor filters $\{0, \pi/4, \pi/2 \text{ and } 3\pi/4\}$ are applied to the input image and the Gabor energy responses are calculated at each pixel location. In order to avoid numerical difficulties in subsequent calculations and to make the filter response values lie in the same interval, some data normalization has to be done. Most often, the filter responses are individually rescaled to lie within a fixed range like $[0, 1]$. However, normalization to a fixed range would make all the Gabor energy responses have the same importance and the resultant combined vector may be inaccurate.

To solve this problem, the following normalization technique is proposed. First, the global maximum ($E_{\max}$) across all four Gabor energy responses $E_{\theta_i, \lambda}$, $\theta_i \in \{0, \pi/4, \pi/2 \text{ and } 3\pi/4\}$ are determined. Then, each Gabor energy response is multiplied by $\frac{1}{E_{\max}}$. This

scaling technique preserves the importance of each Gabor energy response while making them comparable in the same interval. Figure 3.4 illustrates the orientation tuning curves of four Gabor energy responses for all angles of input synthetic images after normalization. It is depicted that each individual filter has a very broad orientation tuning curve. It is similar to the distribution of the orientation selectivity in visual cortex which each neuron often has very broadly tuning curves to a wide range of given orientations [78]. This neurons' characteristic, will be used later in the proposed opponent LDO model to estimate orientation with high precision. On the other hand, Figure 3.4 illustrates the opponent responses of the Gabor energies in orthogonal directions.

Figure 3.4 Four Gabor energy orientation tuning curves for all the angles of synthetic images (0 to 3.14 radians).

Moreover, four Gabor energy responses of a synthetic image containing bars with angle of 30 degrees are shown in Figure 3.5. High intensity pixels (brighter color) depict greater activation while low intensity pixels show less activation. Finally, to measure errors of the

proposed LDO estimation approach and to better understand nonlinear effects of Gabor filtering, 9 points located at the center of each single bar are selected and the local dominant orientations are measured at these locations.

Figure 3.5 Four Gabor energy responses of a synthetic image containing bars with angle of 30 degrees.

The error is calculated based on the difference between estimated orientation $\hat{\theta}_{\text{pop}}(x, y)$ and actual orientation $\theta(x, y)$ as follows:

$$e_{\theta}(x, y) = \hat{\theta}_{\text{pop}}(x, y) - \theta(x, y) \quad (3.32)$$

As it is shown in Figure 3.6, the computed angular estimation error is symmetric and it is bounded with the maximum error of 0.0219 radians with the mean error of 0.0098 radians. Therefore, the proposed model has a finite angular resolution of 0.0219 radians while such

accuracy would require 120 oriented Gabor filters in conventional winner-take-all method.

Figure 3.6 The angular estimation error from 0-3.14 radians of the proposed method based on the true orientations.

In addition, considering that the error function that result from using equation (3.32) contains a systematic offset at specific angular values, the accuracy of this method can be further improved. If the error were known with probability of 1 for all angles, the error analysis would be normally just an addition of error e_θ to the estimation orientation $\hat{\theta}_{pop}$ at each pixel location. In order to apply such error analysis, first equation (3.32) is needed to rewrite based on the estimation orientation $\hat{\theta}_{pop}$.

$$e_{\hat{\theta}}(x, y) = \hat{\theta}_{pop}(x, y) - \theta(x, y) \quad (3.33)$$

Then, we try to find the best fit to the data by assigning a lower complexity function to the entire set of data points. In this work, the angular estimation error is fitted with a cubic spline functions. Each data point was fitted by a cubic polynomial in the interval of 11 points (5 points before and 5 after the fitted point).

Finally, this error function is used to better approximate the true orientation based on the estimated orientation $\hat{\theta}_{\text{pop}}(x, y)$. The angular estimation error after error removal is shown in Figure 3.7. In this case, the maximum angular estimation error is less than 0.0053 radians with the mean error of 0.0023 radians.

Figure 3.7 The angular estimation error of the proposed method after cubic polynomial fit from 0-3.14 radians.

Two-stage LDO Estimation

The angular estimation error can be further reduced by applying hierarchical processing, which it is called in this thesis two-stage LDO estimation technique. Once an initial dominant orientation $\hat{\theta}_{\text{pop}}(x,y)$ is estimated at each pixel location, a set of five Gabor filters separated by an angular spacing of $\frac{\theta_n}{2}$ are applied to each estimation orientation $\hat{\theta}_{\text{pop}}(x,y)$ to estimate the dominant orientations more precisely. Figure 3.8 shows the angular estimation error after two-stage LDO estimation. Notice that a similar level of accuracy was obtained in two-stage LDO as in one stage followed by removal of systematic error (see Figure 3.7).

Figure 3.8 The angular estimation error of two-stage LDO estimation method from 0-3.14 radians.

In the proposed two-stage LDO estimation technique error is reduced to the maximum error of 0.0122 radians with the mean error of 0.0038 radians. For applications, which require the angular error to be very small and the real-time processing is not so crucial, the two-stage LDO estimation technique is recommended.

Classical vs. Opponent LDO Estimation

As it is discussed in the section 3.4, a new model (opponent LDO) based on correlation of Gabor filter responses in the population is introduced. According to the proposed model, it is more robust to noise, thus, it gives better accuracy compared to classical LDO method. Here, in this experiment, sensitivity to additive noise is evaluated by applying zero mean Gaussian white noise with different noise variances to the synthetic stimuli. An example of an original synthetic image with its noisy version with noise variance of 0.6 are shown in Figure 3.9.

Figure 3.9 An example of a synthetic image with an angle of 150 degrees (left), noisy synthetic image with an angle of 150 degrees (right).

The maximum orientation estimation errors with respect to the noise variance are measured across all angles ranging from 0 degrees to 179 degrees at 1 degrees increment. Figure 3.10 displays the results of both opponent LDO and classical LDO as a function of noise variance.

Figure 3.10 The results of opponent LDO and classical LDO as a function of noise variance.

As expected, since the opponent LDO model uses opponency mechanism, its estimation error is lesser and increases slowly by increasing noise variance compared to the classical LDO method.

Effect of the Number of Filters in the Population

To investigate the effect of increasing the population size and determine the optimal number of filters, a set of four experiments are conducted where at each experiment different number of Gabor filters are used in the population. The number of Gabor filters used are 4, 8, 16, and 32. At each experiment, the opponent LDO is applied to the synthetic images with various noise levels. The results of this study are shown in Figure 3.11. It can be seen that by increasing the number of Gabor filters from 4 filters (minimum number of filters required in the opponent LDO model) to 8 filters the

estimation error is significantly reduced. On the other hand, increasing the size of population to more than 8 Gabor filters (e.g. 16, 32) does not have a significant effect on the orientation estimation errors.

Figure 3.11 The effect of increasing the number of Gabor filters in the population.

Contrast Invariance

To further analyses the proposed opponent LDO characteristics, the responses of opponent LDO model in various stimulus contrasts are evaluated. The opponent LDO model is applied to the synthetic images with different contrasts levels. Contrast level varies from 10 to 100 percent of the stimulus full intensity (i.e. 255) in step size of 10. Figure 3.12 shows some examples of synthetic images (in this case 30 degrees stimulus) with three different contrast levels.

Figure 3.12 Examples of synthetic input images with three different contrast levels (10, 50, 80 percent of stimulus full intensity).

The average and standard deviation of orientation estimation errors as a function of contrast level across all input angles are displayed in Figure 3.13. It appears that the opponent LDO model is invariant to stimulus contrast level, which is similar to the contrast-invariant orientation tuning properties of simple cells found in the primary visual cortex [79].

Figure 3.13 Average and standard deviation orientation estimation errors of the opponent LDO model as a function of contrast level.

Multiscale vs. Single-scale LDO model

In the section 3.4.1, two methods for combining the multiscale Gabor filters are proposed. Here, two sets of experiments are conducted to evaluate the effect of multiscale filtering on the performance of opponent LDO model. In the first set of experiment, the effect of combining multiscale Gabor filters on the robustness to noise is explored. A set of five different spatial frequencies is employed in the multiscale LDO models within range from 0.25 (wavelength of 4) to 0.0625 (wavelength of 16). The frequency ratio of $\sqrt{2}$ corresponding to half-octave frequency spacing is chosen as it is suggested by [76]. Table 3.2 summarized the parameters used in the multiscale LDO models. Figure 3.14 illustrates the real parts of Gabor filter kernels for the parameters used in Table 3.2.

TABLE 3.2 FIXED PARAMETERS USED IN THE MULTISCALE MODEL

Parameters	Values
No. of filter orientations	4
No. of filter scales	5
Gabor filter orientations	$\{0, \pi/4, \pi/2 \text{ and } 3\pi/4\}$
Gabor filter frequencies	$\{\frac{1}{4}, \frac{1}{4\sqrt{2}}, \frac{1}{8}, \frac{1}{8\sqrt{2}}, \frac{1}{16}\}$

The result of two multiscale LDO methods vs. single-scale LDO with various noise levels is depicted in Figure 3.15. The plot indicates that the “*Maxfrequency*” method performs better (less maximum estimation error) than both “*Multipopulation*” and single-scale for different noise levels.

Figure 3.14 Real part Gabor filter kernels

Figure 3.15 Multiscale LDO methods vs. single-scale LDO as a function of noise variance.

The next experiment is performed to evaluate the effect of scale (bar thickness) on the performance of multiscale LDO methods vs. single-scale LDO. To estimate the dominant orientation at the center of each single bar, the scale of the filter should be chosen carefully according to the bar thickness. However, it is assumed that no prior knowledge about the input stimulus is known and input stimulus may contain different bars' thicknesses. Hence, the proposed algorithm should be able to estimate the dominant orientation of bars with arbitrary widths. The results of this experiment are depicted in Figure 3.16. Note that the bar thickness varies from 1 to 12 pixels with the fixed contrast level (100%) and without adding any noise. The plots show that “*Maxfrequency*” methods performs better than “*Multipopulation*” and single-scale methods for all levels of the bar thicknesses.

Figure 3.16 Multiscale LDO methods vs. single-scale LDO as a function of bar thickness.

Comparison with State of the Art LDO Methods (Global Dominant Orientation)

To compare the performance of the proposed algorithm with the state of the art LDO methods, an experiment utilizing a single known oriented sinusoidal grating as stimulus is conducted. The reason that sinusoidal grating is employed instead of previous single bars synthetic images is that in order to compare the performance of each algorithm more effectively one needs to find the global dominant orientation across the entire image. Thus a sinusoidal grating, which is a series of repeating bars with the same orientation (called a "grating") across the input image field, is more informative compared to single isolated bars used for error analyses.

Given a set of local dominant orientations within an input image, it is possible to associate a global dominant orientation by calculating the arithmetic linear mean over the entire

local dominant orientations of an input image. However, since angular data has a periodic nature (i.e. $\hat{\theta} = \hat{\theta} + 2\pi$), arithmetic linear mean may lead to the wrong estimation of global dominant orientation. For example, consider the arithmetic linear mean of two local angles of 1 degree and 359 degrees would be 180 degrees while both of them are approximately pointing to the same direction near 0 degrees.

This effect becomes more clear when it comes to axial data (i.e. oriented or bimodal data), where two angles in opposite directions ($\hat{\theta}$, $\hat{\theta} + \pi$) may cancel each other by calculating the arithmetic linear mean while they have the same orientation. Therefore, to estimate the global dominant orientation one needs to consider directional (circular) statistics suitable for analysis of directional data [80] as follows. Each local dominant orientation is considered as a unit vector represented in a polar form:

$$e^{i\hat{\theta}} = \cos \hat{\theta} + i \sin \hat{\theta}. \quad (3.34)$$

In order to make two vectors in opposite directions have the same orientation (i.e. $\hat{\theta}$ and $\hat{\theta} + \pi$), angle of each unit vector is doubled so that the vector with angle of ($\hat{\theta} + \pi$) would have the angle of ($2\hat{\theta} + 2\pi$) which is equal to $2\hat{\theta}$. Then, the mean resultant vector over the entire image is calculated by:

$$\mathbf{r} = \sum_{j=1}^N e^{i2\hat{\theta}_j}, \mathbf{r} = (r_x + ir_y). \quad (3.35)$$

Where N is the total number of estimated local dominant orientations in the input image. The orientation of resultant vector can be defined by:

$$\hat{\theta}_{\text{global}} = \frac{1}{2} \tan^{-1} \frac{r_y}{r_x}, \quad (3.36)$$

where $\hat{\theta}_{\text{global}}$ is the global dominant orientation of the input image between $[0, \pi)$. The angle is divided by 2 because each unit vector is doubled initially before the summation. Figure 3.17 shows a single known sinusoidal grating image ($\pi/6$ radians) and a noisy image with noise variance of 0.6.

Figure 3.17 Sinusoidal grating image ($\pi/6$ radians), noisy image with noise variance of 0.6.

The two state of the art LDO methods PCA/Multiscale Gradient method [50] and steerable filters [51] are compared with the proposed single-scale and multiscale (“*Maxfrequency*”) methods (8 Gabor filters used in the population) on the sinusoidal grating images with single known orientation of $\pi/6$ radians. The average orientation estimation errors as a function of noise variance are shown in Figure 3.18.

It can be seen from Figure 3.18 that the proposed multiscale and single-scale methods are almost robust to noise variances while they perform significantly better in terms of orientation accuracy than PCA/Multiscale Gradient method and steerable filters.

Figure 3.18 Average estimation error as a function of noise variance across different methods.

3.5.2 Experimental Results on Natural Images

In order to further evaluate the performance of the proposed LDO model, the proposed method is applied to natural texture images and compared with various state of the art LDO methods presented in the section 3.2. To compare quantitatively the results of various LDO methods one needs to know the ground truth orientations assigned to each natural texture image. Therefore, various texture samples that are presenting one dominant global orientations (i.e. anisotropic) are selected from the Brodatz and Outex datasets, which are the most popular and widely used benchmarks. Only those images are chosen from these datasets that human experts can identify their ground truth orientations with high precision. Then, each of them are artificially rotated counter clockwise from 0 to 90 degrees with a step size of 30 degrees. Therefore, the dataset is comprised of 32 natural images from two benchmarks. Anisotropic texture samples used in these experiments with

one of their rotated versions (30 degrees) are depicted in Figure 3.19 and Figure 3.20. The rotated version of Brodatz texture images is extracted from USC-SIPI database [81] and OuTex texture images from Outex_TC_00010 [82] test suite.

Each texture image first is downsampled by a factor of 4 in order to reduce the computational load. The average orientation estimation errors (radians) for each of these sample images are depicted in Table 3.3. The results show that the multiscale opponent LDO method outperforms all the other baseline algorithms in the natural texture images in terms of less orientation estimation errors. Although, steerable filters yield reasonably good results they are computationally far more expensive than the opponent LDO method since it requires steering the oriented energy in all orientations with precise angle step size (in this case 1 degree interval) to estimate the local dominant orientation.

Figure 3.19 Sample images from Brodatz textures.

Figure 3.20 Sample images from OuTex textures.

Table 3.3 Average orientation estimation errors (radians)

Image	PCA/Multiscale Gradient	Steerable filters	Opponent LDO Single-scale	Opponent LDO Multiscale
D11 (Brodatz)	0.0965	0.052	0.077	0.052
D51 (Brodatz)	0.058	0.049	0.046	0.046
D68(Brodatz)	0.060	0.038	0.018	0.008
D94 (Brodatz)	0.043	0.007	0.027	0.019
Canvas006(Outex)	0.070	0.014	0.008	0.009
Canvas009(Outex)	0.076	0.010	0.017	0.008
Canvas011(Outex)	0.063	0.033	0.036	0.033
Canvas024(Outex)	0.1084	0.133	0.087	0.083
Mean Error	0.071	0.042	0.039	0.032

To qualitatively analyze the results of opponent LDO model, each estimated LDO is projected back to the original images and sample results for two images from Brodatz textures are displayed in the Figure 3.21. The complete set of results is illustrated at Appendix A. The orientation map is illustrated for each pixel location in the image with

step size of 5 pixels for better visualization. In conclusion, the opponent LDO model proposed in this paper gives the best performance in terms of accuracy, noise robustness with far less computational expenses than the existing LDO methods.

Figure 3.21 LDO estimations are projected back to the sample image from Brodatz textures.

3.6 Chapter Summary

In this Chapter, a biologically inspired mechanism for orientation selectivity characteristics in primary visual cortex (V1) of human visual system was proposed. The proposed model is based on psychophysical evidences that at earlier stages in the primary visual cortex, basic visual feature such as line orientations are encoded using a population of correlated neurons. The proposed model combined the distributed population coding with an opponency mechanism found in ensemble of neurons (opponent LDO model) to estimate the local dominant orientation at each pixel location of input images. The proposed opponent model outperforms the classical LDO method in terms of accuracy and robustness to noise. The model was further extended to take into consideration of multiple scales in an input stimulus. Two multiscale filtering methods for combining the multichannel Gabor filters called the “Maxfrequency” method and “Multipopulation”

method were proposed. Based on experimental results the “Maxfrequency” method performed significantly better than “Multipopulation”. Finally, the effectiveness of the opponent LDO model was demonstrated on a broad set of experiments. The experiments were conducted on both synthetic and natural texture images with associated ground truth. Apart from being biologically plausible, the results demonstrate several advantages of the proposed approach including higher accuracy, more robustness to input noise and scale variations, contrast invariance, and finally less computational cost compared to filtering based LDO methods.

Chapter 4

VANISHING POINT ESTIMATION FOR UNMANNED GROUND VEHICLES

4.1 Introduction

Monitoring driver behaviors has been a subject of research for several decades, yet it is still poorly understood. The one key problem that prevents the idea of fully understanding the driver behaviors is that driving is a very complex behavior comprising of many subtasks. A driver has to identify road trajectory, maintain the vehicle position on the road, anticipate the vehicle's heading, monitor the road in order to avoid any possible hazard, estimate the behavior of other drivers or pedestrians and detect and analyze road signs for further necessary actions. A human driver needs to acquire and process diverse range of sensory information to accomplish the driving task although it is investigated that ninety percent of this sensory information used by a driver is the visual information [83]. Therefore, driving is primarily a visually guided behavior task strongly link to motor control.

There are many hypotheses about what visual features a driver might use while driving. In the computer vision literature, many algorithms have been developed to address this problem. These algorithms often rely heavily on existence of distinct road features such as lane markings, road boundaries [5, 8] or unique road surfaces appearance characteristics (i.e. color and/or texture) [12-13]. While such algorithms work well in the well constructed, paved roads, they would have difficulties in unstructured environments, where there are hardly any discernible and invariant features that characterize the road or its boundaries. Since humans are easily able to drive on such unstructured environments, it is natural to focus on drivers' gaze patterns and fixations to understand where and how long drivers look to obtain important visual information while driving [84-87].

Over the decades, many studies have been conducted to understand the visual behavior of drivers. Among those, Rockwell et al. [88], a pioneer in the eye movement research, developed the first eye tracking system that could monitor and record drivers' on-road eye fixation patterns. In one of their studies [86], they observed that drivers focus their attention (66% of the time) towards the road at a distance greater than 76 meters (250 feet) in front of the car while only 21% of the time the car drivers look at a distance between 22 meters (75 feet) to 76 meters (250 feet). Moreover, they reported that drivers do not look at any road feature less than 22 meters in front of the car. Later, Shinar et al. [89] claimed that as a driver becomes more familiar with the road, the fixation patterns tend to be increasingly constrained and more drifted on the far field. In this case, drivers spend less time looking at the road features such as road edges' lines. One explanation of these findings is that drivers prefer to look close to the end of the road where the road edges converge on the horizon (i.e. the vanishing point). Thus, they can obtain more information about the environment with maximal lead time.

In 1993, Serafin [87] described the drivers' eye fixation patterns on straight and curved rural roads. He outlined a variety of visual behaviors based on road characteristics, road features, drivers' age and gender. His study also revealed that fixation patterns are more directed toward the far field (54%) on a straight road (Figure 4.1) and 35% on a curved road (Figure 4.2). He described "Far Field" as the area around the focus of expansion

where the end of the road meets the horizon. He reported that the probability of fixating on any other feature for all types of roads is less than a 10%. To understand visual information of which sections of the road is highly utilized by driver, Land et al. [84] conducted series of experiments by showing only small segments of the road to the drivers. They concluded that information about the far field of the road helps the drivers to increase their stability and predict the upcoming road trajectory especially during fast driving scenarios while near field information is mostly used for correcting the car position in the lane at lower speed.

Figure 4.1 Drivers' eye fixation patterns on straight roads, Adapted from [87].

Figure 4.2 Drivers' eye fixation patterns on 13 degrees curved roads, Adapted from [87].

In the most recent findings, Rogers et al. [90] provided the first analysis of the relation between gaze, speed and expertise in straight-road driving. They demonstrated that driver gaze patterns are highly correlated with speed of the movement. The distribution of the gaze patterns becomes more constrained by an increase in driving speed while in all speed conditions, the peak of the distribution falls very close to the vanishing point (Figure 4.3). Their study revealed that at higher speed gaze was never directed at the lane markers or road surface in near field.

Based on these empirical findings, it seems that vanishing point is a salient and consistent feature during most of the driving behavior tasks regardless of type of the environments. When a driver becomes more familiar with the road situations, it prefers to look at the far field and close to the end of the road to anticipate the upcoming road trajectory and the car steering. By increasing the driving speed, drivers concentrate more on the end of the road

to obtain visual cues of the upcoming road with maximal lead time. Hence, estimating the location of vanishing point plays a pivotal role in the determination of the road trajectory. In this chapter, a computational model for the vanishing point estimation based on the visual gaze behavior of drivers is developed for autonomous UGVs. The algorithm employs the LDO estimation mechanism proposed in the Chapter 3 to detect the locally oriented patterns such as road edges, ruts and tire tracks left by previous vehicles and then make them vote for the global convergence point of the road in the visual field. This model utilizes only monocular vision information similar to human perception to detect road directions with respect to the vehicle.

Figure 4.3 Gaze distribution in the driving phase as a function of speed a) 30 km/hr, b) 110 km/hr c) 190 km/hr, Adapted from [90].

This chapter is organized as follows. Section 4.2 describes related state of the art computer vision vanishing point detection algorithms. In section 4.3, a new mechanism

for vanishing point estimation is presented. It includes novel voting schemes, which comprises of several voting algorithms with different weighting functions. In section 4.4, the performance of the proposed vanishing point detection algorithm is evaluated using experimental results obtained with different weighting functions and different voting schemes using two-way analysis of variance (ANOVA). It is followed by a series of quantitative and qualitative analyses using real data sets from the DARPA project. Finally, Section 4.5 summarizes the results and draws conclusions.

4.2 Related Work

A set of lines in the image plane that corresponds to a set of parallel lines in the 3D world space converges to a common point in image space known as a vanishing point. Vanishing points are widely used in a diverse range of computer vision applications such as 3D interpretation, scene reconstruction and camera calibration. In general, a man-made environment, has two or three vanishing points which correspond to the different sets of parallels lines of the architectural structures such as buildings, walls, and windows stretched in vertical or horizontal direction [91].

However, in the context of the cross-country road detection, regardless of types of the roads, there is a unique vanishing point, which associated with the most immediate straight road parts in the direction of the optical axis of the forward looking cameras (Figure 4.4 (a)). The vanishing point plays an important role as a global constraint for detecting roads, since all parallel border road lines, road edges, and even ruts and tire tracks left by previous vehicles on the road appear to converge into a single point known as the vanishing point. In the case of curved road, a set of vanishing points is associated with different tangent directions along each section of the road. However, a single best vanishing point can be estimated along the tangent direction of the most immediate road regions in front of the vehicle (Figure 4.4 (b)). By estimating the vanishing point location, the direction of the road in front of the vehicle can be approximately determined. This can be utilized in future for the autonomous vehicle steering [92].

Most of the existing vanishing point detection algorithms rely on three steps [93-95]. Typically, the first step is to perform some sort of edge detection (e.g. Canny) on the image. In order to extract most dominant edges like road borders or lane markings. Edge detection tells us where the edges are, but not what kinds of shape they have (like lines, arcs, etc.). The next step is to determine if there are any line segments in the image. There are many methods for line extraction in the literature. A good review of line extraction methods can be found in [96].

Figure 4.4 (a) Vanishing point in the straight road (b) Vanishing point in the curved road

Once all the line segments are identified, a voting procedure is applied to find the intersections of the lines. The differences between voting procedures come from different ways of defining so called accumulator spaces where votes are accumulated. These accumulator spaces are categorized into three groups. One approach is to consider the line intersections directly in the image space. Since the line segments can intersect anywhere within an image to infinity, it makes the search space unbounded. Many vanishing points detection techniques map the unbounded image R^2 onto a bounded space, like Gaussian Sphere [97] or Hough space [94, 98], and search for possible vanishing points as local maxima in these bounded spaces. The shortcoming of all these methods is that they are based on some necessary preliminary steps of edge detection followed by line extraction,

which may restrain the process of detecting the true vanishing points in unstructured road conditions where there are no apparent boundaries. Moreover, at standard line extraction step, the direction of each detected edge is ignored and only the position of each feature point is used for line fitting.

In addition, numerous vanishing point detection methods have been proposed suitable for man-made environments which do not depend on any specific feature detection step like edge detection [99-100]. These approaches search for similar global structures and repeating patterns (e.g. walls, doors, windows) in the image to define the vanishing point locations. However, first, it is unlikely that one can measure such repeating structures in outdoor, unstructured environments. Next, such sophisticated approaches are inappropriate for real-time navigation systems.

To address the abovementioned drawbacks, Rasmussen [101-102] proposed texture-based methods to replace the noise sensitive edge detection step of vanishing point detection algorithms. Texture-based approaches apply a bank of oriented filters like Gabor filter banks [101-105], or steerable filter banks [106-107] and choose the orientation corresponding to maximum filter response as the dominant texture orientation $\hat{\theta}(p)$ at each pixel location $p(x,y)$. Later, each local dominant orientation votes for the location of road's vanishing points. A location with maximum votes is considered as the vanishing point of the road. Hence, precise orientation estimation of a locally oriented texture is the first and the most vital step in the texture-based vanishing point detection methods. In order to achieve precise orientation estimation one needs to apply a large number of oriented filters in all possible directions from zero to 180 degrees. For instance, to achieve a 1 degree level of precision 180 evenly spaced oriented filters should be applied. To address this problem, Freeman and Adelson [51] proposed steerable filter in which each arbitrary oriented filter can be formed by a linear combination of a fixed set of basis oriented filters. Although the steerable filter is more efficient approach compared to the bank of oriented filters, still it requires steering the basis oriented filters in all orientations with precise angle step size (e.g. 1 degree interval) and studying the outputs of the filters

as a function of their orientations to find the maximum response as the local dominant orientation.

To overcome this major drawback of the existing texture-based vanishing point detection methods, the novel opponency based LDO estimation model proposed in the Chapter 3 is used to estimate the local dominant orientations.

4.3 Vanishing Point Voting

The solution presented here is to estimate the local dominant orientations using the opponent LDO model presented in the Chapter 3. Single-scale opponent LDO model with four Gabor filters in the population $\{0, \pi/4, \pi/2 \text{ and } 3\pi/4\}$ is utilized for computational savings instead of multiscale opponent LDO model with larger population size. The single-scale opponent LDO model is applied to each pixel location $p(x,y)$ of the input image and the local dominant orientations $\hat{\theta}_{pop}(p)$ is estimated.

Once the dominant orientation $\hat{\theta}_{pop}(p)$ is estimated at each pixel location $p(x,y)$ of the image, a ray defined by $r_p = (p, \hat{\theta}_{pop}(p))$ is drawn in an accumulator space, which has the same size of the input image. Accumulator space of all pixels is initialized to zero. If a pixel lies on the ray r_p it gets incremented by 1 provided it's above the point $p(x,y)$. This is because each ray can vote only upward since the candidate vanishing point is expected to receive more votes from pixels below it than above. After rendering all the rays, a pixel with the maximum number of supporting rays, is declared as the candidate vanishing point vp_{max} [101, 105]. However, a small error in orientation of each ray may cause the rays not to intersect at one pixel location, but instead at several pixels within a small neighborhood from the true vanishing point. Hence, a Gaussian filter is applied to smooth the accumulator space and the global maximum is selected as the candidate vanishing point. Figure 4.5 demonstrates an example of the candidate vanishing point vp_{max} with supporting oriented texture voting for its location.

Figure 4.5 The upward voting, the candidate vanishing point $vp_{\max}$ is receiving supports from possible texture orientations voters.

4.3.1 Weighting Method

In the conventional voting methods [101, 105] all the estimated dominant orientations that either come from the strong edge of the road or from any random off-road texture have the same influence on the voting scheme. Therefore, it may lead to incorrect estimation of the vanishing point location. One way of preventing this effect is to assign a weight to each ray r_p . Several weighting methods such as weighting based on the strength of the resultant vector $\mathbf{z}(p)$ ($\|\mathbf{z}\|_p$), weighting based on the trigonometric function of dominant orientations ($\sin \hat{\theta}_{\text{pop}}(p)$) or various combinations of them are considered in this section. The best results come from weighting based on the trigonometric function of dominant orientations ($\sin \hat{\theta}_{\text{pop}}(p)$). The reason is that the vanishing point is expected to determine

mostly by supporting rays with range of angles closer to vertical orientations than horizontal ones. Therefore, by weighting each dominant orientation by its sin function, those horizontal or nearly horizontal orientations are suppressed. On the other hand, one possible reason that weighting based on the strength of the resultant vector $\mathbf{z}(p)$ of four Gabor energy responses yields poor result is that the strongest Gabor energy responses are usually along the strong edges (borders) of bright sky and mountains compared to much weaker responses due to road regions [104]. Figure 4.6 shows an example of a typical image from an unstructured environment where the strongest Gabor energy responses (shown in greater intensity values) are along the edges of bright sky and mountains. All of these weighting methods are evaluated statistically in section 4.4.

Figure 4.6 (a) Input image, (b)-(e) The Gabor energy responses.

4.3.2 Voting Scheme

Although the proposed weighting method resolves the problem of assigning a weight to each ray r_p , it is still more biased towards higher pixels since it allows only pixels to vote upward [104]. Hence, the pixels in the upper part of the image receive more votes than lower image pixels. In order to overcome the biases to effectively estimate the vanishing point location, two voting schemes are proposed. One way is to let each pixel vote not only in the direction of its dominant orientation $\hat{\theta}_{pop}(p)$ but also in the direction of $\pi +$

$\hat{\theta}_{\text{pop}}(p)$ called “Bidirectional voting”. Therefore, each pixel can vote upward and downward. Figure 4.7 illustrates the “Bidirectional” voting method. However, this strategy only works when both upper part of the image (e.g. mountains, trees) and lower part of the image (e.g. road borders) point to the approximately same candidate vanishing point location, which is not always true in unstructured environments. Figure 4.8 and Figure 4.9 illustrate both of these cases.

Figure 4.7 The Bi-directional voting, the candidate vanishing point vp_{max} is receiving supports from possible texture orientations voters from top and bottom of its current location in the image.

Figure 4.8 An example where Bi-directional voting with $\sin \hat{\theta}_{pop}(p)$ weighting method performs well (a) Input image with black dot showing the ground truth and white dot shows the estimated vanishing point, (b) Accumulator space.

Figure 4.9 An example where Bi-directional voting with $\sin \hat{\theta}_{pop}(p)$ weighting method fails (a) Input image with black dot showing the ground truth and white dot shows the estimated vanishing point, (b) Accumulator space.

An alternative approach is to make each pixel's vote based on its distance to candidate vanishing point. This approach implies that each dominant orientation gives higher weight to the points closer to it than to the points further away along its ray r_p . Hence, a function is defined based on the normalized Euclidean distance between each dominant orientation location $p(x,y)$ and each point on its ray r_p . This distance is first normalized by the maximum distance D_p between the pixel $p(x,y)$ and the intersection point of its ray with

the image perimeter (depends on its dominant orientation). Then the distance function is computed as:

$$y_j(\hat{d}) = e^{\frac{-\hat{d}^2}{2\sigma^2}}, \quad \hat{d} = \frac{d}{d_p}, \quad d = \sqrt{(x - x_j)^2 + (y - y_j)^2}, \quad (4.1)$$

where $y_j(\hat{d})$ is the distance function, d is the distance between dominant orientation $p(x, y)$ and any point (x_j, y_j) on its ray r_p , $\hat{d}$ is the normalized distance, and σ^2 is used to measure how fast a weight should be diminished from a dominant orientation location, which is set experimentally to 0.25. Figure 4.10 shows an example of distance based voting function that performs significantly better than Bi-directional voting in Figure 4.9.

Figure 4.10 An example where distance-based voting with $\sin \hat{\theta}_{pop}(p)$ weighting method performs well (a) Input image with black dot showing the ground truth and white dot shows the estimated vanishing point, (b) Accumulator space.

Consequently, each ray r_p with dominant estimated orientation $\hat{\theta}(p)$ is first weighted based on the sine function of its dominant orientations ($\sin \hat{\theta}(p)$) and then it is multiplied by the distance function $y_j(\hat{d})$ to give higher vote to the points closer to its origin than to the points further away along the ray r_p .

4.3.3 *Skyline Detection*

As mentioned earlier, for most road images, there is a vanishing point associated with the local oriented textures belonging to the road regions. In such outdoor unstructured environments, most of the pixels in the upper parts of the images are related to the sky pixels or random off-road regions with no strong points of convergence. Hence, in order to enhance the vanishing-point detection process, we propose to exclude sky pixels from the LDO estimation step and thereafter in the voting scheme by detecting the skyline and removing the pixels above it. Such sky pixels removal will discard parts of the image that are not related to the road and result in a significant computational speed-up.

Skyline is the line with the highest likelihood of separating the image into two regions with different appearances (ground and the sky). Due to the complex variability of appearance of either the ground or the sky in outdoor, unstructured environments, almost any deterministic skyline detection, which assumes a prior characteristic of either the ground or the sky, has proved to be futile as it is not robust to changing environments and not able to adapt to unforeseen ground cover variations.

However, one can assume that most of the sky pixels look similar to other sky pixels and less like ground pixels [108]. Therefore, the algorithm used in this work considers relative differences in appearance between the sky and the ground. The algorithm is based on the following assumption that sky regions are more likely appearing at the upper part of the image. To meet the real-time requirement, only color information is used as the feature space. However, the output color information from the camera is in RGB color space and as such color channels are highly correlated and vary drastically with illumination changes. While, the HSV (Hue, Saturation, Value) color space decouples the luminance changes from the chrominance information (hue and saturation). Therefore, the HSV color space is chosen as color representation. The proposed algorithm contains the following steps. Starting from the top of the image (row=1) and monotonically increasing the “row” with step size of one pixel, a hypothesized horizontal line is drawn to segment the image into two regions. All the pixels above this line are grouped as sky pixels and the pixels

below it as ground. Then, the mean and variance of all the pixels for these regions are computed separately for each HSV color channels. To determine the best skyline, the Mahalanobis distances between each of these means (μ) and variances (σ^2) are utilized as follows:

$$d_H^2 = \frac{(\mu_H^{\text{sky}} - \mu_H^{\text{ground}})^2}{(\sigma_H^{\text{sky}})^2 + (\sigma_H^{\text{ground}})^2} \quad (4.2)$$

$$d_S^2 = \frac{(\mu_S^{\text{sky}} - \mu_S^{\text{ground}})^2}{(\sigma_S^{\text{sky}})^2 + (\sigma_S^{\text{ground}})^2} \quad (4.3)$$

$$d_V^2 = \frac{(\mu_V^{\text{sky}} - \mu_V^{\text{ground}})^2}{(\sigma_V^{\text{sky}})^2 + (\sigma_V^{\text{ground}})^2} \quad (4.4)$$

denote d_H^2 , d_S^2 , d_V^2 is the Mahalanobis distances in H, S, and V color channels respectively. The best skyline is the line that shows maximum differences between the means of the sky and ground distributions while both regions have the lowest variance from the mean. To combine these metrics into a unify function, the Total distance function is proposed in this work as follow:

$$D_{\text{Total}} = w_\tau * d_H^2 + d_V^2. \quad (4.5)$$

Only the features H and V are selected which are also suggested in [109]. The w_τ is a weight used to balance how much to rely on the chrominance component and luminance on the skyline detection. Since the chrominance is more distinct than luminance in outdoor environments, higher weight is given to d_H^2 $w_\tau \gg 1$. Maximizing the Total distance function gives the best skyline in the image. The search region is restricted up to the half of the height of the image to fail on the safe side. Computational speed up is the main reason of adding skyline detection step in the proposed vanishing point detection method since most of the sky pixels do not participate in the vanishing point estimation. Figure 4.11 shows some examples of skyline detection drawn as red line in the images.

Figure 4.11 Examples of skyline detection—skylines are drawn as a straight line.

4.4 Results and Analyses

A series of quantitative and qualitative analysis based on experimental results are conducted to evaluate the performance gains of the best proposed method as against other techniques. The algorithms were tested on images taken from unstructured and cross-country terrains. The vanishing point estimation error was determined by comparing the result of the algorithms against the vanishing point determined manually through human perspective perception. A new metric called normalized Euclidean distance is used to evaluate the performance of the algorithms. The experimental results are analyzed using two-way analysis of variance (ANOVA) followed by a series of t-tests to compare each pairs of means.

4.4.1 *Dataset*

To determine the quality of the vanishing point estimation methods in unstructured environments experiments are conducted on images taken from unstructured and cross-country environments. The dataset used for evaluation of the proposed vanishing point detection consists of 500 images from a set of digital photographs taken on a scouting trip along Grand Challenge routes in the Mojave Desert and from six video sequences logged by the Stanford team during their trial test in the Mojave Desert [36]. The dataset contains

various types of ground covers (soil, rock, etc) with complex geometric characteristics, different illuminations and weather conditions (Figure 4.12). All the images are rectified and undistorted with the size of 320x240 and taken by forward looking camera mounted in front of the vehicle.

Figure 4.12 Representative images from our dataset

4.4.2 *Manual Ground-truth Vanishing Point Detection*

To evaluate the efficiency of the proposed method, it is important to have the ground truth vanishing point locations in the dataset. The approach used in this thesis is to ask human subjects to manually label the location of the vanishing point in each image of dataset. The similar manual vanishing point detection method is also employed by the current state of the art vanishing point detection algorithms [99, 101-105]. Although manual labeling may not be 100% perfect, it reflects the way human subjects perceive the global perspective structure of environments. In fact, the ultimate goal is to model such perspective perception found in human visual system.

In order to remove biases of individual subjects, 20 students from school of EEE at NTU are invited to manually mark the vanishing point in each image of the dataset. A GUI software known as “manual VP detector” is developed that shows an original image to subjects and ask them to find the vanishing point location of the road regions. The detail

description of the GUI manual ground-truth vanishing point detection software is explained in 0Appendix B. After collecting all the points, those points that are located far from the center of the population of points in each image discarded to remove outliers. Then, the distribution of the users mark points is considered as Gaussian with the center P_0 with (μ_{x0}, μ_{y0}) and the standard deviation $(\sigma_{x0}, \sigma_{y0})$ of the points. For comparison simplicity, the center of each ground truth location is used as the vanishing point location. Figure 4.13 shows examples of ground truth vanishing point marked by users with fitted ellipsoids.

Figure 4.13 Representative images with Gaussian fit of a set of human responses marked with ellipses.

4.4.3 Performance Metric

The vanishing point estimation error is determined by comparing the result of the algorithms against the vanishing point ground truth determined manually through human perspective perception. Previous methods either determine pixels difference between the estimated vanishing point location and the ground truth separately in both column and row [101-102, 105] or use one metric like the Euclidean distance in pixels [103-104] to show the quality of the vanishing point estimation. However, when different methods utilize different image resolutions, the error distance in pixels is not representative of true performance of each method. For efficient quantitative evaluation measurements, normalized Euclidean distance is presented, where the Euclidean distance is normalized by the size of the diagonal of the image resolution as follows:

$$\text{NormDist} = \frac{\|P - P_0\|}{\text{Diag Image}}, \quad (4.6)$$

where $P(x_p, y_p)$ is the estimated vanishing point location and $P_0(x_0, y_0)$ is the center of the ground truth, while “Diag Image” is the diagonal of the image. Using the proposed metric, a value near 1 would correspond to incorrectly estimated vanishing point location. A value near to 0 means estimated vanishing point location is close to the location of ground truth.

4.4.4 Experimental Parameters

Two main groups of experimental parameters considered in this study, which can affect the performance of the proposed method, are voting schemes and weighting methods described in section 4.3. Table 4.1 depicts these two groups and their associated methods. They are summarized in the following sections.

TABLE 4.1 TWO MAIN GROUPS OF EXPERIMENTAL PARAMETERS IN THIS STUDY

Group	Method			
Weighting Method	Unweighted (No. of Supporters)	$\sin \hat{\theta}_{\text{pop}}(p)$	Mag	$\text{Mag} * \sin \hat{\theta}_{\text{pop}}(p)$
Voting Scheme	Upward	Bidirectional	Distance based Upward	Distance based Upward with Sky Removal

Voting Scheme

This group of parameters aims to test the performance impact of different voting schemes. Following are the four voting methods:

1. Upward: The conventional method of voting scheme which allows each pixel vote only for all the pixels above its current location in the image plane in the direction of its dominant orientation $\hat{\theta}_{\text{pop}}$, $0 \leq \hat{\theta}_{\text{pop}} \leq \pi$. The voting scheme proposed by Rasmussen [105] is belong to this method.

2. Bidirectional: Each pixel can vote not only in the direction of its dominant orientation $\hat{\theta}_{pop}$ but also in the direction of $\pi + \hat{\theta}_{pop}$. Therefore, each pixel can vote upward and downward.
3. Distance based Upward: This method implies that each pixel gives higher weight to the points closer to its location than to the points further away along its ray r_p while it points upward.
4. Distance based Upward with Sky Removal: This method differs from the previous method, in such a way that the pixels above the skyline, which may belong to the sky regions are removed from the image before finding the dominant orientation.

Weighting Method

Weighting method considers different ways of assigning a weight to each ray r_p versus a simple Unweighted “No. of Supporters” method.

1. Unweighted (No. of Supporters): If a pixel lies on the ray r_p it gets incremented by 1. After rendering all the rays, a pixel with the maximum number of rays passing through it, is the candidate vanishing point vp_{max} . The Unweighted “No. of Supporters” method first described by Rasmussen [105].
2. $\sin \hat{\theta}_{pop}(p)$: Assigning a weight to each ray r_p based on the trigonometric function of its dominant orientation ($\sin \hat{\theta}_{pop}(p)$).
3. Mag: Assigning a weight to each ray r_p based on the strength of the resultant vector $\mathbf{z}$ ($\|\mathbf{z}\|_p$).
4. $\text{Mag} * \sin \hat{\theta}_{pop}(p)$: Assigning a weight to each ray r_p based on the strength of the resultant vector $\mathbf{z}$ multiply by the sine function of its dominant orientation ($\sin \hat{\theta}_{pop}(p)$).

4.4.5 *Experimental Results*

In this section, the performance of proposed methods have been evaluated both quantitatively and qualitatively. In order to decrease the computational demand to meet the real-time requirement of vehicle navigation system, all the images in the dataset first are downsized to 80x60 pixels by Gaussian pyramid and then passed to vanishing point detection methods to find the location of vanishing point. Once the vanishing point location is estimated its location is projected back to the original image size (320x240) to assess the algorithm's performance with human perception ground truth.

The performances of all methods scored in term of normalized Euclidean distance (NormDist) are shown in Table 4.2. It may be observed that assigning a weight based on the sine function of the dominant orientation ($\sin \hat{\theta}_{pop}(p)$) for each ray outperforms other weighting functions for all variations of the voting schemes. The best overall results are obtained with the distance based upward voting weighted by $\sin \hat{\theta}_{pop}(p)$ with sky removal. However, Table 4.2 does not quantify statistical significance of the difference between different methods of each group, hence; a two-way analysis of variance (ANOVA) technique is used to determine the main effects of voting schemes, and different weighting methods.

Analysis of Variance (ANOVA)

This section provides a study on analysis of variance of two groups of parameters in Table 4.1, the weighting methods and voting schemes, which are employed in the experiments. For the purpose of comparison, the results of ANOVA are presented in Table 4.3. It can be observed from the table that the p values are significantly smaller than a confidence level of 95%. This confirms the existence of different effects due to the different methods. A closer look at the table yields the following interpretations:

1. The F value related to the rows states the existence of a significant difference between the voting schemes,

2. In a similar way, significant differences are suggested to be there between the weighting methods, and
3. The small ρ value attained for the interaction indicates that there is little evidence that the difference between the weighting methods depends on the voting schemes.

TABLE 4.2 OVERALL ALGORITHMS PERFORMANCE (NORMDIST) OF ALL METHODS

		Weighting Method			
Voting Scheme		Unweighted	$\sin \hat{\theta}_{\text{pop}} (p)$	Mag	$\text{Mag} * \sin \hat{\theta}_{\text{pop}} (p)$
	Upward voting	0.1062	0.0556	0.1341	0.0771
	Bidirectional Voting	0.0771	0.0497	0.0712	0.0632
	Distance based Upward Voting	0.0576	0.0389	0.0808	0.0604
	Distance based Upward Voting with Sky Removal	0.0428	0.0365	0.0570	0.0485

TABLE 4.3 ANOVA RESULTS

Source	SS	df	MS	F	ρ
Columns	1.7277	3	0.57591	63.17	0.000
Rows	2.3579	3	0.78598	86.21	0.000
Interaction	0.7983	9	0.0887	9.73	6.1062e-015
Error	72.7861	7984	0.00912		
Total	77.67	7999			

Statistical hypothesis test

In this section, statistical hypothesis tests are carried out for pairs of the methods mentioned in Table 4.3. First, first row in Table 4.3, the upward voting scheme is compared with the best proposed method (i.e. last row in Table 4.3). (Distance based Upward Voting with Sky Removal). For this purpose, a matched-pairs t-test has been organized to find the equality of the means of the two groups. The t-test result shows that the two sided ρ -value is significantly smaller than 0.05. The null hypothesis is rejected

and it is accepted that the two methods are significantly different. The mean normalized distance error for the proposed method (0.0462) is better than the upward voting (0.0933).

Next, it is tested whether the sky removal results in any significant performance difference. Therefore, a matched-pairs t-test between the Distance based Upward Voting with Sky Removal (fourth rows in Table 4.3) and the Distance based Upward Voting (third rows in Table 4.3) are conducted. The t-test result shows that the two sided p -value is significantly smaller than 0.05. Therefore, the null hypothesis is rejected and it is accepted that the two methods are significantly different. The mean normalized distance error for the Distance based Upward Voting with Sky Removal (0.0462) is better than the Distance based Upward Voting (0.0594).

Finally, a series of pairwise matched-pairs t-tests between the different weighting methods (columns in Table 4.3) are conducted to determine which pairs of means are significantly different, against the hypothesis that they are all the same. The t-test results show that in all cases the two sided p -values are significantly smaller than 0.05. Therefore, the null hypothesis is rejected in all the cases and it is accepted that all the weighting methods are significantly different. Specially, the mean normalized distance error for the sine function (0.0451) is better than the others. Thus, based on combined results of the hypothesis tests, it is concluded that the optimum performance can be obtained by using distance based upward voting with sky removal combined with $\sin \hat{\theta}_{\text{pop}}(p)$ weighting method.

4.4.6 *Comparison with State of The Art and Representative Outputs*

For quantitative comparison purposes, the best proposed method (the distance based upward voting weighted by $\sin \hat{\theta}_{\text{pop}}(p)$ with sky removal) is assessed relative to the classical edge-based (Canny/Hough) vanishing point detection [95] method and the two best known texture-based vanishing point detection algorithms for unstructured road conditions proposed by Rasmussen [105], and Kong et al. [104]. Kong et al. [104] extend the textured-based vanishing point detection method proposed by Rasmussen [105] and provide good results for different road conditions. Their implementation includes two

steps. One is the confidence-rated texture orientation estimation, and the locally adaptive soft voting scheme. They define a confidence value for each dominant orientation based on how the Gabor filter response values were scattered around the maximum filter response (i.e. dominant orientation). Subsequently, the low confidence pixels are discarded from final voting scheme. Next, to reduce complexity; they introduce a fixed local voting region for each candidate vanishing point, which is the half disk below each candidate vanishing point with radius of one third of the image height.

To ensure a fair comparison, the source code provided by Kong et al. [104] is used with the same parameters mentioned in their implementation [104], and tested with our 500 image dataset. The method implemented by Rasmussen [105] belongs to the upward voting scheme with unweighted (No. of Supporters) method described in section 4.2. Table 4.4 shows the average normalized Euclidean distance error (NormDist) for the 500 images across all methods. As shown in the Table 4.4, the proposed method in this thesis outperforms all methods significantly. As it is expected, the edge-based method is not suitable for clutter unstructured environments, where there is no clear edges and lines to extract.

For better comparison, the results of the best proposed method (the distance based upward voting weighted by $\sin \hat{\theta}_{\text{pop}}(p)$ with sky removal) versus edge-based method, Rasmussen method [105] and Kong et al. method [104] are evaluated on the 500 images dataset in a 11-bin histogram.

TABLE 4.4 THE ACCURACY COMPARISON

	Edge-based method [95]	Rasmussen [105]	Kong et al. [104]	Our proposed method
Error (NormDist)	0.1710	0.0903	0.052	0.036

In the histogram, the distance error of 0.1 is considered as too large error therefore, if the normalized Euclidean distance between the estimated vanishing point location and the ground truth is larger than or equal to 0.1, it is put into the last histogram bin as it is shown in Figure 4.14. The vertical axis shows the number of images in each histogram bin, and the horizontal axis shows the normalized distance error (NormDist). A larger number of images in left part of the histogram depicts better estimation results while the larger value in the right part shows worse cases. The advantage of the proposed method is that it shows the greatest number of images in the left part of histogram (i.e. low error rate) while it has the smallest number of images with large errors at the right side of the histogram.

In particular, in Rasmussen method [105] error rates of more than 142 images (28%) are greater than 0.1 where in the proposed method this number is less than 36 images (7%). Besides, Kong et al. method [104] shows 56 (11%) images with large error, which is better than Rasmussen method [105], while its performance is degraded in the small error portions of the histogram. The NormDist error less than 0.01 in Kong et al. method [104] occurs in only 63 images (12%). In contrast, the proposed method performs significantly better in which 163 images (32%) have the normalized distance error (NormDist) of less than 0.01. Moreover, the edge-based method [95] performs worst compared to texture-based methods, where over 335 images (67%) have the error greater than 0.1.

For qualitative evaluation, Figure 4.15 depicts a variety of images that the proposed method estimates the vanishing point location successfully in many challenging road and illumination conditions while some or all of the state of the art methods have difficulties. Note that, in curved road, the main vanishing point associated with the tangent direction of the most immediate road regions in front of the vehicle is estimated. Figure 4.15 **Error! Reference source not found.** (a) illustrates the results of edge-based method [95]. As you can see most of the time, the edge maps (Canny operator) are very noisy and show poor results. The Rasmussen method [105] are more biased towards higher pixels since it only allows upward voting Figure 4.15 (b). Kong et al. method [104] shows better results than the other while still it has problems with cast shadows in the road regions Figure 4.15 (c).

Figure 4.14 Comparison of vanishing point detection methods in a 11-bin histogram.

Although the algorithm presented in this work is proposed for unstructured road environments, it works perfectly for structured road (straight or curved) as well. Because the structured roads contain well lane markings and road boundaries, the vanishing point estimation is much more easier. Figure 4.16 illustrates some examples of applying the proposed method to structured road condition. As demonstrated, the proposed method accurately estimates the vanishing point in structured environments as well or better than unstructured ones. For more qualitative comparison purpose, we find that the recent work by Kong et al. [104] reported some failed cases caused by extreme illumination conditions or when the vehicle goes up and down steep slopes. Figure 4.17 shows these images and corresponding accumulator spaces from [104]. The first two rows are the results based on the best proposed method in this work and last two rows are from [104].

Figure 4.15 Comparison of vanishing point estimation methods. (a) Edge-based method: first row is the estimated vanishing point, second row is the Canny edge map. (b) Rasmussen method [105]: first row is the estimated vanishing point, second row is the accumulator space. (c) Kong et al. method [104]: first row is the estimated vanishing point, second row is the accumulator space (d) the best proposed method: first row is the estimated vanishing point (red line showing the estimated skyline), second row is the accumulator space.

Figure 4.16 Examples of vanishing point detection for structured road. First row and third row: outputs of the proposed algorithm (black dot is the ground truth and white dot is the estimated vanishing point location and red line is the estimated skyline); second row and fourth row: accumulator space.

The proposed algorithm estimates vanishing point locations correctly for three out of four of the images while the method proposed in [104] fails in all of four images. A possible reason that the proposed method cannot estimate the vanishing point in one of the images is most likely because there is not enough supporting texture from the left part of the image for the vanishing point location while the vehicle is turning. More results are illustrated at Appendix C.

Figure 4.17 Examples of failed vanishing point detection from [104]. First row: output of the proposed algorithm (black dot is the ground truth and white dot is the estimated vanishing point location and red line is the estimated skyline); second row: accumulator space of the proposed algorithm; third row: output of method in [104] (red dot is the estimated vanishing point); fourth row: accumulator space of method in [104].

4.5 Chapter Summary

In this Chapter, a new computational model for the vanishing point estimation based on empirical studies on the visual gaze behavior of drivers in open driving scenarios using only monocular vision was presented. First, the most important findings and their corresponding analyses in the drivers gaze pattern behaviors were reviewed and it is found out that when driver become more familiar with the road situations, they prefer to look at the far field and close to the end of the road to anticipate the upcoming road trajectory and the car steering. By increasing the driving speed, their gaze distributions are concentrated

more on the end of the road where they are able to obtain more visual information about the upcoming road trajectory with maximal lead time.

Therefore, a new methodology for road direction detection based on vanishing point estimation for outdoor, unstructured road conditions was presented in this chapter. The algorithm employed the LDO estimation mechanism proposed in the Chapter 3, which is ideally suited for real-time applications, to detect the locally oriented patterns such as road edges, ruts and tire tracks left by previous vehicles. After dominant orientations are estimated, in order to estimate the vanishing point location a ray is drawn along each dominant orientation and the pixel with the maximum number of supporting rays is declared as the candidate vanishing point. However, in such method all the dominant orientations would have the same influence on the voting scheme. To address this problem assigning weights to each dominate orientation were considered based on different weighting methods. Although the proposed weighting method resolves the problem of assigning a weight to each dominant orientation it is more biased towards higher pixels since it allows only pixels to vote upward. In order to overcome the biases, several voting schemes are proposed.

The influence of different weighting functions and different voting schemes were studied using two-way analysis of variance (ANOVA) followed by a series of t-tests. The performances of all methods were scored in terms of the normalized Euclidean distance metric. It was shown that assigning a weight based on the trigonometric function of the dominant orientation ($\sin \hat{\theta}_{\text{pop}}(p)$) for each ray outperforms other weighting methods for all variations of the voting schemes. On the other hand, the best voting scheme results are obtained with the distance based upward voting with sky removal.

Furthermore, series of quantitative and qualitative analyses were conducted using real data sets from the DARPA project. The performance in terms of accuracy of the proposed method outperforms the state of the art vanishing point detection methods.

Chapter 5

SELF-SUPERVISED LEARNING APPLIED TO UNMANNED GROUND VEHICLES

5.1 Introduction

As it is discussed in the previous chapter, when drivers become more familiar with the road environments they often direct their attention towards to the end of the road where the road edges converge to anticipate the upcoming road trajectory. However, when human drivers for the first time navigate in a new, unseen road situation, their gaze behaviors are completely different according to several empirical studies on human gaze patterns during navigation [110-111]. Psychologists believe that when people are driving a car along a new, unknown route, they spend more time to explore the situation in their near field than when they have seen the environment before. Wagner et al. [112] was first to study the gaze behavior of walking humans in an outdoor environments. They showed

that humans direct their gazes frequently to the local environments to extract useful information from nearby terrains and adapt to unexpected changes in the environments.

On the other hand, human occasionally need to attend to far field terrain and predict the upcoming road trajectory for better goal-directed navigation task. In fact, humans can navigate safely through an environment by avoiding obstacles in the nearby while they are able to efficiently predict the upcoming road for planning a path towards a goal. Indeed, human drivers often link these two processes by incorporating knowledge of near field into far field in a universal framework. Humans have the ability that can continuously learn the association between terrain geometry characteristics and appearances in the near field and then, use such information to make predictions about the upcoming road to the far field of its visual field.

This superior navigation behavior of humans encourages several researchers to develop intelligent autonomous Unmanned Ground vehicles (UGVs) that can operate safely and effectively in unknown unstructured environments. Most of the current UGVs typically use range finding sensors such as a stereo vision system or LIDAR (Light Detection and Ranging) to sense the nearby environments and predict the traversability of the far field terrains based on the essentially learned knowledge in the near field [19]. Such learning mechanism is called self-supervised, near-to-far learning [18].

Self-supervised, near-to-far learning was successfully employed by the winner of 2005 DARPA Grand Challenge [20]. They used LIDAR sensors to detect the traversable regions in the near field and then employed monocular camera to learn the appearance of the nearby regions by a Mixture of Gaussians model. Such information later was used to predict the far field terrains. However, the near-to-far classifier was not reliable enough to be used for autonomous navigation system, thus, it was applied only for speed control system during 2005 DARPA Grand Challenge [20].

Recently, self-supervised, near-to-far learning has been addressed by several research groups in the DARPA-LAGR program [18] by using stereo vision systems to learn the

geometry and appearance of the traversable and non-traversable regions in the near field and then classifying the far field terrain based on the near field knowledge. Although there are many advantages arising from the use of previous self-supervised, near-to-far learning algorithms, they still pose several limitations that are addressed and resolved by the new Self-Supervised Learning framework proposed in this chapter.

Hence, the primary contribution of this chapter is to propose a new Self-Supervised Learning (SSL) framework for autonomous Unmanned Ground vehicles (UGVs) that is composed of a robust ground plane and obstacles estimation, confidence-based auto labeling, top-down attention model, and an adaptive online classifier.

This chapter is organized as follow. Section 5.2 discusses in details each module of the baseline self-supervised, near-to-far learning method. The proposed SSL framework is described in section 5.3. In section 5.4, a series of extensive qualitative and quantitative experimental results are presented to demonstrate the effectiveness, robustness, and accuracy of the proposed learning framework using the natural datasets logged live during runs of vehicles competing in the DARPA-LAGR program. Section 5.5 summarizes the proposed SSL framework.

5.2 Related Work

As mentioned earlier in section 5.1 the term “self-supervised, near-to-far learning” is exploited by several teams in the DARPA-LAGR program including API [113], JPL [114], SRI [115] NY and Net-Scale [116] and NIST [117] to address the shortcomings of the existing supervised and unsupervised learning methods applied to autonomous UGVs.

The self-supervised, near-to-far learning is an approach of generating near field traversability labels from each incoming pair of stereo images using only the near field stereo information. Stereo vision information is used to find the ground plane (traversable region) and obstacles (non-traversable region) in the near field of the vehicle. A

traversability label is assigned to each pixel in the image based on the stereo information in the near field. Once the ground plane and obstacles are estimated in the disparity image, their corresponding image pixels in the reference image (right image) are labeled as either traversable point (negative class -1) or non-traversable point (positive class +1). These labels are then associated with their visual appearance features to train a classifier, and the model obtained is used to classify the far field terrain well beyond the stereo range. Later, this model is discarded. Therefore, one full training and classification cycle is completed on every incoming pair of stereo images [118]. Every processing cycle of the baseline scenario of self-supervised, near-to-far learning involves four main steps. (1) Obstacles and ground plane estimation (2) Feature selection and extraction (3) Classifier training and model generation (4) Applying that model to the entire input image and subsequently discarding it. The block diagram of the algorithm structure for baseline self-supervised, near-to-far learning is shown in Figure 5.1. Each of the steps of this scenario is discussed in detail in following sections. We also summarize efforts carried out by different teams in LAGR phase I and phase II to address each step of self-supervised, near-to-far learning that are similar to the context of this thesis.

Figure 5.1 Block diagram of the algorithm structure for baseline self-supervised, near-to-far learning.

5.2.1 Vision-Based Obstacles Detection and Ground Plane Estimation

A pair of stereo color images is used as an input to the vision-based obstacles and ground plane estimator module. First, rectification is performed on the left and right images. Rectification is the process of correcting a pair of images whose epipolar geometry is

known so that epipolar lines in the original images map to horizontally aligned lines in the transformed images to remove lens distortion. Image rectification simplifies the search space for finding correspondences from a two dimensions space to one dimension. Then, stereo matching is looking for corresponding point in the left and right rectified images. The relative displacement between each point and its correspondence in each image is known as disparity that is used to generate a dense disparity map. The 3D information of the scene can be derived if the disparity map is known. If the baseline of the stereo camera is B (meter), and the camera focal lengths is f (pixels) then the point (u, v) from the centre of the image (u_c, v_c) on the disparity map with disparity d represents a point in the environment (scene) with coordinates:

$$z_d = \frac{fB}{d}, \quad (5.1)$$

$$x_d = \frac{u-u_c}{f} z_d, \quad (5.2)$$

$$y_d = \frac{v-v_c}{f} z_d, \quad (5.3)$$

There are two principal approaches to ground plane and obstacle detection using stereo vision. Those that find the objects and ground plane directly from the disparity map (u, v, d) [119-120], and those that first transform the disparity to 3D point clouds (i.e. range image (x_d, y_d, z_d)) and then employ depth and height from range image to extract ground plane and obstacles [121]. Each pixel of a range image expresses the 3D position of a visible point in the scene directly. Most of the DARPA-LAGR research groups exploit range image to estimate ground plane and obstacles within the 3D point cloud. They apply Hough Transform [116] or RANSAC (Random Sample Consensus) [113] to fit ground plane model to 3D point cloud from range image. After ground plane is estimated in the near field of the vehicle, obstacles or non-traversable regions is determined by height analysis. Any point in the 3D range image that its height lies above the fitted ground plane but lower than the vehicle's height is considered as a potential obstacle. However, working on 3D point clouds is not reliable due to the non-linear transformations from disparity image to 3D point clouds (equation (5.1)).

Matthies et al. [121] address obstacle detection using a slope estimation algorithm applied to columns of the range image. They assume an obstacle as a near-vertical step displacement of minimum height H in the range image. Their algorithm measures changes in height between every pixel (p_1) in the range image and its next pixel (p_2) in the same column number. If the measured change exceeds a given pre-defined threshold, there is an obstacle in the entire interval of the range image from p_1 and p_2 otherwise it is a flat ground plane.

Hadsell et al. from NY and Net-Scale LAGR team [116] estimate the ground plane and obstacles within the 3D point cloud. They use an elaborate two-stage process for finding ground plane. First, a Hough transform is used to fit a single ground plane model to 3D point cloud from range image. For simplicity, they pre-compute all possible flat ground planes parameters $[a, b, c]$ and quantized each of them into 64 bins. The “ c ” (yaw) is set to “1” so the space exploration is only in the three dimensions of pitch, roll, and offset. The pre-computed parameters is set based on the prior information about the types of the terrain vehicle may encounter and the dimension of the vehicle.

After a plane is fitted to the 3 points, the next step is that points in the cloud vote for each of the candidate planes if they are within a defined threshold. After the voting is done, a plane with maximum number of votes is selected as a ground plane. The next step is to use PCA to refine the ground plane parameters by computing eigenvalues decomposition of the covariance matrix of the set of inliers that are within a threshold of the ground plane.

They extend the ground plane fitting by extracting multiple planes for each frame based on the assumption that in outdoor environments the ground plane is nonplanar surface. After the first plane is fitted, all points that are not within a fixed threshold of the first plane are utilized to find second ground plane in one frame. This process is continuing until either no valid point can be found, or there are not enough points to fit a plane, or a maximum number of three planes have been found. Their short-range ground plane fitting is limited to 10 meters. One disadvantage of their method is that they use 3D point clouds,

which are not reliable due to the non-linear transformations from disparity image to 3D point clouds. Another disadvantage of their method is that they use Hough transform for ground plane fitting to 3D point clouds, which is very sensitive in existence of noisy data. On the other hand, in the initial step they pre-compute the parameters of the ground plane in front of the vehicle based on some prior information about the scene which limits their method.

The ground plane and obstacle detection module of API LAGR team [113] also employs 3D range data. They apply RANSAC technique to 3D point cloud that fall between two planes (15-degree angles above and below a flat ground plane) emanating from the center of front wheels on the ground of the vehicle. Pixels with in an image are labeled as ground plane if their corresponding 3D points are within 10 cm of the fitted plane or they are labeled as obstacles if their corresponding 3D points are more than 20 cm above the fitted plane. The points between 10-20cm are marked as unknown.

Konolige et al. [115] from SRI LAGR team use both disparity and 3D point clouds to estimate ground plane and obstacles from stereo information. They use RANSAC technique on the subset of 10000 3D points in 6m around the vehicle to define the ground plane. Then, they apply height analysis to find obstacles that lie above the current ground plane but lower than the robot's height. They further extend this method to find obstacles in the disparity image. Once the ground plane is detection, it is projected back onto the disparity points. Further, the image is divided into several thin columns and each column is searched from bottom to top to find obstacles. When the ground plane ends in a column and enough disparity value differences are found, it is concluded that there must be an obstacle at that point. This technique of sight lines helps to yield the free space to more distance in front of vehicle.

Another group of methods use directly disparity image to estimate ground plane and obstacles. Disparity map allows us to work with data produced directly from the sensor (stereo camera system), obviating the need for intrinsic camera parameters whilst improving robustness. A simple and real-time method known as “v-disparity” is

introduced by Labayrade et al. [120]. The v-disparity image is computed by generating a histogram of disparity values on every row of the disparity image. Since it is reasonable to assume that a majority of the points result from the ground plane, the histogram bin with maximum number of pixels is assumed to correspond to the ground plane. The road surface is estimated as a 2D straight line in v-disparity image using the Hough transform instead of 3D plane estimation in the disparity image. In some cases such as where obstacles occupy the most part of the image, or outdoor environments where in the presence of a slope variation the 2D line representing the ground plane is not a straight line, the v-disparity fails to extract a ground plane correctly.

In contrast, model based ground plane detection algorithms use a geometric model of the plane to find the ground plane in front of the robot and to detect obstacles. This approach models the ground plane based on the disparity map in image coordinates (u, v) with disparity d . In this case, ground plane equation takes the linear form of

$$au + bv + c - d = 0 \quad (5.4)$$

where a , b , and c are the ground plane parameters and (u, v) are the image coordinates and d is the disparity value at pixel location (u, v) in disparity image. Ferrari et al. [119] use this linear approach by predefining ground floor disparity values and do a fast comparison technique between current disparity value and a reference disparity map to detect obstacles and ground plane. One drawback of this method is that each time it needs a free space in front of the robot for initialization for finding the predefined ground floor disparity values. Therefore, it is not adaptive to unknown environments.

The ground plane detection module of JPL LAGR team [115] takes directly disparity image as an input, and uses fast RANSAC algorithm to find dominant ground plane. Once the ground plane has been detected, height of each pixel above the ground plane in the disparity map is computed, resulting in an elevation image. They use only 10 meters in front of the robot for ground plane estimation. Their ground plane module is the most

similar method to our proposed ground plane estimation technique, which is described in section 5.3.2.

Colorado LAGR team [122] also tries to fit a flat ground plane model to the disparity image. They initially assume a ground plane in disparity image corresponds to a flat model relative to the known height of camera. First, each pixel image is projected to the world coordinate using nominal transformation matrix and the known height of camera given by the vehicle calibration. Next, they utilize the nonlinear transformation (equation 5.1) to translate the calculated distance z_d to disparity d . Therefore, they can predict the disparities related to the flat ground plane in front of vehicle. This initial estimation of the ground plane disparity is updated at each disparity frame by fitting disparities weighted by their distance to the current ground plane estimation. Further, obstacles can be defined by measuring the differences between the current predicted ground plane and observed disparities. Although they declare that they use the disparity image to estimate the ground plane and obstacle, they still use the nonlinear transformation from distances z_d of ground plane to the disparity values in their model. Table 5.1 summarized the ground plane and obstacle estimation modules of LAGR teams.

TABLE 5.1 THE GROUND PLANE AND OBSTACLE ESTIMATION MODULES FROM LAGR TEAMS.

LAGR Team	Domain	Method	Extra assumptions
API [113]	3D point clouds	RANSAC	Flat ground plane within ± 15 degrees of vehicle flat front plane
NY/Net-Scale [116]	3D point clouds	Hough transform+ PCA	Possible ground planes parameters are fixed
SRI [115]	3D point clouds+ disparity	RANSAC	Disparity uses only for sight lines detection
NIST [117]	3D point clouds	column wise scanning	Points immediately in front of the vehicle are considered as ground
Colorado [122]	3D point clouds+ disparity	ground plane modeling	Ground plane found using 3D data then mapped to disparity image. Used the vehicle calibration parameters
JPL [114]	disparity	RANSAC	Generate elevation map from disparity image

5.2.2 *Feature Extraction*

The principal objective of feature extraction in terrain classification is to select the optimum d-dimensional representation of a given image that contains as much as possible useful data from the region of interest in optimal and efficient computational time. Once an image is captured, the image is divided into several primitive units. These primitives are mostly referred to as pixels or patches (fixed-size window) and recently *superpixels* [123]. The pixel-based primitive is a patch with a window size of one. Pixel-based primitive is subject to the noise and increase the computational times in the real-time outdoor navigation systems. Patch-based primitives are more suitable for the outdoor feature extraction since each patch is constructed of several pixels by placing a fixed window size centered on a given pixel and therefore may reduce the noises of a pixel-level. For each primitive centered on an image pixel, a list of descriptors known as a feature vector is extracted. The depth (i.e. the number of feature values) of a feature vector is another important parameter that needs to be considered in real-time applications.

There are many possible features that can be extracted from traversable and non-traversable regions and finding optimal features is an open research area now [124]. It is possible to analyze frames in spatial domain, frequency domain or even mixed time-frequency domains such as wavelets. The term spatial domain refers to direct manipulation of pixels in an image. Frequency domain processing techniques are based on modifying the Fourier transform of an image. Wavelet domain processing is based on representing an image in multi-resolution processing such as Gabor wavelets. Features can be from different color spaces (RGB, HSV, and LAB), texture descriptors, and/or different grayscale analysis such as edge or line detection. In the following subsections the most well-known color and texture features and descriptors are reviewed followed by a summary of related feature extraction techniques carried out by the LAGR teams.

Color Features

Color feature vectors are generated directly by the sensor for each pixel as opposed to other features, such as texture or optical flow, which usually require complex preprocessing. Color provides more information than intensity alone. Grayscale based analysis (Edge detection) usually fails because the road does not have specific structure format. Most of the DARPA-LAGR participants perform their classification based on color features of different color spaces. There are many different ways to represent color, and these color representations or models can be transformed from one space to another. The most common color spaces which are good enough to describe all the ranges of colors are RGB (red-green-blue), HSV (Hue, Saturation and value) and L^*a^*b (“L” stands for Luminance, while “a” and “b” represent the two color areas).

However, color has several well-known problems. The main challenging one is that camera only perceives apparent color of a surface, which is a function of light source, the reflection of the object, illumination geometry, and relative geometry of the illumination and the camera properties. One typical partial solution is to use brightness normalization, which transforms the color vector RGB into a normalized rgb. Another shortcoming of using color information is its un-usability under low-light conditions, especially at night.

Texture Features

Visual texture analysis has been a subject of research in computer vision and image processing communities for over several decades now; however, a unified and clear definition of a visual texture cannot be found in related contexts. Due to the unclear definition of visual texture, numerous texture features have been identified and proposed in the literature to characterize perceptual texture information [37-38]. Examples of visual texture features include smoothness, roughness, coarseness, contrast, directionality (orientation) and regularity. What is in common in all of them is that the visual texture is the characteristic of local neighborhood of pixels compared to color features, which are the pixel level image information. On the other hand, Visual texture is defined on gray image. A variety of methods has been successfully employed to extract texture information. These methods can be divided into three main categories: statistical (or

stochastic), model based, and filtering based methods. In the statistical or stochastic method, texture features are extracted from region with statistics properties of grey levels and positions of pixels. Numerous statistical texture features have been identified and proposed in the literature to characterize perceptual texture information. Among those, co-occurrence matrix and LBP (Local Binary Pattern) have been widely used. Model-based approaches try to fit a model (Markov random fields (MRFs)) to local neighborhood pixels of an image. The model parameters are then utilized to represent texture context of the image. One can adopt various models with varieties of neighborhood shapes and sizes to model texture information.

TABLE 5.2 APPEARANCE FEATURES USED BY LAGR TEAMS

LAGR Team	Color features	Color descriptors	Texture features	Texture descriptors	Extra features	Feature dimension
NIST [117]	R/G, G/B, intensity	Normalized 8-bin histograms	LBP	Normalized 8-bin histograms	Contrast - single value	33
JPL [125]	r, g	2D normalized 16*16 histograms	NA	NA	physical bumper-single value	257
SRI [115]	CIE Lab	Center pixel values	Gradient intensities	8 Surround pixel values	NA	11
API [113]	r, g, intensity	64-bin histograms, 16-bin histograms	NA	NA	8-dimension geometry features	152
Colorado [122]	R, G, B	5-bin histograms	NA	NA	NA	15

Finally, the texture analysis based on filtering methods has been investigated more recently. Such methods employ set of filters (i.e. filter banks) to transform the image into components having different spatial properties. Filtering can be applied in different

domains such as spatial domain, Fourier domain, or wavelet domain. The well-known filtering based texture extraction techniques are Discrete Cosine Transform (DCT), Wavelet Transforms and Gabor filters. Recently, Gabor filter bank is attracted more attention. A typical Gabor filter bank contains multiscale multi-orientation filters between 0 and 180 degree. The Gabor filter technique is described in details in Chapter 3.

Feature extraction and selection is also one of the fundamental parts of self-supervised, near-to-far leaning methods proposed by the DARPA-LAGR teams. JPL LAGR team [125] divides the scene into four zones: underfoot, near field, mid-field, and far field and extracts different features in each zone. In the underfoot, the terrain traversability is the only available data from proprioceptive sensors such as IMU, physical bumper, wheel encoders. Although they mention the benefits of using all these sensors, they only use physical bumper to provide 0/1 quantity in order to assess traversability of underfoot. In near field both height and color statistics are used in each cell with the size 20cmx20cm. The height statistics consist of the number of points, mean, and variance of points, as well as a maximum step height, which is the maximum difference between the mean height of a cell and it neighbors. The color statistics are the mean and variance of the RGB channels.

In mid field and far field the appearance from camera is the only sensory information. They use 2D normalized color histograms of r-g with 16x16 bins and the window size of 33x33 computed using $r = \frac{R}{(R+G+B)}$ and $g = \frac{G}{(R+G+B)}$. Next, they improve their feature extraction technique by using a grid of adaptive and uneven bins. They normalize each of the axes of 2D histogram so that the same numbers of pixels fall in each bin. This technique removes all zero values from the histogram and makes the feature vector more robust and adaptive to outdoor environments.

Konolige et al. [115] (SRI) implement very compact feature vector. They use the pixel intensity gradients in the neighborhood of the center pixel of a window dimension (3x3)

to represent texture information. They also add the three color channels from the CIE Lab color space to the center pixel. The result is an 11-dimensional color/texture vector.

API LAGR team represented by Happold et al. [113] describes a combination of color and geometry in single feature vector. Eight geometric features, for each cell (20cmx20cm) are extracted such as Height Variation, Terrain Slope and Mean Vertical Distance. Color feature vector consists of rgI color space. r and g components are the normalized red and green values from RGB image space and I is the intensity components ($I = \frac{R+G+B}{3}$). The r and g represent by the 64 bins histograms while intensity is described by 16 bins histogram.

Shneire et al. from NIST LAGR team [117] represent color by ratio R/G , G/B and intensity (Intensity is calculated as $0.229R + 0.587G + 0.114B$) rather than directly using RGB to provide some degree of protection to variable lightning conditions and shadows. Each color channel ratio is quantized by an 8-bin histogram in a normalized form. Local Binary Patterns (LBP) is also computed as texture feature extractor over a 3x3 windows and like the color ratios, the texture measure is normalized and stored in an 8-bin histogram. Contrast computed over the same window size is also added to the feature vector. The dimension of the feature vector for each pixel is 33.

Procopio et al. [122] from Colorado LAGR team use appearance-based feature (color-histogram) in their learning methods. The three color channels (R , G , and B) in a fixed sized patch are selected to generate 1D color histograms. The fixed number of bins (here 5 bins) are results in a feature image with feature depth of 15 values (three channel x five bins per channel).

The appearance features used for feature extraction module of several teams participating in LAGR projects are summarized in Table 5.2.

5.2.3 Classifier Training

Once the visual appearance features are associated with the traversability label provided by stereo supervision for each primitive, they are passed to a base classifier to learn the models of the traversable and non-traversable regions. This is a standard two-class classification problem, where the two classes are traversable (negative class -1) and non-traversable (positive class +1). The base classifier learns on the training set provided from only the near field of the vehicle. Therefore, the next step of each self-supervised, near-to-far learning is to choose the base classifier. There are many classifiers available for terrain classification. In the following, some effective methods deployed in terrain classification are described followed by the classification methods employed by the LAGR teams.

Multi-layer Perception (MLP)

Multi-Layer Perception (MLP) is a feed-forward neural network with multiple layers of processing neurons called *perceptrons* to overcome the shortcomings of the single-layer networks. Each node outputs an activation function applied over the weighted sum of its input data. The activation function could be linear, Sigmoid, Tanh. MLP is learned by the *backpropagation* algorithm, an efficient gradient descent algorithm, to propagate error back through network recursively and compute weight changes.

In the *forward pass*, the predicted outputs corresponding to the given inputs are evaluated. In the *backward pass*, If we have a function $f_{\mathbf{w}}(\mathbf{x})$ where $\mathbf{x} \in \mathbb{R}^d$ is a d -dimensional input, and $\mathbf{w}$ represents “the weights”, then given a training set of $(\mathbf{x}_i, y_i)$ we would like to minimize the cost function $C(f_{\mathbf{w}}(\mathbf{x}_i), y_i)$ with respect to $\mathbf{w}$. By using the gradient descent approach for each example $(\mathbf{x}_i, y_i)$ the derivative of the cost function with respect to $\mathbf{w}$ is computed. The weights are then updated by:

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial C(f_{\mathbf{w}}(\mathbf{x}_i), y_i)}{\partial \mathbf{w}} \quad (5.5)$$

λ is the learning parameter (usually a constant). MLP is capable of modeling complex functions, well at noisy input data and it can adapt the weights along by the changing of the environments.

Figure 5.2 MLP Neural Network.

Support Vector Machine (SVM)

Support Vector Machines (SVMs) have become the reference for many classification problems because of their flexibility, computational efficiency and capacity to handle high dimensional data [126]. Support vector machine is known as an optimal classifier as it provides a systematic method for separating the data optimally. It constructs an optimal hyperplane that separates different classes by a clear gap, which is as wide as possible. The optimality concept is involved in SVM through selection of a hyper-plane, which gives the maximum margin over all other possible hyperplanes. Larger margin leads to lower probability of misclassification. The parameters of the optimal hyperplane are obtained through an optimization method which tries to maximize the margins between different classes. This maximization is equivalent to minimization of weight matrix norm described as,

Find: W and b

$$\text{Minimizing: } \frac{1}{2} \|W\|^2 + C \sum_{i=1}^N \xi_i \quad (5.6)$$

$$\text{subject to: } \begin{cases} y_i (W^T \mathbf{x}_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \quad (5.7)$$

where, N is the number of input samples, W is the weight matrix of the optimal hyperplane, y_i is the desired label, and ξ_i are the slack variables, which basically allow some misclassification in the training aiming at better overall misclassification in test and train dataset. Value of $C > 0$ reflects the cost of violating constraints. A large C generally leads to smaller margin but also fewer misclassifications of training data and vice versa. The data points lying on the optimal margin are called support vectors, which contain the most relevant and sufficient information of the SVM algorithm.

Next, as the given optimization problem (primal problem) is not computationally convenient to solve, a simpler alternative is often proposed which is called dual problem. The dual problem is derived using the generalized Lagrangian function with the Karush–Kuhn–Tucker conditions, given as below,

Find: Lagrange multipliers $\{\alpha_i\}$

$$\text{Minimizing: } Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (5.8)$$

$$\text{subject to: } \begin{cases} \sum_{i=1}^N \alpha_i d_i = 0 \\ \alpha_i \geq 0 \end{cases} \quad (5.9)$$

where, $k(\mathbf{x}_i, \mathbf{x}_j)$ is called the kernel function which can be defined linearly ($k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$) or nonlinearly. Obtaining the optimal hyperplane through the given optimization with linear kernel is feasible when the classes are linearly separable. However, the outstanding feature of SVM becomes apparent in non-separable cases where the classes

overlap. In these situations, SVM produces nonlinear boundaries by constructing a linear boundary in a large, transformed version of the feature space. This means that SVM maps the data of the input space into a higher dimensional space in a nonlinear manner. The primary theoretical motivation for this function of SVM is the following theorem;

Cover's Theorem

A set of training data which is not linearly separable can be transformed into a linearly separable space with a high probability by projecting the input space into a higher dimensional feature space via non-linear transformations.

These transformations are in fact the kernel functions which are nonlinear in this case. The common nonlinear kernel functions include:

dth degree Polynomial (homogeneous): $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$

dth degree Polynomial (inhomogeneous): $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$

Gaussian or Radial basis function: $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$ for $\gamma > 0$

Hyperbolic tangent: $k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i \cdot \mathbf{x}_j + c)$

Figure 5.3 Training points that are nearest to the separating planes are called support vectors (SV).

The K-Nearest Neighbor Classifier (KNN)

The K-Nearest Neighbor (KNN) classifier is the most popular nonparametric technique for classification. It is a very useful classification approach where a large amount of training data is available. Given a d -dimensional input $\mathbf{x} \in \mathbb{R}^d$ and $y \in \{-1, 1\}$ class label, KNN task is to classify a new point $\mathbf{x}_{\text{new}}$ to the class y_j when the K closest points $\mathbf{x}_j$ to this new point are from class y_j . The performance of nearest neighbor classifier relies on metric or distance function (Euclidean, Mahalanobis distance), the number of neighbors K (3, 5, 7), and the way to combine the neighbor's labels (weighted majority votes). KNN is simple, but it needs a lot of space to store all the training data. The training phase does not take much time, but the classify phase is very slow since it needs to calculate and compare distance from each new example to all other input data. Figure 5.4 shows a KNN classifier with two different K values (3, 5). If $K=3$ the green circle is classified as blue triangle class, but if the outer circle is selected ($K=5$) it is classified as red circle class. This figure clearly shows that KNN is very sensitive to the number of K .

Figure 5.4 Example of KNN with $K=3$ and $K=5$ (outer circle).

The API LAGR team [113] uses a supervised three-layer neural network to learn the relationship between geometry and traversability through collected data from different previous runs. First of all, a human expert is asked to assign 4 labels (low, intermediate, high, lethal) to the 4000 examples. Next, a multi-layer perception (MLP) with one hidden

layer (hyperbolic tangent activation function) is used to learn 8-dimensional geometric feature vector, while the output layer uses a logistic function.

They also quantify their MLP method with several classification methods such as naïve bayes, KNN ($K=7$), SVM polynomial (3), and SVM Gaussian kernel. They report that SVM Gaussian kernel performs slightly better than their method with more computational time. A disadvantage of their work is that it depends on human expert to assign labels for training data.

JPL LAGR group [125] address 2 learning algorithms. The first one is a naïve-Bayes classifier that uses geometry-based features. This learner is used only for underfoot and near field data where geometry information is available. This classifier is fast and robust while it has limited discrimination. The second classifier is a linear SVM with soft margin ($C=500$). The linear Support vector machine (SVM) is learned using the near field labeled images and is used to classify pixels in the mid and far field.

On the other hand, they address different learning algorithms in the first phase of LAGR project. Two different learning methods, an unsupervised K-means clustering followed by regression and a supervised Mixture of Gaussian (MOG) are used. First, each frame is clustered using K-means ($K=4$), then an average traversability of each cluster is measured. The next step is to estimate traversability label of each feature vector using a nearest neighbor regression. In second approach, a two-class discriminant-based MOG is used with full-covariance (3×3) Gaussian mixtures followed by EM maximum likelihood. These two methods cannot broadly apply in unstructured environments since they are parametric classifications which are sensitive to how their parameters are selected.

SRI LAGR team [115] uses the two-class AdaBoost as the online learning method at some of their earlier work. AdaBoost is an adaptive Boosting method that constructs a final strong classifier as a linear combination of a number of weak classifiers outputs. Initially, unweighted training data is passed to a weak classifier (e.g. tree classifier). If a training sample in the output of a weak classifier is misclassified, the weight of that

particular training data is increased so that the next classifier is more focused on that example. This procedure repeats for several weak classifiers and the final classification is constructed as the linear combination of the weak classifiers at each round. A score is assigned to each classifier based on its performance on the training set. Thus, the final linear combination is a weighted linear combination.

NY/Net-Scale LAGR team [116] uses logistic regression with stochastic gradient decent and a Kullback-Liebr loss function to classify 5 categories (supertraversable, traversable, foot-line, obstacle, superobstacle) instead of binary classification. A separate logistic sigmoid function is trained for each of the five categories with one-against-the-rest strategy. For each five categories a balanced data (up to 1000 training examples) is generated in five separate ring buffers. These separate ring buffers not only balanced the data for better classification, but also are used as short-term memory from at least 2 timestamps to 30 timestamps. One advantage of their method is that they use multi-class regression instead of binary classification.

Finally, the learning methods employed by DARPA-LAGR research groups are summarized in Table 5.3.

TABLE 5.3 LEARNING METHOD USED BY LAGR TEAMS

LAGR Team	Learning algorithm	Number of classes
API [113]	Three-layer neural network	4
SRI [115]	AdaBoost	2
NY/Net-Scale [116]	logistic regression	5
JPL [125]	naïve-Bayes, Linear SVM	2
JPL [33]	K-means clustering, mixture of Gaussian (MOG)	4
Colorado [122]	logistic regression/ Linear SVM	2

5.3 Proposed Self-Supervised Learning Framework

Although, the current self-supervised, near-to-far learning methods can adapt to changing environment concurrently, it has several limitations. A common challenging aspect is that

the supervision labels provided by stereo system are often contaminated with noise, which can degrade the classifier performance drastically. In order to address the problem of labeling errors in the training data, a new method called confidence-based auto labeling is proposed to identify the data that are susceptible to be mislabeled followed by two algorithms for handling them efficiently.

Once the auto labeling module provides the stereo supervision labels for all the input training data, these labels are then associated with their visual appearance features to generate feature vectors for classifier training. The current self-supervised, near-to-far learning algorithms assume that all the visual features extracted from the input images (such as intensity, color, or texture) have the same level of prominence. Generally, a set of attributes is extracted from the input image and they are added to a feature vector with the same or different predefined weights. However, in most outdoor and unstructured environments, one or some of the attributes of the feature vectors may be more informative for discrimination of the terrain from non-terrain regions. Hence, inspired by the selective visual attention mechanism found in biological vision systems, a new top-down attention model is introduced, which generates a set of adaptive gains representing the saliency of each feature in the feature vector. These adaptive gains bias the classifier in the sense that it devotes more attention to learn those salient features with larger adaptive gains.

Once the visual appearance features are associated with the traversability label provided by stereo supervision, they are passed to a classifier to learn the models of the traversable (negative class -1) and non-traversable (positive class +1) regions. This is a standard two-class classification problem, where the classifier assumes that examples for both classes are provided and have the same size. However, in practical situations like outdoor terrain

classification, the vehicle may not encounter any negative examples (i.e., non-traversable terrain) during some parts of its course. Moreover, since the system autonomously associates the visual appearances of the near field regions with traversability labels (supervisory labels provided by stereo information), it has no control on the distribution of

class labels in the training data. The former limitation is known as novelty detection or one-class classification problem and the later one is known as unbalanced data problem. To address the current limitations of standard two-class classification, an online adaptive learning mechanism is presented which automating the task of selecting learning strategies based on the nature of the incoming input training data in real-time.

The following subsections provide the details of the proposed SSL framework. The proposed methods are general and can be applied to a variety of applications. Block diagram of the proposed SSL framework is illustrated in Figure 5.5.

Figure 5.5 Block diagram of data flow in the proposed Self-Supervised Learning framework.

Figure 5.6 Original RGB image

5.3.1 *Image Pre-Processing*

The vision sensor used in this work is a short baseline ($B = 12\text{cm}$) Point Grey Bumblebee color stereo vision system. Triclops SDK, a Stereo Vision Software Development Kit (SDK) provided by Point Grey is used for stereo processing [127]. There are two main processing blocks in this library. The first processing block is the image pre-processing block that applies a low-pass filter, rectifies the images and performs edge detection [127]. A sample of rectified RGB image is shown in Figure 5.6. A pair of pre-processed images is used as an input to the second processing block that performs stereo matching using Sum of Absolute Differences correlation method (SAD) to create a disparity map (Figure 5.7). In some cases, such as occlusions and lack of texture in the observed scene, estimated disparity values may contain many errors. In order to minimize such inaccuracies and errors, several validation steps such as Texture Validation and Surface Validation are applied using Triclops SDK library. The image size used in this chapter is the standard size of 320×240 .

Figure 5.7 Colored stereo disparity map

5.3.2 Robust Ground Plane Estimation

In indoor environments with flat ground, where there is no change in position of the ground plane relative to the camera, the ground plane disparity function is fixed. However, in unstructured environments where there is no dominant flat ground, a robust estimator is required that not only tolerates a large outlier percentage but also several discontinuities. RANSAC (Random Sample Consensus) [128] is an effective technique developed within the computer vision community to estimate the ground plane robustly, that takes all the image features (the disparity map d in image coordinates (u, v)) in the region of interest, fits the ground plane and discards the obstacle features as outliers. The obstacles and ground plane estimator module applies fast RANSAC algorithm directly to the disparity image to estimate the dominant ground plane. Firstly, invalid pixels in current disparity map where the disparity values are unavailable are removed (0 and 255 values). Secondly, three points from the disparity image (u_i, v_i, d_i) for $i = 1, 2, 3$ are

Figure 5.8 A plane is fitted to disparity map

randomly selected directly from the near field of the disparity image (up to 7 meters around the vehicle) to fit a plane. A plane is fitted to these 3 points and then ranked by the number of supporting points that are close enough to the plane (inliers). Figure 5.8 shows a hypothesize ground plane fitted to the three random selected points in the near field of the disparity image. The distance of a point P_a in the near field of the disparity image to the nominate plane is calculated by:

$$\text{Minimum distance} = (P_a - P_b) \cdot \frac{\mathcal{N}}{\|\mathcal{N}\|}, \quad (5.10)$$

where $\mathcal{N}$ is the normal of the nominate plane and $P_b = (u_b, v_b, d_b)$ is a point on the nominate plane where

$$au_b + bv_b - d_b + c = 0, \quad (5.11)$$

where a , b , and c are the nominate plane parameters. This procedure is repeated (it) times (computed from equation (5.12)). RANSAC adaptively chooses the number of iterations (it) with probability p by

$$it = \frac{\log(1 - p)}{\log(1 - (1 - e)^s)} \quad (5.12)$$

e denotes the percentage of the region of interest which is occupied with obstacles called contamination fraction. It is often unknown a priori, so the worst case is chosen where it is assumed that 75% of the ROI is occupied with obstacles. s is the size of the sample needed for ground plane fitting. The size of the sample in this case is 3, as we need 3 points to define a plane, and the probability p is set to 0.99 which is high enough to remove any outliers from the fitted plane. The pseudo-code for finding the number of iteration (it) is described in Algorithm 5.1. Finally, a new ground plane, known as dominant ground plane, is fitted to the triple points with the maximum number of inliers (supporting points) by total least squares fitting method.

Algorithm 5.1 The pseudo-code for finding the number of sample

```

 $it = \infty$ , sample_count = 0

While  $it > sample\_count$  repeat

  Choose a sample and count the number of inliers

  Set  $e = 1 - (\text{number of inliers}) / (\text{total number of points})$ 

  Recompute  $it$  from  $e$ 

  Increment the sample_count by 1

Terminate

```

One of the limitations that most of the ground plane estimation techniques would suffer is that when obstacles cover dominant area in front of vehicle (more than 75%) these

Figure 5.9 An example of an input image which obstacles occupy most of the near field of the vehicle.

techniques consider the obstacles as a ground plane. An advantage of RANSAC is that it uses the supporting points instead of the whole area of interest for robustly estimating ground plane. Therefore, it prevents failures by selecting obstacles instead of true ground as model parameters [129]. However, even RANSAC will fail if most of the points in the ROI come from the obstacles regions. Hence, in order to tackle this problem we consider two solutions. First, we increase the probability of percentage of the region which may be occupied with obstacles to 0.80 (i.e. $e = 0.80$). Second, we record the normal of the fitted ground plane ($\mathcal{N} = \langle a, b, c \rangle$) at each subsequent frame and compare it with normal of dominant plane in previous frame. If the sign of direction of the normal vector has been changed, we consider the current estimated ground plane fitted to obstacles, discard the current ground plane model and use the previous frame ground plane parameters. Figure 5.9 illustrates an example that the proposed robust ground plane estimation technique is able of estimating ground plane and obstacles correctly while most of the area in front of the vehicle ($85\% >$) is covered by obstacles.

5.3.3 *Auto Labeling*

Once the dominant ground plane is estimated, it is projected back onto the disparity image to detect obstacles. Any point whose measured disparity is not within a preselected threshold of the expected ground plane disparity range will be considered as an obstacle. This threshold is selected through several validation runs trying to get the minimum error on man-labeled datasets. Consequently, for each pixel of the input image with valid stereo data in the near field of the vehicle we will have,

$$y_i = \begin{cases} 1 & d_m(i) \geq d_e(i) + \varepsilon \\ -1 & d_m(i) < d_e(i) + \varepsilon \\ \text{unkown} & \text{invalid } d_m(i) \end{cases}, \quad (5.13)$$

for $i = 1, 2, \dots, N$, where N denotes the number of pixels in the input image. Moreover, $d_m(i)$ shows the measured disparity, and $d_e(i)$ shows the estimated disparity by RANSAC ground plane fitting at each pixel location, and ε represents the preselected threshold. Label 1 corresponds to an obstacle or non-traversable pixel and -1 to a ground or traversable. The labels for those pixels that have invalid disparity values are considered as unknown and they are discarded from the further processing. The ground plane (shown in green) and obstacles (shown in red) detected in the near field are projected back to the original RGB image are depicted in Figure 5.10. Those points that have invalid disparity values left uncolored in the lower half of the image. Ground plane estimation is the most computational complex part of the algorithm. For a 320x240 image, the estimation takes approximately 100 ms.

Figure 5.10 Ground plane and obstacles overlaid on RGB image in the near field.

In this method, the label propagation depends on disparity differences. Thus, when the disparity difference is not significant enough, such as where the lower parts of obstacles (footlines) meet the ground points, the label propagation has some uncertainty regarding the correct labels of the those points. This uncertainty becomes more severe when the vehicle is maneuvering in the outdoor, unstructured environments where the assumption of perfect flat ground plane cannot be applied. In this case, some part of ground plane may stand above the estimated dominant ground plane and considers as obstacles [116]. Figure 5.10 depicts former type of noise in label propagation where the lower parts of the hay bales (footlines) are mislabeled as ground plane (shown in green color).

This type of error in ground and obstacle detection is not crucial in the case of robot localization and mapping where the labeled ground plane and obstacle points in the image are further projected into a 2-dimensional Cartesian grid map centered on the vehicle. Therefore, the upper part of an obstacle, which is most probably labeled correctly, will cover the grid map cell representing the obstacle location even if its lower parts are mislabeled as ground plane. However, in this thesis, as these noisy labels further will be

used in the learning procedure, difficulties will be imposed. Next section describes the proposed technique for identifying the data that are susceptible to be mislabeled and handling them efficiently.

5.3.4 *Confidence-Based Auto Labeling*

Machine learning approaches generally assume that the information in the training data is noise free. However, in most real world applications either training data attributes or the class labels are contaminated with noise that may introduce serious negative impacts on the classification performance [130]. In the latter case, which is also known as labeling error or mislabeling, the performance of a learning algorithm can be degraded drastically that can lead to highly nonlinear decision surface, over-fitting of the training set, and poor generalization ability of the classifier [131].

If high quality training data is not guaranteed, one cannot expect to get high classification rate. For instance, the generalization ability of SVMs is achieved by finding a large margin between two classes. Since the optimal hyper-plane obtained by the SVM depends on only a small part of the data points that are close to the boundary, i.e. the support vectors, a few mislabeled data points in the training data could reduce classification accuracy significantly and increased the number of required support vectors to define the proper decision surface.

Many recent works in machine learning literature have addressed the problem of labeling errors in the training data [132-135]. Brodley and Friedl [132] consider three different reasons for mislabeling errors. The subjective nature of the labeling task, data-entry error, and the lack of information to determine the correct label for a training sample are the most common reasons for the labeling error in the training data. The problem of labeling error that is occurred in this work is related to the third cause (i.e. the lack of information), as the training points are labeled automatically by stereo supervision from the unknown environments without any human intervention, and it is always prone to contain errors.

Various methods have been proposed to handle the mislabeling problem in the real world training data. These approaches can be classified into two major categories: learning in noisy environments, and techniques based on identifying and handling noisy training data before applying to a classifier. Different learning algorithms have been developed, which are capable of learning in the presence of noise, to reduce the effect of overfitting to class labels noise in the training data [135-136]. However, one drawback of learning techniques in noisy environments is that as the amount of mislabeled training data increases, the impact of pruning becomes less effective and the learning technique becomes more susceptible to the noise level.

Second approaches try to purify the training data before learning model is built. This requires exploring the training data to identify the noisy ones, attempting to enhance them in some way, and finally pass the enhanced training data to the classifier [133]. The study in this work belongs to the latter category.

The first step is to explore the training data in order to identify the susceptible examples. For this purpose, the distance between the measured and estimated disparity, called the “score value” in this proposed method, is calculated for each input instance. Suppose a set of training data is given. Each data point belongs to a pixel in an input image $I(i)$ with $i = 1, 2, \dots, N$, where N is the total number of pixels in the image. Depending on which class the data points belong to, they are labeled by $y_i \in \{1, -1\}$, which were determined through the procedure of stereo supervision auto labeling mentioned before. Note that those points that have invalid disparity are removed.

To identify training data points that are susceptible to be mislabeled, first the training dataset S is split into two subsets S_1 and S_2 with sizes N_1 and N_2 respectively based on the estimated labels $y_i \in \{1, -1\}$, where $S_1 = \{i | y_i = 1\}$, $S_2 = \{i | y_i = -1\}$ subject to $N_1 + N_2 = N$. This means that those pixels of the training data in an input image with label 1 are grouped into subset S_1 and those with label -1 are grouped into subset S_2 . The reason is that the disparity differences range differently in obstacle points than ground

plane points. Since later, statistical information needs to be extracted from each of these subsets they have to be isolated from other class members.

Next, two score functions are defined for each of these subsets with their values determined by the Euclidean norm of the measured and estimated disparity for each labeled data point,

$$f_k(i: i \in S_k) = \|d_m(i) - (d_e(i) + \varepsilon)\| \quad \text{for } k = 1, 2. \quad (5.14)$$

where, $d_m(i)$ shows the measured disparity, and $d_e(i)$ shows the estimated disparity by RANSAC ground plane fitting at each pixel location, and ε represents the preselected threshold. The longer distance a data point has from the estimated ground plane, it gains higher score value. Subsequently the data points, whose score function values are smaller than a threshold, are recognized to have high probability of being mislabeled. The major problem is that finding a predefined global threshold that can be applied to different situations is almost impossible since no prior knowledge about the scene or distribution of the f_k is available. One possible solution is to define an adaptive threshold value based on the statistical information of each score function. The simplest and the most efficient statistical value is the *mean* of the score function values:

$$T_k = \text{mean}_{i \in S_k} = \{f_k(i)\} \quad \text{for } k = 1, 2 \quad (5.15)$$

However, if the score function contains noisy information, the mean would not be a stable threshold value. Therefore, to automatically select an optimal threshold value we present an iterative adaptive thresholding method similar to K-means clustering algorithm. For each subset of training datasets S_k for $k = 1, 2$, the mean of the score function value is selected as an initial threshold value (T_k). Then during each iteration, the score function values are divided into two groups separated by the threshold. The average of each cluster is computed and a new threshold value is reinitiated based on the average of the two clusters means. This procedure is repeated until the new threshold value is very close to its previous one or the maximum number of iterations is reached. The pseudo-code for iterative adaptive thresholding is described in Algorithm 5.2.

Algorithm 5.2. The pseudo-code for iterative adaptive thresholding

1. For each subset of training datasets S_k where $k \in \{1, 2\}$
2. Initiate threshold $T_k = \text{mean}_{i \in S_k} = \{f_k(i)\}$
3. Segment a score function into two sets by

$$\text{I. } S_{k,1}(i) = f_k(i) > T_k$$

$$\text{II. } S_{k,2}(i) = f_k(i) \leq T_k$$
4. Find the mean of each set

$$\text{mean}_{k,1} = \text{mean} \{S_{k,1}(i)\}$$

$$\text{mean}_{k,2} = \text{mean} \{S_{k,2}(i)\}$$
5. A new threshold is the mean of the mean1 and mean2

$$T_k = \frac{\text{mean}_{k,1} + \text{mean}_{k,2}}{2}$$
6. Go to the step two, use the new threshold and keep repeating until either the maximum number of iterations is reached or the new threshold is very similar to the previous one.

Once the mislabeled data points are identified, one needs to handle mislabeled examples. In this work, two methods for handling the mislabeled data is proposed: removing, and assigning probabilistic confidences weights on each instance of training data.

One way of improving the quality of the training data is to remove the unclear instances. Those points that have low score values calculated from equation (5.14) are discarded prior to applying to the classification method. Each score function value is compared with its class related adaptive threshold T_k found by Algorithm 5.2 and if it is smaller than the threshold its respective point is removed from the training data. However, the mislabeled data points often occur in the boundary of two classes and may contain valuable information, which is thrown away by eliminating them. To prevent losing information an alternative approach called *probabilistic weighing method* is proposed in this work.

In *probabilistic weighing method*, each labeled pixel in the input image is assigned a probabilistic weight w_i that represents its confidence level. However, assigning a proper value for a weight of each data instance is a challenging task. In supervised learning techniques, a human expert usually gives a proper value for the quality of each data instance. However, in this case, each data instance should be ranked based on its data quality automatically without any human intervention. Therefore, a new approach is proposed in which the data points are weighted based on their score function values calculated from equation (5.14).

a weight vector $\mathbf{w}^k$ is denoted for each subset of the labeled training points by $\mathbf{w}^k = \{w_i^k | i \in S_k, \text{ and } k = 1, 2\}$. Each probabilistic weight should indicate the confidence level of a label assigned by the stereo supervision. To correctly assign a probabilistic weight to each instance in the training data, a map function from score function values to the probabilistic weights is introduced based on the logistic distribution function, which is denoted as follows:

$$w_i^k(f_k | \mu_k, \sigma_k) = \frac{1}{1 + e^{\frac{(f_k(i:i \in S_k) - \mu_k)}{\sigma_k}}} \quad (5.16)$$

where μ_k is the mean and is calculated from equation (5.15) and σ_k represents the standard deviation of the score function f_k . Each weight vector $\mathbf{w}^k$ for $k = 1, 2$ is then normalized to the range of $[0, 1]$. Therefore, each data point in the training set has a probabilistic weight assigned to it that indicates its relative importance. The instances with lower weights will have a higher probability of being mislabeled. These probabilistic weights will be used in the later stages of self-supervised, near-to-far learning as the confidence level for each instance in the training data passed to a classifier. The classifier uses these weights to bias learning method towards those instances that have higher confidence level (less noisy) and inhibits those samples that susceptible to be mislabeled.

5.3.5 *Optimal Feature Extraction*

Once the ground plane and obstacles are estimated in the disparity image, and mislabeled data are identified and handled through confidence-based auto labeling procedure, their corresponding image pixels in the reference image (right image) are labeled as either traversable (negative class -1) or non-traversable pixel (positive class +1). These labels are then associated with their visual appearance features to generate feature vectors for classifier training. However, in this work, in order to remove noise and reduce computation time associated with pixel-based processing, an input image is decomposed into non-overlapping patches (windows) of size 10x10. In this method, a 20-pixel margin is set for each side of an image of size 320x240 (because invalid disparity values are often located in the side of images) and then the entire image is decomposed into 560 patches of size 10x10. The lower half of the frame that contains valid stereo labels is only used for the classifier training (maximum 280 patches).

The number of ground plane and obstacle labels is counted in each patch, and the patch is assigned traversability label of either obstacle or ground plane based on the number of whichever is greater. In order to fail on the safe side, if the numbers of both labels are the same the patch is marked as an unknown. Figure 5.10 shows a labeled image using patches size of 10x10. Only near field stereo data is labeled (280 patches).

Next, a list of descriptors also known as an appearance feature vector is extracted within each labeled patch. For color features, the color information output from the camera is in RGB color space, which is highly correlated and varies drastically with illumination changes. Instead, the RGB color space is transformed to the CIE Lab, which is nearly uniform color space and close to human visual perception. An advantage of CIE Lab color space is that the intensity (channel L) is separated from color channels (a, and b). The CIE Lab statistics consisting of mean and standard deviation within each patch are chosen as color and intensity descriptors.

However, color and intensity information are not enough for discriminating regions in outdoor environments. Texture is a fundamental property of surfaces. Image texture features represent a structure in terms of smoothness, coarseness, directionally and regularity of an object. In this work, in order to extract optimal and efficient textual information for ML algorithm, the input image is convolved with multiscale multi-orientation Gabor filters explained in Chapter 3 from equations 3.18 to 3.20. The magnitude of Gabor energy responses (from equation (3.20)) are calculated for each pixel location of the input image (see equation 3.20). Later, statistical information (i.e. mean and standard deviation) of Gabor energy responses are measured separately for each scale and orientation and they are added to the feature vector for each patch. Hence, a set of 38 features, which consist of intensity (2 features), color (4 features) and texture (32 features) are extracted for all patches of the input image.

The current self-supervised, near-to-far learning algorithms assume that all the visual features extracted from the input images (such as intensity, color, or texture) have the same level of prominence [18]. Generally, a set of attributes is extracted from each primitive the input image and they are added to a feature vector with the same or different predefined weights. However, in most outdoor and unstructured environments, one or some of the attributes of the feature vectors may be more informative for discrimination of the terrain from non-terrain regions. Recently, Blas et.al [23] assumed that the chrominance is more distinct than luminance in outdoor environments, therefore, they balanced their feature vector to rely more on color than texture or intensity, whereas, in an outdoor unstructured environment, one feature is not constantly dominant compared to others for all types of terrains and weather conditions.

To compensate the mentioned limitation of the current self-supervised, near-to-far learning algorithms, a novel approach is presented which exploits the biological-based procedure of selecting the most salient features based on top-down goal driven, resulting in more efficient and fast classification.

5.3.6 *Computational Model of Top-Down Attention*

An important mechanism of human visual system is its ability to attend selectively to regions of interest within the entire visual field. This mechanism is known as selective visual attention. For example, a driver is able to focus on the road while driving and ignores many other irrelevant visual data presented in the scene. The selective visual attention enables biological systems and in particular human to react to their environment instantaneously.

According to psychophysical and neurobiological findings, while a human observer looks at a scene, attention is driven to both Bottom-Up (BU) salient regions [137] and Top-Down (TD) relevant tasks [138]. When there is no specific goal or intention, the BU saliency guides attention solely by measuring the dissimilarity of the low-level features of the scene such as intensity, color, and orientations. BU saliency model considers a region as a salient if it differs from its surrounding regions. During the last two decades, several computational models for BU saliency have been proposed in the literature [139-140].

On the other hand, Top-Down (TD) guidance enhances the visual processing by prior knowledge of task or target, intentions, and current goals. TD attention biases the low-level features with known target representation [138]. TD task relevance directs attention quickly and reliably to the target; therefore decreases the processing time. Especially if the system has to operate in real-time, (e.g. an autonomous navigation system) the knowledge of the target helps to accelerate the target detection. However, as the neurobiological foundations are not completely understood, the top-down mechanisms, despite their importance in human visual system, are rarely considered in computational attention systems [139].

There are two major computational models of top-down guidance existed in the literature. One is those models that considers only target information and enhances features that are similar to the target [141-142] whereas second groups take into account information related to both target and distracters [138-139].

Recent neurological studies reveal that knowledge of the target is necessary but not enough for optimum search task [139]. For instance, if a task for robot is to drive on the road covered with green grass, color green should be biased and get more attention. However, if the background also contains green distracters such as tall grass and bushes, green color feature is not any more informative and the search task should focus more on another features such as texture of the road regions.

To address this problem, Navalpakkam and Itti [138] and Frintrop [139] consider properties of both target and the surrounding to assign weights to the features. Weight is computed for each feature as the ratio of the expected target's feature mean over the expected distracters' feature mean. While this simple ratio of expected values may be sufficient for searching an object in the static scenes, it is not robust to scene that its visual features have more complex distributions and the background may contain more salience regions like outdoor unstructured environments.

Moreover, they use several training images containing different view of the manually marked target to learn the relative top-down weights and test it on the images with slightly different backgrounds and target views. Therefore, choosing a suitable training image that contains both target and background is a disadvantage of this type of top-down attention model.

In the following, inspired by the selective visual attention mechanism found in biological vision systems, we propose a new top-down attention model (shown in block diagram of Figure 5.5), which generates a set of adaptive gains representing the saliency of each feature in the feature vector. The suggested top-down model is general and can be potentially applied to a wide range of applications such as object detection [143], and visual search [138].

A salient feature is defined as a feature that is highly distinctive between target (i.e. traversable regions) and background (i.e. non-traversable regions). The more salient a feature is the greater gain it receives. Later, these gains are applied to the same feature

vector of training data, therefore; the classifier devotes more attention to those salient features with larger adaptive gains.

To address this problem, we define a gain vector $\mathbf{g} = [g_1, \dots, g_m, \dots, g_M]^T$, where M is the total number of features (i.e. one gain per feature). Each gain is derived based on the dissimilarity measurement between the distributions of each feature in the traversable and non-traversable regions given by the stereo auto labeling in the near field of the vehicle.

Let χ_m, γ_m represent two random variables belong to a certain feature ($m, m \in \{1, \dots, M\}$) of traversable and non-traversable regions in the near field of the vehicle respectively. Rather than explicitly assume the distribution of each visual feature of target (i.e. traversable) and background (i.e. non-traversable) regions are single normal distributions [138-139], the distribution of each feature in each region is modeled as a Mixture of Gaussians (MoG). In general, Gaussian Mixture Model (GMM) is a weighted linear combination of Gaussian probability distribution functions (pdfs) as follows:

$$p(\mathbf{x}|\Theta) = f(\mathbf{x}) = \sum_{i=1}^n \alpha_i \mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i) \quad (5.17)$$

where $\Theta = \{\mu_i, \Sigma_i, \alpha_i\}_{i=1}^n$ are unknown parameters of GMM, n is the total number of mixtures, $\mathbf{x}$ is d -dimensional ($\mathbf{x} \in \mathbb{R}^d$) random variable (i.e. feature), the parameters μ_i and Σ_i are the mean and covariance matrix of i^{th} Gaussian distribution respectively, and α_i denotes the weight assigned to each Gaussian distribution $\mathcal{N}$ based on prior probability of mixture i subjected to $\sum_{i=1}^n \alpha_i = 1$. Each Gaussian probability distribution function (pdf) $\mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i)$, for $i = 1, \dots, n$, in the mixture is given by:

$$\mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^d \Sigma_i}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1}(\mathbf{x} - \mu_i)\right) \quad (5.18)$$

The next step is to estimate the GMM parameters, which are the number of mixtures (n), and the parameters μ_i and Σ_i of each mixture of GMM and the mixtures' prior probability α_i that best match the distribution of input data. The optimum number of mixtures (n) can

be found by several model order selection criteria [144]. In this work, we used Bayesian Information Criterion (BIC) [144] to estimate the number of Gaussian mixtures. Different numbers of Gaussian mixtures are explored from 1 to 10, and then the model with minimum Bayesian Information Criterion (BIC) is chosen as the appropriate number of mixtures. From the experiment, we observed that the three Gaussian mixtures satisfy the real-time requirements of the application without compromising the accuracy of estimation.

Moreover, several techniques have been proposed to estimate the GMM parameters. The most well-known technique is maximum likelihood estimation (MLE). Given a dataset $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, where N is the total number of feature vectors, assuming independence between the vectors, the objective of MLE is to find the GMM parameters that maximize the likelihood of the GMM given the input dataset X :

$$\Theta^* = \operatorname{argmax}_{\Theta} \log L(\Theta|X) = \operatorname{argmax}_{\Theta} (\log \prod_{j=1}^N p(\mathbf{x}_j|\Theta)) \quad (5.19)$$

The Maximum Likelihood Estimation (MLE) can be solved iteratively by using the conventional Expectation-Maximization (EM) algorithm [145]. The EM algorithm consists of iterating two steps, the E-step and M-step. The E-step calculates the expected value of the likelihood function, which is further used in the M-step to update the mixture parameters. This process is iteratively continued until convergence, i.e., reaching a local maximum. For further detail of EM algorithm, readers are referred to [145].

In real-time applications, one can reduce the EM search space; therefore increase the speed of estimation by making certain assumptions about the initial values. For example, it is assumed that all the mixtures are equally likely a priori, $\alpha_i = 1/n$, where n is the number of mixtures in the GMM. Next, given the features are independent the covariance matrices become diagonal that reduce model complexity. In this work, since the aim is to define distribution of each individual feature (m , $m \in \{1, \dots, M\}$) of traversable (χ_m) and non-traversable (γ_m) regions as a mixture of Gaussians ($n=3$), the feature vector dimension is set to $d=1$. In order to meet the requirement of the application for performing

in real-time, and to simulate the human like visual behavior, we assume that the feature responses are independent. Therefore, M-dimensional EM algorithm can be converted to much simpler M one-dimensional EM.

Once GMM parameters for a certain feature (m) of traversable and non-traversable regions are computed one needs to compare these distributions to define whether a feature is highly distinctive between traversable (i.e. target) and non-traversable regions (i.e. background). For the comparison between the distributions of two regions, the Kullback-Liebler divergence [146] is employed in the proposed top-down model. The KL divergence is a distance metric for a dissimilarity measure between two regions represented by Gaussian mixture models. The Monte Carlo sampling is used to approximate the KL-divergence between two GMMs as follows. Let $f(\chi_m)$, $g(\gamma_m)$ represent two probability density functions of a certain feature (m) belong to traversable and non-traversable regions respectively. Each probability density function is a GMM given by equation (5.17). The KL-divergence between these two probability density functions can be approximated as:

$$\text{KL}(f||g) = \int f \log \frac{f}{g} \approx \frac{1}{T} \sum_{t=1}^T \log \frac{f(x_{m,t})}{g(y_{m,t})} \quad (5.20)$$

where T i.i.d. samples are drawn from the density functions. Subsequently, the KL-divergence values are normalized within $[0, 1]$ across all features. If the distributions of a certain feature of two regions are very similar, the KL-divergence value will be near zero, implying that this feature are similarly distributed at both traversable and non-traversable regions. On the other hand, if the KL-divergence value is near one, the current feature is probably distinctive between the two regions and it is selected as a salient feature. Finally, a gain proportional to this distance metric is assigned to each feature (i.e. one gain per feature).

The detail description of the proposed top-down attention model is summarized below.

1. Once the stereo supervision labels for all patches of near field of the vehicle are estimated, they are associated with a set of visual appearance features (M is the total number of features) extracted within each patch.
2. The samples of two classes (i.e. traversable and non-traversable regions) are separated, and the distribution of each feature (m) in each class is modeled as a mixture of Gaussians (three mixtures per feature per class).
3. Once GMM parameters for distributions of a certain feature (m) in traversable and non-traversable regions are identified, the KL-divergence is employed to find the dissimilarity measure between the distributions.
4. A gain g is assigned to each feature based on the KL-divergence value. A gain close to 1 indicates that its corresponding feature is highly distinctive between two classes, and a gain near zero indicates that the distribution of its corresponding feature is similar in both classes thus it is not informative for classification.
5. Once a gain vector $g = [g_1, \dots, g_m, \dots, g_M]^T$, where M is the total number of features (i.e. one gain per feature), is determined it is applied to all feature vectors and new training data from the near field of the vehicle is passed to the classifier to learn the traversability model of the environments. Therefore, the classifier devotes more attention to those salient features with larger adaptive gains. Later, the gain vector is also applied to all the feature vectors of the entire image in the classification stage.

Figure 5.11 Overview of the proposed top-down attention model.

5.3.7 Online Learning

After the visual features with associated labels are extracted and biased through top-down task-specific guidance in the near field of the vehicle, they are employed to train a classifier to learn the models of the traversable (negative class -1) and non-traversable (positive class +1) regions. This is a standard two-class classification problem, where the classifier assumes that examples for both classes are provided and have the same population size. However, in practical situations like outdoor terrain classification, since the system autonomously associates the visual appearances of the near field regions with traversability labels (supervisory labels provided by stereo information), it has no control on the distribution of the class labels in the training data. Generally, the number of examples from negative class (mostly traversable class) outnumbers the number of

instances in the positive class (i.e. non-traversable class). This is known as unbalanced data problem.

Machine learning techniques generally assume that the training examples are evenly distributed among different classes. However, unbalanced training dataset problem arises frequently in many real-life applications, not just in vision-based autonomous navigation domain but also in other applications like fraud detection, and text classification [147]. A training dataset is unbalanced if it contains significantly fewer training instances of one class compared to another class. The terms imbalanced datasets problem, rare classes or learning with skewed class distribution are often used to refer to unbalanced dataset problem in the literature [148-149].

The unbalanced data problem frequently occurs in the autonomous navigation system when small proportion of the training data from the near field stereo labels may constitute examples of the minority class (non-traversable terrain examples) whilst the rest belongs to the majority class (traversable terrain examples). Figure 5.12 depicts an example of this situation where the minority class (red colored regions) only occupies 7% of the training data (near field of the vehicle). In this kind of situation, most of the DARPA-LAGR teams select randomly a predetermined number of samples for both traversable and non-traversable classes and if the number of available samples in either class is less than the target number of training examples, learning is not performed for that frame and the vehicle navigates only based on its near field stereo information [122].

Research has shown that standard classification algorithms yield poor results when the training data is unbalanced [150]. These classifications when dealing with unbalanced data sets simply ignore the minority class or classify them with small distribution. Hence, several algorithms have been proposed to cope with unbalanced training data sets problems. These algorithms are either applied to data level or handle unbalanced problem in classifier level.

Figure 5.12 Only 7% of near field occupies with positive examples.

The first type of algorithms is to preprocess the training data to balance the classes' distributions. Several data preprocessing algorithms have been proposed to handle the unbalanced data problem. Common data preprocessing approaches include under-sampling and over-sampling [151-153]. Under-sampling methods randomly remove the majority class examples while keeping the minority class distribution in order to balance training data. Therefore, the resulting dataset will be reduced in size compared to the original, biased training sets. One drawback of this method is that it discards a portion of training samples from the majority class, despite the fact that they may contain crucial information about terrain traversability.

Over-sampling algorithms duplicate the training instances from minority class until it attains the same size as majority class [147]. Two common methods of over-sampling are random resampling and SMOTE (Synthetic Minority Over-sampling TEchnique) [152]. In random method, over-sampling randomly replicates the data points in the minority class with replacement. In contrast, SMOTE method synthetically creates new data by interpolating new, non-replicated instances around the existing minority class points. Over-sampling does not suffer from information loss though one drawback of over-sampling techniques that makes them unpractical for autonomous navigation system is that it introduces an additional computational cost by increasing the size of training data.

The second type of algorithms for handling unbalanced data problem is to modify the classifiers in which they are able to deal with the unbalanced problems. One popular classifier level technique is to bias the classifier so that it pays more attention to minority class examples [151]. This can be done in many ways while a common approach is to set a high cost to the misclassification of the minority class examples, which is known as Cost Sensitive Learning (CSL).

Cost Sensitive Learning (CSL) has been introduced for several machine learning techniques recently [154]. The learning method used in this thesis is the Support Vector Machines (SVMs) which is currently enjoying much popularity in the machine learning community due to its good performance and less computational demands. SVMs also supports biased penalties for different classes for dealing with unbalanced data problem [151, 155].

5.3.8 *Cost Sensitive SVMs*

Direct application of SVM yields poor results in the presence of unbalanced data. When datasets are unbalanced, the learned class-boundary hyperplane by SVM is closer to the majority/negative examples. This means that the SVM considers positive class examples as noises and learns everything as negative class members. Hence, the false negative rate will be too high. This may be much more costly especially for autonomous navigation systems since the vehicle may consider all the potential obstacles as traversable ground plane regions and tries to drive through them.

To address shortcoming of conventional SVM algorithm, Veropoulos et al. [156] propose to modify the SVM learning method by introducing two loss functions for majority and minority classes. This is the method we adopt for cost sensitive self-supervised learning in this thesis to cope with unbalanced data problem.

Given a training dataset $\{\mathbf{x}_i, y_i\}_{i=1}^N$ where $y_i \in \{-1, +1\}$ is class label and $\mathbf{x}_i$ is the feature vector for i^{th} training data, the number of examples for positive class label $y_i = +1$ are

significantly fewer than the negative class label examples $y_i = -1$. Veropoulos et al. [156] suggest to assign different losses to different classes of data during training. The class-boundary hyperplane is then pushed closer to the positive class by giving more penalty value to minority class examples. This can be done by revising the equation (5.6) and (5.7) as follows:

$$\text{Minimizing: } \frac{1}{2} \|W\|^2 + C_+ \sum_{\{i|y_i=+1\}} \xi_i + C_- \sum_{\{i|y_i=-1\}} \xi_i \quad (5.21)$$

$$\text{Subject to: } \begin{cases} y_i (W^T \mathbf{x}_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0, i = 1, \dots, N \end{cases} \quad (5.22)$$

where C_+ and C_- are two different penalty values for positive and negative classes. By increasing C_+ , the misclassification cost of the positive class becomes more severe relative to the negative class. Such penalties make SVM more sensitive to the positive class examples during training.

Next step is to assign proper penalty values for both negative and positive examples. Since no prior knowledge is available about the level of unbalanced data in outdoor unstructured environments, it is proposed to assign the penalty values based on the distribution of the class labels in the training data at each input frame in real-time. If input training data contains both negative and positive examples, but it is unbalanced, two penalty values C_+ and C_- are selected such that it is inversely proportional to the instances numbers.

$$\frac{C_+}{C_-} = \frac{N_-}{N_+}, \quad (5.23)$$

where N_+ and N_- are the number of instances at each class label respectively. Therefore, when an autonomous vehicle travels along the road, adaptive penalty values are assigned to both negative and positive classes instantly at each input frame based on the number of instances at each class label. Subsequently, the whole framework will be adapted to changing environments concurrently.

Figure 5.13 Near field training data contains only negative examples (green colored projected on original RGB image).

5.3.9 *Novelty Detection*

Although cost sensitive SVMs can cope with unbalanced datasets, it is still a standard two-class classification problem, where the classifier assumes that examples for both classes are always provided. However, in outdoor unstructured environments, the vehicle may not encounter any positive examples (i.e., non-traversable terrain) during some parts of its course as depicted in Figure 5.13. Therefore, the two-class SVM classification cannot be used for traversability prediction since it needs both class examples. This limitation is known as novelty detection or one-class classification problem in the machine learning community.

Recently, one-class classification or novelty detection has received much attention and importance in the practical machine learning research community. One-class classification (distribution estimation) attempts to describe one class of objects (target) and distinguishes it from all other possible objects (outliers). Only negative examples are used for training, as it is hard to identify models for positive examples. One-class SVM proposed by Schölkopf et al. [157] recently extended the SVM methodology to handle training using only negative information (one-class). First, it transforms the feature space

via a kernel function to the higher dimension representation. Then, it attempts to separate the new training data with the largest margin from the origin that corresponds to the second class. The algorithm can be summarized as follows.

Given a training dataset $\{\mathbf{x}_i\}_{i=1}^N$ where $\mathbf{x}_i$ is the feature vector for i^{th} training data with no class information, the task is to solve this problem by:

$$\text{Minimizing: } \frac{1}{2} \|W\|^2 + \frac{1}{vN} \sum_{i=1}^N \xi_i - \rho, \quad (5.24)$$

$$\text{Subject to: } \begin{cases} (W^T \mathbf{x}_i) \geq \rho - \xi_i \\ \xi_i \geq 0, i = 1, \dots, N \end{cases} \quad (5.25)$$

where $v \in (0,1]$ is a upper bound that controls the expected fraction of outliers (i.e. second class members) in the training data. Note that ρ is similar to ρ ($\rho = -b$) in equation (5.7) and it is calculated to define the decision function. Thus, in this work, one-class SVM is employed only for those input frames in which examples of one class label are observed in the near field of the vehicle.

Therefore, the proposed adaptive online learning method can be summarized as below:

1. On every processing cycle, the distribution of the class labels in the training data is first examined.
2. If the distribution contains only one class labels examples, the one-class classification learning strategy (One-class SVM) is selected automatically to predict the upcoming traversability of the road based on only one class examples.
3. Once the second class labels examples is observed in the near field of the vehicle, cost sensitive SVMs is activated which is assigned adaptive penalty values to both negative and positive classes instantly at each input frame based on the number of examples at each class label.

4. However, if the distribution of the two class labels are even or nearly even (between 40-60%) the typical two-class classification (Two-class SVM) are employed to train in the near field and classify the far field terrains.

5.4 Results and Analyses

In this section, a series of quantitative and qualitative analysis are conducted based on experimental results to evaluate the performance gains of the proposed Self-Supervised Learning framework versus the baseline techniques. The results presented in this chapter are carried out on images datasets logged during live runs of the vehicle in the DARPA-LAGR program. The Geometric Mean (G-means) is used as an evaluation metric to take into account the unbalanced nature of the datasets in outdoor unstructured environments. Furthermore, a series of representative images are presented to assess qualitatively the performance of the proposed framework.

5.4.1 *Datasets*

The experimental results presented in this chapter, are conducted on six natural 320x240 resolution frame sequences logged during live runs of vehicles competing in the DARPA-LAGR program. DARPA-LAGR is the current gold standard program for vision-based navigation system using machine-learning techniques. The dataset is referred as the “Hand-Labeled DARPA LAGR datasets” recently contributed by Procopio from Colorado LAGR team and made publicly available [158]. Three types of terrain covers each with two separate lighting conditions are included in the datasets. Figure 5.14 shows representative images from these six datasets. As it is shown the terrain appearance varies drastically from one datasets to another dataset. The ground cover includes mulch, dirt, and foliage where both natural obstacles (trees, dense shrubs) and man-made obstacles (hay bales) are placed in different locations. The difficult lighting conditions vary from cast shadows (e.g., DS2A) to high saturation (e.g., DS1A).

Each frame consists of a raw RGB image, raw disparity information and a hand-labeled image. Each pixel in a hand-labeled image indicates a label: 0 means ground plane, 1 means obstacle, and 2 means unknown. If it is hard for a human to assign a class label for a pixel with probability, or it is from “don’t cares” regions (e.g. sky) is labeled as an unknown pixel. Two samples of hand-labeled ground truth images are illustrated in Figure 5.15. Each pixel in a hand-labelled image (Figure 5.15 (b)) is color coded as: green, red and blue indicate ground plane, obstacle, and unknown respectively. From now on this chapter, green color indicates ground plane (i.e. traversable points or class -1) and red represents obstacles (i.e. non-traversable points or class +1).

(a) DS1A

(a) DS1B

(a) DS2A

(a) DS2B

(a) DS3A

(a) DS3B

Figure 5.14 Representative images from six datasets.

Figure 5.15 (a) Example RGB images (b) Human labeling RGB images overlaid.

5.4.2 *Evaluation Measurements*

The efficiency of the learning applied to computer vision can be evaluated either qualitatively by human visual perception or quantitatively using different evaluation metrics. Human observers simply can tell which classifiers perform qualitatively better, but human visual perception may fail when the classifiers perform almost the same. For quantitative evaluation, there are different ways of measuring a classifier performance in literature depending on the applications and types of classification [122, 151-152]. One widely used evaluation metric is the overall classification accuracy, which is defined as the percentage of corrected classified pixel to total number of pixels. However, this is not a suitable metric when the datasets are highly imbalanced or the error costs of different classes vary drastically. For example, for an autonomous navigation system, the number of instances for traversable terrains (i.e. negative class) outnumbers the number of instances in the other class (non-traversable terrain) during most of its course. If an input dataset contains 99 examples of usual event (i.e. traversable terrain) and only one example

of the rare event (i.e. non-traversable terrain), a classifier that predicts everything as usual traversable terrain will be 99% accurate, while it misclassifies a very important non-traversable terrain which is very critical for an autonomous vehicle to avoid. On the other hand, misclassifying the rare event is much more costly than misclassifying the usual event for an autonomous vehicle.

The performance of a classifier on highly imbalanced datasets can be assessed by a confusion matrix. A typical two-class confusion matrix is illustrated in Table 5.4. The elements of the confusion matrix are the number of True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

TABLE 5.4 CONFUSION MATRIX

		Predicted Class	
		T	N
True Class	T	TP	FN
	N	FP	TN

A variety of evaluation metrics can be derived using the confusion matrix depend on different applications. The most relevant to rare event classification are precision, recall, F-measure, sensitivity, specificity, geometric mean (G-means).

Precision, recall and F-measure:

When the performance of the correctly classified minority class (positive examples) is concerned, Recall and Precision are two suitable metrics.

- The recall (r) or true positive rate is the proportion of positive cases that were correctly identified to all positive cases in data, as calculated using the equation:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5.26)$$

- Precision (p) is the proportion of the predicted positive examples that were correct to examples labeled as positives, as calculated using the equation:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5.27)$$

- Finally, F-measure, a metric used widely by the information retrieval community, is a harmonic average of precision and recall with respect to minority class (i.e. positive examples). The weighted harmonic mean of recall(r) and precision (p) is:

$$\text{F-score} = \frac{(\alpha+1)r \times p}{r + \alpha \times p} \quad (5.28)$$

where α indicates how much weight one wants to assign to Precision vs. Recall. With no a-prior knowledge, F-measure commonly uses “average” of precision and recall (i.e. $\alpha = 1$).

Sensitivity, Specificity, and Geometric Mean:

To evaluate the performance of both classes on imbalanced datasets, two metrics the sensitivity and the specificity are used in several researches.

- Sensitivity (recall) can be defined as the accuracy on the positive examples given as:

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (5.29)$$

- Specificity can be defined as the accuracy on the negative examples given as:

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (5.30)$$

- Kubat and Matwin [153] introduced Geometric Mean (G-means) to take into account both sensitivity and specificity simultaneously in one metric. Several researchers indicated G-means as the best performance metric on the imbalanced data sets because G-means shows only high values when classifier perform well on both majority and minority classes. Therefore, G-means is used in this chapter for evaluation metric. G-means is denoted as :

$$\text{G-means} = \sqrt{\text{Sensitivity} * \text{Specificity}} \quad (5.31)$$

5.4.3 *Experimental results*

A series of extensive qualitative and quantitative experiments are run to evaluate different subparts of the proposed Self-Supervised Learning (SSL) framework against baseline SSL. Baseline SSL comprises of the best self-supervised, near-to-far learning steps proposed by LAGR groups.

With regard to the randomize search step in the RANSAC ground plane estimation, repeated measures are needed to compensate the effect of each individual randomized experiment. Thus, each experiment is repeated 10 times on a dataset and the average and standard deviation is reported across all datasets for all the experiments in this section.

Moreover, a number of parameters are fixed for all experiments conducted in this study. Here, all of them are summarized.

Baseline Classifier: The baseline classifier is a SVM with linear kernel called SVM–Linear. The Support Vector Machines implementation used here is LIBSVM 2.9. All the experiments are conducted in the visual C++ 6.0.

The Near Field: The area less than 7 meters in front of the vehicle considers as the near field. Only this area is employed for classifier training.

Ground Plane and Obstacle Detection: A fast robust RANSAC algorithm directly applied to the disparity image to estimate the dominant ground plane and obstacle in the near field of vehicle.

Image Resolution: The input images in the datasets are in the size of 640x480. However, to meet the real-time requirement, the images first downsized by the factor of 2 in each dimension (i.e. width and height). Thus, the final image resolution using in this study is at 320x240 resolution.

Disparity Image: Raw stereo disparity information included in the dataset is provided in the 16-bit subpixel values. The image size of the disparity image is in the same size as the RGB image.

Feature Vector Dimension: The feature vector dimension is set to 38 comprises of intensity features (2 features), color representation (4 features) and texture information (32 features). This Feature vector is extracted for all the patches of the input images.

Color and Intensity Features: For color features the first and second moments (i.e mean and variance) of the CIE Lab color space are used for each patch.

Texture Features: Using the mean and standard deviation of a bank of Gabor filters for four different orientations and four different scales, the texture attributes are constructed.

Ground plane and obstacle detection accuracy

As it is discussed earlier at the section 5.3.2, the first step of self-supervised, near-to-far learning (SSL) is to estimate ground plane and obstacles from stereo information in the near field of the vehicle. However, this estimation often is noisy and contains mislabeling due to many variations of ground plane and obstacles geometry characteristics in outdoor and unstructured environments. Thus, the performance of the proposed ground plane and obstacle detection module is evaluated against the hand-labeled ground truth image on a frame-by-frame basis. Since the vehicle may not encounter any positive examples (i.e., non-traversable terrains) during some parts of its course G-means is not a good evaluation metric with regards to zero True positives (TP) rates in some frames. Hence, the conventional overall accuracy is used as an evaluation metric in this step only. Note that the ground plane and obstacle detection module evaluates only on the near field of the vehicle (less than 7 meters). The numerical results are given in Table 5.5. The mean and standard deviation are reported across all datasets for 10 repeated runs. Figure 5.15 also shows the ground plane and obstacle detection G-means on a frame-by-frame basis across all datasets. Moreover, Figure 5.17 to Figure 5.22 show representative outputs of the ground plane and obstacle estimation method projected back to the original RGB images

in the near field of the vehicle. Green color indicates ground plane while obstacles are labeled as red color.

TABLE 5.5 GROUND PLANE AND OBSTACLE DETECTION ACCURACY (G-MEANS) IN EACH DATASET.

Dataset	Ground plane and obstacle detection accuracy (overall accuracy)
DS1A	0.9766 $\pm$ 0.019
DS1B	0.9574 $\pm$ 0.031
DS2A	0.9732 $\pm$ 0.017
DS2B	0.9608 $\pm$ 0.028
DS3A	0.9774 $\pm$ 0.014
DS3B	0.9761 $\pm$ 0.011
Overall	0.9702 $\pm$0.02

The results indicate that stereo supervision has the overall accuracy of 0.9702 in the near field of the vehicle. Thus, it confirms the fact that the stereo auto-labeling module introduces mislabeling errors in the training data that have to be identified and handled correctly. Although it seems the mislabeling error is not significant, next experimental results verify that coping with mislabeling error is critical for optimal classifier training, which results in better far field traversability prediction. The level of class labels noises are illustrated by the percentage of noisy stereo labels in each dataset. This is calculated in each training dataset according to following equation:

$$\text{Noise level} = \frac{\text{FN} + \text{FP}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \quad (5.32)$$

TABLE 5.6 MEAN AND STANDARD DEVIATION PERCENTAGE OF NOISY LABELS IN EACH DATASET.

Dataset	Noise Level
DS1A	2.34% $\pm$ 1.98
DS1B	4.26% $\pm$ 3.18
DS2A	2.66% $\pm$ 1.78
DS2B	3.91% $\pm$ 2.81
DS3A	2.26% $\pm$ 1.42
DS3B	2.38% $\pm$ 1.18
Overall	2.96% $\pm$2.05

(a) Ground/Obstacle estimation accuracy DS1A

(b) Ground/Obstacle estimation accuracy DS1B

(c) Ground/Obstacle estimation accuracy DS2A

(d) Ground/Obstacle estimation accuracy DS2B

(e) Ground/Obstacle estimation accuracy DS3A

(f) Ground/Obstacle estimation accuracy DS3B

Figure 5.16 Plots by dataset of ground plane and obstacle estimation accuracy (mean and std) in the near field of the vehicle (less than 7 meters). The results verify the existence of noise in the ground plane and obstacle estimation provided by the stereo supervision.

(a) RGB Image

(b) Stereo Disparity Image

(c) Hand-Labeled Ground Truth image

(d) Ground Plane and Obstacles Overlaid on RGB Image

Figure 5.17 Representative output of the ground plane and obstacle estimation algorithm in the near field of the vehicle (less than 7 meters), (DS1A Frame 30).

Figure 5.18 Representative output of the ground plane and obstacle estimation algorithm in the near field of the vehicle (less than 7 meters), (DS1B Frame 80).

Figure 5.19 Representative output of the ground plane and obstacle estimation algorithm in the near field of the vehicle (less than 7 meters), (DS2A Frame 39).

Figure 5.20 Representative output of the ground plane and obstacle estimation algorithm in the near field of the vehicle (less than 7 meters), (DS2B Frame 71).

Figure 5.21 Representative output of the ground plane and obstacle estimation algorithm in the near field of the vehicle (less than 7 meters), (DS3A Frame 17).

Figure 5.22 Representative output of the ground plane and obstacle estimation algorithm in the near field of the vehicle (less than 7 meters), (DS3B Frame 49).

Auto labeling versus confidence-based auto labeling

Once the ground plane and obstacles are estimated in the disparity image, traversability labels are assigned by stereo supervision to each primitive in the near field of the input image. As it is mentioned in the section 5.3.3 and verified in the Table 5.6, this label propagation may contain noises due to several reasons. A new method is introduced called confidence-based auto labeling to identify the mislabeling data followed by two algorithms for handling them efficiently. One handling algorithm is to discard those input stereo label with low confidence value called “*elimination method*”. Second approach is to

assign probabilistic confidences weights on each point of training data called *probabilistic weighing method*.

Over all six datasets, the *elimination method* and *probabilistic weighing* perform significantly better compared to the standard auto labeling method with no error handling. Table 5.7 summarizes the performance of the Self-Supervised Learning (SSL) with three different auto labeling algorithms.

Moreover, Figure 5.23 is provided as an example of *elimination method* to compare the result of auto labeling before and after the confidence-based method by means of visual perception. The lower parts of the hay bales in the image are detected as ground plane in baseline method (auto labeling method with no error handling), while they are identified as mislabeling patches and eliminated in the proposed confidence-based auto labeling algorithm (*elimination method*) correctly. The ground plane (shown in green) and obstacles (shown in red) detected in the near field are projected back to the original RGB image. Figure 5.24 illustrates the performance of the SSL method with three different auto labeling algorithms frame-by-frame. The overall mean results for all datasets are given in Figure 5.25 respectively. This analysis shows that the SSL with confidence-based auto labeling performs significantly better than the baseline (SSL with basic auto labeling (no error handling)). In addition, it is shown that the SSL is sensitive to noise and errors in its auto labeling module. Finally, Figure 5.26 represents the outputs of the SSL using three different auto labeling algorithms for qualitative analysis. Note that the obstacles in the left part of the image is labeled incorrectly in the baseline while the SSL with confidence based auto labeling algorithms can detect those obstacle regions correctly.

TABLE 5.7 THE PERFORMANCE (G-MEANS) OF THE SELF-SUPERVISED LEARNING (SSL) WITH THREE DIFFERENT AUTO LABELING ALGORITHMS IN EACH DATASET.

Auto Labeling		Datasets						Overall
		DS1A	DS1B	DS2A	DS2B	DS3A	DS3B	
SSL with Basic auto labeling (no error handling)	G-means	0.64 (± 0.02)	0.78 (± 0.01)	0.85 (± 0.01)	0.78 (± 0.01)	0.93 (± 0.01)	0.76 (± 0.01)	0.79 (± 0.09)
SSL with Confidence-based auto labeling (elimination method)	G-means	0.76 (± 0.03)	0.86 (± 0.01)	0.89 (± 0.01)	0.83 (± 0.02)	0.94 (± 0.007)	0.85 (± 0.01)	0.85 (± 0.06)
SSL with Confidence-based auto labeling (probabilistic weighting method)	G-means	0.77 (± 0.03)	0.84 (± 0.01)	0.88 (± 0.02)	0.83 (± 0.01)	0.94 (± 0.006)	0.83 (± 0.01)	0.84 (± 0.05)

Figure 5.23 Representative output of the Basic auto labeling method and the proposed confidence-based auto labeling (elimination method), (Frame 40 DS1B). The mislabeled regions belong to footlines of hay bales are removed from the training labeled data in our proposed confidence based auto-labeling method.

(a) Different Auto Labeling Methods DS1A

(b) Different Auto Labeling Methods DS1B

(c) Different Auto Labeling Methods DS2A

(d) Different Auto Labeling Methods DS2B

(e) Different Auto Labeling Methods DS3A

(f) Different Auto Labeling Methods DS3B

Figure 5.24 Plots by dataset of SSL algorithm with three different auto labeling algorithms. The results depict that the proposed confidence based auto labeling methods outperform the baseline (auto labeling method with no error handling).

(a) SSL with different Auto Labeling Methods
DS1A

(b) SSL with different Auto Labeling Methods
DS1B

(c) SSL with different Auto Labeling Methods
DS2A

(d) SSL with different Auto Labeling Methods
DS2B

(e) SSL with different Auto Labeling Methods
DS3A

(f) SSL with different Auto Labeling Methods
DS3B

Figure 5.25 The overall mean results for the SSL with three auto labeling algorithms across all datasets.

Figure 5.26 Representative outputs of the SSL using three different auto labeling methods (frame 49 DS1B).

Auto Labeling versus Manual Labeling

This section examines whether having access to all correct information in the near field of the scene will improve the classification performance (Manual Labeling). To answer this question, the traversability labels of all patches in the near field of the vehicle is determined by using the hand-labeled ground truth images (Figure 5.27). Thus, all the patches in the image are manually labeled as either traversable or non-traversable. Only the lower half of the hand-labeled ground truth images is used as supervisions to make the manual labeling comparable to the auto labeling method. In contrast, stereo supervision is used to find out traversability labels of each patch in the near field (auto labeling), and then these labeled patches are linked with their visual appearance feature vectors to the

base classifier (SVM-Linear) to compare these two methods. From the result in Table 5.8 one can see that the manual labeling method depicts significant improvement compared to auto labeling method. Note that since in manual labeling the SSL method does not use any random calculation the standard deviation across 10 runs is almost zero.

An important conclusion from this study is that the maximum possible accuracy that one can achieve using the conventional SSL methods with the current fixed parameters (i.e. feature descriptors, base classifier) is bounded to the maximum accuracy of the SSL with manual labeling. Figure 5.27 and Figure 5.28 compare the outputs of these two methods in term of quality. Additionally, the G-means values of the SSL using manual labeling and auto labeling are illustrated on a frame-by-frame basis across all datasets in Figure 5.29.

TABLE 5.8 NUMERICAL RESULTS FOR SELF-SUPERVISED LEARNING (SSL) WITH AUTO LABELING VS. MANUAL LABELING IN EACH DATASET (G-MEANS).

Dataset	SSL with Auto Labeling	SSL with Manual Labeling
DS1A	0.64(± 0.02)	0.85(± 0.0)
DS1B	0.78(± 0.01)	0.94(± 0.0)
DS2A	0.85(± 0.01)	0.89(± 0.0)
DS2B	0.78(± 0.01)	0.82(± 0.0)
DS3A	0.93(± 0.01)	0.95(± 0.0)
DS3B	0.76(± 0.01)	0.86(± 0.0)
Overall	0.79(± 0.09)	0.88(± 0.05)

Figure 5.27 The hand-labeled ground truth image provides supervisory labels for training data (left image), output of SSL using manual labeling (right image) (Frame 74 DS1B).

Figure 5.28 The stereo supervision provides traversability labels for training data (left image), output of SSL using Auto labeling (right image) (Frame 74 DS1B).

Figure 5.29 Plots by dataset of manual labeling vs. auto labeling. G-means values are reported on a frame-by-frame basis.

Top-Down Feature Attention

Once the auto labeling module provides the stereo supervision labels for all the input training data, these labels are then associated with their visual appearance features to generate feature vectors for classifier training. The current self-supervised learning algorithms add all the visual features extracted from the each primitive of the input images (such as intensity, color, or texture) to a feature vector with the same or different predefined weights. However, in most outdoor and unstructured environments, one or some of the attributes of the feature vectors may be more informative for discrimination of the terrain from non-terrain regions and this information is unknown in advance. To solve the mentioned limitation a new top-down attention model is presented that enhances the feature vectors by assigning adaptive gains representing the saliency of each feature in the feature vector for each frame. The proposed Top-Down attention model is compared with the best known Top-Down model proposed by Navalpakkam and Itti [138]. A table of numerical results for the SSL with two Top-Down models is given in Table 5.9.

TABLE 5.9 NUMERICAL RESULTS FOR THE SSL WITH NAVALPAKKAM AND ITTI [138] TOP-DOWN MODEL VS. THE PROPOSED TOP-DOWN MODEL IN EACH DATASET (G-MEANS).

Dataset	Navalpakkam and Itti Top-Down Model [138]	The Proposed Top-Down Model
DS1A	0.75(± 0.03)	0.80(± 0.04)
DS1B	0.84(± 0.01)	0.86(± 0.02)
DS2A	0.88(± 0.01)	0.88(± 0.01)
DS2B	0.82(± 0.02)	0.86(± 0.03)
DS3A	0.94(± 0.007)	0.95(± 0.008)
DS3B	0.85(± 0.01)	0.87(± 0.01)
Overall	0.84(± 0.06)	0.87(± 0.04)

For most datasets (except DS2A), SSL with the proposed Top-down model performs better than SSL with Navalpakkam and Itti Top-Down model [138]. In DS2A both the Navalpakkam and Itti Top-Down model [138] and the proposed Top-down model perform similarly in term of G-means values. However, the main effect of Top-Down attention is

to direct attention quickly and reliably to the salient features; therefore, the classifier spends less time to learn the optimum model from training data which results in more efficient processing time. In order to study the effect of adding Top-down model to SSL the number of Support Vectors (SVs) in the learned model from training data are explored at each frame. It should be mentioned that since the classification time is primarily determined by the number of SVs [159], sparsity in the number of support vectors provides the opportunity to considerably reduce the memory and time requirements of SVM classifier [160]. Figure 5.30 depicts the overall number of SVs utilized in SSL with or without Top-Down (TD) attention model across all datasets. Overall, the SSL with the proposed Top-down model in this chapter significantly reduces the number of SVs, which results in less computational time and increases memory efficiently. The overall comparison of SSL before and after adding Top-Down (TD) models indicates better classification performance, as well as, significantly less number of SVs in the learned model. Note that both Top-Down (TD) models outperform the SSL without any Top-Down biasing, while the proposed Top-Down model requires almost the half the number of SVs used in the Top-Down model proposed by Navalpakkam and Itti [138].

Figure 5.30 The overall mean number of Support Vectors (SVs) utilized in SSL method with or without Top-Down (TD) attention across all datasets.

Cost Sensitive Learning vs. Cost Insensitive Learning

The impact of unbalanced data problem varies based on the level of skewed class distribution in the training dataset. In order to depict that the distribution of the class labels in the training data provided by the stereo supervision in the near field of the vehicle are subjected to unbalanced data problem, the positive class labels ratios are calculated for each frame across all datasets and they are reported in the Figure 5.31. The ratio is calculated based on the percentage of positive class labels over total number of samples in the training data. As it can be seen, the level of unbalanced data ratio varies across different frames within a dataset. A ratio close to 50% is considered as balanced data while a ratio close to zero or above 70% shows high unbalanced level (skewed ratio). The results are summarized in the Table 5.10 that shows the overall percentage of positive class labels varies significantly across different datasets. Therefore, any prior cost values for the classifier penalties will be failed. In contrast, the method proposed in this work uses an adaptive biasing method that first calculates two penalty values based on the distribution of the class labels in unbalanced training dataset of each input frame and then assigns those penalties to bias the classifier. Figure 5.32 illustrates an example of high unbalanced level in the training data. As it is shown, the cost sensitive learning SVMs performs significantly better to classify the far field terrain than the conventional cost insensitive learning SVMs.

TABLE 5.10 OVERALL PERCENTAGE OF POSITIVE CLASS LABELS ACROSS ALL DATASETS.

Dataset	Overall No. of Positive Samples	Overall No. of Negative Samples	Total No. of Samples	Overall Unbalanced ratio %
DS1A	6.4	154.1	160.5	3.76%
DS1B	36.4	149.27	185.67	18.28%
DS2A	19.11	174.45	193.5	9.38%
DS2B	20.04	171.58	191.6	10.15%
DS3A	80.65	141.22	221.87	35.60%
DS3B	30.52	142.57	173.69	17.75%

Figure 5.31 Positive class labels ratio frame-by-frame across all datasets.

Figure 5.32 An example of high unbalanced level in the training data (Frame 12 DS2A).

Novelty Detection vs. Two-class Classification

Although the SSL with cost sensitive learning can handle different level of unbalanced training data, still it requires the existence of both class labels in the training datasets. While in real-life outdoor unstructured environments, the vehicle may not encounter any positive examples (i.e., non-traversable terrain) during some parts of its course. Therefore, the two-class classification cannot be used for traversability prediction since it needs both class examples. This limitation is shown in Figure 5.33. As it is shown, only one class examples are observed in the near field of vehicle by stereo vision systems (Figure 5.33 (a)). Therefore, if this information is passed to the classifier (conventional two-class

Figure 5.33 An example of near field training data that contains only negative examples (Frame 1 DS2A).

SVMs), it will not be able to predict any pixel with positive class labels (i.e. all pixels are classified as ground plane or green colors (Figure 5.33 (c)). However, the SSL with proposed One-Class SVMs can handle this situation efficiently by predicting both examples of traversable and non-traversable points in far field of the input frame (Figure 5.33 (d)).

Baseline SSL Method vs. The Proposed SSL Framework

In previous sections, the impacts of different parts of the proposed SSL framework were investigated individually through extensive numerical evaluation and qualitative comparison. However, the final goal is to verify the effectiveness of the proposed SSL

framework compared to the baseline SSL method. Therefore, in this section, both baseline SSL methods and proposed SSL framework are applied to the six datasets and results are shown frame-by-frame across all datasets in Figure 5.34. The numerical results is also summarized in Table 5.11. As it is shown the proposed SSL framework significantly outperforms the baseline SSL method in all datasets.

TABLE 5.11 NUMERICAL RESULTS FOR PROPOSED SELF-SUPERVISED LEARNING (SSL) FRAMEWORK VERSUS BASELINE SSL METHOD IN EACH DATASET (G-MEANS).

Dataset	Baseline SSL Method	Proposed SSL Framework
DS1A	0.64(± 0.02)	0.90(± 0.01)
DS1B	0.78(± 0.01)	0.92(± 0.003)
DS2A	0.85(± 0.01)	0.93(± 0.003)
DS2B	0.78(± 0.01)	0.90(± 0.003)
DS3A	0.93(± 0.01)	0.95(± 0.005)
DS3B	0.76(± 0.01)	0.89(± 0.008)
Overall	0.79(± 0.09)	0.91(± 0.02)

Figure 5.34 Proposed SSL Framework versus baseline SSL Method frame-by-frame across all datasets.

Finally, Figure 5.35 to Figure 5.40 show six qualitative examples of self-supervised learning applied to terrain classification. The original RGB image, hand labeled ground truth overlaid in original image, baseline SSL output and the proposed SSL framework output are shown for the frame 50 across all datasets. Red color indicates non-traversable terrains and green color indicates traversable points in the visual field of the vehicle. The middle frames (frame number 50) across all datasets are selected to remove any bias towards any particular frame number in each dataset. Overall, the proposed SSL framework qualitatively outperforms the baseline SSL.

(a) RGB Image

(b) Hand labeled ground truth

(c) Baseline SSL Method

(d) Proposed SSL Framework

Figure 5.35 Representative outputs of the baseline SSL Method and the proposed SSL Framework, (Frame 50 DS1A).

(a) RGB Image

(b) Hand labeled ground truth

(c) Baseline SSL Method

(d) Proposed SSL Framework

Figure 5.36 Representative outputs of the baseline SSL Method and the proposed SSL Framework, (Frame 50 DS1B).

Figure 5.37 Representative outputs of the baseline SSL Method and the proposed SSL Framework, (Frame 50 DS2A).

Figure 5.38 Representative outputs of the baseline SSL Method and the proposed SSL Framework, (Frame 50 DS2B).

Figure 5.39 Representative outputs of the baseline SSL Method and the proposed SSL Framework, (Frame 50 DS3A).

Figure 5.40 Representative outputs of the baseline SSL Method and the proposed SSL Framework, (Frame 50 DS3B).

5.5 Chapter Summary

In this chapter, a new Self-Supervised Learning framework for vision-based autonomous Unmanned Ground Vehicles (UGVs) was described. At first, the biological motivation of the proposed framework is explained followed by the current state of the art in mimicking such intelligent behaviors. Next, a new Self-Supervised Learning framework was explained in detail that addressed several challenging aspects of baseline self-supervised, near-to-far learning methods.

The learning framework was able to robustly identify the traversable and non-traversable points in the near field of the vehicle. Next, those points that are susceptible to be mislabeled were identified and handled efficiently by proposed confidence-based auto labeling techniques. Next, a new top-down attention model was presented that integrates the visual appearance features with a set of adaptive gains representing saliency of each of them in the feature vector. Moreover, an online adaptive learning mechanism was proposed to address the current limitation of standard two-class classification. The learning mechanism integrated several learning strategies into a unified framework and selected the proper learning method based on the distribution of class labels in each frame training data in real-time. Finally, a series of extensive qualitative and quantitative experimental results were presented to demonstrate the effectiveness, robustness, and accuracy of the proposed SSL framework using the natural datasets logged live during runs of vehicles competing in the DARPA-LAGR program.

Chapter 6

CONCLUSION AND RECOMMENDATION

This thesis sought to design artificial intelligent perceptual systems for Unmanned Ground Vehicles (UGVs) that currently exists only in biological vision systems. To achieve this goal, several computational models of early, mid and high level visual processing were proposed through investigation of the computational principles behind the Human Visual System (HSV). First, a novel estimation scheme was presented to computationally model the orientation selectivity characteristic discovered at the majority of neurons in the early stages of the primary visual cortex (V1). The novel scheme for the estimation of Local Dominant Orientation (LDO) developed in this work was based on the psychophysical findings that basic visual information such as orientation of external stimuli are encoded through distributed activities within population of correlated neurons in V1. Next, the LDO model was employed to present a new method, which mimics the visually guided behavior of human drivers for autonomous UGVs. The proposed approach formed a midlevel computational model utilizing monocular vision information similar to human perception to detect upcoming road directions with respect to the vehicle. Finally, based on several empirical studies on human visual processing during navigation, a novel and robust high level visual intelligent framework was developed for understanding the

traversability of new, unknown environments by integrating learning mechanism into the visual perception system. The core components of this framework were a new mechanism of confidence-based auto labeling, a computational model of Top-down (TD) attention mechanism found on visual cortex, and online adaptive learning mechanism.

The rest of this chapter summarizes the main results presented in this thesis and discusses the contributions made to the field of autonomous Unmanned Ground Vehicles. Some research directions opened as a result of the contributions are then discussed. This chapter is organized as follows. In section 6.1, we list the contributions per chapter in more detail. Section 6.2 briefly discusses the scope for future research.

6.1 Summary

In Chapter 1, we introduced the main objective of this thesis, which was to develop biologically inspired visual intelligence for autonomous Unmanned Ground Vehicles. The motivation behind this thesis was the way human navigates in dynamic outdoor environments mainly using its visual system.

In Chapter 2, related work in the area of Unmanned Ground Vehicles (UGVs) was reviewed. The focus of this chapter was on the perception systems of autonomous UGVs. A brief review of machine vision and machine learning techniques applied to Unmanned Ground Vehicles was described.

Chapter 3 presented a novel biologically inspired mechanism for Local Dominant Orientation (LDO) estimation. First, we described the orientation selectivity characteristic discovered at the majority of neurons in the primary visual cortex (V1) of the mammalian brain. Based on this orientation tuning properties, 2D Gabor filters were utilized to model Respective Field (RF) of individual simple cells in V1. Then, responses of pairs of simple cells were combined and modeled by the local Gabor energy function to represent the complex cells RF in V1. Next, activities of ensemble of Gabor energy responses were

grouped together by introducing the opponency mechanism to the distributed population coding in accordance with the activities of the population of neurons in V1. The proposed method was implemented on both synthetic and natural datasets to evaluate its robustness and accuracy. Apart from being biologically plausible, the results demonstrated several advantages of the approach including higher accuracy, more robustness to input noise and scale variations, contrast invariance, and finally less computational cost compared to the state of the art filter-based LDO estimation methods.

The LDO model was the essential part of our proposed method for the vanishing point estimation based on the visual gaze behavior of drivers in Chapter 4. First, we addressed several empirical studies on human drivers gaze patterns in the open driving scenarios, and we concluded that when human drivers become more familiar with the road situations, they prefer to look at the far field where the road edges converge (called the vanishing point) to anticipate the upcoming road trajectory and the car steering with maximal lead time. Hence, we proposed a novel methodology based on LDO estimation model for fast estimation of the vanishing point in challenging and unstructured roads. The key attributes of the methodology consisted of using proposed LDO estimation model to estimate the local dominant orientation at each pixel location in the image plane, weighting of each pixel by a sine function of its dominant orientation, and an adaptive distance based voting scheme for estimation of the vanishing point. Robustness, and speed were further enhanced through exclusion of sky pixels in the processing by our proposed skyline detection technique. A series of quantitative and qualitative analyses were presented using natural data sets from the DARPA Grand Challenge projects to demonstrate the effectiveness and accuracy of the proposed methodology.

Chapter 5 addressed another aspect of visual guided behavior of drivers. Psychologists believe that human can adapt to changing environments concurrently by incorporating knowledge of near field into far field, using its ability that can continuously learn the association between terrain geometry characteristics and appearances in the near field and then, transfer such knowledge to make predictions about the upcoming road to the far

field of its visual field. Hence, we proposed a robust Self-Supervised Learning (SSL) framework for understanding the traversability of the outdoor unstructured terrains by integrating learning mechanism into visual perception system. The proposed SSL framework is composed of a robust ground plane and obstacles estimation, confidence-based auto labeling, biologically inspired top-down attention model, and an adaptive online classifier. Finally, a series of extensive qualitative and quantitative experimental were carried out on the natural datasets logged during DARPA-LAGR programs to demonstrate the effectiveness, robustness, and accuracy of the proposed learning framework.

6.2 Recommendations for Further Research

A few possible promising directions for further areas of research are:

- In Chapter 3, we have considered estimating a dominant orientation at each pixel location for the purpose of Local Dominant Orientation (LDO) estimation. Future direction can be to detect multiple orientations, like corners, t-junctions, cross or multiple overlapping objects. It would be interesting to incorporate knowledge of mixture orientations into the proposed model. We believe that combining the current proposed LDO model with hierarchical processing model will help to estimate more complex features for applications such as object recognition.
- The proposed vanishing point estimation model presented in Chapter 4 performs on frame-to-frame basis while when the vehicle moves one can use the information from the sequence of frames leading up to the current frame to improve estimation of the vanishing point location. Therefore, the next possible extension is to track the locations of the vanishing point over a sequence of frames using particle filtering techniques. This cannot be reliably and accurately achieved unless one has access to the vehicle motion information, which was not available at the current standard outdoor unstructured dataset (DARPA Grand Challenge)

that we have used in this thesis. Another way to estimate which way the vehicle is heading to is to use the optical flow information from the sequence of the images. Therefore, in the future work information from both the vanishing point and optical flow estimators are integrated to build more efficient vanishing point estimation method for UGVs.

- Although the Self-Supervised Learning framework proposed in Chapter 5 can adapt to the changing environments concurrently, it lacks memory or history of past information. Future work is to investigate strategies, and algorithms for an online, self-supervised learning algorithm that exploits multiple frames to develop adaptive models that can classify multitude of terrain types.

APPENDIX A

LDO ESTIMATION ON NATURAL IMAGES

Figure A.1 LDO estimations are projected back to the sample image from Brodatz textures.

Figure A.2 LDO estimations are projected back to the sample image from OuTex texture.

APPENDIX B

MANUAL VP DETECTOR

Manual VP detector software is a GUI software written by the author in Microsoft visual C++ 6.0 to facilitate the manual labeling of the Vanishing Point (VP) ground truth by human subjects.

Each image of the dataset is shown to the subject in its full resolution (320x240) and ask them to mark the vanishing point in each image. Each subject can mark the vanishing point location in an image by the left click of mouse. For each image, a vector stores the pixel location (x and y location) of the vanishing point labeled by the subject. Only one location is accepted for each image. Therefore, we ask the users if they by any chance select more than one locations or if they do not satisfy with the current selected location, they can remove the selected VP in each image and modify the correct location accordingly. Once they satisfy with the marked VP they can proceed to the next frame. A screenshot of the GUI and the output of the Manual VP detector by a user is shown in the Figure B.1 and Figure B.2.

Figure B.1 Screenshot of the GUI for Manual VP detector.

Figure B.2 The output of the Manual VP detector shown in a red dot.

APPENDIX C

VANISHING POINT ESTIMATION

Figure C.1 The outputs of the vanishing point estimation algorithm.

APPENDIX D

AUTHOR'S RELATED PUBLICATIONS

D.1 Journal Papers

P. Moghadam, J. A. Starzyk, W. S. Wijesoma, "Fast Vanishing-Point Detection in Unstructured Environments," IEEE Transactions on Image Processing, vol.21, no.1, pp.425-430, Jan. 2012.

D.2 Conference Papers

P. Moghadam, S. Salehi, W. S. Wijesoma, "Computationally Efficient Navigation System for Unmanned Ground Vehicles," IEEE TePRA 2011, MA, USA, 2011, pp. 133-138.

P. Moghadam, J. A. Starzyk, W. S. Wijesoma, "Computational Model of Driver Gaze Direction" Decade of the Mind VI conference 18-20 OCT, 2010.

P. Moghadam, W. S. Wijesoma, M.D.P. Moratuwage, "Towards A Fully-Autonomous Vision-based Vehicle Navigation System in Outdoor Environments," IEEE ICARCV, pp. 597-602, 2010.

M.D.P. Moratuwage, W.S. Wijesoma, B. Kalyan, N. M. Patrikalakis, P. Moghadam
“Collaborative Multi-Vehicle Localization and Mapping in High Clutter Environments,”
IEEE ICARCV, pp. 1422-1427, 2010.

P. Moghadam, W. S. Wijesoma, “Online, Self-Supervised Vision-Based Terrain
Classification in Unstructured Environments,” IEEE SMC, pp.3100-3105, 11-14 Oct.
2009.

P. Moghadam, W. S. Wijesoma, J. F. Dong, “Improving Path Planning and Mapping
Based on Stereo Vision and Lidar,” IEEE ICARCV, pp. 384-389, 2008.

BIBLIOGRAPHY

- [1] *Packbot Robots*. Available: <http://www.irobot.com/>
- [2] *Frontline Robotics Teleoperated UGV (TUGV)*. Available: <http://www.frontline-robotics.com/>
- [3] *VisLab's autonomous car* Available: <http://vislab.it/>
- [4] *Google driverless car*. Available: http://en.wikipedia.org/wiki/Google_driverless_car
- [5] 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, pp. 62-81, 1998.
- [6] W. Wijesoma, K. Kodagoda, and A. Balasuriya, "Road-boundary detection and tracking using ladar sensing," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 456-464, 2004.
- [7] K. Kaliyaperumal, S. Lakshmanan, and K. Kluge, "An algorithm for detecting roads and obstacles in radar images," *IEEE Transactions on Vehicular Technology*, vol. 50, pp. 170-182, 2001.
- [8] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, "Vision and navigation for the Carnegie-Mellon Navlab," *IEEE transactions on pattern analysis and machine intelligence*, vol. 10, pp. 362– 372, 1988.
- [9] K. Kodagoda, W. Wijesoma, and A. Balasuriya, "CuTE: curb tracking and estimation," *IEEE Transactions on Control Systems Technology*, vol. 14, pp. 951-957, 2006.
- [10] D. Pomerleau and T. Jochem, "Rapidly adapting machine vision for automated vehicle steering," *IEEE expert*, vol. 11, pp. 19-27, 1996.
- [11] M. Turk, D. Morgenthaler, K. Gremban, and M. Marra, "VITS-A vision system for autonomous land vehicle navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 342-361, 1988.

-
- [12] J. Crisman and C. Thorpe, "SCARF: a color vision system that tracks roads and intersections," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 49-58, 1993.
 - [13] J. Crisman and C. Thorpe, "UNSCARF-a color vision system for the detection of unstructured roads," in *IEEE International Conference on Robotics and Automation*, pp. 2496-2501, 1991.
 - [14] D. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," *In Advances in Neural Information Processing Systems*, pp. 305-313, 1989.
 - [15] C. Shoemaker and J. Bornstein, "The Demo III UGV program: a testbed for autonomous navigation research," in *IEEE International Symposium on Intelligent Control, Gaithersburg, MD.*, pp. 644-651, 1998.
 - [16] K. Iagnemma and M. Buehler, "Special issue on the darpa grand challenge, part 1," *Journal of Field Robotics*, vol. 23, pp. 461-652, 2006.
 - [17] K. Iagnemma and M. Buehler, "Special issue on the darpa grand challenge, part 2," *Journal of Field Robotics*, vol. 23, pp. 655-835, 2006.
 - [18] L. Jackel, *et al.*, "The DARPA LAGR program: Goals, challenges, methodology, and phase I results," *Journal of Field Robotics*, vol. 23, pp. 945-973, 2006.
 - [19] P. Bellutta, *et al.*, "Terrain perception for DEMO III," in *Intelligent Vehicles Symposium*, pp. 326-331, 2000.
 - [20] S. Thrun, *et al.*, "Stanley: The robot that won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, pp. 661-692, 2006.
 - [21] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 237-267, 2002.
 - [22] R. Aufrere, *et al.*, "Road sides recognition in non-structured environments by vision," in *IEEE Intelligent Vehicles Symposium*, pp. 329-334, 2004.
 - [23] M. Blas, M. Agrawal, A. Sundaresan, and K. Konolige, "Fast color/texture segmentation for outdoor robots," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 4078-4083, 2008.
 - [24] J. Zhang and H. Nagel, "Texture-based segmentation of road images," in *IEEE Intelligent Vehicles Symposium*, pp. 260-265, 1994.
 - [25] Y. Alon, A. Ferencz, and A. Shashua, "Off-road path following using region classification and geometric projection constraints," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 689-696 2006.
 - [26] H. Cheng, *et al.*, "Hierarchical Lane Detection for Different Types of Roads," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1349-1352, 2008.
 - [27] R. Pagnot and P. Grandjean, "Fast cross-country navigation on fair terrains," in *IEEE International Conference on Robotics and Automation*, pp. 2593-2598, 1995.
 - [28] A. Talukder, R. Manduchi, A. Rankin, and L. Matthies, "Fast and reliable obstacle detection and segmentation for cross-country navigation," in *IEEE Intelligent Vehicles Symposium*, pp. 610-618, 2002.
-

-
- [29] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous Robots*, vol. 18, pp. 81-102, 2005.
 - [30] C. Dima, M. Hebert, and A. Stentz, "Enabling learning from large datasets: Applying active learning to mobile robotics," in *IEEE International Conference on Robotics and Automation*, pp. 108-114, 2004.
 - [31] Z. Ghahramani, "Unsupervised Learning," in *Advanced Lectures on Machine Learning*, vol. 3176, O. Bousquet, U. von Luxburg, and G. Rätsch, Eds., ed: Springer Berlin / Heidelberg, 2004, pp. 72-112.
 - [32] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *the Seventeenth International Conference on Machine Learning (ICML)*, pp. 593-600, 2005.
 - [33] A. Howard, *et al.*, "Towards learned traversability for robot navigation: From underfoot to the far field," *Journal of Field Robotics*, vol. 23, pp. 1005-1017, 2006.
 - [34] H. Dahlkamp, *et al.*, "Self-supervised monocular road detection in desert terrain," in *Robotics Science and Systems 2006*.
 - [35] C. Wellington and A. Stentz, "Learning predictions of the load-bearing surface for autonomous rough-terrain navigation in vegetation," in *the International Conference on Field and Service Robotics*, pp. 83-92, 2003.
 - [36] A. Lookingbill, *et al.*, "Reverse optical flow for self-supervised adaptive autonomous robot navigation," *International Journal of Computer Vision*, vol. 74, pp. 287-302, 2007.
 - [37] R. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, pp. 786-804, 1979.
 - [38] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 8, pp. 460-473, 2007.
 - [39] A. Guérin-Dugué and A. Oliva, "Classification of scene photographs from local orientations features," *Pattern Recognition Letters*, vol. 21, pp. 1135-1140, 2000.
 - [40] C. Grigorescu, N. Petkov, and M. Westenberg, "Contour and boundary detection improved by surround suppression of texture edges," *Image and Vision Computing*, vol. 22, pp. 609-622, 2004.
 - [41] F. Lovergine, A. Branca, G. Attolico, and A. Distante, "Leather inspection by oriented texture analysis with a morphological approach," in *IEEE Proc. of International Conference on Image Processing*, pp. 669-671, 2002.
 - [42] D. Chetverikov, "Detecting defects in texture," in *International Conference on Pattern Recognition*, pp. 61-63, 1988.
 - [43] Y. Hamamoto, "A Gabor filter-based method for fingerprint identification," in *Intelligent Biometric Techniques in Fingerprint and Face Recognition (L.C. Jain, U. Halici et al., Eds.)*, pp. 137-151, 1999.
 - [44] D. Hubel and T. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, p. 106, 1962.
-

-
- [45] J. Bigun and G. Granlund, "Optimal orientation detection of linear symmetry," in *IEEE First International Conference Computer Vision*, pp. 433-438, 1987.
 - [46] B. Chaudhuri, P. Kundu, and N. Sarkar, "Detection and gradation of oriented texture," *Pattern Recognition Letters*, vol. 14, pp. 147-153, 1993.
 - [47] M. Kass and A. Witkin, "Analyzing oriented patterns," *Computer vision, graphics, and image processing*, vol. 37, pp. 362-385, 1987.
 - [48] A. Rao and B. Schunck, "Computing oriented texture fields," *CVGIP: Graphical Models and Image Processing*, vol. 53, pp. 157-185, 1991.
 - [49] J. Bigun, G. Granlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 775-790, 1991.
 - [50] X. Feng and P. Milanfar, "Multiscale principal components analysis for image local orientation estimation," in *the 36th Asilomar Conference on Signals, Systems and Computers*, pp. 478-482, 2003.
 - [51] W. Freeman and E. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 891-906, 2002.
 - [52] R. De Valois, D. Albrecht, and L. Thorell, "Cortical cells: bar and edge detectors, or spatial frequency filters," *Frontiers in visual science Eds. S. J. Cool and E. L. Smith III.*, pp. 544-556, 1978.
 - [53] R. De Valois, D. Albrecht, and L. Thorell, "Spatial frequency selectivity of cells in macaque visual cortex," *Vision Research*, vol. 22, pp. 545-559, 1982.
 - [54] D. Marr and S. Ullman, "Directional selectivity and its use in early visual processing," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, pp. 151-180, 1981.
 - [55] B. Ma, S. Lakshmanan, and A. Hero, "Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, pp. 135-147, 2000.
 - [56] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *Journal of the Optical Society of America A*, vol. 7, pp. 923-932, 1990.
 - [57] N. Petkov and P. Kruizinga, "Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells," *Biological cybernetics*, vol. 76, pp. 83-96, 1997.
 - [58] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *nature neuroscience*, vol. 2, pp. 1019-1025, 1999.
 - [59] A. Teich and N. Qian, "Comparison among some models of orientation selectivity," *Journal of Neurophysiology*, vol. 96, p. 404, 2006.
 - [60] J. Movshon, I. Thompson, and D. Tolhurst, "Spatial summation in the receptive fields of simple cells in the cat's striate cortex," *The Journal of Physiology*, vol. 283, p. 53, 1978.
 - [61] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A*, vol. 2, pp. 1160-1169, 1985.
-

-
- [62] T. Lee, "Image representation using 2D Gabor wavelets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 959-971, 1996.
 - [63] T. Lee, "Image representation using 2D Gabor wavelets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, pp. 959-971, 2002.
 - [64] D. A. Pollen, J. P. Gaska, and L. D. Jacobson, "Responses of simple and complex cells to compound sine-wave gratings," *Vision Research*, vol. 28, pp. 25-39, 1988.
 - [65] D. Pollen and S. Ronner, "Phase relationships between adjacent simple cells in the visual cortex," *Science*, vol. 212, pp. 1409-1411, 1981.
 - [66] L. Abbott, "Decoding neuronal firing and modelling neural networks," *Quarterly reviews of biophysics*, vol. 27, pp. 291-331, 1994.
 - [67] A. Georgopoulos, A. Schwartz, and R. Kettner, "Neuronal population coding of movement direction," *Science*, vol. 233, p. 1416, 1986.
 - [68] G. Kreiman, C. Koch, and I. Fried, "Category-specific visual responses of single neurons in the human medial temporal lobe," *nature neuroscience*, vol. 3, pp. 946-953, 2000.
 - [69] A. Pouget, K. Zhang, S. Deneve, and P. Latham, "Statistically efficient estimation using population coding," *Neural Computation*, vol. 10, pp. 373-401, 1998.
 - [70] N. Lüdtkke, R. Wilson, and E. Hancock, "Tangent fields from population coding," in *IEEE Int. Workshop Biologically Motivated Computer Vision*, pp. 727-734, 2000.
 - [71] A. Luczak, P. Barthó, and K. Harris, "Spontaneous events outline the realm of possible sensory responses in neocortical populations," *Neuron*, vol. 62, pp. 413-425, 2009.
 - [72] E. Adelson and J. Bergen, "Spatiotemporal energy models for the perception of motion," *Journal of the Optical Society of America A*, vol. 2, pp. 284-299, 1985.
 - [73] D. Heeger, *et al.*, "Motion opponency in visual cortex," *Journal of Neuroscience*, vol. 19, p. 7162, 1999.
 - [74] E. Simoncelli and D. Heeger, "A model of neuronal responses in visual area MT," *Vision Research*, vol. 38, pp. 743-761, 1998.
 - [75] A. Bovik, M. Clark, and W. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 55-73, 1990.
 - [76] F. Bianconi and A. Fernández, "Evaluation of the effects of Gabor filter parameters on texture classification," *Pattern Recognition*, vol. 40, pp. 3325-3335, 2007.
 - [77] D. Clausi and H. Deng, "Design-based texture feature fusion using Gabor filters and co-occurrence probabilities," *IEEE Transactions on Image Processing*, vol. 14, pp. 925-936, 2005.
 - [78] D. Ringach, M. Hawken, and R. Shapley, "Dynamics of orientation tuning in macaque primary visual cortex," *Nature*, vol. 387, pp. 281-284, 1997.
 - [79] I. Finn, N. Priebe, and D. Ferster, "The emergence of contrast-invariant orientation tuning in simple cells of cat visual cortex," *Neuron*, vol. 54, pp. 137-152, 2007.
 - [80] K. Mardia, P. Jupp, and E. Corporation, *Directional statistics*: Wiley Chichester, 2000.
-

-
- [81] A. Weber, "The USC-SIPI image database version 5," *USC-SIPI Report*, vol. 315, pp. 1-24, 1997.
 - [82] T. Ojala, *et al.*, "Outex-new framework for empirical evaluation of texture analysis algorithms," *Pattern recognition*, vol. 1, p. 10701, 2002.
 - [83] K. Takemura, J. Ido, Y. Matsumoto, and T. Ogasawara, "Drive monitoring system based on non-contact measurement system of driver's focus of visual attention," in *IEEE Symposium on Intelligent Vehicles*, pp. 581-586, 2003.
 - [84] M. Land, J. Horwood, and T. Cornwell, "Fast driving reduces eye movement amplitude and frequency," *Investigative Ophthalmology & Visual Science*, vol. 35, p. 2033, 1994.
 - [85] M. F. Land and D. Lee, "Where we look when we steer," *Nature*, vol. 369, pp. 742-744, 1994.
 - [86] R. R. Mourant and T. H. Rockwell, "Mapping eye-movement patterns to the visual scene in driving: An exploratory study," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 12, pp. 81-87, 1970.
 - [87] C. Serafin, "Preliminary examination of driver eye fixations on rural roads: Insight into safe driving behavior (Rep. No. UMTRI 85843)," *Ann Arbor: University of Michigan, Transport Research Institute*, 1993.
 - [88] T. Rockwell, C. Overby, and R. Mourant, "Drivers' eye movements: An apparatus and calibration," *Highway Research Record No.*, vol. 247, pp. 29-41, 1968.
 - [89] D. Shinar, E. D. McDowell, N. J. Rackoff, and T. H. Rockwell, "Field dependence and driver visual search behavior," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 20, pp. 553-559, 1978.
 - [90] S. D. Rogers, E. E. Kadar, and A. Costall, "Gaze patterns in the visual control of straight-road driving and braking as a function of speed and expertise," *Ecological Psychology*, vol. 17, pp. 19-38, 2005.
 - [91] C. Rother, "A new approach to vanishing point detection in architectural environments," *Image and Vision computing*, vol. 20, pp. 647-655, 2000.
 - [92] R. Schuster, N. Ansari, and A. Bani-Hashemi, "Steering a robot with vanishing points," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 491-498, 1993.
 - [93] T. Suttorp and T. Bucher, "Robust vanishing point estimation for driver assistance," in *IEEE Intelligent Transportation Systems Conference*, pp. 1550-1555, 2006.
 - [94] T. Tuytelaars, L. Van Gool, M. Proesmans, and T. Moons, "The cascaded Hough transform as an aid in aerial image interpretation," in *The Sixth International Conference on Computer Vision*, pp. 67-72, 1998.
 - [95] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-Snake," *Image and Vision computing*, vol. 22, pp. 269-280, 2004.
 - [96] J. Burns, A. Hanson, and E. Riseman, "Extracting straight lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 8, pp. 425-455, 1986.
 - [97] S. Barnard, "Interpreting perspective images," *Artificial Intelligence*, vol. 21, pp. 435-462, 1983.
-

-
- [98] A. Matessi and L. Lombardi, "Vanishing point detection in the hough transform space," *Euro-Par'99 Parallel Processing*, pp. 987-994, 1999.
 - [99] H. Kogan, R. Maurer, and R. Keshet, "Vanishing points estimation by self-similarity," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 755-761, 2009.
 - [100] F. Stentiford, "Attention-based vanishing point detection," in *IEEE International Conference on Image Processing*, pp. 417-420, 2006.
 - [101] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 470-477, 2004.
 - [102] C. Rasmussen, "Texture-based vanishing point voting for road shape estimation," in *British Machine Vision Conference*, 2004.
 - [103] H. Kong, J. Audibert, and J. Ponce, "Vanishing point detection for road detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 96-103, 2009.
 - [104] H. Kong, J. Audibert, and J. Ponce, "General Road Detection from a Single Image," *IEEE Transactions on Image Processing*, vol. 19, pp. 2211-2220, 2010.
 - [105] C. Rasmussen, "RoadCompass: following rural roads with vision+ lidar using vanishing point tracking," *Autonomous Robots*, vol. 25, pp. 205-229, 2008.
 - [106] 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, pp. 20-37, 2006.
 - [107] M. Nieto and L. Salgado, "Real-time vanishing point estimation in road sequences using adaptive steerable filter banks," in *Advanced Concepts for Intelligent Vision Systems (ACIVS07)*, pp. 840-848, 2007.
 - [108] S. Ettinger, M. Nechyba, P. Ifju, and M. Waszak, "Vision-guided flight stability and control for micro air vehicles," *Advanced Robotics*, vol. 17, pp. 617-640, 2003.
 - [109] S. Todorovic, M. Nechyba, and P. Ifju, "Sky/ground modeling for autonomous MAV flight," in *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1422-1427, 2003.
 - [110] W. O. Readinger, *et al.*, "Gaze-eccentricity effects on road position and steering," *Journal of Experimental Psychology Applied* vol. 8, pp. 247-258, 2002.
 - [111] F. C. Fortenbaugh, J. C. Hicks, L. Hao, and K. A. Turano, "High-speed navigators: Using more than what meets the eye," *Journal of Vision*, vol. 6, 2006.
 - [112] M. Wagner, J. C. Baird, and W. Barbaresi, "The locus of environmental attention," *Journal of Environmental Psychology*, vol. 1, pp. 195-206, 1981.
 - [113] M. Happold, M. Ollis, and N. Johnson, "Enhancing supervised terrain classification with predictive unsupervised learning," in *In G. S. Sukhatme, S. Schaal, W. Burgard, & D. Fox (Eds.), Robotics: Science and Systems II*, pp. 41-48, 2006.
 - [114] M. Bajracharya, *et al.*, "Autonomous off-road navigation with end-to-end learning for the LAGR program," *Journal of Field Robotics*, vol. 26, pp. 3-25, 2008.
-

-
- [115] K. Konolige, *et al.*, "Mapping, navigation, and learning for off-road traversal," *Journal of Field Robotics*, vol. 26, pp. 88-113, 2009.
 - [116] R. Hadsell, *et al.*, "Learning long-range vision for autonomous off-road driving," *Journal of Field Robotics*, vol. 26, pp. 120-144, 2009.
 - [117] M. Shneier, *et al.*, "Learning traversability models for autonomous mobile vehicles," *Autonomous Robots*, vol. 24, pp. 69-86, 2008.
 - [118] J. Mulligan and G. Grudic, "Editorial for Journal of Field Robotics—Special issue on machine learning based robotics in unstructured environments," *Journal of Field Robotics*, vol. 23, pp. 943-944, 2006.
 - [119] F. Ferrari, E. Grosso, G. Sandini, and M. Magrassi, "A stereo vision system for real time obstacle avoidance in unknown environment," in *IEEE International Workshop on Intelligent Robots and Systems*, pp. 703-708, 1990.
 - [120] R. Labayrade, D. Aubert, and J. P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation," in *IEEE Intelligent Vehicles Symposium*, pp. 646–651, 2002.
 - [121] L. Matthies and P. Grandjean, "Stochastic performance, modeling and evaluation of obstacle detectability with imaging range sensors," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 783-792, 2002.
 - [122] M. J. Procopio, J. Mulligan, and G. Grudic, "Learning terrain segmentation with classifier ensembles for autonomous robot navigation in unstructured environments," *Journal of Field Robotics*, vol. 26, pp. 145-175, 2009.
 - [123] D. Kim, S. M. Oh, and J. M. Rehg, "Traversability classification for ugv navigation: A comparison of patch and superpixel representations," in *International Conference on Intelligent Robots and Systems* pp. 3166-3173, 2007.
 - [124] M. J. Procopio, J. Mulligan, and G. Grudic, "Long-term learning using multiple models for outdoor autonomous robot navigation," in *International Conference on Intelligent Robots and Systems*, pp. 3158-3165, 2007.
 - [125] M. Bajracharya, *et al.*, "Autonomous off-road navigation with end-to-end learning for the LAGR program," *Journal of Field Robotics*, vol. 26, pp. 3-25, 2009.
 - [126] M. H. Nguyen and F. de la Torre, "Optimal feature selection for support vector machines," *Pattern Recognition*, vol. 43, pp. 584-591, 2010.
 - [127] PTGrey. Point grey research inc. Available: <http://www.ptgrey.com/>,
 - [128] 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, pp. 381-395, 1981.
 - [129] S. Se and M. Brady, "Ground plane estimation, error analysis and applications," *Robotics and Autonomous Systems*, vol. 39, pp. 59-71, 2002.
 - [130] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial Intelligence Review*, vol. 22, pp. 177-210, 2004.
 - [131] W. Hu, Y. Liao, and V. R. Vemuri, "Robust support vector machines for anomaly detection in computer security," in *the International Conference on Machine Learning and Applications (ICMLA)*, 2003.
 - [132] C. Brodley and M. Friedl, "Identifying mislabeled training data," *Journal of Artificial Intelligence Research*, vol. 11, pp. 131-167, 1999.
-

-
- [133] D. Gamberger, N. Lavrac, and S. Dzeroski, "Noise detection and elimination in data preprocessing: experiments in medical domains," *Applied Artificial Intelligence*, vol. 14, pp. 205-223, 2000.
 - [134] G. John, "Robust decision trees: Removing outliers from databases," in *The First International Conference on Knowledge Discovery and Data Mining*, pp. 174–179, 1995.
 - [135] C. Schaffer, "Overfitting avoidance as bias," *Machine Learning*, vol. 10, pp. 153-178, 1993.
 - [136] J. Schlimmer and R. Granger, "Incremental learning from noisy data," *Machine Learning*, vol. 1, pp. 317-354, 1986.
 - [137] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Hum Neurobiol*, vol. 4, pp. 219-27, 1985.
 - [138] V. Navalpakkam and L. Itti, "Search goal tunes visual features optimally," *Neuron*, vol. 53, pp. 605-617, 2007.
 - [139] S. Frintrop, *VOCUS: A visual attention system for object detection and goal-directed search*: Springer, 2006.
 - [140] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1254-1259, 1998.
 - [141] K. Cave, "The FeatureGate model of visual selection," *Psychological Research*, vol. 62, pp. 182-194, 1999.
 - [142] J. Tsotsos, *et al.*, "Modeling visual attention via selective tuning," *Artificial intelligence*, vol. 78, pp. 507-545, 1995.
 - [143] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, pp. 1395-1407, 2006.
 - [144] S. J. Roberts, D. Husmeier, I. Rezek, and W. Penny, "Bayesian approaches to Gaussian mixture modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1133-1142, 2002.
 - [145] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, pp. 1-38, 1977.
 - [146] S. Kullback, *Information theory and statistics*: Dover Publications, New York, 1968.
 - [147] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, pp. 429-449, 2002.
 - [148] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter*, vol. 6, pp. 20-29, 2004.
 - [149] G. M. Weiss, "Mining with rarity: a unifying framework," *ACM SIGKDD Explorations Newsletter*, vol. 6, pp. 7-19, 2004.
 - [150] G. M. Weiss and H. Hirsh, "Learning to predict extremely rare events," in *learning from Imbalanced Data Sets, AAI Workshop. Technical Report WS-00-05* pp. 64-68, 2000.
-

- [151] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," *Machine Learning: ECML 2004*, pp. 39-50, 2004.
- [152] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002.
- [153] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," in *The Fourteenth International Conference on Machine Learning*, pp. 179-186, 1997.
- [154] C. Elkan, "The foundations of cost-sensitive learning," in *The Seventeenth International Joint Conference on Artificial Intelligence*, pp. 973-978, 2001.
- [155] K. Morik, P. Brockhausen, and T. Joachims, "Combining statistical learning with a knowledge-based approach-a case study in intensive care monitoring," in *The 16th International Conference on Machine Learning*, pp. 268-277, 1999.
- [156] K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," in *the International Joint Conference on AI*, pp. 55-60, 1999.
- [157] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Computation*, vol. 12, pp. 1207-1245, 2000.
- [158] M. J. Procopio, *Hand-labeled DARPA LAGR datasets*. Available: <http://www.mikeprocopio.com/research/labeledlagrdata/>, 2007
- [159] V. Vladimir and V. Vapnik, "The nature of statistical learning theory," ed: Springer, 1995.
- [160] L. Bottou and C. J. Lin, "Support vector machine solvers," *Large scale kernel machines*, pp. 301-20, 2007.