

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Learning, Modeling, and Understanding Vehicle Surround Using Multi-Modal Sensing /

Permalink

<https://escholarship.org/uc/item/05h602hb>

Author

Sivaraman, Sayanan

Publication Date

2013

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

**Learning, Modeling, and Understanding Vehicle Surround Using
Multi-Modal Sensing**

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Intelligent Systems, Robotics, and Control)

by

Sayanan Sivaraman

Committee in charge:

Professor Mohan M. Trivedi, Chair
Professor Serge Belongie
Professor Sanjoy Dasgupta
Professor Kenneth Kreutz-Delgado
Professor Bhaskar Rao

2013

Copyright
Sayanan Sivaraman, 2013
All rights reserved.

The dissertation of Sayanan Sivaraman is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2013

TABLE OF CONTENTS

	Signature Page	iii
	Table of Contents	iv
	List of Figures	vii
	List of Tables	xiv
	Acknowledgements	xv
	Vita	xvi
	Abstract of the Dissertation	xvii
Chapter 1	Introduction	1
	1.1 Contributions and Outline	2
Chapter 2	Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis	5
	2.1 Introduction	5
	2.2 On-Road Environmental Perception	7
	2.3 Vision-Based Vehicle Detection	11
	2.3.1 Monocular Vehicle Detection	12
	2.3.2 Stereo-vision for Vehicle Detection	17
	2.4 On-Road Vehicle Tracking	21
	2.4.1 Monocular Vehicle Tracking	21
	2.4.2 Stereo-vision Vehicle Tracking	22
	2.4.3 Fusing Monocular and Stereo-Vision Cues	24
	2.4.4 Real-time Implementation and System Architecture	24
	2.4.5 Fusing Vision with other Modalities	25
	2.5 On-Road Behavior Analysis	27
	2.5.1 Context	27
	2.5.2 Maneuvers	28
	2.5.3 Trajectories	29
	2.5.4 Behavior Classification	29
	2.6 Discussion and Future Directions	30
	2.6.1 Vehicle Detection	30
	2.6.2 Vehicle Tracking	31
	2.6.3 On-Road Behavior Analysis	32
	2.6.4 Benchmarks	34
	2.7 Concluding Remarks	38
	2.8 Acknowledgment	38
Chapter 3	A General Active Learning Framework for on-road Vehicle Detection and Tracking Systems	39
	3.1 Introduction	39
	3.2 Related Research	40
	3.2.1 Vehicle Detection and Tracking	41
	3.2.2 Active Learning for Object Recognition	42
	3.3 Active Learning: Motivation	43

	3.3.1	Overview	43
	3.3.2	Implementation	44
3.4		Training Framework	44
	3.4.1	Initialization	44
	3.4.2	Query and Retraining	45
3.5		Vehicle Recognition Using Rectangular Features and Adaboost Classifier	46
3.6		Vehicle Tracking with Condensation Filter	48
3.7		Experimental Evaluation	49
	3.7.1	Experimental Datasets	49
	3.7.2	Performance Metrics	51
	3.7.3	Static Images: Caltech 1999 Dataset	53
	3.7.4	Video Sequences: LISA-Q Front FOV Datasets	53
3.8		Remarks	58
3.9		Acknowledgment	59
Chapter 4		Active Learning for On-Road Vehicle Detection: A Comparative Study . .	60
	4.1	Introduction	60
	4.2	Related Research	61
	4.2.1	Active Learning for Object Detection	62
	4.2.2	On-road Vehicle Detection	64
	4.3	Active Learning for On-road Vehicle Detection	65
	4.3.1	Query by Confidence	65
	4.3.2	Query by Misclassification	68
	4.4	Experimental Evaluation	68
	4.4.1	Learning Considerations	68
	4.4.2	Feature and Classifiers Sets	70
	4.4.3	Training Sets	70
	4.4.4	Experiment 1: HOG-SVM Vehicle Detection	70
	4.4.5	Experiment 2: Haar features and Adaboost	71
	4.4.6	Analysis	75
	4.4.7	Analysis	76
	4.5	Remarks	77
	4.6	Acknowledgments	77
Chapter 5		Integrated Lane and Vehicle Detection, Localization, and Tracking: A Syn-	
		ergistic Approach	78
	5.1	Introduction	78
	5.2	Related Research	81
	5.2.1	Lane Detection and Tracking	81
	5.2.2	Vehicle Detection and Tracking	81
	5.2.3	Integrating Lane and Vehicle Tracking	82
	5.3	Lane Tracking and Vehicle Tracking Modules	82
	5.3.1	Lane Tracking using Steerable Filters	82
	5.3.2	Active Learning for Vehicle Detection with Particle Filter Tracking	85
	5.4	Synergistic Integration	87
	5.4.1	Improved Lane Tracking Performance	88
	5.4.2	Improved Vehicle Detection	88
	5.4.3	Localizing and Tracking Vehicles and Lanes	89
	5.5	Experimental Validation and Evaluation	90
	5.5.1	Lane Tracking Performance	92
	5.5.2	Vehicle Tracking Performance	95
	5.5.3	Localizing Vehicles with Respect to Lanes	98

	5.5.4 Processing Time	100
	5.6 Remarks	100
	5.7 Acknowledgments	101
Chapter 6	Vehicle Detection by Independent Parts for Urban Driver Assistance	102
	6.1 Introduction	102
	6.2 Related Research	106
	6.2.1 On-road Vehicle Detection and Tracking	106
	6.2.2 Part-based Object Detection	107
	6.3 Vehicle Detection by Independent Parts	108
	6.3.1 Active Learning for Detecting Independent Parts	110
	6.3.2 Semi-supervised Labeling for Part-Matching Classification . . .	112
	6.3.3 Tracking Vehicle Parts and Vehicles	115
	6.4 Experimental Evaluation	116
	6.4.1 Training Data	117
	6.4.2 Part Detection	117
	6.4.3 VDIP System Performance and Comparative Evaluation	120
	6.5 Remarks	124
	6.6 Appendix: Validation Sets	124
	6.7 Acknowledgments	125
Chapter 7	Dynamic Probabilistic Drivability Maps for Lane Change and Merge Driver Assistance	126
	7.1 Introduction	126
	7.2 Related Research	129
	7.2.1 Compact Representations	129
	7.2.2 Decision-Making During Maneuvers	130
	7.3 Dynamic Probabilistic Drivability Maps	131
	7.3.1 Drivability Cell Geometry	132
	7.3.2 Drivability Cell Probabilities	134
	7.4 Lane Change and Merge Recommendations	136
	7.4.1 Cost Function	137
	7.4.2 Min-Cost Solution via Dynamic Programming	139
	7.5 Experimental Results	140
	7.5.1 Merges	142
	7.5.2 Highway Lane Changes: Free-flow Traffic	144
	7.5.3 Highway Lane Changes: Dense Traffic	144
	7.5.4 Urban Lane Changes	147
	7.6 Remarks and Future Work	149
	7.7 Acknowledgments	149
Chapter 8	Conclusions	150
Bibliography	153

LIST OF FIGURES

Figure 2.1:	Illustrating the ascending levels of vision for semantic interpretation of the on-road environment. At the lowest level, features such as appearance, disparity, motion, and size are used to detect vehicles in images and video. One level up, data association, temporal coherence, and filtering are used for tracking, to re-identify and measure the dynamic parameters, and estimate the positions of the vehicles. At the highest level, an aggregate of spatio-temporal features are used to learn, model, classify, and predict the behavior and goals of other vehicles on the road. This area of research includes identification of specific maneuvers [1], and modeling typical on-road behavior [2].	6
Figure 2.2:	a) Radar for on-road vehicle detection uses radar antennas, emitting millimeter wavelength radio signals. The frequency shift in the reflected signal is used to determine distance to an object. b) Lidar for on-road vehicle detection uses laser scanners, emitting illumination at 600-1000nm wavelength, detecting backscattered energy with an imaging receiver, which is used to segment obstacles. c) Vision for on-road vehicle detection uses cameras, which sense the ambient light. Points in the camera’s field of view are mapped to pixels via perspective projection. Computer vision techniques, further detailed in this paper, recognize and localize vehicles from images and video. While radar and lidar detect objects, vision explicitly differentiates vehicles vs. non-vehicles.	10
Figure 2.3:	Images from representative vehicle detection studies, highlighting real-world system performance. Top Row: Monocular a) Sivaraman and Trivedi, 2010 [3]. b) Niknejad et al., 2012 [4]. c) O’Malley et al., 2010.[5] d) Jazayeri et al., 2011[6]. Bottom Row: Stereo-vision e) Erbs et al., 2011 [7]. f) Barth and Franke, 2010. [1] g) Danescu et al., 2011 [8].	18
Figure 2.4:	The ascending levels vehicle behavior interpretation. At the lowest level, vehicles are detected using vision. Vehicle tracking estimates the motion of previously-detected vehicles. At the highest level of interpretation, vehicle behavior is characterized. Behavior characterization includes maneuver identification, on-road activity description, and long-term motion prediction.	25
Figure 2.5:	A depiction of trajectory prediction, aiming to map the most likely future vehicle motion, based on observed motion [9].	28
Figure 2.6:	Performance metrics. a) Overlap criterion used for labeling detections as true positives or false positives in the image plane [10]. b) Plotting the recall vs. $1 - precision$ for monocular vehicle detection [11]. c) Plotting the estimated yaw rate vs. time, along with ground truth [1].	36
Figure 3.1:	Active Learning based Vehicle Recognition and Tracking. The general active learning framework for vehicle recognition and tracking systems consists of an off-line learning portion [white and red], and an online implementation portion [green]. Prior works in vehicle recognition and tracking have not utilized active learning [red].	40
Figure 3.2:	a) QUAIL- QUery & Archive Interface for active Learning. The QUAIL interface evaluates the passively trained vehicle recognition system on real-world data, and provides an interface for a human to label and archive ground truth. Detections are automatically marked green. Missed detections are marked red by the user. False positives are marked blue by the user. True detections are left green. b) QUAIL outputs. A true detection and a false positive, archived for retraining.	45
Figure 3.3:	Examples of false positive outputs queried for retraining using QUAIL	46

Figure 3.4:	Examples of true positives queried for retraining using QUAIL.	46
Figure 3.5:	Schematic of framework for Active Learning Based Vehicle Recognition Training. An initial, passively trained vehicle detector is built. Using the QUAIL interface, false positives, false negatives, and true positives are queried and archived. A new classifier is trained using the archived samples.	47
Figure 3.6:	a) Examples of Haar-like features used in the vehicle detector. b) Cascade of boosted classifiers	47
Figure 3.7:	Left: Detector outputs for a single vehicle [top], and multiple vehicles [middle and bottom]. Middle: Multiple location hypotheses generated by the Condensation filter. Note the multi modal distribution of the hypotheses when tracking multiple vehicles [bottom]. Right: Best tracking results, as confirmed by detections in the consequent frame.	48
Figure 3.8:	Full ALVERT system overview. Real on-road data is passed to the active learning based vehicle recognition system, which consists of Haar-like rectangular feature extraction and the boosted cascade classifier. Detections are then passed to the Condensation multiple vehicle tracker. The tracker makes predictions, which are then updated by the detection observations.	50
Figure 3.9:	a) Plot of TPR vs. FDR for passively trained recognition [blue], the active learning vehicle recognition [red]. We note the improvement in performance due to active learning on this dataset. b) Plot of TPR vs. Number of False Detections per Frame for passively trained recognition [blue], the active learning vehicle recognition [red]. We note the reduction in false positives due to active learning.	52
Figure 3.10:	Sample vehicle recognition results from the Caltech 1999 dataset.	53
Figure 3.11:	a) Recognition output in shadows. Five vehicles were detected, and two were missed due to their distance and poor illumination. We note that poor illumination seems to limit the range of the detection system; vehicles farther from the ego vehicle are missed. b) Vehicle recognition output in sunny highway conditions.	54
Figure 3.12:	Left: The vehicle in front was not detected in this frame on a cloudy day, due to smudges and dirt on the windshield. Right: The vehicle’s track was still maintained in the following frame.	54
Figure 3.13:	Top Left: The vehicle recognition system has output two true vehicles, one missed vehicle, and one false positive. Top Right: These detection outputs are passed to the tracker. We note that the missed vehicle still has a track maintained, and is being tracked despite the missed detection. Bottom Left: In the following frame, all three vehicles are detected. Bottom Right: In the corresponding tracker output, we note that a track was not maintained for the false positive from the previous frame.	55
Figure 3.14:	a) Recognition output in sunny conditions. Note the vehicle detections across all six lanes of traffic during San Diego’s rush hour. The recognizer seems to work best in even, sunny illuminations, as such illumination conditions have less scene complexity compared to scenes with uneven illuminations. b) Recognition output in dense traffic. We note that the vehicle directly in front is braking, and the system is aware of vehicles in adjacent lanes.	56
Figure 3.15:	a) Recognition output in a non-uniformly illuminated scene. Six vehicles were detected. No false positives, and no missed detections. b) Recognition output in cloudy conditions. Note the reflections, glare, and smudges on the windshield.	57

Figure 3.16:	a) Plot of TPR vs. FDR for passively trained recognition [blue], the active learning vehicle recognition [red], and the ALVeRT system [black]. We note the large improvement in performance due to active learning for vehicle detection. b) Plot of TPR vs. Number of False Detections per Frame for passively trained recognition [blue], the active learning vehicle recognition [red], and the ALVeRT system [black]. We note the reduction in False Positives due to active learning.	58
Figure 4.1:	Examples of the varied environments where on-road vehicle detectors must perform.	65
Figure 4.2:	a) Training image from which the query function from equation 4.7 returned no informative training examples. b) Training image from which multiple informative image subregions were returned.	67
Figure 4.3:	Interface for Query by Misclassification. The interface evaluates the initial detector, providing an interface for a human to label ground truth. Detections are automatically marked green. Missed detections are marked red by the user, and false positives blue.	69
Figure 4.4:	Example from <i>LISA_2009_Dense</i> validation set. The dataset consists of 1600 consecutive frames, captured during rush hour, over a distance of roughly 2km. The ground-truthed dataset is publicly available to the academic and research communities at http://cvrr.ucsd.edu/LISA/index.html	72
Figure 4.5:	Recall vs. 1-Precision for each vehicle detector, evaluated on the <i>LISA_2009_Dense</i> dataset. a) Classifiers trained with 2500 samples b) Classifiers trained with 5000 samples c) Classifiers trained with 10000 samples.	73
Figure 4.6:	Frame 1136 of <i>LISA_2009_Dense</i> . Detectors trained with 2500 samples. a) Random Samples. b) Query by Misclassification. c) Query by Confidence-Independent Samples . d) Query by Confidence- Labeled Samples.	74
Figure 4.7:	Frame 1136 of <i>LISA_2009_Dense</i> . Detectors trained with 5000 samples.a) Random Samples. b) Query by Misclassification. c) Query by Confidence-Independent Samples . d) Query by Confidence- Labeled Samples.	74
Figure 4.8:	Frame 1136 of <i>LISA_2009_Dense</i> . Detectors trained with 10000 samples. a) Random Samples. b) Query by Misclassification. c) Query by Confidence-Independent Samples . d) Query by Confidence- Labeled Samples.	74
Figure 5.1:	Framework for integrated lane and vehicle tracking, introduced in this study. Lane tracking and vehicle tracking modules are executed on the same frame, sharing mutually beneficial information, to improve the robustness of each system. System outputs are passed to the integrated tracker, which infers full state lane and vehicle tracking information.	79
Figure 5.2:	Typical performance of integrated lane and vehicle tracking on highway with dense traffic. Tracked vehicles in the ego lane are marked green. To the left of the ego lane, tracked vehicles are marked blue. To the right of the ego lane, tracked vehicles are marked red. Note the curvature estimation.	80
Figure 5.3:	Lane tracking framework used in this study. Feature extraction is achieved by applying a bank of steerable filters. The road model is fit using RANSAC, and lane position tracked with Kalman filtering.	83
Figure 5.4:	An illustration of the variables used in lane tracking, further explained in Table 5.1	85
Figure 5.5:	Active learning for vehicle detection and tracking, module originally presented in [3]	86
Figure 5.6:	Comparison of initial classifier with active learning based vehicle detection, in scenes with complex shadowing.	86

Figure 5.7:	Selected parameters from the CANbus over the 5000 frame sequence. Note how the vehicle’s speed decreases, and driver’s braking increases, as the segment progresses, coinciding with increasing traffic density.	91
Figure 5.8:	Estimated position of the right lane marker vs. frame number. The ground truth is shown in green. The estimated position using just lane tracker is shown in red. The result of the integrated lane and vehicle tracking system is shown in blue. Note that for the last 1000 frames, the lane tracker alone loses track of the lane position, due to high density traffic and a tunnel.	93
Figure 5.9:	Lane change maneuvers. In low-density traffic, both stand-alone lane tracking and integrated lane and vehicle tracking perform equally well. a) Lane Tracking outputs, including two lane changes b) Steering angle during this segment.	93
Figure 5.10:	a) Poor lane localization due to a missed vehicle detection. The missed vehicle detection leads the lane tracker to integrate erroneous lane markings into the measurements, resulting in worse lane estimation for the right marker. b) Example misclassification of lane position. The jeep in the right lane [green] has been classified as in the ego lane. This is due to the fact that the jeep is farther ahead than the lane tracker’s look-ahead distance.	94
Figure 5.11:	a) We note a spike in the ground truth, and corresponding error around Frame 2550. Large estimation error due to rapid, severe variation in road pitch, due to a large bump in the road, which severely alters the pitch for a very short period, less than a second. The ego vehicle was traveling at 35 meters per second. b) Selected frames from this one-second span. The beginning and end frames show normal lane estimation. The middle frames show lane estimation errors due to the bump in the road. c) Horizon estimation using lane estimation. The red line is the estimated horizon.	95
Figure 5.12:	a) Right lane marker estimation error. b) Lane tracking in dense traffic, frame 4271, The pink lines indicate estimated lane positions. Note the large estimation error due to the presence of vehicles in dense traffic. c) Integrated Lane and Vehicle Tracking in dense traffic, frame 4271. The red and blue lines indicate estimated lane positions. Note the tracked vehicles and accurate lane estimation.	96
Figure 5.13:	5.13(a) Recall vs. False Positives per Frame, comparing vehicle detection and tracking alone [3], and integrated lane and vehicle tracking. Performance is evaluated over a 1000-frame sequence, which features 2790 vehicles. While both systems perform quite well over the dataset, Integrated Lane and Vehicle Tracking has better performance in terms of false positives per frame. 5.13(b) Estimated lane position during vehicle localization validation sequence, integrated lane and vehicle tracking. Note the two lane changes towards the end of the sequence.	96
Figure 5.14:	a) Buildings off the road result in false positives. b) By enforcing the constraint that tracked vehicles must lie on the ground plane, the false positives are filtered out.	97
Figure 5.15:	Ambiguities in lane/vehicle positions. The vehicle on the left is in the midst of a lane change. a) The vehicle is determined to still be in the ego-lane. b) The vehicle is determined to have changed lanes in to the left lane.	98
Figure 5.16:	Illustrating lane-level localization of vehicles during an ego lane change. a) Frame 2287, immediately prior to lane change. b) Lane Change. We note that the truck on the left has been incorrectly assigned to the ego-lane. c) The truck on the left has been correctly assigned to the left lane, a few frames later.	99

Figure 6.1:	The urban driving environment features oncoming, preceding, and sideview vehicles [top]. Additionally, vehicles appear partially-occluded as they enter and exit the camera’s field of view [bottom].	103
Figure 6.2:	Vehicle Detection by Independent Parts [VDIP]. The learning approach detailed in this study. An initial round of supervised learning is carried out to yield initial part detectors, for the front and rear parts of vehicles. We use the initial detectors to query informative training examples from independent data, performing active learning to improve part detection performance. While we query informative training examples, we label side-view vehicles in a semi-supervised manner, using the active learning annotations to form fully-visible vehicles.	105
Figure 6.3:	Vehicle Detection by Independent Parts [VDIP]: Data information flow for real-time vehicle detection and tracking by parts.	110
Figure 6.4:	VDIP illustrative example. As a vehicle enters the camera’s field of view, its front part is detected [blue]. As it becomes fully-visible, its rear part is detected too [red]. Parts are tracked, and a part matching classifier is applied, to detect fully-visible side-view vehicles [purple].	111
Figure 6.5:	Active learning interface used for part detection sample query, and for semi-supervised labeling side-view vehicles for detection by parts. a) All training examples, positive and negative, collected from a single frame, for training part matching, as well as part detection. b) Active learning sample query for front parts, featuring true positives [blue] and false positives [red]. c) Active learning sample query for rear parts, featuring true positives [purple] and false positives [cyan]. d) Part-matching examples for side vehicle detection, positive [green] and negative yellow]. e) All positive training examples collected from this frame, for front part, rear part, and part matching. f) All negative training examples collected from this frame front part, rear part, and part matching.	114
Figure 6.6:	Using the tracking velocities of the vehicle parts eliminates erroneous side-view vehicles. a) A side-view vehicle is erroneously constructed from an oncoming and a preceding vehicle. b) Using velocity information from tracking, the erroneous side-view vehicle is not constructed.	116
Figure 6.7:	Showing the full track of a vehicle in the camera’s field of view. In the first frame, the front part of the vehicle is detected, but most of the vehicle is occluded. A couple of frames later, the rear part of the vehicle is detected as well. The full side-view vehicle is detected, and identified with a purple bounding box. The vehicle and its parts are tracked while they remain in the camera’s field of view. As the vehicle leaves the camera’s field of view, its rear part is still detected.	117
Figure 6.8:	Part detection performance, LISA-X Downtown. True Positive Rate vs. False Positives per Frame, for the initial part detectors [red], and the active learning based part detectors [blue].	118
Figure 6.9:	Part detection performance, LISA-X Intersection. True Positive Rate vs. False Positives per Frame, for the initial part detectors [red], and the active learning based part detectors [blue].	119
Figure 6.10:	True positive rate vs. False Positives per frame, for the part matching classifier, compared to the system presented in [12].	121
Figure 6.11:	Showcasing VDIP system performance in various scenarios. a) Detecting preceding vehicle. b) Detecting oncoming, side-view, and partially-occluded vehicles at an intersection. c) Detection of occluded vehicle parts, while a pedestrian walks in front of the camera. d) Oncoming and sideview vehicles. e) Oncoming and preceding vehicles. f) Oncoming and sideview vehicles.	122

Figure 6.12: Showcasing VDIP system difficulties. a) False positives in urban traffic, due to multiple poorly-localized detections of vehicles. b) Ambiguous classification between oncoming and preceding vehicles. c) and d) False positives due to complex background trees. g) False positives due to road texture h) Poorly-localized side-view bounding box [purple], due to poor localization of the rear part [red].	122
Figure 7.1: a) Dynamic Probabilistic Drivability Map, and lane change recommendations [left]. The probability of drivability is indicated by the color of map cell, with green areas carrying a high probability, and red areas a low probability of drivability. The DPDM integrates information from lidar, radar, and vision-based systems, including lane estimation and vehicle tracking. Recommendations for lane changes are made using this information. [Right] Camera view of the road. b) HMI concept for presenting recommendations to the driver via heads-up display.	127
Figure 7.2: The full spectrum of maneuver-based decision systems in intelligent vehicles, with implications for driving. At one end, there is fully manual driving. Active safety systems, such as lane departure warning [LDW] and side warning assist [SWA] are already becoming more commercially-available. Predictive driver assistance remains an open area of research. Cooperative driving will integrate predictive systems, and seamlessly allow hand-offs of control between driver and autonomous driving. At the far end of the spectrum is fully autonomous driving, with no input from the driver.	128
Figure 7.3: a) Lane changes commonly take place in both urban driving b) and highway driving. c) Merges take place in the transition from urban to highway driving.	128
Figure 7.4: a) The Audi automotive testbed used in this study. b) A depiction of the sensing capabilities of the instrumented testbed.	132
Figure 7.5: Drivability cells are physically modeled as convex quadrilaterals, which adapt their geometry to the geometry of the roads and lanes. Their width adapts to the lane width, and they also adapt to accommodate lane curvature.	133
Figure 7.6: Histograms of recommended accelerations during maneuvers. a) Merges. Most of the recommendations require the ego-vehicle to accelerate, which is to be expected during merge maneuvers. b) Free-flow highway lane changes. Most of the recommendations involve constant-velocity lane changes, or decelerating to safely accommodate slower vehicles. c) Dense highway lane changes. Most of the recommendations require a positive acceleration. This is due to the fact that there is often lane-specific congestion in dense traffic, which results in high relative velocities between adjacent lanes. d) Urban lane changes. Urban driving features a roughly equal proportion of acceleration, deceleration, and constant-velocity lane changes.	140
Figure 7.7: We demonstrate the system recommendations in a merge scenario with no pertinent vehicles in the surround. a) We plot the cost of merging to left vs. time, and the system's recommended acceleration during the sequence. Given the lack of surround vehicles, the only on-road constraints to consider are the lane markings. b) At the beginning of the merge sequence, the DPDM cells to the left of the ego-vehicle have a low probability of drivability, because of the solid lane boundary. This coincides with a very high cost, and a null recommendation to merge. c) After the lane boundary has transitioned to dashed markings, the system recommends a constant-velocity merge, with very low cost.	141

Figure 7.8:	System recommendations during a merge scenario that requires acceleration. a) The cost, recommended acceleration, and vehicle velocity vs. time during the merge sequence. b) At the beginning of the sequence, the DPDM cells to the left carry low probability of drivability, as there is a vehicle in the left blind-spot. c) As the sequence progresses, the system recommends an acceleration in order to create safe distance between the ego-vehicle and vehicle to the left. d) The recommended acceleration drops to 0.	143
Figure 7.9:	Lane-change recommendations as the ego-vehicle overtakes a slower vehicle [right] in free-flowing traffic. a) At the beginning of the sequence, the right-lane recommendation involves deceleration with some cost. As the ego-vehicle overtakes the slower vehicle, the cost and required acceleration both go to zero, and a constant-velocity lane-change is possible. b) The slower vehicle is in front of the ego-vehicle in the right lane. c) The ego-vehicle has overtaken the slower vehicle, which has subsequently exited from the highway.	145
Figure 7.10:	Dense highway segment. a) Cost, recommended acceleration, and velocity vs. time. b) DPDM from the beginning of the segment. c) DPDM from the end of the segment.	146
Figure 7.11:	An urban sequence during which the system recommends acceleration to change into the left lane. a) We plot cost, recommended acceleration, and velocity vs. time. b) A vehicle approaches the ego-vehicle with positive relative velocity in the left lane. As the ego-vehicle speeds up, the recommended acceleration and cost go to 0.	148

LIST OF TABLES

Table 2.1:	Comparison of Sensors for Vehicle Detection	8
Table 2.2:	Vehicle Detection Benchmarks	35
Table 2.3:	Monocular Vehicle Detection Metrics	35
Table 2.4:	Stereo-vision Tracking Metrics	36
Table 3.1:	Selected Active Learning Based Object Recognition Approaches.	42
Table 3.2:	Datasets Used in this Study	50
Table 3.3:	Experimental Dataset 1 : Jan 28, 2009, 4pm, highway, sunny	56
Table 3.4:	Experimental Dataset 2: March 9, 2009, 9am, urban , cloudy	56
Table 3.5:	Experimental Dataset 3: April 21, 2009, 12pm, highway, sunny	57
Table 4.1:	Definition of active learning approaches compared in this paper	63
Table 4.2:	Comparison of Labeling Time for Vehicle Detectors Trained with 1000 Samples, HOG-SVM	69
Table 4.3:	Active Learning Results: HOG-SVM Vehicle Detection, Caltech 1999 Vehicle Database	71
Table 4.4:	HOG-SVM Vehicle Detection, Training Parameters	71
Table 4.5:	Comparison of Annotation Time for Vehicle Detectors Trained with 2500 Samples, Haar+Adaboost	72
Table 4.6:	Comparison of Annotation Time for Vehicle Detectors Trained with 5000 Samples, Haar+Adaboost	73
Table 4.7:	Comparison of Annotation Time for Vehicle Detectors Trained with 10000 Samples, Haar+Adaboost	74
Table 4.8:	Haar-Adaboost Vehicle Detection, Training Parameters	75
Table 5.1:	Variables used for Lane Tracking	84
Table 5.2:	Lane Localization Results	92
Table 5.3:	Lane Localization Results, Last 1000 Frames	95
Table 5.4:	Performance Comparison, Low False Positives per Frame	97
Table 5.5:	Performance Comparison, High Recall	97
Table 5.6:	Confusion Matrix of Tracked Vehicle Lane Assignments	99
Table 5.7:	Processing Time for Vehicle, Lane, and Integrated Systems	100
Table 6.1:	Vision-based Vehicle Detection	104
Table 6.2:	Taxonomy of On-Road Vehicle Detection	109
Table 6.3:	Number of Training Examples	117
Table 6.4:	LISA-Q Urban, 300 Frames, Preceding Vehicles Only	118
Table 6.5:	LISA-X Downtown, 500 Frames, Preceding, Oncoming, Partially Occluded	121
Table 6.6:	LISA-X Intersection, 1500 Frames, Preceding, Oncoming, Side, Partially Occluded	123
Table 6.7:	Validation Sets Used in this Study	124
Table 7.1:	Drivability Cell Attributes	137
Table 7.2:	Merge Data, 50 Merges	142
Table 7.3:	Free-Flow Highway Lane-Changes, N=25	144
Table 7.4:	Dense Highway Lane-Changes, N=25	147
Table 7.5:	Urban Lane-Changes, N=50	147

ACKNOWLEDGEMENTS

I thank my parents, who supported me all throughout my studies, with empathy, sound advice, and the best of intentions. I also thank my brother and sister, and the rest of my family, for their support throughout these years.

I thank my love Melissa, my anchor and inspiration.

Thanks to all the friends I've met over the years, for all the good times we've had in San Diego.

I thank my advisor, Prof. Mohan Trivedi, for his mentorship and guidance in research over these many years.

I thank the committee members, Prof. Serge Belongie, Prof. Sanjoy Dasgupta, Prof. Ken Kreutz-Delgado, and Prof. Bhaskar Rao. Their scholarship, teaching, questions, and guidance greatly influenced this body of research.

I thank my labmates for their guidance, collaboration, and assistance over the years. In particular, I'd like to acknowledge Dr. Joel McCall, Dr. Shinko Cheng, Dr. Erik Murphy-Chutorian, Dr. Shankar Shivappa, Dr. Anup Doshi, Dr. Brendan Morris, Mr. Ashish Tawari, Mr. Derick Johnson, Mr. Justin Li, Ms. Sujitha Martin, Mr. Larry Ly, and Mr. Eshed Ohn-Bar for their help and contributions.

I thank the University of California Discovery Grant, Volkswagen Group of America's Electronics Research Laboratory, and Audi AG for funding this research.

Publication acknowledgements: Chapter 2 of this dissertation is a partial reprint of material published in IEEE Transactions on Intelligent Transportation Systems, 2013. Chapter 3 is a partial reprint of material published in IEEE Transactions on Intelligent Transportation Systems, 2010. Chapter 4 is a partial reprint of material published in Machine Vision and Applications, 2011. Chapter 5 is a partial reprint of material published in IEEE Transactions on Intelligent Transportation Systems, 2013. Chapter 6 is a partial reprint of material published in IEEE Transactions on Intelligent Transportation Systems, 2013. Chapter 7 is a partial reprint of material submitted to IEEE Transactions on Intelligent Transportation Systems, 2013. Each of these chapters and acknowledged publications were authored by Sayanan Sivaraman and Mohan Trivedi. The dissertation author was the primary investigator and author of these papers.

VITA

- 2007 B. S. in Electrical Engineering, University of Maryland, College Park
- 2009 M. S in Electrical Engineering (Intelligent Systems, Robotics, and Control), University of California, San Diego
- 2013 Ph. D. in Electrical Engineering (Intelligent Systems, Robotics, and Control), University of California, San Diego

PUBLICATIONS

Sayanan Sivaraman and Mohan M. Trivedi, Dynamic Probabilistic Drivability Maps for Lane Change and Merge Driver Assistance, *IEEE Transactions on Intelligent Transportation Systems*, in submission 2013.

Sayanan Sivaraman and Mohan M. Trivedi, Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis, *IEEE Transactions on Intelligent Transportation Systems*, 2013.

Sayanan Sivaraman and Mohan M. Trivedi, Vehicle Detection by Independent Parts for Urban Driver Assistance, *IEEE Transactions on Intelligent Transportation Systems*, 2013.

Sayanan Sivaraman and Mohan M. Trivedi, Integrated Lane and Vehicle Detection, Localization, and Tracking: A Synergistic Approach, *IEEE Transactions on Intelligent Transportation Systems*, 2013.

Sayanan Sivaraman and Mohan M. Trivedi, Active Learning for On-Road Vehicle Detection: A Comparative Study, *Machine Vision and Applications: Special Issue on Car Navigation and Vehicle Systems*, 2011.

Sayanan Sivaraman and Mohan M. Trivedi, A General Active Learning Framework for On-road Vehicle Recognition and Tracking, *IEEE Transactions on Intelligent Transportation Systems*, 2010.

ABSTRACT OF THE DISSERTATION

**Learning, Modeling, and Understanding Vehicle Surround Using
Multi-Modal Sensing**

by

Sayanan Sivaraman

Doctor of Philosophy in Electrical Engineering (Intelligent Systems, Robotics,
and Control)

University of California, San Diego, 2013

Professor Mohan M. Trivedi, Chair

This dissertation seeks to enable intelligent vehicles to see, to infer context, and to understand the on-road environment.

We provide a review of the literature in on-road vision-based vehicle detection, tracking, and behavior understanding. Placing vision-based vehicle detection in the context of sensor-based on-road surround analysis, we discuss monocular, stereo-vision, and active sensor-vision fusion for on-road vehicle detection. We discuss vehicle tracking in the monocular and stereo-vision domains, analyzing filtering, estimation, and dynamical models. We introduce relevant terminology for treatment of on-road behavior, and provide perspective on future research directions in the field.

We introduce a general active learning framework for on-road vehicle detection and tracking. Active learning consists of initial training, query of informative samples, and retraining,

yielding improved performance with data efficiency. In this work, active learning reduces false positives by an order of magnitude. The generality of active learning for vehicle detection is demonstrated via learning experiments performed with detectors based on Histogram of Oriented Gradient features and SVM classification [HOG-SVM], and Haar-like features and Adaboost Classification [Haar-Adaboost] . Learning approaches are assessed in terms of the time spent annotating, data required, recall, and precision.

We introduce a synergistic approach to integrated lane and vehicle tracking for driver assistance. Integration improves lane tracking accuracy in dense traffic, while reducing vehicle tracking false positives. Further, system integration yields lane-level localization, providing higher-level context.

We introduce vehicle detection by independent parts for urban driver assistance, for detecting oncoming, preceding, side-view, and partially occluded vehicles in urban driving. The full system is real-time capable, and compares favorably with state-of-the-art vehicle detectors, while operating 30 times as fast.

We present a novel probabilistic compact representation of the on-road environment, the Dynamic Probabilistic Drivability Map (DPDM), and demonstrate its utility for predictive lane change and merge [LCM] driver assistance during highway and urban driving. A general, flexible, probabilistic representation, the DPDM readily integrates data from a variety of sensing modalities, functioning as a platform for sensor-equipped intelligent vehicles. Based on the DPDM, the real-time LCM system recommends the required acceleration and timing to safely merge or change lanes with minimum cost.

Chapter 1

Introduction

As Moore’s law marches onward, computation is beginning to pervade many facets of modern life, including how we communicate, socialize, and produce goods. One area where advanced computation has large unrealized potential is in the area of transportation. Annually, between 1 and 3 percent of the world’s GDP is spent on the medical costs, property damage, and other costs associated with automotive accidents, and each year, some 1.2 million people die worldwide as a result of traffic accidents [13]. In the United States, tens of thousands of drivers and passengers die on the roads each year, with most fatal crashes involving more than one vehicle [14]. The National Highway Transportation Safety Administration reports that 49% of fatal crashes feature a lane or roadway departure, and the majority of crashes feature more than one vehicle [14].

Complex maneuvers such as lane changes and merges, which require the driver to maintain an awareness of the vehicles and dynamics in multiple lanes, account for a disproportionate number of collisions. According to NHTSA, lane change crashes account for some 500,000 crashes per year in the United States [15]. Merge maneuvers at highway ramps account for far more crashes per mile driven than other highway segments [16].

In recent years, there has been great progress in sensing and computation for intelligent vehicles. Sensors have become higher in fidelity and cheaper over time. Computation has become cheaper and faster, while the advent of multi-core architectures and graphical processing units allows for parallel processing. Research utilizing intelligent vehicles, equipped with advanced sensing and computing technology, has proliferated in recent years, resulting in robust environmental perception using computer vision [11, 8, 17], radar [18], and lidar [19]. Using the perception modules available, researchers have begun to address decision-making and assistance for lane changes, and to a lesser extent, merges.

In this work, we detail several fundamental research contributions to the environmental perception and driver assistance for intelligent vehicles. These contributions use a variety of

sensing modalities, including computer vision, radar, and lidar. Using advanced sensing, computation, modeling, and machine learning, we can develop machines that understand the on-road environment, and human-machine interfaces to assist drivers.

1.1 Contributions and Outline

In this dissertation, we provide a review of vision-based vehicle detection, tracking, and on-road behavior analysis. We concentrate our efforts on works published since 2005, referring the reader to [20] for earlier works. We place vision-based vehicle detection in context of on-road environmental perception, briefly detailing complimentary modalities that are commonly used for vehicle detection, namely radar and lidar. We then review vision-based vehicle detection, commenting on monocular vision, stereo-vision, monocular-stereo combination, and sensor-fusion approaches to vision-based vehicle detection. We discuss vehicle tracking using vision, detailing image-plane and 3D techniques for modeling, measuring, and filtering vehicle dynamics on the road. We then discuss the emerging body of literature geared towards analysis of vehicle behavior using spatio-temporal cues, including modeling, learning, classification, and prediction of vehicle maneuvers and goals. We provide our insights and perspectives on future research directions in vision-based vehicle detection, tracking, and behavior analysis.

In the computer vision domain, we detail for the first time, the contributions provided by active learning for training an appearance-based vehicle detection system using machine learning. The main novelty and contributions of this work include the following. A general active learning framework for on-road vehicle recognition and tracking is introduced. Using the introduced active learning framework, a full vehicle recognition and tracking system has implemented, and a thorough quantitative performance analysis has been presented. The vehicle recognition and tracking system has been evaluated on both real world video, and public domain vehicle images. In this study, we introduce new performance metrics for assessing on road vehicle recognition and tracking performance, which provide a thorough assessment of the implemented system’s recall, precision, localization, and robustness.

In this dissertation, a comparative study of active learning for on-road vehicle detection is presented. We have implemented three separate active learning approaches for vehicle detection, comparing the annotation costs, data costs, recall, and precision of the resulting classifiers. The implications of querying examples from labeled vs. unlabeled data is explored. The learning approaches have been applied to the task of on-road vehicle detection, and a quantitative evaluation is provided on a real world validation set. we have implemented three widely used active learning frameworks for on-road vehicle detection. Two main sets of experiments have been performed. In the first set, HOG features and Support Vector Machine [21, 22] classification have been used. In the second set of experiments, the vehicle detector was trained using Haar-like features and Adaboost classification. Based on the initial classifiers, we have employed various

separate querying methods. Based on confidence-based active learning [23], we have implemented Query by Confidence [QBC] with two variations. In the second, informative independent examples are queried from unlabeled data corpus, and a human oracle labels these examples. This method is compared against simply querying labeled examples from the initial corpus are queried based on a confidence measure, as in [24, 25]. The third learning framework we term Query by Misclassification [QBM], in which the initial detector is simply evaluated on independent data, and a human oracle labels false positives and missed detections. This approach has been used in [26, 27].

In this work, we introduce a synergistic approach to integrated lane and vehicle tracking for driver assistance. Utilizing systems built upon works reported in the literature, we integrate lane and vehicle tracking and achieve the following. Lane tracking performance has been improved by exploiting vehicle tracking results, eliminating spurious lane marking filter responses from the search space. Vehicle tracking performance has been improved by utilizing the lane tracking system to enforce geometric constraints based on the road model. By utilizing contextual information from two modules, we are able to improve the performance of each module. The entire system integration has been extensively quantitatively validated on real-world data, and benchmarked against the baseline systems.

Beyond improving the performance of both vehicle tracking and lane tracking, this research study introduces a novel approach to localizing and tracking vehicles with respect to the ego-lane, providing lane-level localization of other vehicles on the road. This novel approach adds valuable safety functionality, and provides a contextually relevant representation of the on-road environment for driver assistance, previously unseen in the literature. Figure 5.1 depicts an overview of the approach detailed in this study, and Figure 5.2 shows typical system performance.

In this study, we introduce Vehicle Detection by Independent Parts [VDIP]. Vehicle part detectors are trained using active learning, wherein initial part detectors are used to query informative training examples from unlabeled on-road data. The queried examples are used for retraining, in order to improve the performance of the part detectors. Training examples for the part matching classifier are collected using semi-supervised annotation, performed during the active learning sample query process. After retraining, the vehicle part detectors are able to detect oncoming, preceding vehicles, and front and rear parts of cross-traffic vehicles. The semi-supervised labeled configurations are used to train a part-matching classifier for detecting full side-view vehicles. Vehicles and vehicle parts are tracked using Kalman filtering. The final system is able to detect and track oncoming, preceding, side-view, and partially-occluded vehicles. The full system lightweight, robust, and runs in real time. Figure 6.2 depicts the learning process for vehicle detection by independent parts.

In this study, we introduce a novel compact representation of the on-road environment, the Dynamic Probabilistic Drivability Map [DPDM], and demonstrate its utility in predictive driver assistance for lane changes and merges [LCM]. The DPDM is a data structure that contains

spatial information, dynamics, and probabilities of drivability, readily integrating measurements from a variety of sensors. In this work, we develop a general predictive LCM assistance system which efficiently solves for the minimum cost maneuver, using dynamic programming over the DPDM. The full system provides timing and acceleration recommendations, designed to advise the driver *when* and *how* to merge and change lanes. The LCM system has been extensively tested using real-world on-road data from urban driving, dense highway traffic, free-flow highway traffic, and merge scenarios.

The contributions in this work derive from scientific exploration into the potential for advanced sensing and computation to enable a machine or vehicle to perceive and understand its surroundings. An intelligent vehicle needs to detect other vehicles on the road, to track them, to understand their behavior, and ultimately to make decisions on how, where and when to maneuver. In this work, we explore each of these aspects of intelligent vehicle cognition, and perform experiments that elucidate techniques and frameworks for enhancing a machine’s ability to understand the on-road environment, and assist the driver.

Among the contributions of this dissertation:

- A comprehensive survey of the state-of-the-art in computer vision for on-road vehicle detection, vehicle tracking, and behavior analysis (Chapter 2).
- Active learning for on-road vehicle detection using computer vision, and a comparative study of active learning techniques for this problem (Chapters 3 and 4).
- Integrated lane and vehicle tracking using monocular computer vision for driver assistance (Chapter 5).
- Vehicle detection by independent parts using computer vision (Chapter 6).
- Dynamic probabilistic drivability maps for lane change and merge assistance (Chapter 7).

The following chapter provides a comprehensive review of the state of the literature in computer vision for on-road vehicle detection, vehicle tracking, and vehicle behavior analysis. Chapter 3 and chapter 4 detail active learning for vision-based on-road vehicle detection, and perform a comparative study of active learning for this problem. Chapter 5 details the synergistic integration of vision-based lane and vehicle tracking for driver assistance. Chapter 6 introduces vision-based vehicle detection and tracking by independent parts. Chapter 7 details Dynamic Probabilistic Drivability Maps and their application to lane change and merge maneuvers. Chapter 8 offers concluding remarks, and discussion on future research areas.

Chapter 2

Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis

2.1 Introduction

In the United States, tens of thousands of drivers and passengers die on the roads each year, with most fatal crashes involving more than one vehicle [14]. Research and development efforts in advanced sensing, environmental perception, and intelligent driver assistance systems seek to save lives and reduce the number of on-road fatalities. Over the past decade, there has been significant research effort dedicated to the development of intelligent driver assistance systems and autonomous vehicles, intended to enhance safety by monitoring the on-road environment.

In particular, the on-road detection of vehicles has been a topic of great interest to researchers over the past decade [20]. A variety of sensing modalities have become available for on-road vehicle detection, including radar, lidar, and computer vision. Imaging technology has progressed immensely in recent years. Cameras are cheaper, smaller, and of higher quality than ever before. Concurrently, computing power has increased dramatically. Further, in recent years, we have seen the emergence of computing platforms geared towards parallelization, such as multi-core processing, and graphical processing units [GPU]. Such hardware advances allow computer vision approaches for vehicle detection to pursue real-time implementation.

With advances in camera sensing and computational technologies, advances in vehicle

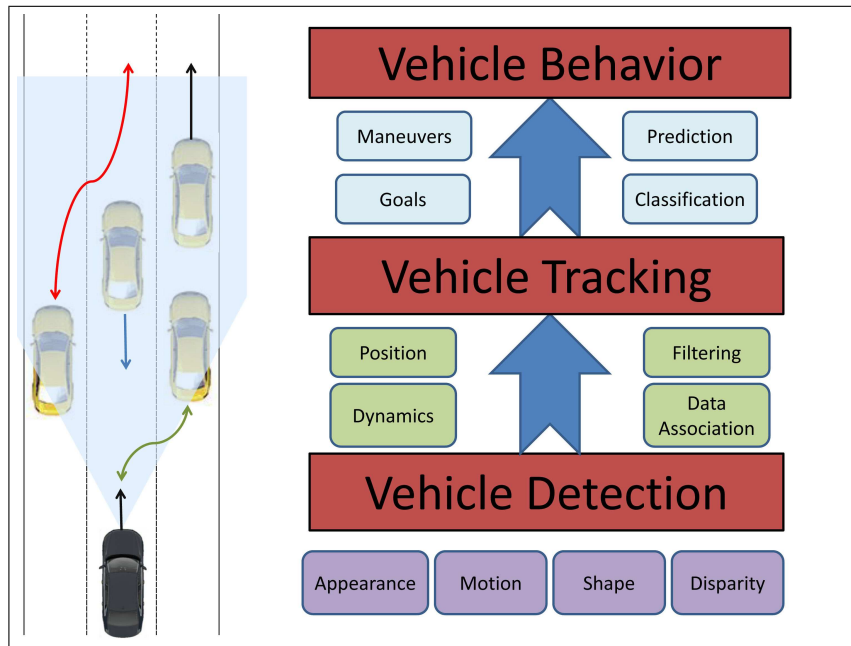


Figure 2.1: Illustrating the ascending levels of vision for semantic interpretation of the on-road environment. At the lowest level, features such as appearance, disparity, motion, and size are used to detect vehicles in images and video. One level up, data association, temporal coherence, and filtering are used for tracking, to re-identify and measure the dynamic parameters, and estimate the positions of the vehicles. At the highest level, an aggregate of spatio-temporal features are used to learn, model, classify, and predict the behavior and goals of other vehicles on the road. This area of research includes identification of specific maneuvers [1], and modeling typical on-road behavior [2].

detection using monocular vision, stereo-vision, and sensor fusion with vision have been an extremely active research area in the intelligent vehicles community. On-road vehicle tracking has also been extensively studied. It is now commonplace for research studies to report the ability to reliably detect and track on-road vehicles in real-time, over extended periods [1, 28]. Theoretical, practical, and algorithmic advances have opened up research opportunities that seek higher level of semantic interpretation of on-road vehicle behavior. The aggregate of this spatio-temporal information from vehicle detection and tracking can be used to identify maneuvers, and to learn, model, and classify on-road behavior.

Figure 2.1 depicts the use of vision for on-road interpretation. At the lowest level, various motion and appearance cues are used for on-road vehicle detection. One level up, detected vehicles are associated across frames, allowing for vehicle tracking. Vehicle tracking measures the dynamics of the motion of detected vehicles. At the highest level, an aggregate of spatio-temporal features allows for characterization of vehicle behavior, recognition of specific maneuvers, behavior classification, and long-term motion prediction. Examples of work in this nascent area include prediction of turning behavior [1], prediction of lane changes [29], and modeling typical on-road behavior [2].

In this paper, we provide a review of vision-based vehicle detection, tracking, and on-road behavior analysis. We concentrate our efforts on works published since 2005, referring the reader to [20] for earlier works. We place vision-based vehicle detection in context of on-road environmental perception, briefly detailing complimentary modalities that are commonly used for vehicle detection, namely radar and lidar. We then review vision-based vehicle detection, commenting on monocular vision, stereo-vision, monocular-stereo combination, and sensor-fusion approaches to vision-based vehicle detection. We discuss vehicle tracking using vision, detailing image-plane and 3D techniques for modeling, measuring, and filtering vehicle dynamics on the road. We then discuss the emerging body of literature geared towards analysis of vehicle behavior using spatio-temporal cues, including modeling, learning, classification, and prediction of vehicle maneuvers and goals. We provide our insights and perspectives on future research directions in vision-based vehicle detection, tracking, and behavior analysis.

2.2 On-Road Environmental Perception

While the focus of this paper lies in vision-based vehicle detection, it is pertinent to include a brief treatment of complimentary modalities currently used in on-road vehicle detection. We discuss general sensor-based vehicle detection to place vision-based vehicle detection in the overall context of on-road environmental perception. We take this occasion to discuss conceptual similarities and differences that the various sensing modalities bring to vehicle detection, and discuss the emerging avenues for data fusion and systems integration. Namely, we briefly discuss the use of millimeter-wave radar, and of lidar, alongside computer vision, for on-road vehicle

Table 2.1: Comparison of Sensors for Vehicle Detection

Sensing Modality	Perceived Energy	Raw Measurement	Units	Recognizing Vehicles vs. Other Objects
Radar	Millimeter-wave radio signal [emitted]	Distance	Meters	Resolved via tracking
Lidar	600-1000 nanometer-wave laser signal [emitted]	Distance	Meters	Resolved via spatial segmentation, motion
Vision	Visible light [ambient]	Light intensity	Pixels	Resolved via appearance, motion

detection. Table 2.1 summarizes the comparison between radar, lidar, and vision for vehicle detection.

Millimeter-wave radar is widely used for detecting vehicles on the road. Radar technology has made its way into production-mode vehicles, for applications including adaptive cruise control [ACC] and side-warning assist [SWA] [18, 30]. Typically, a frequency-modulated continuous waveform signal is emitted. Its reflections are received and demodulated, and frequency content is analyzed. The frequency shift in the received signal is used to measure the distance to the detected object. Detected objects are then tracked and filtered based on motion characteristics to identify vehicles and other obstacles [18]. The radar sensing used for adaptive cruise control generally features a narrow angular field of view, well-suited to detecting objects in the ego vehicle’s lane. Figure 2.2(a) depicts the operation of radar for on-road vehicle detection.

Radar sensing works quite well for narrow field-of-view applications, detecting and tracking preceding vehicles in the ego lane. Radar vehicle tracking works fairly consistently in different weather and illumination conditions. However, vehicle-mounted radar sensors cannot provide wide field-of-view vehicle detection, struggling with tracking cross traffic at intersections. Further, measurements are quite noisy, requiring extensive filtering and cleaning. Radar-based vehicle tracking does not strictly detect vehicles; rather it detects and tracks objects, classifying them as vehicles based on relative motion.

Lidar for on-road vehicle detection has increased in popularity in recent years, due to improved costs of lasers, sensor arrays, and computation. Lidar has been used extensively for obstacle detection in autonomous vehicles [31], and are beginning to make their way into driver assistance applications such as adaptive cruise control [19]. Lidar sensing systems emit laser at wavelengths beyond the visual light spectrum, generally between 600-1000 nanometers, typically

scanning the scene at 10-15 Hz [32]. The receiver of the range-finder then senses backscattered energy. Using occupancy grid methods [19], objects of interest are segmented from the background. Segmented objects are then tracked, and classified as vehicles based on size and motion constraints. Figure 2.2(b) depicts the operation of lidar for on-road vehicle detection.

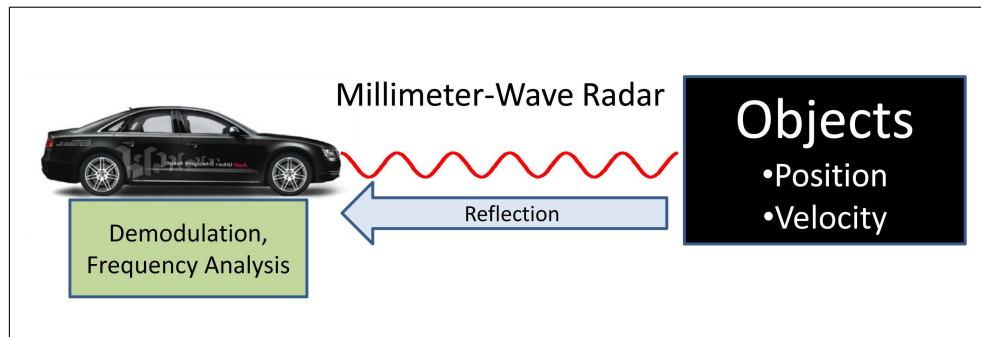
Vehicle-mounted lidar sensing is emerging as a leading technology for research-grade intelligent vehicles, providing cleaner measurements and much wider field-of-view than radar, allowing for vehicle tracking across multiple lanes. However, lidar sensing is more sensitive to precipitation than radar. While cost remains a factor for lidar systems, the price will continue reduce over the next decade. Lidar-based vehicle tracking does not strictly detect vehicles; rather it detects, segments, and tracks surfaces and objects, classifying them as vehicles based on size and motion.

Vision-based vehicle detection uses one or more cameras as the primary sensor suite. Unlike lidar and radar, cameras do not emit electromagnetic energy, rather measuring the ambient light in the scene. In its simplest form, a digital imaging system consists of a lens, and an imaging array, typically CCD or CMOS. Within the field of view of an ideal camera, a point X in the 3D world is mapped to a homogeneous pixel in a digital image via perspective projection, as shown in equation 2.1 [33].

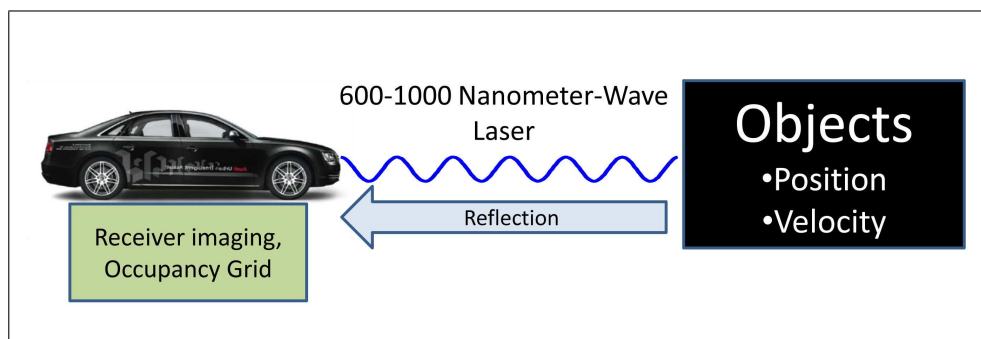
$$\begin{aligned} \mathbf{x} &= K\Pi_0g\mathbf{X} \\ \mathbf{x} &= \begin{bmatrix} x & y & 1 \end{bmatrix}^T, \mathbf{X} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T \\ g &= \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}, \Pi_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned} \tag{2.1}$$

K contains the camera's intrinsic parameters, and g the camera's extrinsic parameters. The mapping converts objects in the real world, to representations in the image plane, converting units from meters to pixels. If multiple cameras are used, image rectification according to epipolar constraints is applied [33], followed by stereo-matching. The end result of capturing an image from a camera is an array of pixels. In the case of stereo-vision, we are left with two arrays of pixels, and an array of disparities, used to calculate distance, after stereo-matching. Figure 2.2(c) depicts vehicle detection using vision.

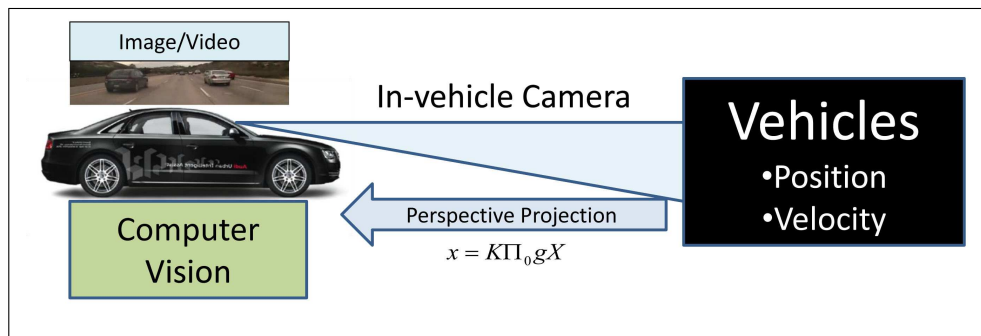
Going from pixels to vehicles is not straight-forward. A visual object detection system requires camera-based sensing to measure the scene's light, as well as computational machinery to extract information from raw image data [33]. Unlike lidar or radar, detection cannot rely on a reflected reference signal. Computer vision techniques are necessary to detect the vehicles in images and video. While vehicle detection using cameras often requires more sophisticated computation, it also features several advantages.



(a)



(b)



(c)

Figure 2.2: a) Radar for on-road vehicle detection uses radar antennas, emitting millimeter wavelength radio signals. The frequency shift in the reflected signal is used to determine distance to an object. b) Lidar for on-road vehicle detection uses laser scanners, emitting illumination at 600-1000nm wavelength, detecting backscattered energy with an imaging receiver, which is used to segment obstacles. c) Vision for on-road vehicle detection uses cameras, which sense the ambient light. Points in the camera’s field of view are mapped to pixels via perspective projection. Computer vision techniques, further detailed in this paper, recognize and localize vehicles from images and video. While radar and lidar detect objects, vision explicitly differentiates vehicles vs. non-vehicles.

Images and video provide a rich data source, from which additional information and context can be surmised. Cameras provide a wide field-of-view, allowing for detection and tracking across multiple lanes. Cameras feature lower costs than active sensors, and are already commonly used for tracking lanes, allowing for system integration [17], shared hardware, and low costs. While active sensors identify objects, vision definitively recognizes objects as *vehicles*. Vision integrates nicely with active sensor suites, allowing sensors like lidar to provide physical measurements, while vision classifies objects as vehicle or non-vehicle [34]. The visual domain is also highly intuitive for humans, making vision-based systems attractive for on-road interactivity and driver assistance. The drawbacks to vision-based vehicle detection include sensitivity to light and weather conditions, and increased computational cost.

As the intelligent vehicles field advances, computer vision will certainly play a prominent sensing role, either as a primary sensor, or as part of multi-modal sensor-fusion suites. In the following sections, we detail the recent advances in vision-based vehicle detection, tracking, and behavior analysis. The recent major advances in monocular vision-based vehicle detection have mirrored advances in computer vision, machine learning, and pattern recognition. There has been immense improvement, from template-matching to sophisticated feature extraction and classification. In stereo-vision, we have seen major advances in stereo-matching, scene segmentation, and detection of moving and static objects. The next section of this paper details the recent advances in monocular, stereo-vision, and fusion of vision with other sensors for on-road vehicle detection.

2.3 Vision-Based Vehicle Detection

The lowest level depicted in Figure 2.1 involves detecting vehicles using one or more cameras. From a computer vision stand-point, on-road vehicle detection presents myriad challenges. The on-road environment is semi-structured, allowing for only weak assumptions to be made about the scene structure. Object detection from a moving platform requires the system to detect, recognize, and localize the object in video, often without reliance on background modeling.

Vehicles on the road are typically in motion, introducing effects of ego and relative motion. There is variability in the size, shape, and color of vehicles encountered on the road [20]. The on-road environment also features variations in illumination, background, and scene complexity. Complex shadowing, man-made structures, and ubiquitous visual clutter can introduce erroneous detections. Vehicles are also encountered in a variety of orientations, including preceding, oncoming, and cross traffic. The on-road environment features frequent and extensive scene clutter, limiting the full visibility of vehicles, resulting in partially-occluded vehicles. Further, a vehicle detection system needs to operate at real-time speeds, in order to provide the human or autonomous driver with advanced notice of critical situations.

In this section, we review on-road vehicle detection. We detail the various cues, assumptions, and classification approaches taken by researchers in the field. We split this section into studies using monocular vision, and those using stereo-vision for on-road vehicle detection. Figure 2.3 shows qualitative results from monocular and stereo-vision based vehicle detection studies.

2.3.1 Monocular Vehicle Detection

We divide vehicle detection approaches into two broad categories: appearance-based, and motion-based methods. Generally speaking, appearance-based methods are more common in the monocular vehicle detection literature. Appearance-based methods recognize vehicles directly from images, that is to say that they go directly from pixels to vehicles. Motion-based approaches, by contrast, require a sequence of images in order to recognize vehicles. Monocular images lack direct depth measurements. Even though ego-motion compensation and structure from motion methods have been used for vehicle detection in [35], generally speaking, appearance-based methods are more direct for monocular vehicle detection.

In this subsection, we discuss camera placement and the various applications of monocular vehicle detection. We then detail common features, and common classification methods. We detail motion-based approaches. We discuss night-time vehicle detection, and monocular pose estimation.

Camera Placement

Vehicle detection using a single camera aims to detect vehicles in a variety of locations with respect to the ego-vehicle. The vast majority of monocular vehicle detection studies position the camera looking forward, to detect preceding and oncoming vehicles, as detailed in later subsections. However, various novel camera placements have yielded valuable insight and safety-critical applications.

Mounting the camera on the side-view mirror, facing towards the rear of the vehicle, has allowed for monitoring of the vehicle’s blind spot. Detecting vehicles with this camera placement presents difficulty because of the field of view, and high variability in the appearance of vehicles, depending on their relative positions. In [36], this camera placement was used to detect overtaking vehicles using optical flow. In the absence of a vehicle in the blind spot, the optical flow of static objects moves backwards, with respect to the ego vehicle. Oncoming vehicles exhibit forward flow. Vehicles were tracked using Kalman filtering [36]. Optical flow was also used for blind spot detection in [37]. Blind spot vehicle detection is also presented in [38], using edge features and Support Vector Machine classification. In [39], a camera is mounted on the vehicle to monitor the blind spot area, The study uses a combination of SURF features, and edge segments. Classification is performed via probabilistic modeling, using a Gaussian-weighted

voting procedure to find the best configuration.

Mounting an omni-directional camera on top of the vehicle has been used to acquire full panoramic view of the on-road scene. In [40], omni-directional vision was used to estimate the the vehicle’s ego-motion, detecting static objects using optical flow. Moving objects were also detected, and tracked over long periods of time using Kalman filtering. In [41], a pair of omni-directional cameras were mounted on the ego-vehicle, performing binocular stereo matching on the rectified images for a dynamic panoramic surround map of the region around the vehicle.

Detection of vehicles traveling parallel to the ego-vehicle, on either side, has also been pursued. In [42], using a camera looking out the side passenger’s window, vehicles in adjacent lanes are detected by first detecting the front wheel, and then the rear wheel. The combined parts are tracked using Kalman filtering. In [43], the camera was similarly mounted on the side of the Terramax autonomous experimental vehicle testbed. An adaptive background model of the scene was built, and motion cues were used to detect vehicles in the side-view.

In [44], the camera was positioned looking backwards, out of the rear windshield. The application was detection the front faces of following vehicles, to advise the driver on the safety of ego lane change. Symmetry and edge operators were used to generate regions of interest, vehicles detected using Haar wavelet feature extraction and SVM classification.

Appearance: Features

A variety of appearance features have been used in the field to detect vehicles. Many earlier works used local symmetry operators, measuring the symmetry of an image patch about a vertical axis in the image plane. Often the symmetry was computed on image patches after evaluating edge operators over the image, to recognize the vertical sides of the rear face of a vehicle [45, 46, 47]. Edge information helps highlight the sides of the vehicle, as well as its cast shadow [48, 49, 38, 50]. Symmetry was used along with detected circular headlights and edge energy to detect vehicles at nighttime in [51]. Symmetry and edges were also used in [52, 53], with longitudinal distance and time-to-collision [TTC] estimated using assumptions on the 3D width of vehicles, and the pinhole camera model 2.1.

In recent years, there has been a transition from simpler image features like edges and symmetry, to general and robust features sets for vehicle detection. These feature sets, now common in the computer vision literature, allow for direct classification and detection of objects in images. Histogram of oriented gradient [HOG] features and Haar-like features are extremely-well represented in the vehicle detection literature, as they are in the object detection literature [21, 54].

HOG features [21] are extracted by first evaluating edge operators over the image, and then discretizing and binning the orientations of the edge intensities into a histogram. The histogram is then used as a feature vector. HOG features are descriptive image features, exhibiting good detection performance in a variety of computer vision tasks, including vehicle detection,

but they are generally slow to compute. HOG features have been used in a number of studies [55, 11]. In [56], the symmetry of the HOG features extracted in a given image patch, along with the HOG features themselves, was used for vehicle detection. Beyond vehicle detection, HOG features have been used for determining vehicle pose [57]. The main drawback of HOG features, is that they are quite slow to compute. Recent work has tackled the speed bottleneck by implementing HOG feature extraction on a graphical processing unit [GPU] [58].

Haar-like features [54] are comprised of sums and differences of rectangles over an image patch. Highly efficient to compute, Haar-like features are sensitive to vertical, horizontal, and symmetric structures, making them well-suited for real-time detection of vehicles or vehicle parts. In [44], Haar features were extracted to detect the front faces of following vehicles, captured with a rear-facing camera. Haar-like features have been extensively used to detect the rear faces of preceding vehicles, using a forward-facing camera [59, 60, 61, 62, 63, 64, 65, 66, 11, 3]. Side profiles of vehicles have also been detected using Haar-like features [42], by detecting the front and the rear wheel. Haar-like features have also been used to track vehicles in the image plane [67]. In [68], Haar features were used to detect parts of vehicles.

While studies that use either HOG or Haar-like features comprise a large portion of recent vehicle detection works, other general image feature representations have been used. In [69], a combination of HOG and Haar-like features [54] were used to detect vehicles. SIFT features [70] were used in [71] to detect the rear faces of vehicles, including during partial occlusions. In [39], a combination of speeded-up robust features [SURF] [72] and edges is used to detect vehicles in the blind spot. In [73] Gabor and Haar features were used for vehicle detection. Gabor features were used in [59], in concert with HOG features. Dimensionality reduction of the feature space, using a combination of PCA and ICA was used in [74] for detecting parked sedans in static images.

Appearance: Classification

Classification methods for appearance-based vehicle detection have followed the general trends in the computer vision and machine learning literature. Classification can be broadly split into two categories: discriminative and generative. Discriminative classifiers, which learn a decision boundary between two classes, have been more widely-used in vehicle detection. Generative classifiers, which learn the underlying distribution of a given class, have been less common in the vehicle detection literature.

While in [59, 75], artificial neural network classifiers were used for vehicle detection, they have recently fallen somewhat out of favor. Neural networks can feature many parameters to tune, and the training converges to a local optimum. The research community has moved towards classifiers whose training converges to a global optimum over the training set, such as Support Vector Machines [22] and Adaboost [76].

Support vector machines [22] have been widely used for vehicle detection. In [44], SVM classification was used to classify feature vectors consisting of Haar wavelet coefficients. The

combination of HOG features and SVM classification has also been used [59, 11, 55]. The HOG-SVM formulation was extended to detect and calculate vehicle orientation using multiplicative kernels in [77]. Edge features were classified for vehicle detection using SVM in [49, 38]. In [73], vehicles were detected using Haar and Gabor features, using SVM classification.

Adaboost [76] has also been widely used for classification, largely owing to its integration in cascade classification in [54]. In [78], Adaboost classification was used for detecting vehicles based on symmetry feature scores. In [79], edge features were classified using Adaboost. The combination of Haar-like feature extraction and Adaboost classification has been used to detect rear faces of vehicles in [62, 80, 81, 82, 64]. In [42], Haar features and Adaboost classification were used to detect the front and rear wheels of vehicles from the side-view. The combination of Haar features and Adaboost classification was used to detect parts of vehicles in [68]. Adaboost classification was used in an active learning framework for vehicle detection in [3, 11]. In [83], online boosting was used to train a vehicle detector. In [84], Waldboost was used to train the vehicle detector.

Generative classifiers have been less common in the vehicle detection literature. This is because it often makes sense to model the classification boundary between vehicles and non-vehicles, rather than the distributions of each class. In [39] a probabilistically-weighted vote was used for detecting vehicles in the blind spot. In [6], motion-based features were tracked over time, and classified using hidden Markov models. In [74], Gaussian mixture modeling was used to detect vehicles in static images. In [71], hidden random field classification was used to detect the rear faces of vehicles.

Recently, there has been interest in detecting vehicles as a combination of parts. The motivation consists of two main goals: encoding the spatial configuration of vehicles for improved localization, and using the parts to eliminate false alarms. In [39], a combination of SURF and edge features are used to detect vehicles, with vehicle parts identified by keypoint detection. In [71], vehicles are detected as a combination of parts, using SIFT features and hidden Conditional Random Field classification. In [85], spatially-constrained detectors for vehicle parts were trained; the detectors required manual initialization of a reference point. The deformable parts-based model [12, 86], using HOG features and the Latent-SVM, has been used for on-road vehicle detection in [87, 88, 4]. In [68, 89], the front and rear parts of vehicles were detected independently, and matched using structural constraints, encoded by an SVM.

Motion-Based Approaches

Motion-based monocular vehicle detection has been less common than appearance-based methods. It is often more direct to use appearance cues in monocular vision, because monocular images do not directly provide 3D depth measurements. Adaptive background models have been used in some studies, in an effort to adapt surveillance methods to the dynamic on-road environment. In [43], an adaptive background model was constructed, with vehicles detected

based on motion that differentiated them from the background. Adaptive background modeling was also used in [90], specifically to model the area where overtaking vehicles tend to appear in the camera’s field of view. Dynamic modeling of the scene background in the area of the image where vehicle typically overtake was implemented in [91]. A similar concept, dynamic visual maps, were developed in [92] for detecting vehicles, and identifying unusual maneuvering in the scene. In [93], a homography matrix was computed between adjacent video frames; image regions that didn’t cleanly map between frames were assumed to be vehicles. This method seems likely to return many false alarms, but quantitative performance analysis was not included.

Optical flow [94], a fundamental machine vision tool, has been used for monocular vehicle detection [95]. In [47], a combination of optical flow and symmetry tracking was used for vehicle detection. In [6], interest points that persisted over long periods of time were detected as vehicles traveling parallel to the ego vehicle. Optical flow was used in conjunction with appearance-based techniques in [62]. Ego-motion estimation using optical flow, and integrated detection of vehicles was implemented in [96, 97, 98]. Ego-motion estimation using omni-directional camera, and detection of vehicles was implemented in [40]. In [36], optical flow was used to detect overtaking vehicles in the blind spot. A similar approach for detecting vehicles in the blind spot was reported in [37]. Cross traffic was detected in [99]. In [100], optical flow was used to form a spatio-temporal descriptor, able to classify the scene as either intersection or non-intersection. Optical flow was used in [101] for segmentation of the on-road scene using video. In [102], ego-motion compensation and motion cues were used for tomographical reconstruction of objects in the scene.

Nighttime Vision

The vast majority of vision-based vehicle detection papers are dedicated to daytime conditions. Nighttime conditions may be dealt with in a few ways. Using high dynamic range cameras [59] allows for the same appearance and classification model to be used during daytime or nighttime conditions. Well-illuminated night-time scenes can also accommodate vehicle detection models that have been designed for daytime conditions [48]. Absent specialized hardware, or illumination infrastructure, various studies have trained specific models for detecting vehicles at nighttime, often by detecting the headlights and taillights of vehicles encountered on the road.

Colorspace thresholding can often serve as the initial segmentation step in detecting vehicle lights in low-light conditions. In [103], vehicles are detected at nighttime using stereo-vision to extract vertical edges and 3D, and color in the $L^*a^*b^*$ colorspace. In [104], taillights are localized by thresholding the grayscale image. Vehicles are detected based on fitting a bounding box around pairs of detected taillights, and 3D range information is inferred by making assumptions on the typical width of a vehicle, and solving for the longitudinal distance using the common pinhole model. In [51], symmetry, edge energy, and detected circles are used to track vehicles using particle filters. In [105], vehicle taillights are paired using cross-correlation, and validated

by tracking. Using the pinhole model, and assumptions on the 3D dimensions of vehicles, the time to collision [TTC] is computed for forward collision warning [FCW] applications. Use of the pinhole model to compute longitudinal distance for night-time vehicles is also featured in [106]. In [5], vehicles are detected by localizing pairs of red taillights in the hue-saturation-value [HSV] color space. The camera has been configured to reliably output colors by controlling the exposure, optimizing the appearance of taillights for segmentation. The segmented taillights are detected as pairs using cross-correlation and symmetry. Vehicles are then tracked in the image plane using Kalman filtering. In [107], multiple vehicles are detected by tracking headlight and taillight blobs, a detection-by-tracking approach. The tracking problem is formulated as a maximum a posteriori inference problem over a random Markov field.

Vehicle Pose

Determining vehicle pose can be useful for understanding how a detected vehicle is oriented, with respect to the ego-vehicle. The orientation can serve to predict the vehicle’s motion. In the absence of 3D measurements, tracking information, coupled with a set of geometric constraints was used in [50] to determine the vehicles’ pose information. In [108], color, location, and texture features were used, with detection and orientation using Conditional Random Field classification.

Simultaneously detecting vehicles and determining their orientations has been pursued with the use of HOG features [21] in various works. HOG features, while expensive to compute, are descriptive and well-suited to distinguishing orientations within an object class. In [109], a set of HOG-SVM classifiers was trained for several orientations. Vehicles were detected in static frames using the all-vs.-one trained detectors. However, it was found that a general HOG-SVM detector performed better at detecting vehicles than the oriented detectors. In [77], multiplicative kernels were used to train a family of HOG-SVM classifiers for simultaneous vehicle detection and orientation estimation. In [57], HOG features were used to discover vehicle orientations in a partition-based unsupervised manner, using simple linear classifiers.

2.3.2 Stereo-vision for Vehicle Detection

Motion-based approaches are more common than appearance-based approaches to vehicle detection using stereo-vision. Multi-view geometry allows for direct measurement of 3D information, which provides for understanding of scene, motion characteristics, and physical measurements. The ability to track points in 3D, and distinguish moving from static objects, affects the direction of many stereo-vision studies. While monocular vehicle detection often relies on appearance features and machine learning, stereo vehicle detection often relies on motion features, tracking, and filtering. Stereo-vision approaches have access to the same image pixels as monocular approaches, but two views allows spatial reasoning, and many research studies

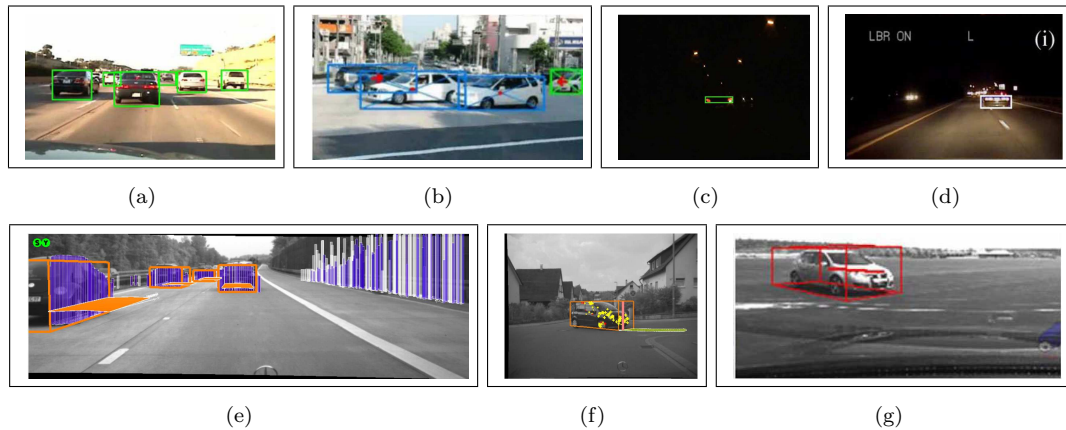


Figure 2.3: Images from representative vehicle detection studies, highlighting real-world system performance. Top Row: Monocular a) Sivaraman and Trivedi, 2010 [3]. b) Niknejad et al., 2012 [4]. c) O’Malley et al., 2010.[5] d) Jazayeri et al., 2011[6]. Bottom Row: Stereo-vision e) Erbs et al., 2011 [7]. f) Barth and Franke, 2010. [1] g) Danescu et al., 2011 [8].

concentrate their efforts on this problem domain.

While [110] places stereo-cameras looking sideways for cross traffic, most studies place the stereo rig looking forwards out the front windshield, to detect vehicles ahead of the ego vehicle. In this subsection, we discuss stereo-matching, appearance-based approaches, and motion-based approaches to vehicle detection using stereo-vision.

Stereo-Matching

Epipolar rectification between the two cameras in a stereo rig transforms the epipolar lines into horizontal scan lines in the respective image planes. This transformation confines the search for point correspondences between two images to the horizontal direction. The set of solved point correspondences yields a disparity map, and is achieved by performing stereo-matching [33]. Various techniques are available in the vision community for dense matching. Advances in dense stereo matching, filtering, and interpolation, have been of great interest in the intelligent vehicles community [111], as better stereo matching allows for better interpretation of the on-road scene. While classic correlation-based stereo-matching has been implemented and highly optimized [112], new advances in stereo-matching are actively pursued in the computer vision and intelligent vehicles communities. In particular, there has been a transition from local, correlation-based approaches [112], towards semi-global matching [113, 114, 115], which features denser disparity maps and lower errors.

Compact Representations

Stereo-vision studies have made extensive use of compact representations of measured data, including occupancy grids [116], elevation maps [117], free space understanding [118],

ground surface modeling [119], and dynamic stixels [7]. Compact representations serve to facilitate segmentation of the scene [119], identify obstacles [120], and reduce computational load. We discuss compact representations in the following subsections, dividing them between appearance and motion-based methods.

Appearance-Based Approaches

Exclusive reliance on appearance cues for vehicle detection is not as common in stereo-vision as monocular vision. While motion-based approaches are more common, even studies that rely on motion for vehicle detection often utilize some appearance-based stereo-vision techniques for initial scene segmentation.

The v-disparity [119] has been widely used to model the ground surface, in order to identify objects that lie above the ground. The v-disparity forms a histogram of disparity values for pixel locations with the same v , vertical image coordinate. Starting with an $n \times m$ disparity map, the result is an image consisting of n stacked histograms of disparity for the image. Using curve-fitting techniques such as the Hough transform [121] or RANSAC [122], disparity can be modeled as a function of the v coordinate of the disparity map, and pixel locations can be classified as belonging to the ground surface if they fit this model [119]. The v-disparity has been widely used in stereo-vision for intelligent vehicles [119, 123, 124, 125, 126, 127, 128, 129].

Free space understanding from disparity maps has been implemented using the u-disparity [130], which forms a similar histogram of stacked disparities, for pixel locations sharing the same u coordinate. Instead of fitting a road model, the u-disparity is used to infer free space directly. Free space computation features heavily in the stereo-vision literature, for scene segmentation and highlighting of potential obstacles. In [131, 118], free space was computed directly from the disparity and depth maps using dynamic programming. In [132], convolutional noise and image degradation are added to stereo image pairs to model the corresponding errors introduced to stereo matching and 3D localization of tracked interest points. Corresponding methods are introduced to compensate for the errors in localization.

Monocular appearance features are sometimes used for vehicle detection using a stereo rig, including color [133], and image intensity [134]. Disparity and depth appearance features are generally more common. In [134], features such as size, width, height, and image intensity were combined in a Bayesian model to detect vehicles using a stereo rig. In [135], a histogram of depths, computed from stereo matching, was used to segment out potential vehicles. Operations directly on the monocular frame also include Delauney triangulation [136].

Various studies have utilized clustering in the depth map for object detection, often using euclidean distance to cluster point clouds into objects [137, 138]. Clustering was also used for object detection in [130]. In [139], clustering was implemented using a modified version of iterative closest point, using polar coordinates to segment objects. The implementation was able to detect vehicles, and infer the vehicle's pose with respect to the ego vehicle. Clustering was used

in tandem with image-based mean shift algorithm for vehicle detection in [140]. The mean-shift algorithm was also used in [141] for object detection.

Motion-Based Approaches

The use of motion features heavily in stereo-based vehicle detection. The foundation for a large portion of stereo-vision analysis of the on-road scene starts with optical flow [94]. In many studies, interest points are tracked in the monocular image plan of one of the stereo rig's cameras, and then localized in 3D using the disparity and depth maps [142]. In [142], the concept of 6D-vision, the tracking of interest points in 3D using Kalman filtering, along with ego-motion compensation, is used to identify moving and static objects in the scene. Optical flow is also used as a fundamental component of stereo-vision analysis of the on-road scene in [143, 123, 136, 135, 144, 145, 146, 8, 142, 140]. A 3-dimensional version of optical flow, in which a least-squares solution to 3D points' motion is solved, is used in [147].

There are various modifications and uses of optical flow point-tracking in the literature. In [145], block-based coarse-to-fine optical flow is compared with classical Lucas-Kanade optical flow, and is found to be more robust to drifting. The object-flow descriptor [100] is used to understand whether the ego vehicle is at an intersection or arterial road, by modeling the aggregate flow of the scene over time. Scene flow is used to model the motion of the background, and regions whose motion differs from the scene flow are categorized as candidate vehicles in [143], where integration with geometric constraints improves vehicle detection performance.

Via tracking, ground plane estimation is implemented by tracking feature points from optical flow [143, 136]. In [123], the ground plane model is fit using total least squares. In [135][143], the ground plane is estimated using RANSAC to fit the plane parameters [122]. The ground is estimated as a quadratic surface in [117], which serves as a scene segmentation for obstacle detection using rectangular digital elevation maps [120]. This work is enhanced in [148] by radial scanning of the digital elevation map, for detecting static and dynamic objects, which are tracked with Kalman filtering. Interest points are also tracked in order to utilize structure from motion techniques for scene reconstruction and understanding. The Longuet-Higgins equations are used for scene understanding in [136] [1] [149]. In [136], tracked interest points are used to estimate ego-motion. In [137], ego-motion estimation is performed by tracking SURF interest points.

In [7], tracked 3D points, using 6D vision, are grouped into an intermediate representation consisting of vertical columns of constant disparity, termed stixels. Stixels are initially formed by computing the free space in the scene, and using the fact that structures of near-constant disparity stand upon the ground plane. The use of the stixel representation considerably reduces the computation expense over tracking all the 6D vision points individually. The tracked stixels are classified as vehicles using probabilistic reasoning and fitting to a cuboid geometric model.

Occupancy grids are widely used in the stereo-vision literature for scene segmentation

and understanding. Static and moving points are tracked in 3D, used to populate an occupancy grid, and compute the free space in the scene using dynamic programming in [118]. Dynamic programming is also used in [150] for computing the free space and populating the occupancy grid. The comparison of spatial representation of the scene is presented, detailing cartesian coordinates, column-disparity, and polar coordinates, in a stochastic occupancy framework. We note that the column-disparity representation of [118] is equivalent to the u-disparity representation of [130]. In [130][151], scene tracking and recursive Bayesian filtering is used to populate the occupancy grid each frame, while objects are detected via clustering. In [123], the occupancy grid’s state is inferred using recursive estimation technique termed the sequential probability ratio test. In [116], the occupancy grid is filtered both temporally and spatially. In [149], the occupancy grid is set up in polar coordinates, the cells assigned depth-adaptive dimensions to model the field of view and depth resolution of the stereo rig. In [8], the occupancy grid is populated using motion cues, with particles representing the cells, their probabilities the occupancy, and their velocities estimated for object segmentation and detection.

2.4 On-Road Vehicle Tracking

Beyond recognizing and identifying vehicles from a given captured frame, vehicle tracking aims to re-identify, and measure dynamics and motion characteristics, and predict and estimate the upcoming position of vehicles on the road. Implicit in vehicle tracking are issues like measurement and sensor uncertainty, data association and track management. In this section, we detail the common vehicle tracking methods employed in the research literature. We split our discussion into portions detailing monocular approaches, and stereo-vision approaches. While there are estimation and filtering methods common to both camera configurations, often the estimated parameters differ, based on the available measurements. Many monocular tracking methods measure and estimate dynamics in terms of pixels, while stereo-vision methods estimate dynamics in meters.

We continue this section by discussing works that combine or fuse monocular and stereo-vision for on-road vehicle detection and tracking. We then discuss papers that focus on optimized system architecture, and real-time implementation of on-road vehicle detection and tracking. We conclude this section with discussion of studies that fuse vision with other sensing modality for on-road vehicle detection and tracking.

2.4.1 Monocular Vehicle Tracking

Using monocular vision, vehicles are typically detected and tracked in the image plane. Tracking using monocular vision serves 2 major purposes. Tracking facilitates estimation of motion, and prediction of vehicle position in the image plane. Secondly, tracking enforces tem-

poral coherence, which helps to maintain awareness of previously-detected vehicles that were not detected in a given frame [67], while filtering out spurious false positives [3].

The goal in monocular vision is to measure the motion and predict the position of vehicles in pixel position and pixel velocity. The observation space, based on pixels, gives way to uniquely vision-based tracking methods, based on the appearance of the vehicle in the image plane. An example of uniquely vision-based tracking is template matching. In [44], vehicles were detected in the image plane using Haar wavelet coefficients and SVM classification. Vehicles were tracked from frame to frame by taking a measurement of the similarity in appearance.

Often the appearance-based tracking is based on cross-correlation scores. Vision-based tracking is taken one step further using feature-based tracking [152]. In [67], vehicles were detected using Haar-like features and Adaboost cascade classification. Candidate vehicles locations were predicted using Kalman filtering in the image plane. The measurements in the image plane were determined by a local search over the image patch for similar feature scores, allowing for a measurement even if the detector failed in a particular frame. Optical flow has also been used to track vehicles, by directly measuring the new position and the displacement of interest points [153].

Conventional tracking and Bayesian filtering techniques have been widely used in the monocular vehicle tracking literature. The state vector typically consists of the pixel coordinates that parametrize a rectangle in the image plane, and their inter-frame pixel velocities [48]. In [42, 47, 53], Kalman filtering was used to estimate the motion of detected vehicles in the image plane. Particle filtering has also been widely used for monocular tracking in the image plane [51, 3, 4, 87, 48, 154].

Estimating longitudinal distance, and 3D information, from monocular vision has been attempted in various vehicle tracking studies. Typically, the ground plane is assumed flat [62, 155], or its parameters estimated using interest point detection and a robust mode-fitting step, such as RANSAC [122]. In [50], a set of constraints and assumptions were used to estimate 3D coordinates from monocular vision, and Kalman filtering was used to track vehicles in 3D. In [45], 3D information was inferred using ground plane estimation, and interacting multiple models were used to track vehicles, each model consisting of a Kalman filter. In [97, 98], ego-motion was estimated using monocular vision, and moving objects were tracked in 3D using Kalman filtering. While various tracking studies have estimated 3D vehicle position and velocity information from monocular measurements, few such studies have compared their measurements to a ground-truth reference 3D measurement, from radar, lidar, or stereo-vision for example.

2.4.2 Stereo-vision Vehicle Tracking

Vehicle tracking using stereo-vision concerns itself with measuring and estimating the position and velocity, in meters, of detected vehicles on the road. The state vector often consists

of the vehicle’s lateral and longitudinal position, width and height, as well as velocity. Estimation is most often implemented using Kalman filtering, which is considered optimal assuming linear motion and Gaussian noise [111]. In reality, vehicle motion is non-linear, with the vehicle’s yaw rate describing the vehicle’s turning behavior. Using the extended Kalman filter [EKF] is often used for estimating non-linear parameters, by linearizing the motion equations for estimation [156]. Particle filtering has been used as an alternative to measure both linear and non-linear motion parameters [147], using sample importance re-sampling [SIR] in place of the linearization of the EKF.

Kalman filtering for stereo-vision vehicle tracking has been widely used [144] for vehicle tracking, as well as disparity filtering. Noise in stereo-matching is generally modeled as white Gaussian noise [142, 132], and filtering over time can produce cleaner disparity maps [111]. Kalman filtering is used to track individual 3D points in [118, 142]. Kalman filtering is used to track stixels, intermediate vertical elements of near-constant depth, which are fit to cuboid vehicle models [7]. In [135, 28], vehicles are detected in the monocular plane using an Adaboost-based classification, and tracked in 3D using Kalman filtering in the stereo domain. In [146], vehicles’ positions and velocities are estimated using Kalman filtering. In [157], Kalman filtering is used to track objects detected by clustering, stereo matching linear cameras. In [1], Kalman filtering is used to estimate the vehicles’ yaw rate, as well as position and velocity.

The extended Kalman filter has also been used widely in stereo-vision vehicle tracking, specifically to account for non-linearity in motion and observational model quantities. The extended Kalman filter was used to estimate the yaw rate, and corresponding turning behavior of vehicles in [156, 158]. Extended Kalman filtering for vehicle tracking was particularly apt due to camera positioning in [110], with the side-mounted stereo rig observing particularly non-linear motion of tracked vehicles, with respect to the camera’s frame of reference. Extended Kalman filtering was used to model the non-linearity of mapping a vehicle’s 3D position into stereo image position and disparity in [123]. In [136], extended Kalman filtering was used to estimate the ego-motion, with independently-moving objects’ position and motion estimated using Kalman filtering. Vehicles were also tracked using extended Kalman filtering in [124].

Particle filtering for vehicle tracking has also been fairly widely used. In particular, particle filtering offers an alternative to the extended Kalman filter’s estimation of non-linear parameters, as the particle filter’s multiple hypotheses are weighted by a likelihood function. In [147], vehicles were tracked using particle filter, estimating their 3D position and yaw rate. In [140], the motion of tracked vehicles was estimated using a particle filter that mapped the motion to full trajectories, learned from prior observational data. In [8], the on-road environment is modeled using particles that serve a dual purpose, as occupancy cells, and as tracking states for detected objects and obstacles.

Interacting multiple models have been used in tracking, to estimate the motion of a vehicle given different motion modes. In [1], four different predominating modes were used to

model the motion of oncoming vehicles at intersections, in terms of their velocity and yaw rate characteristics. The goal was to identify whether the velocity was constant or accelerated, and whether the yaw rate was constant or accelerated. The model fit was determined using the error covariance of each estimator. A state transition probability was used to switch between competing modes after model fit was determined [1]. Interacting multiple models were also used in [110]. The use of interacting multiple modes will likely increase in popularity, as measurements become more precise, and as it becomes apparent that all the motion parameters cannot be well-estimated by a single linear or linearized filter.

2.4.3 Fusing Monocular and Stereo-Vision Cues

Various studies have fused monocular and stereo-vision for on-road vehicle tracking. We draw a distinction between papers that use optical flow and stereo-vision for vehicle detection, and those papers that use monocular computer vision for full vehicle detection, typically relying on machine learning, and stereo-vision for tracking in 3D.

The use of both monocular and stereo-vision cues typically manifests itself in the use of monocular vision for detection, and stereo-vision for 3D localization and tracking. In [159], it was noted that monocular vision can detect objects that stereo-vision approaches typically miss, such as disambiguation of two objects that lie close together in 3D space. This problem was addressed by detecting in the monocular plane, but localizing in 3D using stereo-vision. In [160] monocular symmetry was used to generate vehicle candidate regions, and stereo-vision to verify those regions as vehicles, by searching for vertical objects in the 3D domain. In [28], vehicles were tracked in the image plane using a monocular vehicle detector [3], and tracked in 3D using stereo-vision and Kalman filtering. Clustering on aggregates vehicle tracking data from the system presented in [28] was used for learning typical vehicle behavior on highways in [2].

In [124], vehicles were detected using an Adaboost classifier on the monocular plane. The v-disparity was used to estimate the ground surface, and vehicles were tracked using extended Kalman filtering in the stereo-vision domain. Track management for reduction of false alarms, and improved precision was presented. In [135], vehicles candidates regions were selected in the image plane using a set of Adaboost detectors, trained for multiple vehicle views. The candidate regions were verified by looking for peaks in the disparity range, Stereo-vision was used for 3D ranging, and for estimating the ground plane.

2.4.4 Real-time Implementation and System Architecture

Eventually, for a vehicle detection and tracking system to be of utility on the road, real-time implementation is necessary, typically processing above 10 frames per second. While some detection algorithms run in real-time on standard CPU's, many do not, and further efforts are necessary to optimize the implementation in hardware and software.

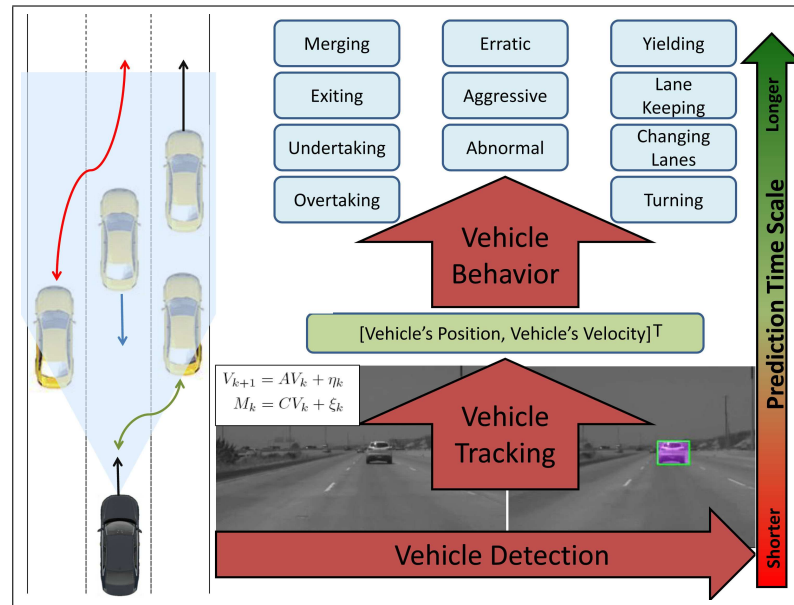


Figure 2.4: The ascending levels vehicle behavior interpretation. At the lowest level, vehicles are detected using vision. Vehicle tracking estimates the motion of previously-detected vehicles. At the highest level of interpretation, vehicle behavior is characterized. Behavior characterization includes maneuver identification, on-road activity description, and long-term motion prediction.

Efforts to implement vehicle detection algorithms using embedded systems implementations have garnered attention in recent years. In [161], vehicle detection using shadow features was implemented embedded system. In [162], a boosting based vehicle detector was implemented on an embedded system. In [163], nighttime vehicle detection was implemented on an embedded system. Embedded implementation of stereo-vision based lane and vehicle detection and tracking was reported in [164]. Commercialization of embedded vision-based vehicle detection has also hit the market [165]. Embedded systems for correlation-based stereo-matching have also been commercialized [112].

In recent years, the availability of graphical processing units [GPU] has enabled real-time implementation and parallelization of computationally expensive vision algorithms for vehicle detection and tracking. In [166], an early GPU was used for monocular vehicle detection. In [167], real-time stereo matching using semi-global matching [113] is implemented on the GPU. In [58], the GPU was used to implement real-time vehicle detection using HOG [21] features. GPU implementation was used in [116] for real-time occupancy grid computation. In [168], vehicle detection was implemented using a fusion of stereo-vision and lidar on the GPU.

2.4.5 Fusing Vision with other Modalities

Over the past decade, the availability and cost of a variety of sensors has become favorable for integration in intelligent vehicles, for driver assistance and for autonomous driving. It is widely

accepted that fully-autonomous vehicles will need an advanced sensor suite, covering a variety of sensing modalities, in order to sense, perceive, and respond to the on-road environment in a safe and efficient manner. Leading research in autonomous vehicles feature sensor suites that include cameras, lidars, and radar sensing arrays [31]. Often, the vision algorithms used in sensor-fusion studies closely resemble those in vision-only studies, with information fusion performed across modalities, to reduce uncertainty, cover blind spots, or perform ranging with monocular cameras.

Radar-vision fusion for on-road vehicle detection and perception has received quite a bit of attention in recent years [169]. Radars have good longitudinal ranging coupled with crude lateral resolution; monocular vision can localize well in the camera’s field of view, but lacks ranging. The combination of the two can ameliorate the weakness of each sensor [170, 171]. In [172, 173], information fusion between radar and vision sensors was used to probabilistically estimate the positions of vehicles, and to propagate estimation uncertainty into decision-making, for lane-change recommendations on the highway. In [174], vision and radar were combined to detect overtaking vehicles on the highway, using optical flow to detect vehicles entering the camera’s field of view. Radar and vision were combined in [175], with radar detecting side guardrails, and vision detecting vehicle using symmetry cues.

Several studies perform extrinsic calibration between radar and camera sensors. In [176], obstacles are detected using vision operations on the inverse perspective mapped image, and ranged using radar. In [177], vehicles are detected with a boosted classifier using Haar and Gabor features, and ranged using radar. In [170], camera and radar detections were projected into a common global occupancy grid, vehicles tracked using Kalman filtering in a global frame of reference. In [178], potential vehicles were detected using saliency operations on the inverse perspective mapped image, and combined with radar. In [179], vehicles were detected using a combination of optical flow, edge information, and symmetry, ranged with radar, and tracked using interacting multiple models with Kalman filtering. In [180], symmetry was used to detect vehicles, with radar ranging. In [181], vehicles were detected using HOG features and SVM classification, ranged using radar. In [35], monocular vision was used to solve structure from motion, with radar providing probabilities for objects and the ground surface. In [182], a radar-vision online learning framework was utilized for vehicle detection. Stereo-vision has also been used in conjunction with radar sensing [183, 184].

Fusion of lidar with monocular vision has been explored in recent years. Several studies perform extrinsic calibration between lidar and camera sensors, using monocular vision for vehicle detection, and lidar for longitudinal ranging. In [185], monocular vision was used to detect vehicles using Haar-like features, and ranging was performed using lidar. A similar system was presented in [34, 186]. Saliency was used as the vision cue in [187], fused with lidar in a Bayesian framework. Fusion of stereo-vision with lidar was performed in [188, 189, 190, 191].

2.5 On-Road Behavior Analysis

Analysis of the behavior of tracked vehicles has emerged as an active and challenging research area in recent years. While considerable research effort has been dedicated to on-road detection and tracking of vehicles in images and video, going from pixels to vehicles with positions and velocity, the highest level of semantic interpretation lies in characterizing the behavior of vehicles on the road. In order to analyze the on-road behavior of other vehicles, robust vehicle detection and tracking are prerequisite. While various studies have modeled the vehicle dynamics [192] and driver gestures [193] associated with the ego-vehicle’s maneuvering, research into the on-road behavior of other vehicles is a relatively recent development. Figure 2.4 depicts on-road behavior analysis in the context of vision-based understanding of the driving environment. At the lowest level, detection takes place, recognition and localizing vehicles on the road. One level up, tracking re-identifies vehicles, measures their motion characteristics using a motion model. Often, linear or linearized models are used. At the highest level, using spatial and temporal information from vehicle detection and vehicle tracking, vehicle behavior analysis is performed.

Research studies in this area take a variety of approaches to characterize on-road behavior. Certain studies try to categorize observed vehicle behavior as normal or abnormal [92], identifying and highlighting critical situations. Other studies try to identify specific maneuvers, such as overtaking [174], turning [1], or lane changes [29]. Most recently, studies in the literature have tried to make long term classification and prediction of vehicle motion. While vehicle tracking, often based on Kalman filtering, can make optimal estimation of the vehicle state one frame [$\frac{1}{25}$ sec.] ahead of time, trajectory modeling approaches try to predict vehicle motion up to 2 seconds ahead, based on models of typical vehicle trajectories [194]. Figure 2.5 depicts trajectory prediction.

Broadly speaking, we categorize studies that address the characterization of on-road vehicle behavior based on four main criteria. Firstly, we consider the role of context in the analysis of on-road behavior, loosely defined to encompass considerations such as urban driving vs. highway driving, or intersection vs. non-intersection driving. Secondly, we consider the identification of pre-specified maneuvers, such as turning, lane change, or overtaking maneuvers of tracked vehicles on the road. Thirdly, we consider the use of *trajectories*, long-term sequences of positions and velocities, in characterizing on-road behavior. Finally, we consider classification and modeling found in the literature.

2.5.1 Context

The use of context is a vital component of many studies that characterize on-road vehicle behavior. The motion model used in [6] models the distribution of vehicles in the image plane, using it as a prior probability on vehicle detections. The vehicle detection in [6] can be viewed as a detection-by-tracking approach, enabled by spatio-temporal modeling of the driving context. In

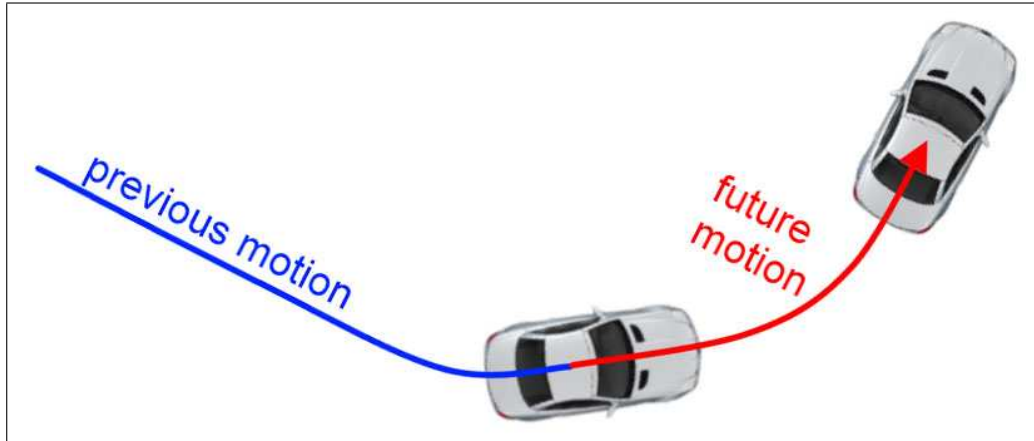


Figure 2.5: A depiction of trajectory prediction, aiming to map the most likely future vehicle motion, based on observed motion [9].

[100], histograms of scene flow vectors are used to classify the driving environment as intersection or non-intersection driving, modeling the driving context using spatio-temporal information. In [2], a context-specific spatio-temporal model of highway driving is developed by performing clustering on observed vehicle trajectories on highways. In [140], the trajectories of vehicles are recorded as they navigate a roundabout, and are used to model the long-term behavior of vehicles in roundabouts. In [1], the context of interest is intersections, the turning behavior of oncoming vehicles inferred. In [92], a dynamic visual model is developed of the driving environment, with saliency alerting the system to unusual and critical on-road situations.

2.5.2 Maneuvers

A body of work has been dedicated to the detection of specific maneuvers of vehicles on the road. In [36], overtaking behavior is detected, by detecting vehicles in the blind spot of the ego-vehicle. Overtaking behavior is specifically detected in [91], this time with the camera pointing forward, and vehicles detected as they overtake in front of the ego-vehicle. In [174], overtaking behavior is also detected in front of the ego-vehicle, using a fusion of vision and radar. Overtaking behavior is also detected in [90], also for vehicles in front of the ego-vehicle. In these studies, the overtaking maneuver is detected by virtue of detecting the vehicle, as the search space includes only vehicles that are in the process of overtaking.

By contrast, specific maneuvers are identified in other works via inference on tracking information. In [156], the turning behavior of tracked vehicles is identified by measuring the yaw rate using extended Kalman filtering. Using the yaw rate in the vehicle motion model, the system is able to detect turning behavior. In [1], turning behavior is further addressed, using interacting multiple models to characterize the motion of the oncoming vehicle. The model with the highest likelihood, based on observations, characterizes the turning behavior of the oncoming

vehicle, with a transition probability handling change of states. Turning behavior is addressed in [139] by solving the vehicle’s pose, with respect to the ego vehicle using clustering of 3D points.

On-road vehicle behavior is modeled in [195] as a Markov process, and inferred using a dynamic Bayesian network, based on tracking observations. However, the experimental evaluation is performed using simulation data. In [29], the lane change behavior of tracked vehicles is modeled using a dynamic Bayesian networks, and the experiments are performed on real-world vision data.

2.5.3 Trajectories

The use of vehicle trajectories, to characterize and learn on-road vehicle behaviors, has emerged in the past few years. A trajectory is typically defined as a data sequence, consisting of several concatenated state vectors from tracking, meaning an indexed sequence of positions and velocities over a given time window. Using a time window of 1 second, for example, can mean trajectories consisting of 25-30 samples, depending on the frame rate of the camera.

In [9], variational Gaussian mixture modeling is used to classify and predict the long-term trajectories of vehicles, using simulated data. In [2], highway trajectories are recorded using stereo-vision, and clustering is performed to model the typical trajectories encountered in highway driving, with classification performed using hidden Markov modeling. In [194], trajectories are classified using a rotation-invariant version of the longest common subsequence as the similarity metric between trajectories. Vehicle trajectories are used to characterize behavior at roundabouts in [140], using the QRLCS metric to match observed trajectories to a database of pre-recorded trajectories. Similar work is carried out in [196] for vehicles at intersections.

2.5.4 Behavior Classification

Classification of vehicle behavior is performed using a variety of techniques, dependent on the objective of the study. In [91, 90, 174, 36], the vehicle detection task encompasses the classification of vehicle behavior. This is to say, that these studies aim to detect vehicles that overtake the ego-vehicle, so in these cases, vehicle detection is synonymous with vehicle behavior characterization. By contrast, yaw-rate information is used in [156] to characterize the turning behavior of tracked vehicles. In this case, measured motion characteristics describe the specific maneuver.

In studies that explicitly classify vehicle behavior, we see a preponderance of generative modeling. In [9], Gaussian mixture modeling is used, which provides distribution over the prediction, complete with a point-estimate [the conditional mean], and a covariance to convey uncertainty. In [1], the likelihood of the interacting multiple model tracking is used to classify the tracked vehicle’s turning behavior, complete with a transition probability. In [29], Bayesian networks are used for classifying the vehicle’s behavior, and predicting the vehicle’s lane change. In

[2], hidden Markov modeling is used to model each of the prototypical trajectories, learned using clustering. In [195], the vehicle behavior is also modeled as a Markov process, with observations coming from the vehicle’s instantaneous state vector.

2.6 Discussion and Future Directions

In this section, we provide discussion, critiques, and perspective on vision-based vehicle detection, tracking, and on-road behavior analysis.

2.6.1 Vehicle Detection

In recent years, the feature representations used in monocular vehicle detection have transitioned from simpler image features like edges and symmetry, to general and robust features sets for vehicle detection. These feature sets, now common in the computer vision literature, allow for direct classification and detection of objects in images. HOG and Haar-like features are extremely-well represented in the vehicle detection literature, as they are in the object detection literature [21, 54]. While early works heavily relied upon symmetry, symmetry is typically not robust enough to detect vehicles by itself. The on-road environment features many objects that feature high symmetry, such as man-made structures and traffic signs. Many daytime vehicle detection studies that have used symmetry as the main feature do not provide an extensive experimental evaluation. Symmetry is more likely to serve to generate regions of interest for further feature extraction and classification [78, 38, 44, 55]. In [56], a novel analysis of the symmetry of HOG features is used to detect vehicles.

Learning and classification approaches have also transitioned in recent years. While neural networks can deliver acceptable performance, their popularity in research communities has waned. This is mainly due to the fact that neural network training features many parameters to tune, and converges to a local optimum over the training set. Competing discriminative methods converge to a global optimum over the training set, which provides nice properties for further analysis, such as fitting posterior probabilities [197]. Studies that use SVM and Adaboost classification are far more prevalent in the modern vehicle literature, a trend which mirrors similar movement in the computer vision and machine learning research communities.

While monocular vehicle detection has been an active research area for quite some time, open challenges still remain. It is challenging to develop a single detector that works equally well in all the varied conditions encountered on the road. Scene-specific classifiers, categorizing the on-road scene as urban vs. highway, cloudy vs. sunny could augment the performance of vehicle detectors, utilizing image classification as a preprocessing step [198].

Monocular vehicle detection largely relies on a feature extraction-classification paradigm, based on machine learning. This approach works very well when the vehicle is fully-visible. In

particular, robustly detecting partially-occluded vehicles using monocular vision remains an open challenge. Early work in this area, is ongoing, based on detecting vehicles as a combination of independent parts [68], but detecting partially-occluded vehicles remains a challenging research area. Using parts to detect vehicles has been implemented in [4], but the recognition still has difficulty with occlusions. Future works will need to include motion cues into monocular vehicle detection, to identify vehicles as they appear, while seamlessly integrating them into machine learning frameworks.

Object detection using stereo-vision has also made great progress over the past decade. Advances in stereo matching yield much cleaner, less-noisy, denser disparity maps [113]. Improved stereo matching enables more robust scene segmentation, based on motion and structural cues [7]. While stereo matching and reconstruction has improved, stereo-vision methods typically recognize vehicles in a bottom-up manner. This is to say that the typical paradigm consists of ego-motion compensation, tracking feature points in 3D, distinguishing static from moving points, and associating moving points into moving objects [8]. Finally, moving objects are labeled as vehicles by fitting a cuboid model [146], or clustering [139]. While these methods have made great progress, complex scenes still present difficulty [7]. Integration of machine learning methodology could increase the robustness of existent stereo-vision approaches, and has the potential to simplify the vehicle detection task. Research along these lines has been performed by using machine learning based detection on the monocular plane, integrating stereo-vision for validation and tracking [124, 28, 135]. Future work could involve a more principled machine learning approach, learning on motion cues, image cues, and disparity or depth cues.

As the cost of active sensors, such as radar and lidar, continue to reduce, integration of these sensing modalities with vision will continue to increase in prevalence. Automotive radar and lidar systems are fairly mature in their ability to detect objects and obstacles, but their ability to distinguish vehicles from other objects is limited. Currently, radar and lidar systems distill their detections into multiple object lists. As lane tracking cameras become standard options on serial production vehicles, the opportunity to integrate vision with active sensing technology will present itself. Vision allows an intuitive level of semantic abstraction, that is not otherwise available with lidar or radar. Many studies detect vehicles with one modality, and validate with the other [174, 188]. Others detect vehicles with vision, and range with radar or lidar [34]. Future works will need a principled, object-level fusion of vision and radar/lidar for vehicle detection [187]. Such an information fusion could reduce estimation covariance and enhance robustness, although the asynchronous nature of the multiple modalities will need to be handled [199].

2.6.2 Vehicle Tracking

While early works in monocular on-road vehicle tracking used template matching [44], recursive Bayesian filtering approaches, such as Kalman filtering [67] and particle filtering [4]

have become the norm. Estimation of a tracked vehicle’s position in the image plane can be augmented by using optical flow, as vision-based vehicle detectors can fluctuate in their pixel locations from frame to frame, even for a static object. Future work in monocular vehicle tracking will pursue information fusion of the optical flow motion, and the motion from vehicle detections in consecutive frames. To this end, low-level motion cues can improve monocular vehicle tracking, and provide a basis for enhanced track management, when detections are dropped in certain frames.

Vehicle tracking in the stereo-vision domain, by contrast, is extremely mature. Indeed, most vehicle detection approaches using stereo-vision are based on motion and tracking, from interest points, to 3D points, to clustered objects, to cuboid vehicle models [7]. Estimation of the vehicle’s yaw rate has emerged as a very important cue, for identifying turning behavior, and for improved prediction of the vehicle’s motion [156]. Extended Kalman filtering for vehicle tracking has increased in popularity to accommodate the non-linear observation and motion models [124, 156]. Particle filtering has also increased in popularity for vehicle tracking, while dispensing with some assumptions required for Kalman filtering [8]. Interacting multiple models, however, seem to best account for the different modes exhibited by vehicle motion on the road. In [1], motion modes for constant velocity and yaw rate, constant velocity and yaw acceleration, constant acceleration and yaw rate, and constant acceleration and yaw acceleration were all available to best model the motion of a tracked vehicle, with a likelihood measurement to choose the best fit. Such modeling allows for sudden changes in vehicle motion, without simply compensating for the changes with the noise terms. Including a transition probability between modes increases the estimation stability, and draws powerful parallels with established techniques in the analysis of Markov chains.

2.6.3 On-Road Behavior Analysis

On-road behavior analysis speaks to a higher level of semantic interpretation of the driving environment, and is the least mature area of research. While this level of analysis is dependent on robust vehicle detection and vehicle tracking, the aim is to answer questions beyond those answered by detection and tracking. These issues include identification of maneuvers, characterization of vehicle behavior, and long-term motion prediction. Only in the past few years have vehicle detection and tracking methodologies become sufficiently mature to enable the exploration of these deeper questions.

Recognition of specific maneuvers has so far been coincident with detection of vehicles in a particular location relative to the ego-vehicle [36], or directly inferred by dynamical information from tracking vehicles [156]. While dynamical models of vehicle motion are well established, identification of vehicle maneuvering has so far been implemented in a maneuver-specific manner. Future works will need to formulate general models of vehicle maneuvering, which allow for

picking the most likely current maneuver from a pool of available classes [29]. This could be implemented using generative models [195], or all-vs-one discriminative classification for maneuver detection. Ongoing research will also have to account for various traffic and road conditions, with different models for urban vs. highway driving, arterial vs. intersections, free-flow vs. congested driving conditions. Predicting a vehicle’s maneuver requires making a decision with a partially-observed sequence of tracking data. Integration of latent variable models [200, 12] will play a role in identification of maneuvers.

Recognition of overtaking maneuvers have been an active research area [36, 174]. The main difference between an overtake and a lane change is the presence of a vehicle in the target lane, and acceleration to keep a safe distance. A similar distinction exists between so-called undertaking and lane changing. In real-world settings, vehicle motion is constrained by traffic, infrastructure, and other vehicles. Modeling the interactions between vehicles is an open area of research, in the context of vehicle behavior characterization. In [201], the distances and timegaps between vehicles were used as a feature for predicting the driver’s intent to change lanes. Further research will need to be conducted to characterize the interactions between vehicles, and their role in on-road behavior. The vehicle dynamics associated with a specific maneuver can be learned, in a data-driven manner, from the controller area network bus [CANbus] of the ego vehicle, presence and absence of vehicles in the target lane. It would then be a research question to determine an appropriate and sufficient distillation of the data to classify and predict those maneuvers in unlabeled data. The inertial sensors available on the ego-vehicle provide more data signals, each of higher precision, than can reasonably be measured using vision. The research question will be concerned with detecting maneuvers based on the parameters that are observable and robustly estimated.

In the analysis and characterization of vehicle behavior, a major challenge will be in identifying erratic, abnormal, and aggressive driving by other vehicles. While identifying specific maneuvers can be formulated as a well-defined problem, for example turning [1], characterizing another vehicle’s behavior remains an open question. Given the tracking of vehicles with respect to their own lanes, weak cues such as a vehicle’s veering within its lane, or crossing over lane boundaries could be used. More likely, research studies will try to characterize normal driving behavior for a given context in a data-driven manner, and identify abnormal trajectories by measuring the model fit of an observed vehicle trajectory [2].

Filtering approaches like the Kalman filter can provide a good estimate of a vehicle’s position, velocity, and other parameters one frame ahead of time, with typical camera frame rates between 10 and 25 frames per second. Long term motion prediction requires an estimate of the vehicle’s motion 1-2 seconds, or 25-50 frames, ahead of time, outside the capabilities of conventional filtering techniques. Long-term motion classification and prediction will involve further research into learning and modeling of vehicle trajectories. An enhanced understanding of vehicle trajectories will allow on-board systems to infer the intent of other vehicle’s drivers,

based on sequential tracking measurements from vision-based systems.

A transition will need to take place, from measuring a tracked vehicle’s motion in the coordinate frame of the ego-vehicle, to position-independent coordinate frames, such as the sequence of angles approach used in [202]. While vision-based vehicle tracking research tends to measure the position and motion of vehicles in the coordinate frame of the ego-vehicle, there needs to be a movement towards understanding the motion and behavior of other vehicles as independent traffic agents. Trajectories, sequences of vehicle tracking data, may be well modeled by Markov chains, but there will need to be a transition probability between sequences, to account for drivers changing their minds last minute. To this end, we foresee learned trajectory models working in concert with established tracking approaches like interacting multiple models. A full vehicle motion understanding engine would include multiple trackers with distinct motion models to estimate vehicle state in the short-term, interacting multiple models to identify vehicle maneuvering in the medium term, and trajectory learning to predict vehicle motion in the long term. Associated issues, such as data windowing, and online model updates, will also need to be addressed.

2.6.4 Benchmarks

We briefly discuss the benchmarks that are publicly available, and commonly used performance metrics in vehicle detection, tracking, and behavior analysis. Table 2.2 provides a summary of some of the publicly available, and widely used datasets. While these datasets are available, we note that it is still common practice for research groups to capture their own video data, for use in training and testing. Like many computer vision and machine learning research areas, vehicle detection is not so easily summarized by one standard dataset. The driving environment features high variability in illumination, weather, and road conditions. Further, vehicle models and road structure differ across the globe, meaning European, Asian, and North American datasets will certainly differ.

Until recently, few vehicle detection studies, especially in the monocular domain, evaluated their performance in realistic conditions, using real-world on-road video. Prior works would report results on static images, often subsets of databases that were created for object recognition, such as Caltech [203, 204]. A lot of research is funded by automotive manufacturers, and their CAN signals, inertial measurements of production vehicles are proprietary information. As such, on-road vehicle detection does not have a strong history of standard benchmark datasets, unlike other computer vision disciplines [10]. In the past few years, this has changed, and published research works now regularly report results on real-world video. It is now becoming a standard practice for researchers to release datasets, source code, and even camera calibration parameters, which will help the research community make further progress in on-road vehicle detection. However, only very recently have published works started to make their datasets publicly available,

Table 2.2: Vehicle Detection Benchmarks

Dataset	Description
Caltech 1999 [203], 2001 [204]	Static images of vehicles in a variety of poses.
PETS 2001 [205]	Testing set of some 2867 frames, from two cameras. Includes videos of preceding vehicles viewed through the front windshield, and a video of oncoming vehicles viewed through the rear windshield.
LISA, 2010. [3]	Three short videos, 1500, 300, and 300 frames, comprising highway and urban driving. Monocular detection of preceding vehicles only.
Caraffi, 2012. [84]	Several videos, comprising some 20 minutes of driving on italian highways
KITTI, 2012[206]	Extended videos from stereo pairs, complete with lidar data. Monocular, stereo-vision, and sensor fusion evaluation is possible.

Table 2.3: Monocular Vehicle Detection Metrics

Performance Metric	Definition
Detection Rate/True Positive Rate/ Recall	$\frac{\# \text{ True Positives}}{\# \text{ Vehicles}}$
False Positive Rate	$\frac{\# \text{ False Positives}}{\# \text{ Possible Bounding Boxes}}$
False Detection Rate/ 1 – Precision	$\frac{\# \text{ False Positives}}{\# \text{ True Positives} + \# \text{ False Positives}}$
False Positives per Frame/ False Positives per Image	$\frac{\# \text{ False Positives}}{\# \text{ Frames}}$

for evaluation and benchmarking by others in the research community.

In monocular vehicle detection, commonly-used benchmarks quantify recall of true positive vehicles, and false alarms. Given a ground truth annotation G_i in a frame, and a detection D_i , the detection is deemed a true positive if the overlap of the two exceeds a threshold τ , as shown in equation 2.2. Figure 2.6 depicts the overlap criterion for detections and ground truth bounding boxes in the image plane.

$$D_i = \begin{cases} \text{True positive,} & \text{if } \frac{G_i \cap D_i}{G_i \cup D_i} > \tau \\ \text{False positive,} & \text{otherwise} \end{cases} \quad (2.2)$$

A detection D_i that does not have sufficient overlap with the ground truth annotation, including zero overlap, is deemed a false positive. Detection and false positives are counted over a video sequence. Dividing the number of true and false positives by a variety of denominators yields a set of metrics that have been used in the field. For detections, the true positive rate, or recall is almost uniformly used. For false alarms, common metrics include $1 - \textit{Precision}$ or false detection rate, false positive rate, false positives per frame, and false positives per object. Table 2.3 defines these terms. Monocular tracking studies typically use the same metrics for tracking as

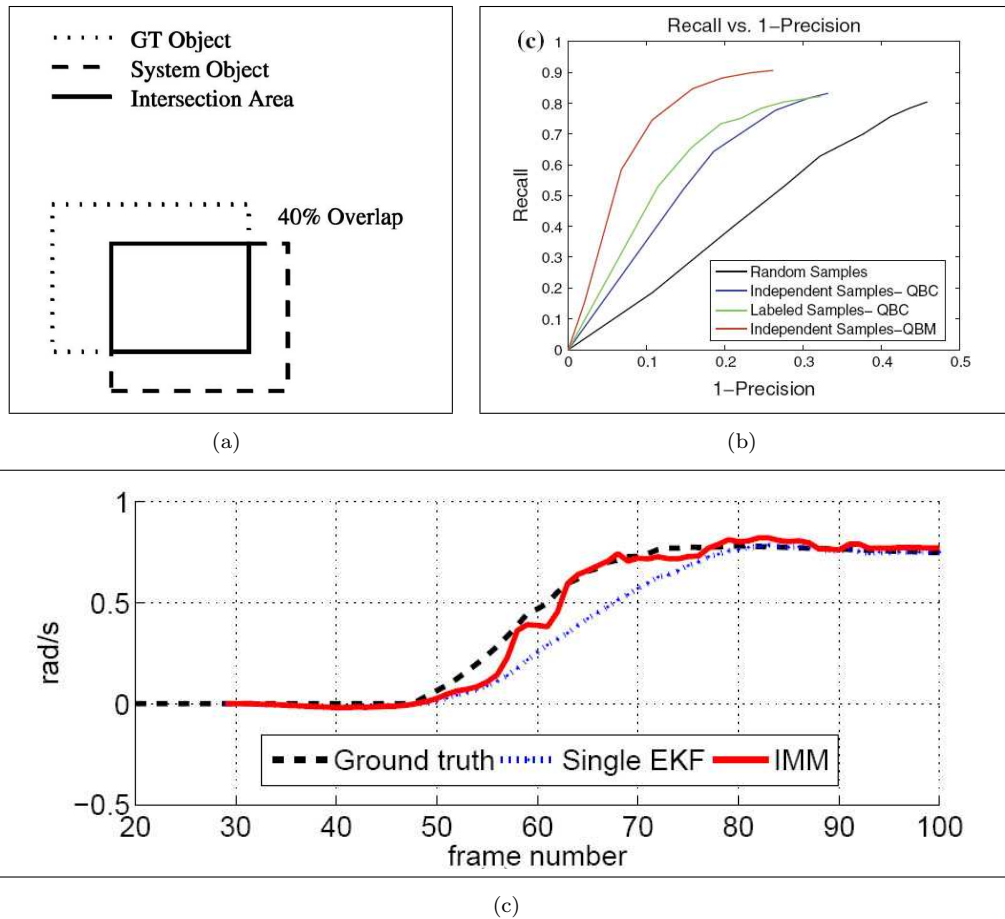


Figure 2.6: Performance metrics. a) Overlap criterion used for labeling detections as true positives or false positives in the image plane [10]. b) Plotting the recall vs. $1 - \text{precision}$ for monocular vehicle detection [11]. c) Plotting the estimated yaw rate vs. time, along with ground truth [1].

Table 2.4: Stereo-vision Tracking Metrics

Performance Metric	Definition
Mean Absolute Error	$\frac{1}{N} \sum x_{\text{estimated}} - x_{\text{ground truth}} $
Standard Deviation of Error	$\sqrt{\frac{1}{N} \sum (x_{\text{estimated}} - x_{\text{ground truth}})^2}$

for detection. While [10] defined useful metrics to quantify the consistency of track identification numbers, their use is virtually unseed in on-road vehicle tracking works.

Stereo-vision vehicle detection studies typically do not report true positive and false positive rates, although this has recently begun to change [124]. Instead, stereo-vision studies tend to focus on estimation accuracy for the motion parameters of a tracked vehicle, including position, velocity, and yaw rate [8, 1]. The performance is typically quantified with the mean absolute error, and the standard deviation in estimation. Table 2.4 defines these terms, using x for the various parameters estimated using a given tracker, and N is the number of frames.

Vehicle behavior studies, lacking a uniformly established system definition, lack a standard set of performance metrics. Context-based detection [36, 132] studies will often use similar metrics to those used in monocular vision. Studies concerned with trajectories [140, 2], will often report classification accuracy, by casting the problem as a multi-class classification task. As this area of study matures, a standard set of performance metrics will emerge. Performance evaluation will need to include a combination of metrics for motion prediction e.g. mean absolute error, and metrics for classification accuracy for multi-class inference e.g. confusion matrices.

Publicly-available vehicle detection benchmarks are becoming more common, but they are still quite rare for vehicle tracking. Part of the difficulty has lied with generating ground truth for the 3D positions and velocities of vehicles in video sequences. The newly-released KITTI database [206] contains extensive video data captured with a calibrated stereo rig, as well as synchronized lidar data, which can be used as a ground truth for vehicle localization. The recently released dataset in [84] also contains lidar data, so that vision-based detection accuracy can be evaluated as a function of longitudinal distance. However, ground truth for dynamical parameters such as velocity and vehicle yaw rate must necessarily come from the tracked vehicle’s own controller area network [CAN], which is not feasible outside of controlled and orchestrated trials.

Benchmark datasets for on-road behavior analysis do not currently exist. This lack of benchmarks largely has to do with the infancy of this research area. Semantically-meaningful labels for ground truth are still not standard. The various studies in the field pursue different objectives: identifying critical and abnormal situations [92], detecting specific maneuvers [174, 1], and predicting long-term motion [140]. As such, there is currently not a unified objective, or set of performance metrics for this area of research. Further, capturing and labeling a relevant dataset is a challenging task, as such a benchmark requires all steps, from vehicle detection to tracking to behavior analysis, all labeled and ground-truth, with globally-applicable and accepted performance metrics and interpretations. As this area of research matures, meaningful and widely-accepted research objectives, goals, and performance metrics will emerge, and standard benchmarks will become more common. More comprehensive benchmark datasets will need to be published, in order to streamline the efforts of the research community.

2.7 Concluding Remarks

In this study, we have provided a review of the literature addressing on-road vehicle detection, vehicle tracking, and behavior analysis using vision. We have placed vision-based vehicle detection in the context of sensor-based on-road perception, and provided comparisons with complimentary technologies, namely radar and lidar. We have provided a survey of the past decade's progress in vision-based vehicle detection, for monocular and stereo-vision sensor configurations. Included in our treatment of vehicle detection is treatment of camera placement, nighttime algorithms, sensor-fusion strategies, and real-time architecture. We have reviewed vehicle tracking in the context of vision-based sensing, addressing monocular applications in the image plane, and stereo-vision applications in the 3D domain, including various filtering techniques and motion models. We have reviewed the state-of-the art in on-road behavior analysis, addressing specific maneuver detection, context analysis, and long-term motion classification and prediction. Finally, we have provided critiques, discussion, and outlooks on the direction of the field. While vision-based vehicle detection has matured significantly over the past decade, a deeper and more holistic understanding of the on-road environment will remain an active area of research in coming years.

2.8 Acknowledgment

Chapter 2 of this dissertation is a partial reprint of material published in IEEE Transactions on Intelligent Transportation Systems, 2013. The dissertation author was the primary investigator and author of these papers.

Chapter 3

A General Active Learning Framework for on-road Vehicle Detection and Tracking Systems

3.1 Introduction

Although active learning for object recognition has been an area of great recent interest in the machine learning community [207], no prior research study has used active learning to build an on-road vehicle recognition and tracking system. In this paper, a general framework for robust active learning based vehicle recognition and tracking, is introduced. The vehicle recognition system has learned in two iterations, using the active learning technique of selective sampling [208] to query informative examples for retraining. Using active learning yields a significant drop in false positives per frame and false detection rates, while maintaining a high vehicle recognition rate. The robust on-road vehicle recognition system is then integrated with a Condensation [209] particle filter, extended to multiple vehicle tracking [210], to build a complete vehicle recognition and tracking system. A general overview of the complete framework can be seen in Figure 3.1.

The main novelty and contributions of this paper include the following. A general active learning framework for on-road vehicle recognition and tracking is introduced. Using the introduced active learning framework, a full vehicle recognition and tracking system has implemented, and a thorough quantitative performance analysis has been presented. The vehicle recognition and tracking system has been evaluated on both real world video, and public domain vehicle images. In this study, we introduce new performance metrics for assessing on road vehicle recognition and tracking performance, which provide a thorough assessment of the implemented

system's recall, precision, localization, and robustness.

The rest of the paper is organized as follows. In the following section we present a brief overview of recent related works in vehicle recognition and active learning. In Section III we review active learning concepts. In Section IV we detail the active learning framework for vehicle recognition. In Sections V and VI we detail the classification and tracking algorithms. In Section VII we provide experimental studies and performance analysis. Finally in Section VIII we provide concluding remarks.

3.2 Related Research

In this section, we present a brief overview of two categories of papers relevant to the research presented in this study. The first set of papers deals with vehicle detection and tracking. The second set of papers deals with active learning for object recognition.

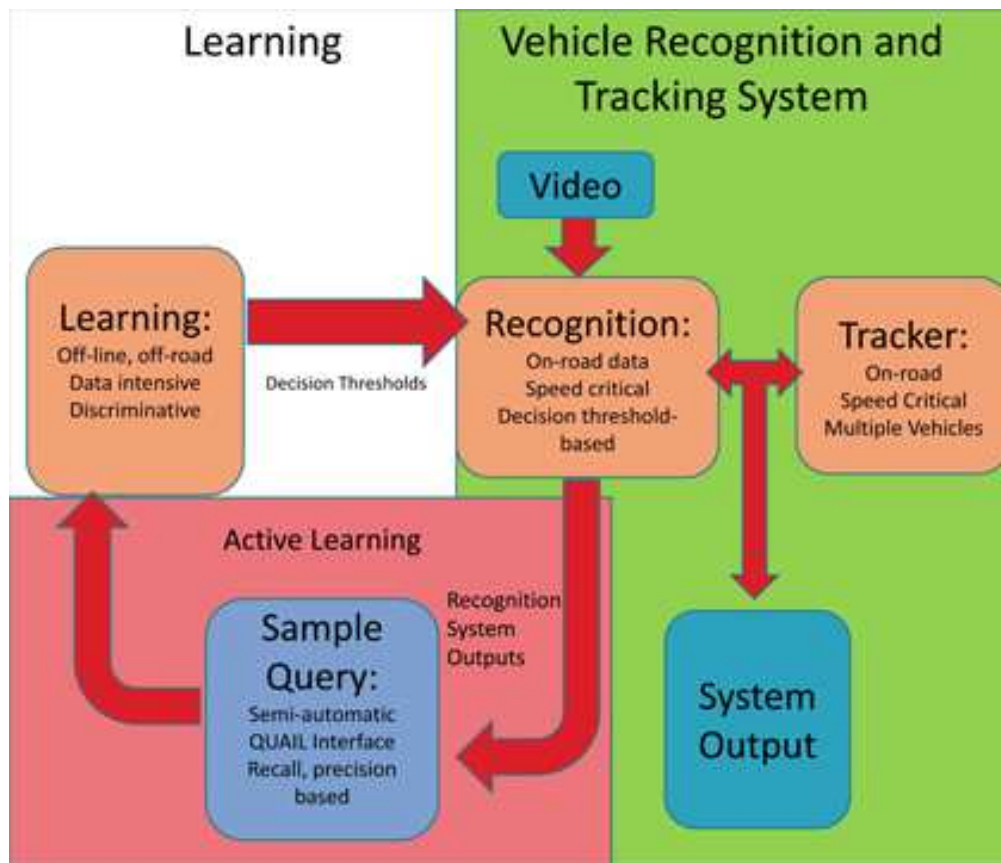


Figure 3.1: Active Learning based Vehicle Recognition and Tracking. The general active learning framework for vehicle recognition and tracking systems consists of an off-line learning portion [white and red], and an online implementation portion [green]. Prior works in vehicle recognition and tracking have not utilized active learning [red].

3.2.1 Vehicle Detection and Tracking

Vision-based vehicle detection is an active area of research in the intelligent transportation systems community [211]. In the literature, many studies have performed experimental validation on static images [74], [65], [211].

A statistical approach has been used in [74], performing vehicle detection using PCA and ICA to do classification on a statistical model, and increased its speed by modeling the PCA and ICA vectors with a weighted Gaussian Mixture Model [74]. This methodology showed very strong performance on static images of parked vehicles, but had slow execution times, and the study limited its scope to sedans.

A neural networks approach has been used in [212]. In [212] a multilayer feedforward neural networks based approach was presented for vehicle detection, the linear output layer replaced by a Mahalanobis kernel. This study showed strong, fast results on pre-cropped, illumination normalized 32×32 image regions.

An SVM approach was used in [211]. Sun et al. [211] built multiple detectors using Gabor filters, Haar wavelets, PCA, truncated wavelets, and a combination of Gabor and wavelet features, using neural networks and SVM classifiers. A thorough comparison of feature and classifier performance was presented, with the conclusion that feature fusion of Haar and Gabor features can result in robust detections. Results were presented for pre-cropped 32×32 pixel, illumination-normalized image regions.

A similar feature analysis was performed in [69]. Negri et al. [69] compared the performance vehicle detectors with Adaboost classification trained using Haar-like features, Histogram of Oriented Gradient [HoG] features, and a fusion of the two feature sets. In this study it was also found that a feature fusion can be valuable. Results were presented for static images.

Haselhoff et al. [65] tested the performance of classifiers trained with images of various resolutions, from the smallest resolution of 18×18 pixels to the largest of 50×50 pixels, using Haar-like feature extraction and Adaboost, discussing the trade-off between classifier performance and training time, as a function of training image resolution. Results were presented on static images. Such a signal-theoretic analysis of Haar features is invaluable to the study of on-road vehicle detection.

Khammari et al. [79] implemented a detector that first applied a 3 level Gaussian pyramid, and used Control Point features, classified by Adaboost, tracked using an Adaboost-based cascaded tracking algorithm. Quantitative analysis was reported for the vehicle detector pre-cropped image subregions. The full detection and tracking system system was then implemented on-road.

A growing number of on-road vehicle studies are reporting results for video datasets [47], [48]. Arrospide et al. [47] performed detection and tracking that evaluated the symmetry-based quality of the tracking results. While tracking based on symmetry metric sounds to be a

Table 3.1: Selected Active Learning Based Object Recognition Approaches.

Research Study	Feature Extraction	Learning, Classification	Selective Sampling Query	Target Object
Abramson and Freund [26], 2005.	Control Points	Adaboost	SEVILLE Visual Interface	Pedestrians
Kapoor et al. [207] 2007	SIFT+PCA	SVM	Probabilistic Selective Sampling	Various Objects
Enzweiler and Gavrilu [24], 2008.	Haar Wavelets	SVM	Probabalistic Selective Sampling	Pedestrians
Roth and Bischof [213], 2008.	Haar Wavelets	Online Boosting	Manual Initialization+ Tracking	Faces
Vijayanarasimhan and K Grauman [214], 2008.	Local features	SVM	Semi-automatic Annotation-based Selective Sampling	Various Objects
This study, 2009.	Haar Wavelets	Adaboost	QUery and Archiving Interface for active Learning [QUAIL] Visual Interface	Vehicles

promising idea, no quantitative performance analysis was provided in this study.

Chan et al. [48] built a detector that used vertical and horizontal symmetry, as well as tail light cues in a statistical model for detection, tracking detected vehicles on the road using particle filtering. This study illustrated the great potential for the use of particle filters in robust on-road vehicle recognition and tracking. Results were presented for on-road video sequences, but false alarm rates were not reported.

3.2.2 Active Learning for Object Recognition

Active learning for object recognition has gained in popularity [26],[24],[3],[213], [207], [214]. Active learning has been used to improve classifier performance by reducing false alarms [3], [26], [208], to increase a classifier’s recall [213], to semi-automatically generate more training data [24], [26], [3], and to perform training with fewer examples [23], [213], [208].

Recently, Enzweiler and Gavrilu [24] used active learning to train a pedestrian detector. A generative model of pedestrians was built, from which training examples were probabilistically selectively sampled for training. In addition, the generative pedestrian models were used to synthesize artificial training data to enhance the pedestrian detector’s training set [24].

Roth and Bischof [213] used a manually initialized tracker to generate positive training

samples for online active learning to develop a face detector that outperformed the face detector trained using passive learning.

Abramson and Freund [26] used selective sampling to drastically reduce the false positives output by a pedestrian detector using iterative training with Adaboost and Control Point features. Table I contains a summary of recent active learning based object recognition studies.

3.3 Active Learning: Motivation

3.3.1 Overview

Passive learning consists of the conventional supervised learning of a binary classifier learned from labeled ‘positive’ examples, and random ‘negative’ examples. To build a vehicle recognition system in this manner, the positive training examples consist of vehicles, and the negative training examples consist of random non-vehicles.

Active learning is a general term that refers to a paradigm in which the learning process exhibits some degree of control over the inputs on which it trains. Active learning has been shown to be more powerful than learning from random examples in [208], [213], [23]. In vision-based object recognition tasks, a common approach is selective sampling [207], [24], [214]. Selective sampling aims to choose the most informative examples for training a discriminative model. Cohn et. al [208] demonstrated that selective sampling is an effective active learning technique by sequentially training robust neural network classifiers. Li and Sethi [23] used a confidence-based sampling for training robust SVM classifiers.

Training aims to build a system that correctly recognizes vehicles in video frames by learning the decision threshold between vehicles and non-vehicles. Consider the concept of a vehicle, $v(s) = 1$, as an image subregion, s classified as a vehicle, and $v(s) = 0$ an image subregion classified as non-vehicle. A concept is *consistent* with a training example s if $v(s) = t(s)$, the true class of s . Consider the set S^m consisting of m training examples used in the initial passive training. Assume all the examples from S^m are consistent with concept v , i.e. the training error is zero. In classification tasks, there exists a region of uncertainty, $R(S^m)$, where the classification result is not unambiguously defined. This is to say that discriminative model can learn a multitude of decision thresholds for the given training patterns, but disagree in certain regions of the decision space [24]. Areas that are *not* determined by the model trained using S^m are of interest for selective sampling, as they constitute more informative examples [208]. Given a random test sample x with true class $t(x)$ and training data S^m , we define the region of uncertainty, $R(S^m)$ as follows:

$$\begin{aligned}
R(S^m) &= \{x : \exists v, \\
&v \text{ is consistent with all } s \in S^m, \\
&\text{and } v(x) \neq t(x)\}
\end{aligned}
\tag{3.1}$$

It is of note that both the trained decision boundary and the region of uncertainty are a function of the training data S^m . In general, the decision boundary or threshold between positive examples and negative examples resides in $R(S^m)$, the region of uncertainty. If we use a point that lies outside of $R(S^m)$ to update the classifier, the classifier will remain unchanged. If we use a point inside the region of uncertainty, the region of uncertainty will reduce [208]. As the region of uncertainty reduces, the classifier becomes less likely to report false positives, and gains precision.

3.3.2 Implementation

Representing $R(S^m)$ exactly is generally a difficult, if not impossible task. A good approximation of the region of uncertainty is a superset, $R^+(S^m) \supseteq R(S^m)$, as we can selectively sample from $R^+(S^m)$ and be assured that we do not exclude any part of the domain of interest [208]. This is the approach taken in many object recognition studies [207], [3], [24], [214].

In general, active learning consists of two main stages: an initialization stage, and a stage of query and retraining [23]. In the initialization stage, a set of training examples is collected and annotated to train an initial classifier. This is the same process as passive learning [213], [208]. Once an initial classifier has been built, a query function is used to query unlabeled examples, and a human or ground truth mechanism is used to assign a class label to the queried examples. The newly labeled training examples are then used to retrain the classifier [23].

The query function is central to the active learning process [23], and generally serves to select difficult training examples, which are informative in updating a decision boundary [208]. In [24], this was achieved by building a generative model of the classes, and samples with probabilities close to the decision boundary were queried for retraining. In [23], confidence outputs from SVM classifiers were used to map to error probabilities, and those examples with high error probabilities were queried for retraining.

3.4 Training Framework

3.4.1 Initialization

The initial passively trained Adaboost [76] cascaded classifier was trained using 7,500 positive training images and 20,500 negative training images. The passively trained cascade

consisted of 30 stages. The positive training images were collected from many hours of real driving data on San Diego highways in the LISA-Q testbed [215]; the testbed is described in Section VII-D.

3.4.2 Query and Retraining

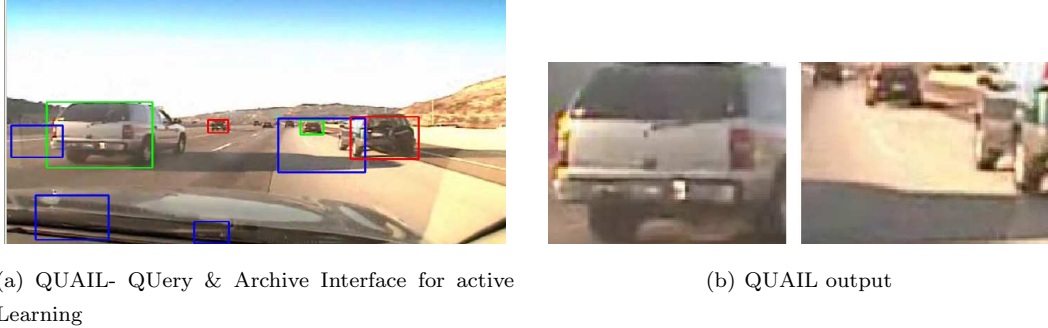


Figure 3.2: a) QUAIL- QUery & Archive Interface for active Learning. The QUAIL interface evaluates the passively trained vehicle recognition system on real-world data, and provides an interface for a human to label and archive ground truth. Detections are automatically marked green. Missed detections are marked red by the user. False positives are marked blue by the user. True detections are left green. b) QUAIL outputs. A true detection and a false positive, archived for retraining.

Efficient visual query and archival of informative examples has been performed via the QUery & Archiving Interface for active Learning [QUAIL]. The interface is able to evaluate the vehicle recognition system on a given random video frame, marking all system outputs, and allowing the user to quickly tag false positives and missed vehicles. The interface then archives missed vehicles and true positives as positive training examples, and false positives as negative training examples, performing selective sampling in an intuitively visual, user-friendly, efficient manner.

Strictly speaking, only missed detections and false positives are known to lie within the region of uncertainty, which we call $R(S^m)$. As it is not possible to represent the region of uncertainty exactly, including correctly identified vehicles in the archived training data ensures that the data comprises a superset of the region of uncertainty. Maintaining a superset $R^+(S^m)$ of the region of uncertainty helps protect against oversampling one part of the domain [208]. We are assured that we are not excluding examples from the domain of interest, but acknowledge that we retrain on some positive examples that are not of interest.

Figure 3.2(a) shows a screen shot of the QUAIL interface, and Figure 3.2(b) shows corresponding cropped true detection and false positive image regions. Figure 3.3 shows false positives that were queried and archived for retraining using QUAIL. Figure 3.4 shows true detections that were queried and archived for retraining using QUAIL.



Figure 3.3: Examples of false positive outputs queried for retraining using QUAIL



Figure 3.4: Examples of true positives queried for retraining using QUAIL.

For the retraining we had 10,000 positive images, and 12,172 negative images. The negative training images consisted exclusively of false positives from the passively trained detector output. Adaboost training was used to build a cascade of 20 stages. Figure 3.5 shows a schematic of the general framework for training an active learning based vehicle detector.

3.5 Vehicle Recognition Using Rectangular Features and Adaboost Classifier

For the task of identifying vehicles, a boosted cascade of simple Haar-like rectangular features has been used, as was introduced by Viola and Jones [54] in the context of face detection. Various studies have incorporated this approach in on-road vehicle detection systems such as [65],[60]. The set of Haar-like rectangular features is well-suited to the detection of the shape of vehicles. Rectangular features are sensitive to edges, bars, vertical and horizontal details, and symmetric structures [54]. Figure 3.6(a) shows examples of Haar-like rectangular features. The algorithm also allows for rapid object detection that can be exploited in building a real-time system, partially due to fast and efficient feature extraction using the integral image [54]. The resulting extracted values are effective weak learners [54], which are then classified by Adaboost.

Adaboost is a discriminative learning algorithm, which performs classification based on a weighted majority vote of weak learners [76]. We use Adaboost learning to construct a cascade

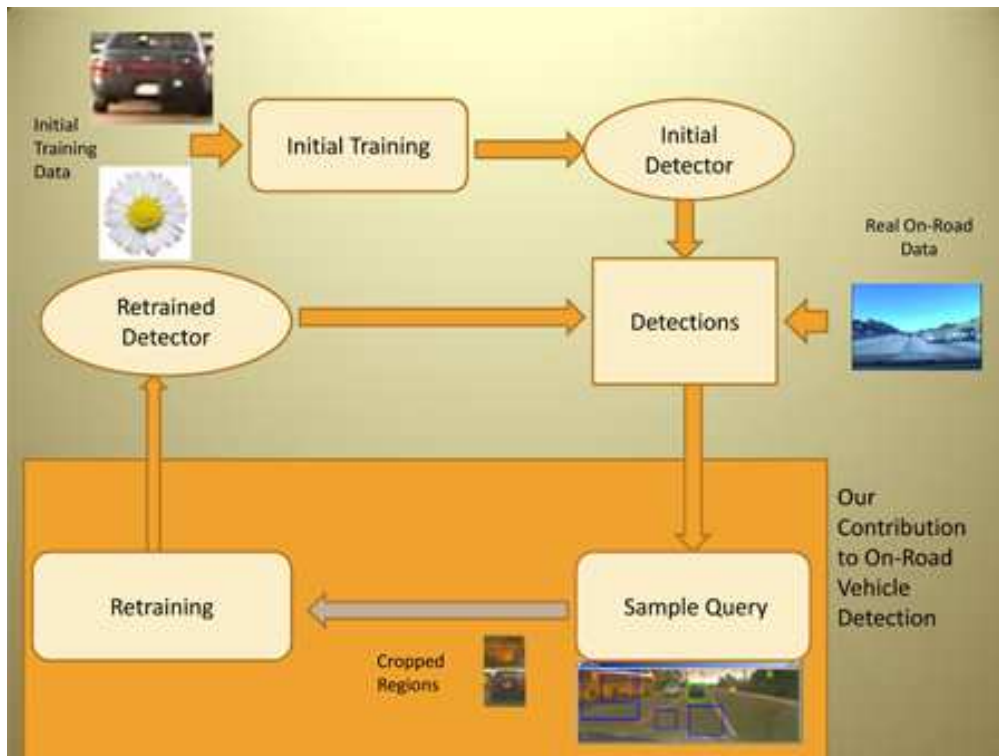


Figure 3.5: Schematic of framework for Active Learning Based Vehicle Recognition Training. An initial, passively trained vehicle detector is built. Using the QUAIL interface, false positives, false negatives, and true positives are queried and archived. A new classifier is trained using the archived samples.

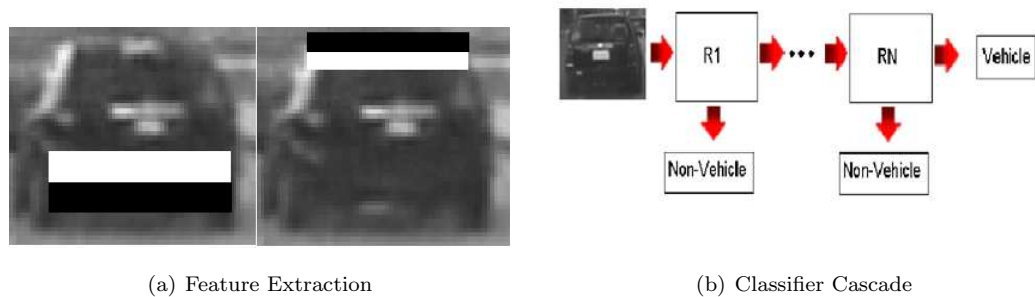


Figure 3.6: a) Examples of Haar-like features used in the vehicle detector. b) Cascade of boosted classifiers



Figure 3.7: Left: Detector outputs for a single vehicle [top], and multiple vehicles [middle and bottom]. Middle: Multiple location hypotheses generated by the Condensation filter. Note the multi modal distribution of the hypotheses when tracking multiple vehicles [bottom]. Right: Best tracking results, as confirmed by detections in the consequent frame.

of several binary classifier stages. The earlier stages in the cascade eliminate many non-vehicle regions with very little processing [54]. The decision rule at each stage is made based on a threshold of scores computed from feature extraction. Each stage of the cascade reduces the number of vehicle candidates, and if a candidate image region survives until it is output from the final stage, it is classified as a positive detection. Figure 3.6(b) shows a schematic of the cascade classifier.

3.6 Vehicle Tracking with Condensation Filter

We integrate a particle filter for vehicle tracking. The probability densities of possible predictions of the state of the system are represented by a randomly generated set, and multiple hypotheses are used to estimate the density of the tracked object. The original Condensation algorithm was designed to track one object. First, a random sample set of hypotheses is generated, based on the previous sample state and a representation of the previous stage's posterior probability, $p(x_t|Z_{t-1})$, where x is the state and Z is the set of all observed measurements. Then, the random samples are used to predict the state using a dynamic system model. Finally, a new measurement is taken and each of the multiple position hypotheses is weighted, yielding a representation of the observation density $p(z_t|x_t)$ [209].

The basic Condensation algorithm was not designed to track an arbitrarily changing number of objects. Koller-Meier and Ade [210] proposed extensions to the algorithm to allow for tracking of multiple objects, and to track objects entering and leaving the camera’s field of view using one Condensation tracker. Maintaining the tracks of multiple objects is achieved by including a representation of the probability distribution of all tracked objects in the Condensation tracker itself, as in equation (2).

$$p(x_t) = \sum_i \alpha^{(i)} p^{(i)}(x_t) \quad (3.2)$$

During tracking, all tracked objects are given equal weightings $\alpha^{(i)}$ to ensure that the sample set does not degenerate [210]. To track newly appearing objects into the Condensation tracker, the observed measurements are integrated directly into the sampling. An initialization density $p(x_{t-1}|z_{t-1})$, a representation of the probability of the state at time $t - 1$ given just one measurement, is calculated, and combined with the posterior density from the previous step, as shown in equations (3) and (4).

$$p'(x_{t-1}|Z_{t-1}) = (1 - \gamma)p(x_{t-1}|Z_{t-1}) + \gamma p(x_{t-1}|z_{t-1}) \quad (3.3)$$

$$\gamma = \frac{M}{N} \quad (3.4)$$

In this implementation of the Condensation algorithm, $N - M$ samples are drawn from the representation of the previous stage’s posterior density, and M new samples are drawn from the initialization density [210]. This method ensures that there is a re-initialization of the probability density every time there is a new measurement observed, allowing for very fast convergence of the tracks [48], [209]. Figure 3.7 shows example detections, multiple particle tracking hypotheses, and the best tracking hypotheses, confirmed by the consequent vehicle detection observations. Figure 3.8 shows a flowchart depicting the ALVeRT system overview.

3.7 Experimental Evaluation

3.7.1 Experimental Datasets

Two main datasets were used to quantify the performance of vehicle recognition. The first dataset, the publicly available Caltech 1999 dataset, consists of 126 distinct static images of rear-facing vehicles.

To test the full ALVeRT system, video datasets are necessary. The second dataset, the LISA-Q Front FOV datasets, consist of three video sequences, consisting of 1600, 300, and 300 consecutive frames, respectively. Table 3.2 briefly describes the datasets used in this study.

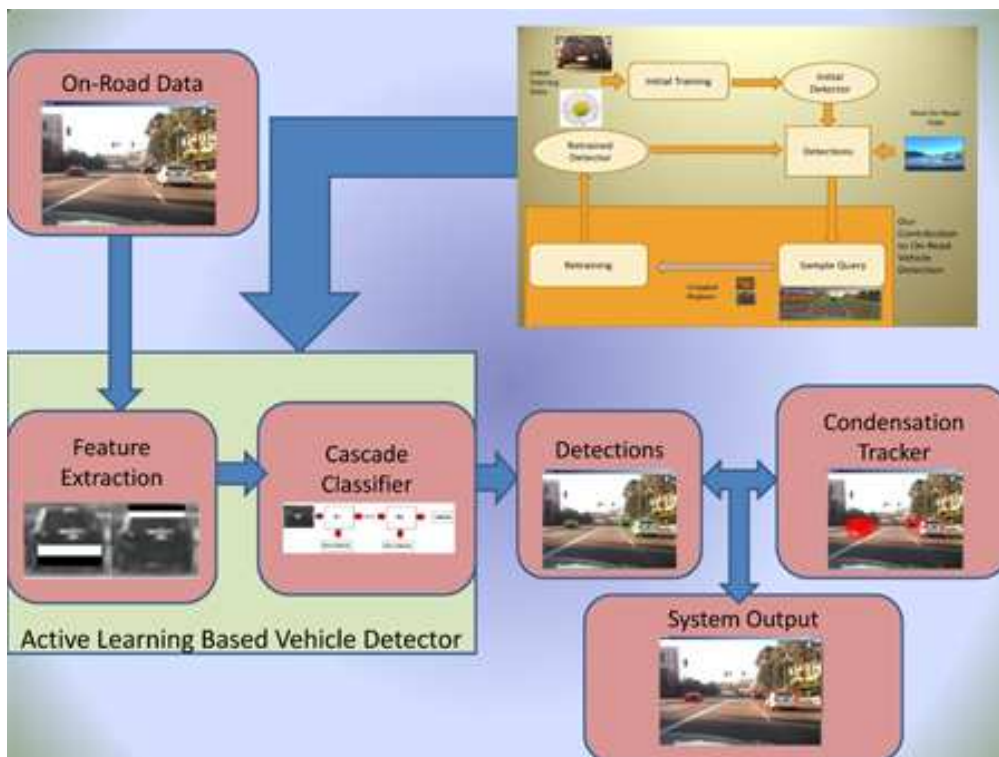


Figure 3.8: Full ALVERT system overview. Real on-road data is passed to the active learning based vehicle recognition system, which consists of Haar-like rectangular feature extraction and the boosted cascade classifier. Detections are then passed to the Condensation multiple vehicle tracker. The tracker makes predictions, which are then updated by the detection observations.

Table 3.2: Datasets Used in this Study

Dataset	Description
Caltech Vehicle Image 1999	This dataset consists of 126 distinct static images of vehicles. The dataset is publicly available, and has been used in research studies such as [74] and [211]. The image dataset can be accessed at Caltech's Computational Vision website, http://www.vision.caltech.edu/archive.html
LISA-Q Front FOV Video Datasets 1-3	The LISA-Q Front FOV Datasets are videos taken from the LISA-Q testbed [215]. Dataset 1 consists of 1600 consecutive frames, captured during sunny evening rush hour traffic. Dataset 2 consists of 300 consecutive frames, captured on a cloudy morning on urban roads. Dataset 3 consists of 300 consecutive frames, captured on a sunny afternoon on highways. The datasets will be available for academic and research communities, pending approvals, at the Laboratory for Intelligent and Safe Automobiles website, http://cvrr.ucsd.edu/LISA/index.html

3.7.2 Performance Metrics

The performance considerations we present consist of the following: precision, recall, localization, robustness, efficiency, scalability.

Many vehicle detection studies test their classifiers on static images, by first cropping down test samples into candidate subregions, normalizing the illumination and contrast, and then quantifying the performance of the classifier as a binary classifier [20], [212]. This evaluation procedure can report lower false positive rates and higher detection rates than the system would output on full, consecutive video frames, because the system is presented with fewer vehicle-like regions than if required to search through a video frame. This evaluation does not give information about localization or robustness, because test image intensities and contrasts have been normalized [20], [212], does not indicate scalability.

Other studies do not crop down test images for validation, but quantify their true negatives as being all image sub-regions not identified as false positives [74], [65]. For an $n \times n$ image, there are potentially n^4 rectangular subregions [216]. Using this definition of a false positive, studies on vehicle detection present false positive rates on the order of 10^{-5} [74], [65]. Such false positive rates have little practical meaning. This evaluation method gives a practical assessment of recall, precision, and efficiency, but not of robustness, localization, or scalability.

Recent vehicle tracking papers either do not offer numerical evaluation of their systems [47], or provide numerical values for successfully tracked vehicles but do not provide counts of erroneously tracked objects [48]. Such evaluations do indicate recall and efficiency, but do not provide information on precision, scalability, localization, and robustness.

In our research, the performance of a detection module is quantified by the following metrics: True Positive Rate, False Detection Rate, Average False Positives per Frame, Average True Positives per Frame, False Positives per Vehicle.

The true positive rate, TPR , is the percentage of non-occluded vehicles in the camera’s view that are detected. TPR is assessed by dividing the number of truly detected vehicles by the total number of vehicles. This takes into account the vehicles preceding the ego-vehicle, and those in adjacent lanes. This quantity measures recall and localization. TPR is defined in equation (5).

$$TPR = \frac{\text{Detected Vehicles}}{\text{Total Number of Vehicles}} \quad (3.5)$$

The false detection rate, FDR , is the proportion of detections that were not true vehicles. We assess the FDR by dividing the number of false positives by the total number of detections. This is the percentage of erroneous detections. FDR is a measure of precision and localization; it is defined in equation (6).

$$FDR = \frac{\text{False Positives}}{\text{Detected Vehicles} + \text{False Positives}} \quad (3.6)$$

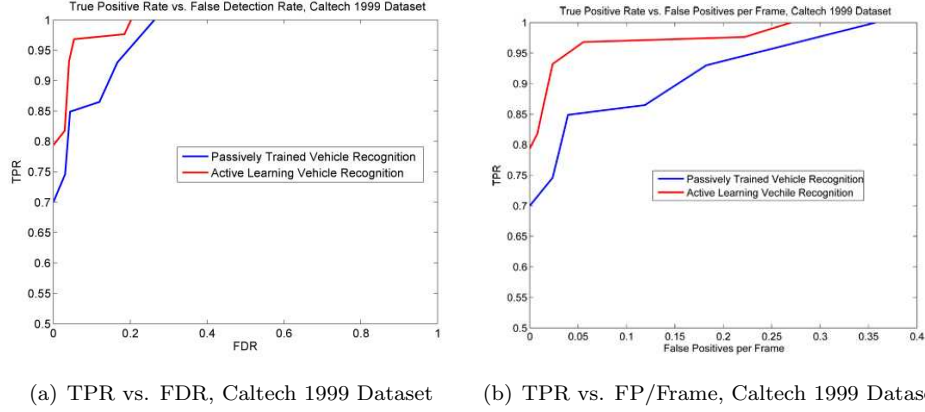


Figure 3.9: a) Plot of TPR vs. FDR for passively trained recognition [blue], the active learning vehicle recognition [red]. We note the improvement in performance due to active learning on this dataset. b) Plot of TPR vs. Number of False Detections per Frame for passively trained recognition [blue], the active learning vehicle recognition [red]. We note the reduction in false positives due to active learning.

The average false positives per frame, average $FP/Frame$, quantity describes how susceptible a detection module is to false positives, and gives an informative measure of the credibility of the system. This quantity measures robustness, localization, and scalability. $FP/Frame$ is defined in equation (7).

$$Avg. FP/Frame = \frac{False\ Positives}{Total\ Number\ of\ Frames\ Processed} \quad (3.7)$$

The average false positives per object, $FP/Object$, describes, on average, how many false positives are observed. This quantity measures robustness. This performance metric was first used in [217]. It is defined in equation (8).

$$Avg. FP/Object = \frac{False\ Positives}{TrueVehicles} \quad (3.8)$$

The average true positives per frame, $TP/Frame$, describes how many true vehicles are recognized on average. This quantity indicates robustness. It is defined in equation (9).

$$Avg. TP/Frame = \frac{True\ Positives}{Total\ Number\ of\ Frames\ Processed} \quad (3.9)$$

The overall performance of the system, including efficiency, indicates how scalable a system is. If the system performs with high precision, high recall, good localization, and in a robust and efficient manner, it is a viable system to be used as part of a larger, more sophisticated framework.

The performance metrics give an informative assessment of the recall, precision, localization, and robustness. We have formally defined the metrics in equations (5)-(9) above.



Figure 3.10: Sample vehicle recognition results from the Caltech 1999 dataset.

3.7.3 Static Images: Caltech 1999 Dataset

We first evaluate the passively trained vehicle recognition and active learning based vehicle recognition systems on the publicly available Caltech 1999 dataset. The dataset consists of 126 static images of vehicles. We note that using this dataset, we are not able to evaluate the full ALVeRT system. Figures 3.9(a) and 3.9(b) plot the TPR versus FDR and $FP/Frame$. We note that the active learning based vehicle recognition performance was stronger than the passively trained recognition system. Figure 3.10 shows sample recognition results on this dataset. It is of note that on this dataset, our vehicle detection achieves lower false positive per frame rate when compared with [74].

3.7.4 Video Sequences: LISA-Q Front FOV Datasets

In the Laboratory for Intelligent and Safe Automobiles, on-road data is captured daily and archived as part of ongoing studies in naturalistic driving and intelligent driver assistance systems. The data used in this paper were captured in the LISA-Q testbed, which has synchronized capture of vehicle CAN data, GPS, and video from six cameras [218]. Video from the front-facing camera comprise the LISA-Q Front FOV datasets.

LISA-Q Front FOV 1 was captured around 4pm on January 28, 2009, during San Diego’s rush hour, so there are many vehicles on the road performing complex highway maneuvers. It was a sunny evening, but at this time of year, the sun was within an hour or so of setting. The time of day and geographical attributes like hills and canyons result in complex illumination and shadowing. Poor illumination can limit the range of detections, and complex shadows can result in false positives. The ALVeRT system performed robustly in this difficult environment.

LISA-Q Front FOV 2 was taken around 9am on March 9, 2009, on an urban road in La Jolla, California. The combined early morning and clouds result in poor illumination conditions.



Figure 3.11: a) Recognition output in shadows. Five vehicles were detected, and two were missed due to their distance and poor illumination. We note that poor illumination seems to limit the range of the detection system; vehicles farther from the ego vehicle are missed. b) Vehicle recognition output in sunny highway conditions.



Figure 3.12: Left: The vehicle in front was not detected in this frame on a cloudy day, due to smudges and dirt on the windshield. Right: The vehicle's track was still maintained in the following frame.

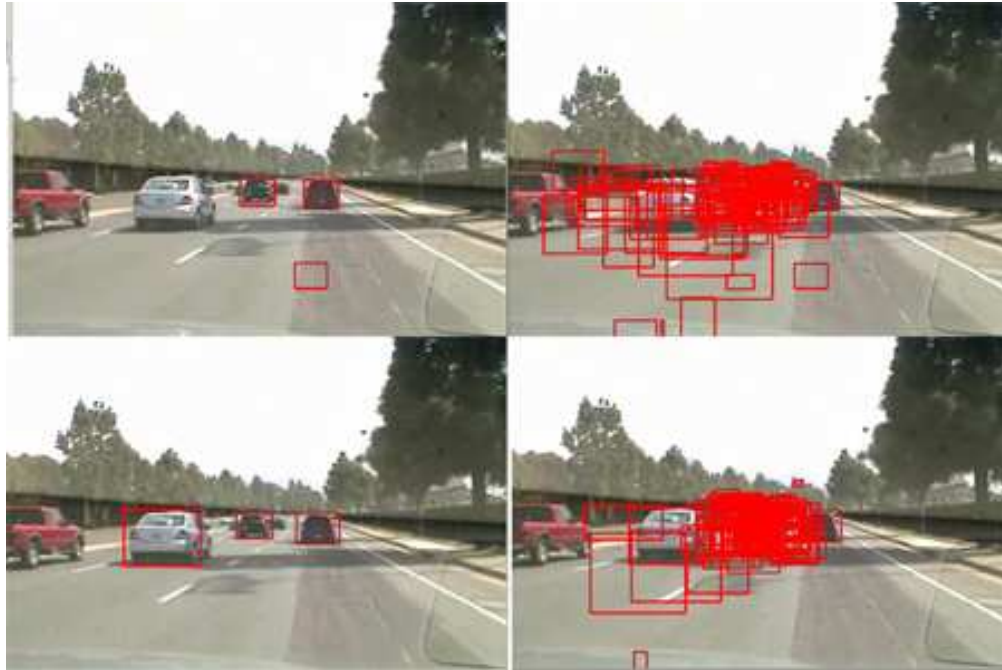


Figure 3.13: Top Left: The vehicle recognition system has output two true vehicles, one missed vehicle, and one false positive. Top Right: These detection outputs are passed to the tracker. We note that the missed vehicle still has a track maintained, and is being tracked despite the missed detection. Bottom Left: In the following frame, all three vehicles are detected. Bottom Right: In the corresponding tracker output, we note that a track was not maintained for the false positive from the previous frame.

In addition, the vehicle's windshield is smudged/dirty. The active learning based recognizer did not perform as well on this data set as Datasets 1 and 3. The full ALVeRT system performed significantly better than the vehicle recognition system alone, because the Condensation particle tracker maintains tracks from frame to frame. Inconsistent vehicle detections from the active learning based recognizer can still result in smooth vehicle tracks in the Condensation tracking framework, without introducing false positives that result in erroneous tracks. Figure 3.15(b) shows vehicle recognition outputs from Dataset 2, and Figure 3.12 shows tracking output.

LISA-Q Front FOV 3 was captured around 12.30pm on April 21, 2009 on a highway. The weather was sunny and clear. These conditions can be thought of as ideal. We note that the active learning based recognizer had a much better false positive per frame rate than the passively trained recognizer. Figure 3.11(b) shows detection results from this dataset.

We note that the passively trained recognizer is more susceptible to false positives, with *FDR* of over 45% in each of the three datasets. This means that almost one out of every two detections is indeed erroneous. In addition, $FP/Frame$ between 2.7 and 4.2 false positives per frame indicates that the system is generally not credible. However, the passively trained recognizer had quite a high detection rate, which indicates that the Haar-like wavelet feature set

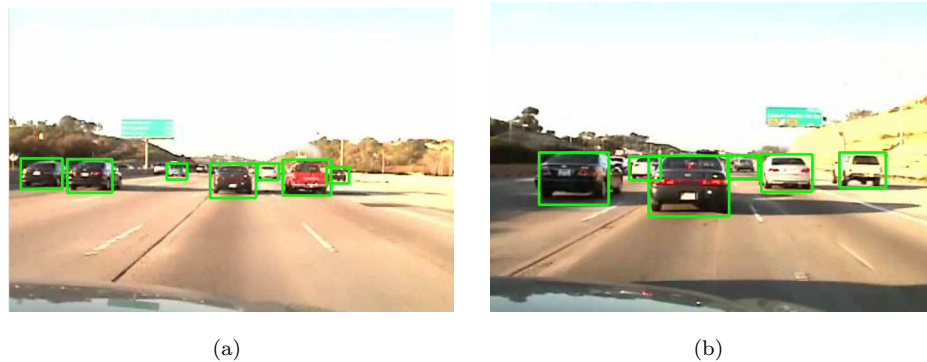


Figure 3.14: a) Recognition output in sunny conditions. Note the vehicle detections across all six lanes of traffic during San Diego’s rush hour. The recognizer seems to work best in even, sunny illuminations, as such illumination conditions have less scene complexity compared to scenes with uneven illuminations. b) Recognition output in dense traffic. We note that the vehicle directly in front is braking, and the system is aware of vehicles in adjacent lanes.

Table 3.3: Experimental Dataset 1 : Jan 28, 2009, 4pm, highway, sunny

System	TPR	FDR	FP/Frame	TP/Frame	FP/Object
Passively Trained Vehicle Recognition	89.5%	51.1%	4.2	4.1	0.94
Active Learning Vehicle Recognition	93.5%	7.1%	0.32	4.2	0.07
ALVeRT	95.0 %	6.4%	0.29	4.2	0.06

Table 3.4: Experimental Dataset 2: March 9, 2009, 9am, urban , cloudy

System	TPR	FDR	FP/Frame	TP/Frame	FP/Object
Passively Trained Vehicle Recognition	83.5%	79.7%	4.0	1.0	3.3
Active Learning Vehicle Recognition	80.2%	41.7%	0.72	0.98	0.57
ALVeRT	91.7 %	25.5%	0.39	1.14	0.31

Table 3.5: Experimental Dataset 3: April 21, 2009, 12pm, highway, sunny

System	TPR	FDR	FP/Frame	TP/Frame	FP/Object
Passively Trained Vehicle Recognition	98.1%	45.8%	2.7	3.16	0.83
Active Learning Vehicle Recognition	98.8%	10.3%	0.37	3.18	0.11
ALVeRT	99.8 %	8.5%	0.28	3.17	0.09



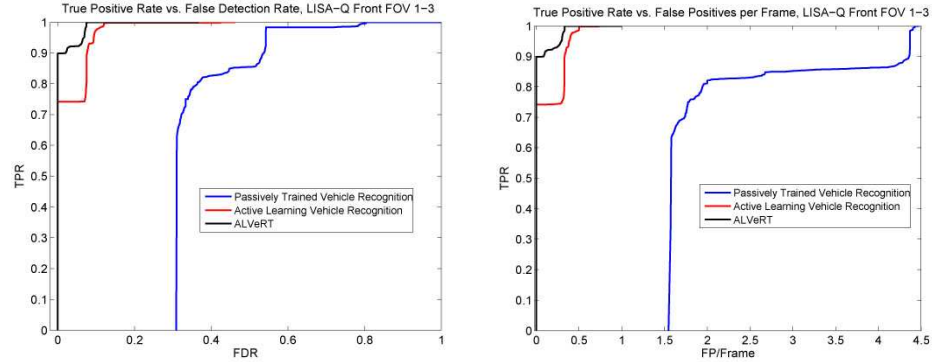
Figure 3.15: a) Recognition output in a non-uniformly illuminated scene. Six vehicles were detected. No false positives, and no missed detections. b) Recognition output in cloudy conditions. Note the reflections, glare, and smudges on the windshield.

has potential use in effective on-road vehicle recognition, and validates the initialization approach.

The active learning based recognizer had a higher detection rate than the passively trained recognizer, and an impressive reduction in the false detection rate. We note that the *FDR* value depends both on the detection rate and the false positives generated. This classifier produced about one false positive every three frames, achieving a significant reduction in *FP/Frame*. The dramatic reductions in false positive rates are a result of selective sampling. We have trained the classifier using difficult negative training examples obtained using the QUAIL system detailed in Section IV.

Integrating multi-vehicle tracking improved the performance of the overall system, with increased detection rates and fewer false positives per frame than the recognizer alone. Tables 3.3, 3.4, and 3.5 detail the performance of each system on each of the test datasets.

To discuss scalability, it is useful to use tracking as an example of scaling up a system. If a detector outputs between 2.7 and 4.2 false positives per frame, those false positives can create erroneous tracks. This means that the passively trained vehicle recognizer is not scalable. However, the active learning based recognizer outputs between 0.32 and 0.72 false positives per frame, not consistent enough for a tracker to create erroneous tracks. Thus, the active learning



(a) TPR vs. FDR, LISA-Q Front FOV Dataset 1-3 (b) TPR vs. FP/Frame, LISA-Q Front FOV Dataset 1-3

Figure 3.16: a) Plot of TPR vs. FDR for passively trained recognition [blue], the active learning vehicle recognition [red], and the ALVeRT system [black]. We note the large improvement in performance due to active learning for vehicle detection. b) Plot of TPR vs. Number of False Detections per Frame for passively trained recognition [blue], the active learning vehicle recognition [red], and the ALVeRT system [black]. We note the reduction in False Positives due to active learning.

based recognition system is scalable, as shown by the performance of the full detection and tracking system. This is demonstrated by Figure 3.13.

Figure 3.16(a) shows plots of the TPR versus the FDR for each of the three systems per frame. Figure 3.16(b) shows plots of TPR versus the $FP/Frame$.

3.8 Remarks

A general active learning framework for robust on-road vehicle recognition and tracking has been introduced. Using active learning, a full vehicle recognition and tracking system has been implemented, and a thorough quantitative analysis has been presented. Selective sampling was performed using the QUery and Archiving Interface for active Learning [QUAIL], a visually intuitive, user-friendly, efficient query system. The system has been evaluated on both real-world video, and public domain vehicle images. We have introduced new performance metrics for assessing on road vehicle recognition and tracking performance, which provide a thorough assessment of the implemented system’s recall, precision, localization, and robustness. Given the success of ALVeRT framework, a number of future studies are planned. These research efforts include pedestrian protections systems [219], trajectory learning [220], and integrated active safety systems [221, 222].

3.9 Acknowledgment

Chapter 3 is a partial reprint of material published in IEEE Transactions on Intelligent Transportation Systems, 2010. The dissertation author was the primary investigator and author of these papers.

Chapter 4

Active Learning for On-Road Vehicle Detection: A Comparative Study

4.1 Introduction

In the past decade, active learning has emerged as a powerful tool in building robust object detection systems in the computer vision community. Incorporating active learning approaches into computer vision systems promises training with fewer samples, more efficient use of data, less manual labeling of training examples, and better system recall and precision. In particular, various active learning approaches have demonstrated encouraging results for the detection of pedestrians [24, 25], vehicles [3], faces [223], and other objects of interest.

Broadly speaking, active learning consists of initializing a classifier using conventional supervised learning, and querying informative samples to retrain the classifier. The query function is central to the active learning process [23]. Generally, the query function serves to select difficult training examples, which are informative in updating a decision boundary [208, 224]. Various approaches to the general query of samples in a binary problem have been explored [225].

While utilizing active learning for object detection promises training with fewer samples [225], less manual labeling of training examples [27], and better system performance [27, 226], there remain several open research issues. While it is generally accepted that active learning can reduce human annotation time in data labeling, few studies have performed the sort of annotation time experiments found in [227]. In addition to documenting the necessary labeling effort, few studies provide a side-by-side analysis of human time costs as a function of the sample query.

While it is also accepted that active learning promises training with fewer samples, *how many* samples is not known. If a particular active learning approach outperforms than its competitors with 1000 training samples, does this necessarily mean that it will outperform them with 200, 5000, or 10000 training examples? In particular, when developing vision systems for vehicles, knowledge of the performance, labor, and data implications is very important.

In this study, we have implemented three widely used active learning frameworks for on-road vehicle detection. Two main sets of experiments have been performed. In the first set, HOG features and Support Vector Machine [21, 22, 228] classification have been used. In the second set of experiments, the vehicle detector was trained using Haar-like features and Adaboost classification. Based on the initial classifiers, we have employed various separate querying methods. Based on confidence-based active learning [23], we have implemented Query by Confidence [QBC] with two variations. In the second, informative independent examples are queried from unlabeled data corpus, and a human oracle labels these examples. This method is compared against simply querying labeled examples from the initial corpus are queried based on a confidence measure, as in [24, 25]. The third learning framework we term Query by Misclassification [QBM], in which the initial detector is simply evaluated on independent data, and a human oracle labels false positives and missed detections. This approach has been used in [26, 27].

The contributions of this paper include the following. A comparative study of active learning for on-road vehicle detection is presented. We have implemented three separate active learning approaches for vehicle detection, comparing the annotation costs, data costs, recall, and precision of the resulting classifiers. The implications of querying examples from labeled vs. unlabeled data is explored. The learning approaches have been applied to the task of on-road vehicle detection, and a quantitative evaluation is provided on a real world validation set.

The remainder of this paper is organized as follows: Section 4.2 provides a survey of related works in active learning, and on-road vehicle detection. Section 4.3 provides background and theoretical justification for the active learning approaches evaluated in this research study. Section 3 provides implementation details for the object detection systems. Section 4.4.4 provides a quantitative experimental analysis of learning framework performance, which includes system performance, data necessity, and human annotation time involved in each. Section 4.5 provides concluding remarks.

4.2 Related Research

We divide our literature review into subsections dealing with active learning, and with vision-based on-road vehicle detection.

4.2.1 Active Learning for Object Detection

There is a rich literature in the machine learning community on active learning [208, 225, 229, 23]. In certain separable learning scenarios, the decision boundary obtained by sequential active learning provably converges much faster than learning by random sampling from the distributions [208]. Active learning makes efficient use of training examples, which is especially useful when training examples are rare, or when extensive human annotation time is necessary to label images [24], which is generally known to be high cost [229]. Moreover, active learning provides a learning framework in which classifiers can be updated after the initial batch training, and adapted to the environment to which they are deployed. This is of particular interest in vision-based object detection, as there may be significant differences between the training set and real-world conditions [230, 27]. Active learning also provides a solid framework for adaptive extensions, such as online learning for object detection [27, 230, 231].

Recent studies in the field address two overlapping questions. The first deals with formulating a query function for identifying informative examples [23, 207, 24]. The second deals with the labeling costs in active and online learning [207, 227, 27].

Sample Query

Many active learning studies invoke the concept of the *Region of Uncertainty* [208, 229, 24], a region in the decision space where classification is not unambiguously defined. The Region of Uncertainty is a function of the training data, whose size can be reduced by querying uncertain samples, identified by their probabilities or confidences [207, 24, 25], or the result of misclassification [208]. Defining a query protocol to identify informative examples is integral to active learning [23, 24].

Confidence-based query uses a confidence metric to query examples that lie near the classification boundary in the decision space [23]. This can be done with a variety of metrics. In [23], such informative scores were identified by using a dynamic histogram of discriminant function values over the entire training set. In [227], entropy was used to query the most uncertain examples. The euclidean distance from the decision boundary was used to query informative samples in [207].

The score-based query may simply be a threshold on the value of a classifier's discriminant function evaluated on given samples. An implicit probability or confidence measure for binary classifiers can also be obtained by feeding the value of the discriminant function to the logistic function as in [232, 233, 230]. In this case, the learning process queries samples x with class conditional probabilities near 0.5. This methodology has been used even when an explicitly probabilistic formulation has been used to model the data. In [24], a generative model of the target class was built. Random samples were generated from the generative model, and those samples lying near the decision boundary were queried using the logistic function for retraining.

Table 4.1: Definition of active learning approaches compared in this paper

Sample Query	Description	Object Detected
Query by Confidence, Labeled Examples	Examples with known class membership are queried based on a confidence or probability score.	Pedestrians [24]
Query by Confidence, Unlabeled Examples	Unlabeled examples are queried based on a confidence or probability score, and labeled by human.	Various objects [207]
Query by Misclassification	Unlabeled examples are queried based on raw classifier output, and labeled by human.	Vehicles [3], Pedestrians [230]

Training samples x were selectively sampled based on their class conditional probability, and these samples were used to retrain the SVM classifier [24, 22].

Explicitly probabilistic approaches for active learning in vision are also explored in [207, 24]. An online explicitly probabilistic formulation is used in [207], where sample query was implemented online using Gaussian Processes for regression. In this study, query is performed using both the mean and variance of the expected class membership, which allows for integrating measures of uncertainty directly into selective sampling query [207].

However, in many object detection studies misclassification is utilized [26, 231, 3, 230, 27]. In this case, the human oracle labels false positives and missed detections, which are then archived for retraining. In [26, 3], the learning process consists of an offline batch training, followed by a series of semi-supervised annotations, and a batch retraining. In [230, 27, 231], this is augmented by integrating online learning, so that each newly annotated frame immediately updated the classifier. In [213, 223, 226], tracking is used as a metric to identify regions of interest and gather more training samples. Table 4.1 summarizes the active learning approaches.

Labeling Costs in Active Learning

Robust object detection often requires tens of thousands of training examples, which can require extensive annotation time [27, 230, 231]. As such, research studies address the cost of annotating unlabeled examples in terms of data required and human annotation time [227, 207, 27]. In [207], the number of necessary annotations is shown to be quite small with the use of online Gaussian Process regression for object recognition.

In [227], the human cost of annotation is measured over multiple datasets, time frames, and individuals. It is found that annotation times are in general not constant, and that while a more precise query function may require fewer annotations, the time an individual takes may increase as the queried examples become more difficult [227]. In [214], the cost annotation costs

per sample are predicted using an uncertainty model.

Identifying mis-classified examples is a simple yet powerful approach to sample query than may increase annotation speed and lend itself easily to online learning for object detection [27, 226]. While online learning for object detection is shown to be quite effective [231], it is shown that combining offline initial training with online updates [27] can reduce annotation time and improve detection performance in deployment scenarios.

4.2.2 On-road Vehicle Detection

While the most popular vehicle detection systems found in consumer products are the radars used for adaptive cruise control, it is known that commonly used commercial grade sensors may have limited angular range and temporal resolution [211]. Many of the radar-based systems for adaptive cruise control are meant only to detect vehicles directly in front of the ego vehicle. This may present problems during lane changes, or when the road has non-zero grade or curvature. In addition, such systems often do not provide information on vehicles in neighboring lanes. Using vision for vehicle detection can recognize and track vehicles across multiple lanes [234].

Robust recognition of other vehicles on the road using vision is a challenging problem, and has been an active area of research over the past decade [20]. Highways and roads are dynamic environments, with ever-changing backgrounds and illuminations. The ego vehicle and the other vehicles on the road are generally in motion, so the sizes and locations of vehicles in the image plane are diverse. There is high variability in the shape, size, color, and appearance of vehicles found in typical driving scenarios [20].

Vehicle detection and tracking has been widely explored in the literature in recent years [234]. In [211], a variety of features were used for vehicle detection, including rectangular features and Gabor filter responses. The performance implications of classification with SVM's and NN classifiers was also explored. In [87], Histogram of Oriented Gradient features were used for vehicle localization.

The set of Haar-like features, classified with Adaboost has been widely used in the computer vision literature, originally introduced for detection of faces [54]. Various subsequent studies have applied this classification framework to vehicle detection [60, 67]. In [65], the effect of varying the resolution of training examples for vehicle classifiers was explored, using rectangular features and Adaboost classification [76]. Rectangular features and Adaboost were also used in [3], integrated in an active learning framework for improved on-road performance.

In [66], vehicle detection was performed with a combination of triangular and rectangular features. In [67], a similar combination of rectangular and triangular features was used for vehicle detection and tracking, using Adaboost classification. In [48], a statistical model based on vertical and horizontal edge features was used for vehicle detection. In [106], vehicles are tracked at nighttime by locating the taillights.



Figure 4.1: Examples of the varied environments where on-road vehicle detectors must perform.

4.3 Active Learning for On-road Vehicle Detection

Vision-based detection of other vehicles on the road is a difficult problem. Given that the end goal of on-road vehicle detection pertains to safety applications such systems require robust performance from an automotive platform. Roads and highways are dynamic environments, with rapidly varying backgrounds and illuminations. The ego vehicle and the other vehicles on the road are in motion, so the sizes and locations of vehicles in the image plane are diverse. There is high intra-class variability found in vehicles, in the shape, size, color, and appearance of vehicles found in typical driving scenarios [211]. There are also motion artifacts, bumps and vibrations from the road, and changes in pitch and yaw due to hills, curves, and other road structures. In addition, real-world on road scenes present many vehicle-like regions such as cast shadows, trees, man-made structures, which tend to spur false positives. In this study, we have applied three active learning approaches to the task of on-road vehicle detection. We discuss the approaches below.

4.3.1 Query by Confidence

Query by Confidence [QBC] is based on the notion that the most uncertain and informative samples are those samples that lie closest to the decision boundary. These examples can be queried based a confidence measure[23]. In [227] uncertainty was calculated using entropy over

the label posteriors. While generative models used throughout the learning process can facilitate a confidence based query based on probabilities as in [207, 24], often visual object detectors are based on discriminative classification. In particular, Support Vector Machines [22], and Adaboost [76] have been widely used [230, 3, 24].

Using common discriminative classifiers such as linear Support Vector Machines and Adaboost, binary classification is based on a weighted sum of extracted features, which can be viewed as an inner product [76]. This is to say that a sample x 's class y is given by the sign of a discriminant function $H(x)$ as given in equations 4.1 and 4.2, where $h_n(x)$ are extracted features, and w_n are the weights.

$$y(x) = \text{sgn}\{H(x)\} \quad (4.1)$$

$$H(x) = \sum_{n=0}^{N-1} w_n h_n(x) \quad (4.2)$$

Using equations 4.1 and 4.2 results in a hard classification. While discriminative classifiers are not explicitly probabilistic, a measure of confidence can be obtained by feeding $H(x)$ to a logistic function. The logistic function is monotonic and maps $H(x)$ to a value on the interval $[0,1]$. For Adaboost classification, the result of the logistic function can be interpreted as a class conditional probability $p(y = 1|x)$ [230], as in equation 4.3.

$$p(y = 1|x) = \frac{1}{1 + e^{-2H(x)}} \quad (4.3)$$

A rigorous proof of these equations can be found in [233]. For Support Vector Machines using a linear kernel, with primal , the class conditional probability is computed via the following equation [235, 197], where the parameters A and B are learned using maximum likelihood. Further discussion can be found in [197].

$$p(y = 1|x) = \frac{1}{1 + e^{-\{AH(x)+B\}}} \quad (4.4)$$

$$Q(x) = \{x : |p(y = 1|x) - 0.5| < \epsilon\} \quad (4.5)$$

With a confidence or probabilistic measure of a sample's class membership as defined by the classifier, we can define a query function $Q(x)$ that returns those samples that lie close to the decision boundary as in equation 4.5.

Query of Unlabeled Samples

Confidence based query may also be used with independent samples by prompting a human oracle for annotations [207] of examples that satisfy equation 4.5. In this case, there are

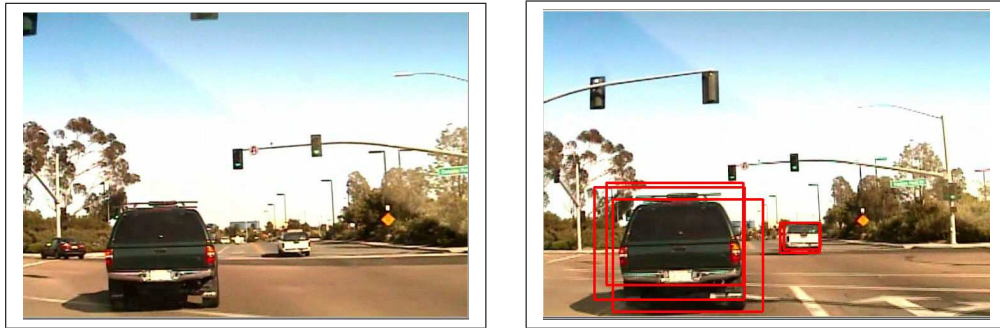


Figure 4.2: a) Training image from which the query function from equation 4.7 returned no informative training examples. b) Training image from which multiple informative image subregions were returned.

no prior hand-labels for the data, and a human must decide whether the examples correspond to objects of interest. A human oracle’s judgment of the appropriateness of the examples may add semantically meaningful examples to the data corpus that otherwise might be missed by automatic methods. This query method incorporates no prior knowledge of structure. In this case it is assumed that while the initial corpus may not be adequate for training a classifier with good generality, that confidence regression properties of the classifier serve to query informative independent samples and minimize human annotation time. It is of note that fewer queried samples do not necessarily mean less human time spent on annotation, as humans may take longer to make decisions on annotating difficult samples [227].

By evaluating the query function $Q(x)$ on a corpus of already labeled training examples, we can retrieve those that lie closest to the decision boundary for retraining with no extra human effort. It is shown in equations 4.1-4.5 that queried samples are those with almost equal class-conditional probabilities.

Query of Labeled Samples

While in [24] pre-cropped, hand-labeled training examples were used for confidence query, in this study, the entire original image has been retained. For each location and scale in the search space of the image, we calculate the confidence value using the trained model. Using equation 4.5, we query those image subregions that lie close to the trained model’s decision boundary. In the next step, we exploit structural information, using the location of the subregion queried using 4.5 and prior image annotations to compute the *Overlap*, as shown in equation 4.6. In this case, x' signifies a prior annotation, and x signifies the image subregion sample returned by 4.5.

Using equation 4.7, if $Q'(x)$ is non-zero, subregion x is retained for retraining. This approach provides a principled way for the learning process to query training examples that lie near the decision boundary and satisfy the structural constraint defined by hand labeling. It also allows us to obtain multiple informative training examples from a single hand-labeled example.

This also means that queried examples are not strictly a subset of the initial training examples.

$$Overlap(x', x) = \frac{Area(x' \cap x)}{Area(x' \cup x)} \quad (4.6)$$

$$Q'(x) = Q(x) \text{ if } Overlap(x', x) > \tau, \text{ else } Q'(x) = 0 \quad (4.7)$$

Query of negative training examples is performed using equations 1-4. We simply query negative image regions whose confidence lies close to the trained model’s decision boundary, which are most informative for updating the classifier [24].

4.3.2 Query by Misclassification

There exist scenarios where although a classifier may have excellent performance over the training examples, the environments encountered in deployment differ substantially from the training examples, to the detriment of system recall and precision [230, 27].

To ameliorate this problem, Query by Misclassification has been used in [26, 230, 27, 3, 231]. This method requires a human in the loop to label queried examples. The system typically presents the user with the results of evaluating an initially trained classifier on independent datasets. Often the independent data is more pertinent to actual deployment scenarios than the data that has been used in initial training [230]. Users then mark these results as correct detections and false positives, and are able to also mark missed detections [3, 213]. The annotation time used in sample query is significantly faster than annotation associated with gathering initial training examples.

Figure 4.3 depicts the interface used in this study for Query by Misclassification. The interface evaluates the initial detector, providing an interface for a human to label ground truth. Detections are automatically marked green. Missed detections are marked red by the user, and false positives blue.

In this active learning paradigm, the most informative samples are those that result in misclassifications by the object detector [26, 230, 3]. However, correct detections are generally retained for retraining, to maintain a superset of the region of uncertainty, and avoid overfitting [208]. Using this methodology, efficient on-line learning systems have been implemented in [231, 230, 27] in surveillance deployments for object detection.

4.4 Experimental Evaluation

4.4.1 Learning Considerations

Active learning is employed for object detection to train using less data [227], to minimize human annotation time [207], and to improve classifier performance [208, 230, 231, 3]. We briefly

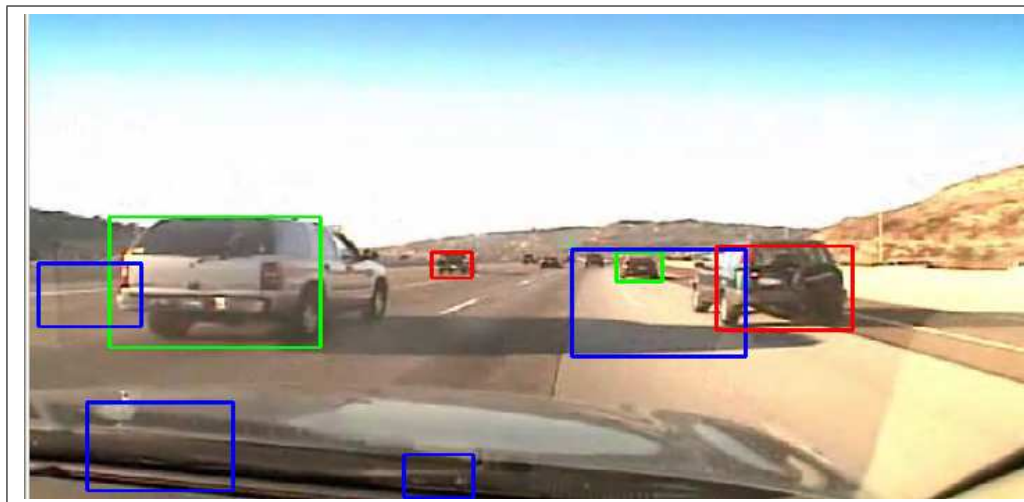


Figure 4.3: Interface for Query by Misclassification. The interface evaluates the initial detector, providing an interface for a human to label ground truth. Detections are automatically marked green. Missed detections are marked red by the user, and false positives blue.

examine the implications of these three factors in building robust active learning based object detection systems.

Table 4.2: Comparison of Labeling Time for Vehicle Detectors Trained with 1000 Samples, HOG-SVM

Method	Data Samples	Total Data	Labeling Time	Total Time	Indep. Samples?
Random Samples	1000	1000	27.8 hours	27.8 hours	No
Query by Misclassification	1000	11000	2.3 hours	30.1 hours	Yes
Unlabeled Query by Confidence	1000	11000	3.0 hours	30.8 hours	Yes

Data Considerations

The size and quality of available training data has major implications for active learning strategies. In certain applications, labeled data may be scarce, and learning methods may be aimed at using the scarce data as efficiently as possible, building models based on this data to query informative unlabeled samples [207]. However, if a large labeled data corpus is available, then learning methods can be aimed at sampling a subset of the extant corpus for retraining [24]. Of course it is favorable to minimize the size of the training data, reducing computational load, while maintaining strong classifier performance.

Further data considerations include characteristics of the features, dimensionality of the

features, the resolution of training examples. These considerations present intertwined issues. A given feature set can be well suited for a given object detection task, but the features themselves may require a certain image resolution. For example, HOG features have been widely used for pedestrian detection [21], but require a higher image resolution than Haar features, which have been used widely for vehicles [65, 3]. The required image resolution and feature set influences the dimensionality of the feature vector, which in turn influences the the required number of training examples for a given performance benchmark.

Supervision Costs

Compiling a set of training examples requires much labor and time spent on collecting and annotating video sets [24]. It is favorable to minimize the effort spent on annotation while still maintaining high quality data for training. While methods for querying the most difficult examples from unlabeled data corpuses have been proposed in [207, 227], it is shown in [227] that querying the most difficult examples may increase the time a user spends on annotation.

4.4.2 Feature and Classifiers Sets

Two sets of experiments have been conducted to evaluate active learning approaches, using two different feature-classifier pairs for vehicle detection. The first set of experiments have used Histogram of Oriented Gradient features with linear Support Vector Machine classification [21, 235, 22]. The second set of experiments have used Haar-like features and Adaboost cascade classification [54, 76].

4.4.3 Training Sets

We begin with a hand-labeled initial data corpus of 10000 positive and 15000 negative training examples, which was used to train the initial classifier. These examples were taken from video captured on highways and urban streets. For implementing Query by Confidence for labeled samples, we queried uncertain examples that satisfied equation 4.7. In querying independent unlabeled samples, Query by Misclassification and Query by Confidence, we queried examples using the methods described in the previous section, applied to an independent data corpus. A human oracle performed the semi-supervised labeling. All labeling was timed. The independent data video corpus was acquired by concatenating 9 high density urban and highway traffic scenes, each lasting some 2 minutes.

4.4.4 Experiment 1: HOG-SVM Vehicle Detection

In this set of experiments, we train vehicle detectors using HOG features and linear SVM classifiers. The goal here was to evaluate what the potential performance would be with a very

Table 4.3: Active Learning Results: HOG-SVM Vehicle Detection, Caltech 1999 Vehicle Database

Sample Query	Recall	Precision
Random Exam- ples	44.4%	59.0%
Query by Confi- dence	54.8%	60%
Query by Mis- classification	77.8%	81.4%

small number of training examples, in this case, 1000. Vehicle detectors were evaluated on the Caltech 1999 vehicle database at <http://www.vision.caltech.edu/archive.html>, which consists of 127 still images of vehicles. Table 4.2 shows the labeling time for querying 1000 samples. We note that Query by Confidence took somewhat longer than Query by Misclassification. This is due to the fact that ambiguous samples are often more difficult for users to decide labels [227]. Parameters used for the training are provided in table 4.4, trained with LibSVM [235].

Table 4.4: HOG-SVM Vehicle Detection, Training Parameters

Parameter	Value
Number of Orienta- tions	4
Cell size	8×8
Training sample reso- lution	96×72
Kernel	Linear

Table 4.3 shows the precision and recall of the classifiers, trained with 1000 samples. We note that both active learning methods showed improved recall and precision, over the vehicle detector trained with random examples. However, the results from Query by Misclassification were far better than those from Query by Confidence.

4.4.5 Experiment 2: Haar features and Adaboost

In this set of experiments, we document the relative merits and trade-offs of active learning for vehicle detection using Haar-like features and Adaboost. Detectors are learned using training sets of 2500, 5000, and 10000 training examples. We evaluate the vehicle detectors on a publicly available dataset, *LISA.2009.Dense*, which can be found at <http://cvrr.ucsd.edu/LISA/index.html>. This is a difficult dataset consisting of 1600 consecutive frames. Captured during rush hour, this scene contains complex shadows, dynamic driving maneuvers, and five lanes of traffic. There are 7017 vehicles to be detected in this clip. The distance covered is roughly 2km. Parameters used in training are provided in 4.8



Figure 4.4: Example from *LISA_2009_Dense* validation set. The dataset consists of 1600 consecutive frames, captured during rush hour, over a distance of roughly 2km. The ground-truthed dataset is publicly available to the academic and research communities at <http://cvrr.ucsd.edu/LISA/index.html>.

Table 4.5: Comparison of Annotation Time for Vehicle Detectors Trained with 2500 Samples, Haar+Adaboost

Method	Data Samples	Total Data	Labeling Time	Total Time	Indep. Samples?
Random Samples	2500	2500	7 Hours [projected]	7 hours [projected]	No
Labeled Query by Confidence	2500	12500	0 hours	27.8 hours	No
Query by Misclassification	2500	12500	2.5 hours	30.3 hours	Yes
Independent Query by Confidence	2500	12500	2.8 hours	30.6 hours	Yes

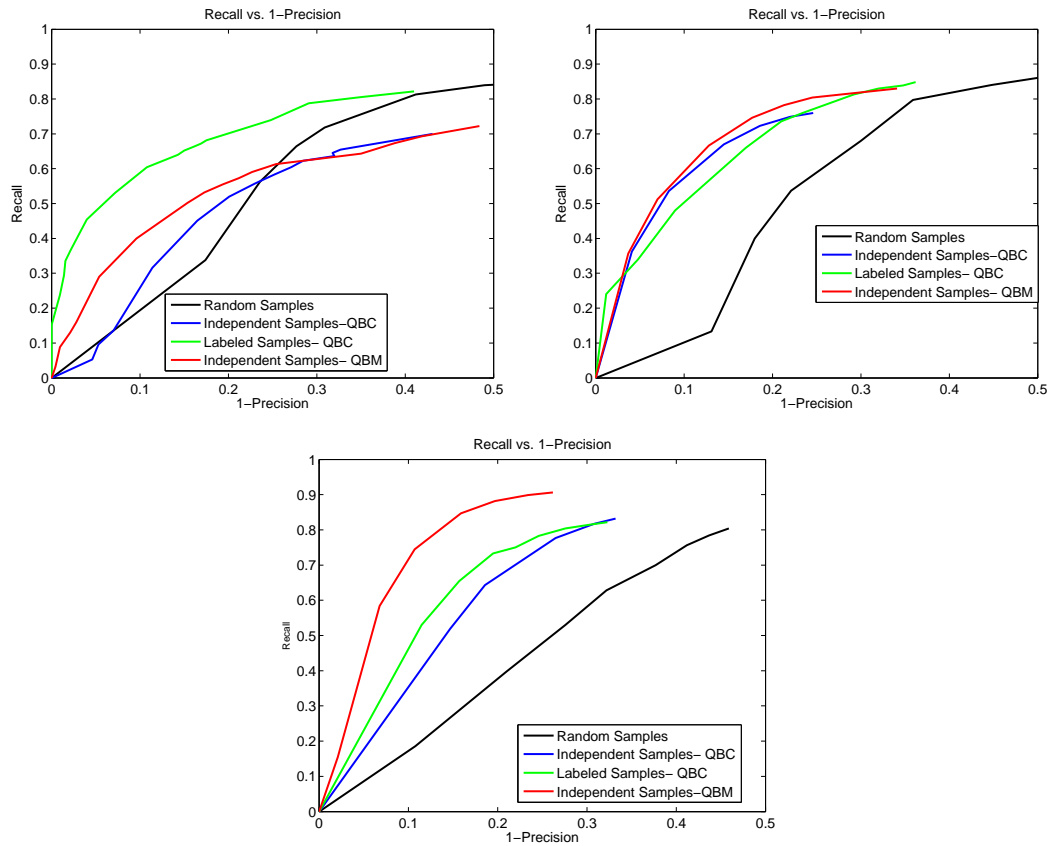


Figure 4.5: Recall vs. 1-Precision for each vehicle detector, evaluated on the *LISA_2009_Dense* dataset. a) Classifiers trained with 2500 samples b) Classifiers trained with 5000 samples c) Classifiers trained with 10000 samples.

Table 4.6: Comparison of Annotation Time for Vehicle Detectors Trained with 5000 Samples, Haar+Adaboost

Method	Data Samples	Total Data	Labeling Time	Total Time	Indep. Samples?
Random Samples	5000	5000	14 Hours [projected]	14 hours [projected]	No
Labeled Query by Confidence	5000	15000	0 hours	27.8 hours	No
Query by Misclassification	5000	15000	4.0 hours	31.8 hours	Yes
Independent Query by Confidence	5000	15000	4.6 hours	32.4 hours	Yes

Table 4.7: Comparison of Annotation Time for Vehicle Detectors Trained with 10000 Samples, Haar+Adaboost

Method	Data Samples	Total Data	Labeling Time	Total Time	Indep. Samples?
Random Samples	10000	10000	27.8 hours	27.8 hours	No
Labeled Query by Confidence	10000	20000	0 hours	27.8 hours	No
Query by Misclassification	10000	20000	7.0 hours	34.8 hours	Yes
Independent Query by Confidence	10000	20000	7.6 hours	35.4 hours	Yes

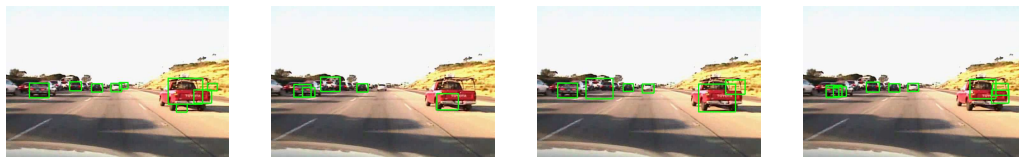


Figure 4.6: Frame 1136 of *LISA_2009_Dense*. Detectors trained with 2500 samples. a) Random Samples. b) Query by Misclassification. c) Query by Confidence- Independent Samples . d) Query by Confidence- Labeled Samples.

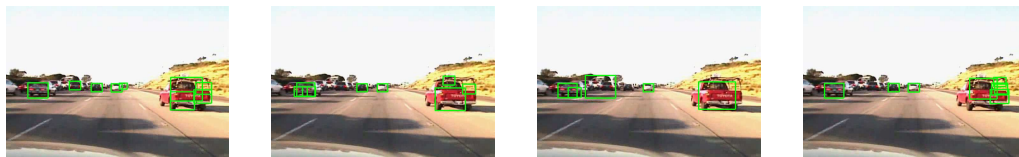


Figure 4.7: Frame 1136 of *LISA_2009_Dense*. Detectors trained with 5000 samples. a) Random Samples. b) Query by Misclassification. c) Query by Confidence- Independent Samples . d) Query by Confidence- Labeled Samples.

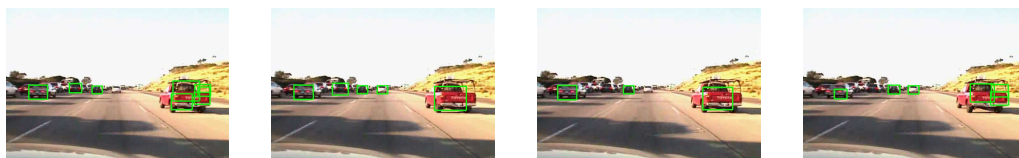


Figure 4.8: Frame 1136 of *LISA_2009_Dense*. Detectors trained with 10000 samples. a) Random Samples. b) Query by Misclassification. c) Query by Confidence- Independent Samples . d) Query by Confidence- Labeled Samples.

4.4.6 Analysis

Human Labeling Time

Table 4.8: Haar-Adaboost Vehicle Detection, Training Parameters

Parameter	Value
Number of cascade stages	15
Training error per stage	0.5
Expected training error	3e-5
Training sample resolution	24×18

Tables 4.5, 4.6, and 4.7 detail the time costs associated with each learning method. While Query by Misclassification and Query by Confidence methods require extra labeling to assign class membership to independent examples, it is of note that even to query and label 10000 independent training examples, it only took some 7 additional hours of annotation. This is due to the fact that the respective query functions make labeling much more efficient.

For all of the active learning methods, we note that the bulk of human labor was spent labeling the initialization set. We also note labeling independent examples using Query by Confidence consistently took longer than Query by Misclassification. This is a similar phenomenon as reported in [227]. Query by Confidence uses a more sophisticated query criterion to return the most difficult examples. A human annotator requires more time to annotate the most difficult examples. Querying the most difficult examples may result in a better trained classifier, but it does not reduce the time spend labeling.

Data Implications and System Performance

As we have trained each classifier with the same number of examples, using their respective active learning query functions, discussion of data implications and system performance are intertwined. Tables 4.5, 4.6, and 4.7 show the number of training examples used to train each classifier. Each classifier has used the same number of positive and negative training examples for each instantiation, 2500, 5000, or 10000. However, active learning methods that use independent samples, Query by Confidence and Query by Misclassification, are based on the initial classifier. As such, we add the size of the initial training corpus to their data requirements.

We have plotted Recall vs. 1-Precision for each classifier and each dataset. The performance of classifiers trained with 2500 examples is plotted in figures 4.5(a). The performance of classifiers trained with 5000 examples is plotted in 4.5(b). The performance of classifiers trained with 10000 examples is plotted in 4.5(c).

4.4.7 Analysis

We observe that in general, as the number of training examples increases, so does the performance of each classifier. Figures 4.5(a), 4.5(b), and 4.5(c) are all plotted on the same axes scale. The overall improvement in system performance with increased training data is shown there.

In general, we find that each of the active learning methods outperforms training with random examples, for both HOG-SVM, and Haar-Adaboost detection algorithms. This confirms the valuable contribution of active learning to vehicle detection. In the HOG-SVM experiment 1, QBM performed the best, followed by QBC, and finally training with random examples from the labeled corpus. We observe a similar performance trend in the Haar-Adaboost experiments.

In experiment 2, as the number of training examples increases, the rankings of the three active learning methods changes. Using only 2500 training examples, the best classifier is that built with labeled examples, using Query by Confidence as 2500 samples is not enough to build a rich, representative data corpus for retraining at such a low resolution. Query by Confidence queries the most informative labeled examples from the initial training corpus, and performs the best. In fact, training with random examples results in better performance than using independent samples for active learning, for such a small number of training examples, as shown in 4.5(a).

In the next training set, we see changes in the performance rankings. Each of the active learning methods using 5000 training examples outperforms training with random samples, as shown in 4.5(b). Using 5000 training samples, each of the active learning methods perform comparably well. Query by Misclassification yields the best performing classifier, but Query by Confidence of labeled and independent examples perform almost as well.

Throughout the experiments, we note that classifiers using Query by Confidence for labeled or independent samples perform similarly. This is to say that the inclusion of independent training samples doesn't make a large difference in the classifier's performance when using Query by Confidence. The reason for this lies in the similar query criterion. While one methods queries automatically from a labeled corpus, and the other queries from unlabeled independent samples, they both use the same query function, defined in equations 1-4. As such, it makes sense that the classifiers that these methods yield perform comparably.

We find that Query by Misclassification performs the best of the learning approaches for vehicle detection, using HOG-SVM, and Haar-Adaboost. Each of the active learning methods far outperforms the initial classifier, training with random examples. Query by Misclassification has been used widely in the literature [26, 230, 213, 27, 226, 3], and so it is expected that this method would perform well. This strong performance comes at the price of 7 extra hours of annotation, and an extra independent data requirement. Of the active learning methods that use independent data that we've examined in this study, Query by Misclassification is the best

choice for system performance and human labor.

4.5 Remarks

In this study, we have compared the labeling costs and performance pay-offs of three separate active learning approaches for on-road vehicle detection. Initial vehicle detectors were trained using on-road data. Using the initial classifiers, informative examples were queried using three approaches: Query by Confidence from the initial labeled data corpus, Query by Confidence of independent samples, and Query by Misclassification of independent samples. The human labeling costs have been documented. The recall and precision of the detectors have been evaluated on static images, and challenging real world on-road datasets. The generality of the findings have been demonstrated by using detectors comprised of HOG-SVM and Haar-Adaboost. We have examined the time, data, and performance implications of each active learning method. The performance of the detectors has been evaluated on publicly available vehicle datasets, as part of long term research studies in intelligent driver assistance.

4.6 Acknowledgments

Chapter 4 is a partial reprint of material published in *Machine Vision and Applications*, 2011. The dissertation author was the primary investigator and author of these papers.

Chapter 5

Integrated Lane and Vehicle Detection, Localization, and Tracking: A Synergistic Approach

5.1 Introduction

Annually, between 1 and 3 percent of the world's GDP is spent on the medical costs, property damage, and other costs associated with automotive accidents. Each year, some 1.2 million people die worldwide as a result of traffic accidents [13]. Research into sensing systems for vehicle safety promises safer journeys by maintaining an awareness of the on-road environment for driver assistance. Vision for driver assistance has been a particularly active area of research for the past decade [218].

Research studies in computer vision for on-road safety have involved monitoring the interior of the vehicle [228], the exterior [215, 236], or both [193, 218, 237]. In this research study, we focus on monitoring the exterior of the vehicle. Monitoring the exterior can consist of estimating lanes [215, 238], pedestrians [219, 239, 240], vehicles [211, 3, 5, 241, 242], or traffic signs [236]. Taking a human-centered approach is integral for providing driver assistance [243]; using the visual modality allows the driver to validate the system's output, and to infer context.

Many prior research studies monitoring the vehicle exterior address one particular on-road concern. By integrating information from across systems, complimentary information can be exploited, and more contextually relevant representations of the on-road environment can be attained.

In this paper, we introduce a synergistic approach to integrated lane and vehicle tracking for driver assistance. Utilizing systems built upon works reported in the literature, we integrate

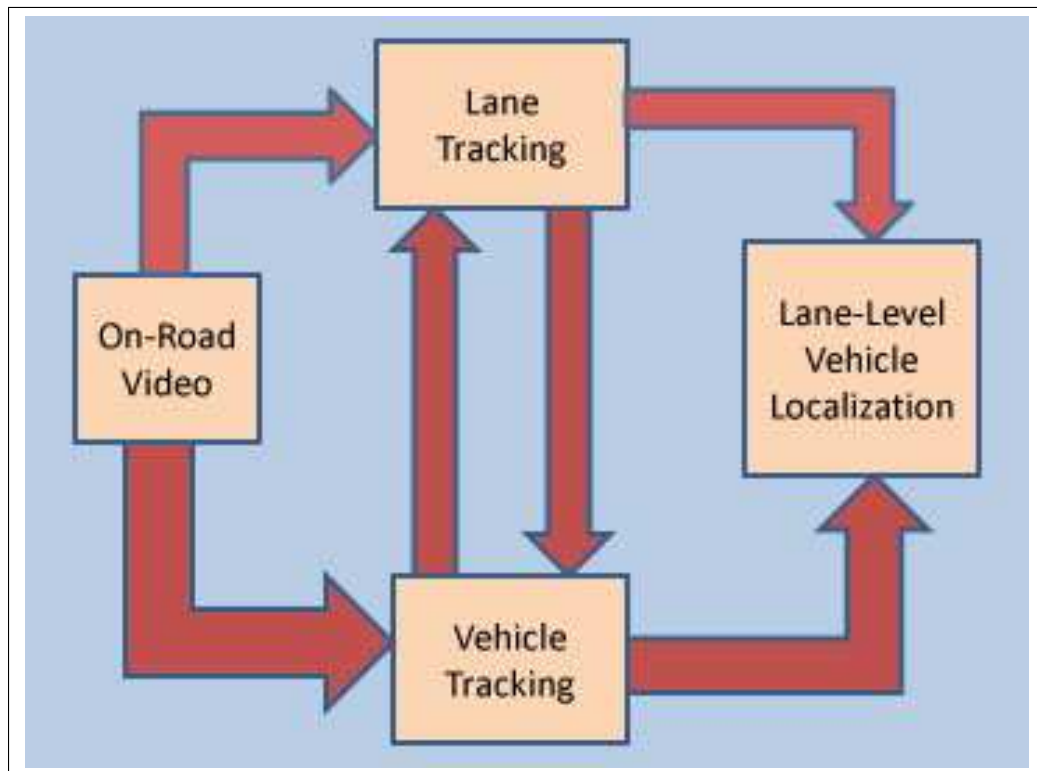


Figure 5.1: Framework for integrated lane and vehicle tracking, introduced in this study. Lane tracking and vehicle tracking modules are executed on the same frame, sharing mutually beneficial information, to improve the robustness of each system. System outputs are passed to the integrated tracker, which infers full state lane and vehicle tracking information.

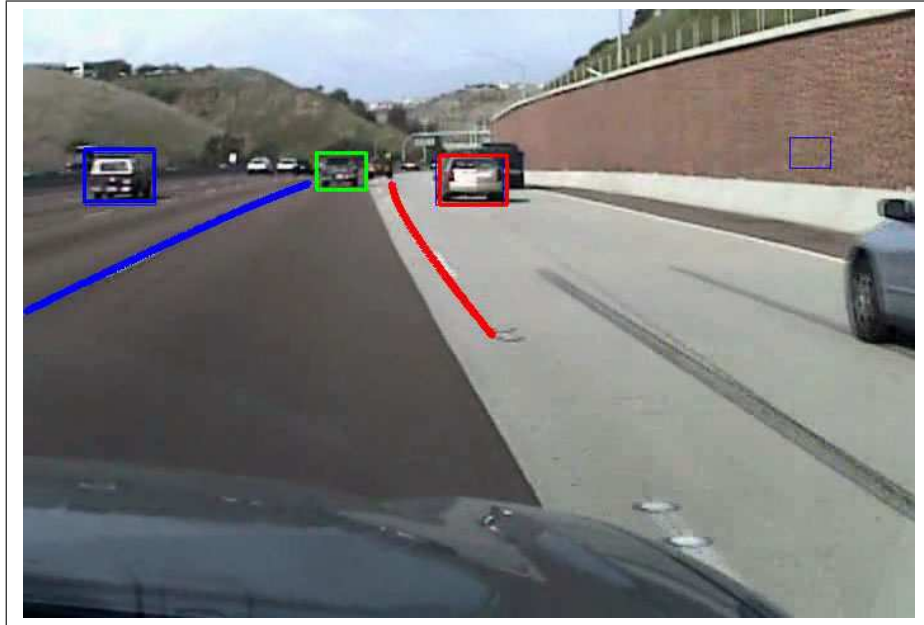


Figure 5.2: Typical performance of integrated lane and vehicle tracking on highway with dense traffic. Tracked vehicles in the ego lane are marked green. To the left of the ego lane, tracked vehicles are marked blue. To the right of the ego lane, tracked vehicles are marked red. Note the curvature estimation.

lane and vehicle tracking and achieve the following. Lane tracking performance has been improved by exploiting vehicle tracking results, eliminating spurious lane marking filter responses from the search space. Vehicle tracking performance has been improved by utilizing the lane tracking system to enforce geometric constraints based on the road model. By utilizing contextual information from two modules, we are able to improve the performance of each module. The entire system integration has been extensively quantitatively validated on real-world data, and benchmarked against the baseline systems.

Beyond improving the performance of both vehicle tracking and lane tracking, this research study introduces a novel approach to localizing and tracking vehicles with respect to the ego-lane, providing lane-level localization of other vehicles on the road. This novel approach adds valuable safety functionality, and provides a contextually relevant representation of the on-road environment for driver assistance, previously unseen in the literature. Figure 5.1 depicts an overview of the approach detailed in this study, and Figure 5.2 shows typical system performance.

The remainder of this paper is structured as follows. In Section 2, we discuss relevant research in the literature pertaining to on-road lane tracking and vehicle tracking for driver assistance. In Section 3, we detail the lane tracking and vehicle tracking modules that have been utilized in this research study. In Section 4, we introduce a synergistic framework for integrated lane and vehicle tracking. In Section 5, we provide thorough experimental evaluation of the introduced framework, via three separate classes of experiments. Finally, in Section 6, we

provide concluding remarks and discuss future research directions.

5.2 Related Research

5.2.1 Lane Detection and Tracking

Lane tracking has been an active area of research for over a decade [244]. At its most basic level, lane keeping for driver assistance consists of locating lane markings, fitting the lane markings to a lane model, and tracking their locations temporally with respect to the ego vehicle. Image descriptors reported in the literature for lane marking localization include adaptive thresholds [245, 246], steerable filters [215, 221, 238], ridges [247] edge detection, global thresholds, and top hat filters [246]. In [248], a classifier-based lane marker detection is employed. A thorough side-by-side segmentation comparison of lane feature extractors can be found in [246].

Road models used in lane detection and tracking systems often try to approximate the clothoid structure which is often used in road construction [215]. This is often done via a parabolic or cubic fitting of the lane markings to a parametric road model [245]. In [215], this is achieved via fitting an adaptive road template to the viewed data. In recent studies [248, 247], RANSAC has been used to fit lane markings to parametric road models. Rural and urban roads may contain various discontinuities, which can require more sophisticated road modeling [249].

Lane tracking has often been implemented using Kalman filters, or variations such as the EKF, which tend to work well for continuous, structured roads [215, 221, 245, 238]. The state vector tracks the positions of the lane markings, heading, curvature, and the vehicle's lateral position [215]. Particle filtering [250] has gained popularity in lane tracking, as it natively integrates multiple hypotheses for lane markings [251, 248]. In [249], a hybrid Kalman Particle filter has been implemented for lane tracking, which combines the stability of the Kalman filter with the ability to handle multiple cues of the particle filter.

5.2.2 Vehicle Detection and Tracking

Vehicle detection and tracking has been widely explored in the literature in recent years [3, 5, 252, 253]. In [211], a variety of features were used for vehicle detection, including rectangular features and Gabor filter responses. The performance implications of classification with SVM's and NN classifiers was also explored. In [87], deformable part-based modeling was used for vehicle localization.

The set of Haar-like features, classified with Adaboost has been widely used in the computer vision literature, originally introduced for detection of faces [54]. Various subsequent studies have applied this classification framework to vehicle detection [60, 67], using Adaboost [76]. Rectangular features and Adaboost were also used in [3], integrated in an active learning framework for improved on-road performance.

In [66], vehicle detection was performed with a combination of triangular and rectangular features. In [67], a similar combination of rectangular and triangular features was used for vehicle detection and tracking, using Adaboost classification. In [48], a statistical model based on vertical and horizontal edge features was integrated with particle filter vehicle tracking. Particle filter tracking was also used in [3] and [87]. Night-time detection of vehicles has been explored in [5].

5.2.3 Integrating Lane and Vehicle Tracking

While dense traffic has been reported as challenging for various lane tracking [215] and vehicle tracking systems [3], few studies have explored integration of lane and vehicle tracking. In [254], lanes and vehicles were both tracked using PDAF. The study showed that coupling the two could improve vehicle detection rates for vehicles in the ego lane. However, [254] does not quantify lane tracking performance, and does not infer other vehicles' lane positions. In [234], vehicle and lane tracking were combined for improved lane localization. However, [234] did not use lane or road information to improve vehicle detection, or localize vehicles with respect to lanes.

While [254] and [234] have explored some level of integration of vehicle and lane tracking, neither has demonstrated a full integration to benefit both vehicle and lane tracking, and neither study has utilized lane and vehicle tracking to infer any higher-level information about the traffic scene, such as local lane occupancy. This paper offers several contributions that have not been reported in prior works, and provides an extensive quantitative validation and analysis. Further, this paper specifically tests the system in dense traffic, which is known to be a difficult scenario for vision-based driver assistance systems.

5.3 Lane Tracking and Vehicle Tracking Modules

In this section, we first briefly review the lane tracking and vehicle tracking modules utilized in this study. The modules used in this study are based on prior works that have been reported in the literature [215, 3]. Building upon tracking systems already reported in the literature serves two main purposes. First, it allows us to demonstrate the generality of our approach, using established techniques. Second, it provides a benchmark against which to compare the performance of the integrated systems approach.

5.3.1 Lane Tracking using Steerable Filters

For lane marking localization, we work with the inverse perspective mapped image of the ground plane, which has been widely used in the literature [248, 238]. The camera's intrinsic parameters are determined using standard camera calibration. Using the camera parameters, a ground-plane image can be generated given knowledge of the real-world coordinate origin and

the region on the road we want to project the image onto [238]. Real-world points lying on the ground plane are mapped into the camera's frame of reference using a rotation and a translation, as in equation 5.1. Given a calibrated camera, 3D points in the camera's frame of reference are mapped to pixels using a pinhole model, as in equation 5.2. Using real-world points of known location on the ground plane, a homography is computed using DLT [33] to map the image plane projections of real-world ground plane points to a ground-plane image, shown in equation 5.3. Pixel locations of points in the flat-plane image and the actual locations on the road are related by a scale factor and offset. H is a 3×3 matrix of full rank, mapping homogeneous points from the image plane to the ground plane [33].

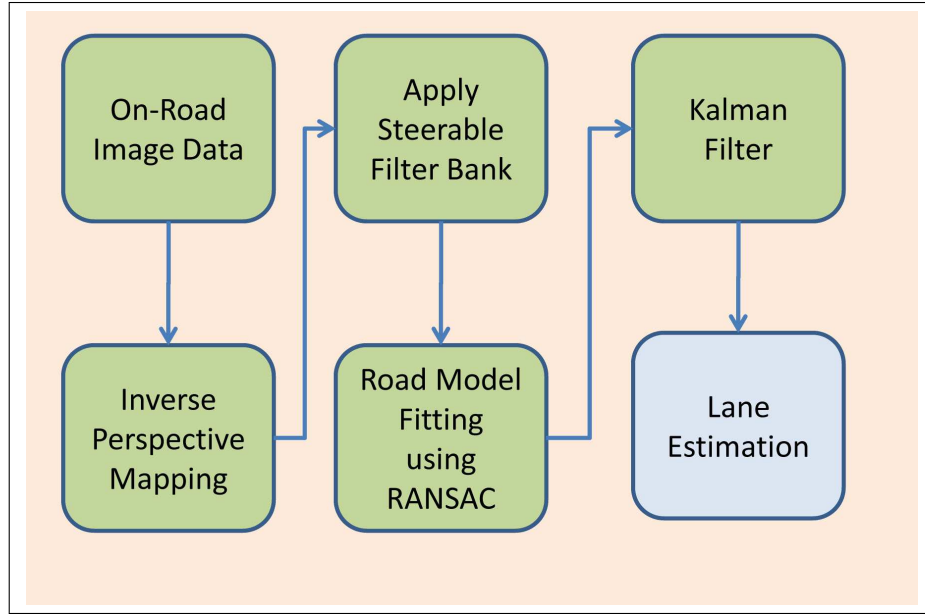


Figure 5.3: Lane tracking framework used in this study. Feature extraction is achieved by applying a bank of steerable filters. The road model is fit using RANSAC, and lane position tracked with Kalman filtering.

$$\begin{bmatrix} X & Y & Z \end{bmatrix}^T = \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X_{world} \\ 0 \\ Z_{world} \\ 1 \end{bmatrix} \quad (5.1)$$

$$x_{image} = \begin{bmatrix} i_{image} \\ j_{image} \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (5.2)$$

$$x_{ground} = Hx_{image} \quad (5.3)$$

We apply a bank of steerable filters to the IPM image. Steerable filters have been used

in prior lane tracking studies [215, 221, 238], and have been shown to detect various types of lane markings in a robust manner. Steerable filters are separable, and capable of localizing lane markings at various orientations. They are constructed by orienting the second derivative of Gaussian filters.

It can be shown that the response of any rotation of the second derivative of Gaussian filter by an angle θ can be computed using equation 5.4. G_{xx} , G_{yy} , and G_{xy} correspond to the second derivatives in the x , y , and x - y directions, respectively.

$$\begin{aligned}
 G2^\theta &= G_{xx}\cos^2\theta + G_{yy}\sin^2\theta - 2G_{xy}\cos\theta\sin\theta \\
 G2^{\theta_{min/max}} &= G_{yy} - \frac{2G_{xy}^2}{G_{xx} - G_{yy} \pm B} \\
 B &= \sqrt{G_{xx}^2 - 2G_{xx}G_{yy} + G_{yy}^2 + 4G_{xy}^2}
 \end{aligned} \tag{5.4}$$

We solve for the maximum and minimum response angles θ_{min} and θ_{max} . Using the filter responses, we then aggregate the observed measurements, and fit them to the road model using 100 iterations of RANSAC [255], removing outliers from the measurement. RANSAC has been used in various lane tracking studies for model fitting [248, 247]. In this study, we use a parabolic model for the road, given in equation 5.5. Table 5.1 defines the variables used in lane tracking, and figure 5.4 illustrates the coordinate system.

$$\begin{aligned}
 X_l(Z) &= \phi - \frac{1}{2}W + b_lZ + CZ^2 \\
 X_r(Z) &= \phi + \frac{1}{2}W + b_rZ + CZ^2
 \end{aligned} \tag{5.5}$$

We track the ego vehicle's position within its lane, the lane width, and lane model parameters using Kalman Filtering. The system's linear dynamic model is given in equation 5.6. Observations come from passing the steerable filters over the ground plane image, and fitting the lane model in equation 5.5 using RANSAC.

Table 5.1: Variables used for Lane Tracking

Variable	Meaning
X_l, X_r	Left and right lane boundaries
Z	Longitudinal distance
ϕ	Lateral position within the lane
W	Lane width
b_l, b_r	Left and right lane boundary slope
C	curvature
l_k	lane state estimate
v	vehicle's velocity

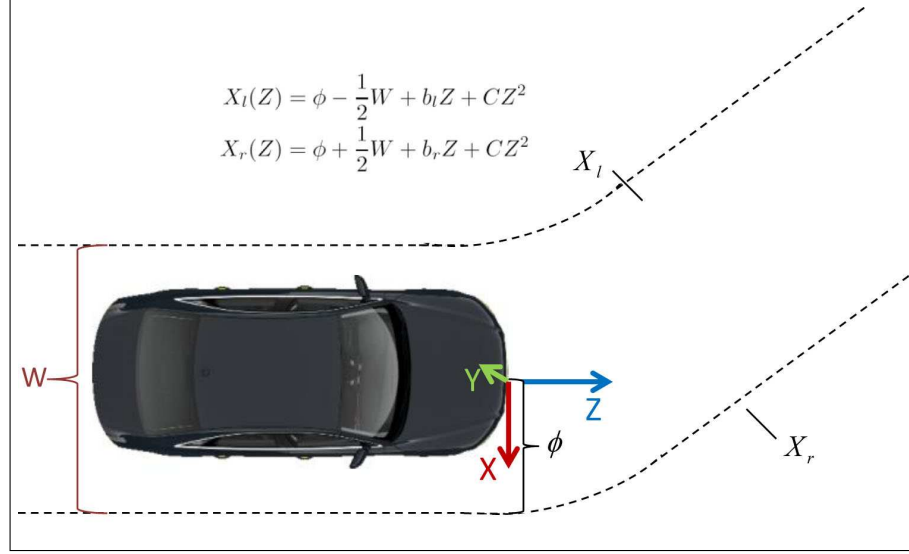


Figure 5.4: An illustration of the variables used in lane tracking, further explained in Table 5.1

$$\begin{aligned}
 l_{k|k-1} &= Al_{k-1} + w_{k-1} \\
 y_k &= Ml_k + v_k \\
 l &= [\phi \quad \dot{\phi} \quad b_l \quad b_r \quad C \quad W]^T \\
 A &= \begin{bmatrix} 1 & v\Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -\frac{1}{2} \\ 1 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.6)
 \end{aligned}$$

5.3.2 Active Learning for Vehicle Detection with Particle Filter Tracking

We have based the on-road vehicle detection and tracking module in this study on the one introduced in [3]. It consists of an active learning based vehicle detector, integrated with particle filtering for vehicle tracking [3, 250]. A comparative study of the performance of active learning approaches for vehicle detection can be found in [256]

For the task of identifying vehicles, a boosted cascade of simple Haar-like rectangular features has been used, as was introduced by Viola and Jones [54] in the context of face detection. Various studies have incorporated this approach in on-road vehicle detection systems such as [67, 60]. Rectangular features are sensitive to edges, bars, vertical and horizontal details, and

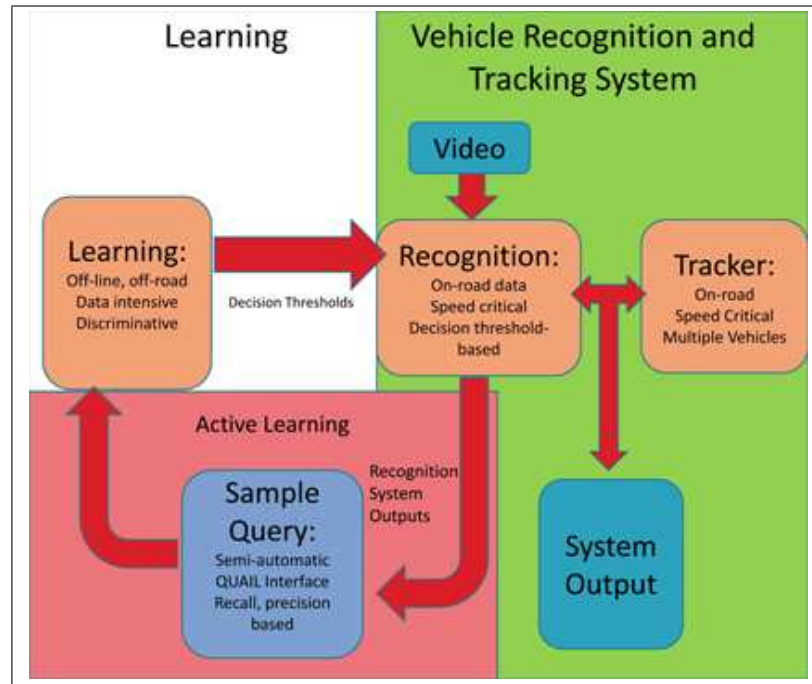


Figure 5.5: Active learning for vehicle detection and tracking, module originally presented in [3]

symmetric structures [54]. The resulting extracted values are effective weak learners [54], which are then classified by Adaboost [76]. In [3] active learning was utilized for training an on-road

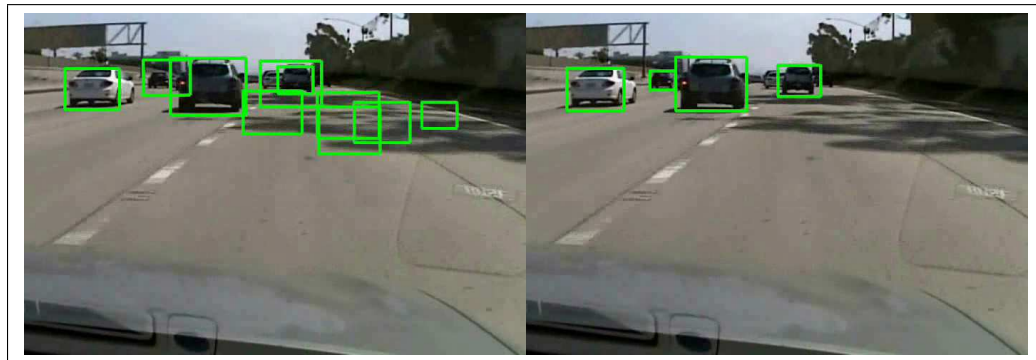


Figure 5.6: Comparison of initial classifier with active learning based vehicle detection, in scenes with complex shadowing.

vehicle detector. An initial classifier was trained using conventional supervised learning, then evaluated on independent real-world datasets. Misclassifications, e.g. false positives and missed vehicles, were queried, along with correct detections, and archived for a retraining stage [24]. The active learning based classifier showed significant improvements in recall and precision. Figure 5.6 shows a side by side comparison of vehicle detector outputs with complex shadowing. On

the left, the output of the initial detector is shown, and on the right, the output of the active learning based detector. Vehicles that persist as detections over three frames are then tracked. Particle filter tracking has been implemented using the Condensation algorithm [250].

5.4 Synergistic Integration

There are two intertwined motivations for integration of lane and vehicle tracking. The first deals with improving the tracking performance of each module via system integration. The second deals with utilizing higher-level information for traffic scene understanding. In the real-world context, dense traffic presents a challenging scenario for vision-based driver assistance, as it presents extensive visual clutter, occlusions, complex texture and shadowing, and dynamic factors. These characteristics lead to false positives, and poor localization. Integrating lane and vehicle tracking can provide robustness in dense traffic scenarios, improving tracking performance for vehicles and lanes. Combining complimentary information from the trackers augments valuable contextual information. The integration of the two systems can be framed in terms of a feedback loop in a partially-observed system, where lane and vehicle estimates are information states [257]. Lane observations augment estimation of the vehicles, while vehicle observations augment lane estimation.

While prior works in vehicle tracking provide only relative position about vehicles, in this study we infer other vehicles' lane position. Lane-level localization of other vehicles provides informational representations that are not possible with relative position alone. Unlike vehicle tracking in prior works, in this study the lane positions and lane changes of other vehicles can be identified, providing safety critical information for short-term and long-term collision prediction. In particular, the system maintains awareness of other vehicles' lane position, identifying when a vehicle merges or deviates from a neighboring lane into the ego-vehicle's lane. By providing a discrete state-based representation of vehicle location on the road, advanced techniques in trajectory learning and classification can be applied [258, 2]. Additionally, traffic density can be locally assessed with respect to the lanes, based on the lane occupancy. This can serve as a basis for traffic-dependent path planning, or for studying driver behavior and perceptions of traffic.

Before we further detail the individual components of the proposed approach, we make the following observations. The system described has no thresholds or parameters to tune. The system described does not need to iterate multiple times over the same input frame. Each frame is processed once, and temporal tracking and coherence result in consistent system tracking outputs. We divide the contributions of the proposed approach into three main categories: Improved Lane Tracking Performance, Improved Vehicle Tracking Performance, and Vehicle Localization and Tracking with Respect to Lanes.

5.4.1 Improved Lane Tracking Performance

It is known that in dense traffic, vision-based lane tracking systems may have difficulty localizing lane positions, due to the presence of vehicles. This phenomenon has been reported in [215, 234, 254]. The reasons for this are two-fold. First, vehicles on the road can occlude lane markings. Second, highlights and reflections from vehicles themselves may elicit false positive lane marking responses, resulting in erroneous lane localization.

To improve the lane estimation and tracking performance, we integrate knowledge of vehicle locations in the image plane. We first pass a bank of steerable filters over the IPM image, as detailed in Section 5.3.1, using equations 5.3 and 5.4. At this point, we have a list of pixel locations in the ground plane, corresponding to filter responses from equation 5.4. Using the inverse of the homography matrix, H^{-1} , we can map potential lane markings from the ground plane into the image plane, as shown in equation 5.7.

$$x_{image} = H^{-1}x_{ground} \quad (5.7)$$

$$Overlap = r_1 \cap r_2 \quad (5.8)$$

While equation 5.7 maps the centroid of the lane marking into the image plane, for convenience we represent each potential lane marking as a small $n \times n$ rectangle in the image plane, centered at the mapped centroid. Vehicle tracking also provides a list of rectangles, corresponding to the tracked vehicle locations in the image plane. Using the Pascal criterion in equation 5.8 for the overlap of rectangles r_1 and r_2 , we can filter out those mapped lane markings that have overlap with the locations of tracked vehicles in the image plane. This effectively eliminates lane markings that correspond to vehicles in the traffic scene.

In practice, this approach produces the result that highlights from the vehicle, including reflections, taillights, and other features resembling lane markings, are excluded from the model-fitting state of the lane estimation, as shown in Figure 5.3. We handle occlusions caused by vehicles in the traffic scene, as well as false positive lane markings caused by vehicles, which is especially pertinent to dense traffic scenes. Using the knowledge of vehicle locations in the image plane, we distill the lane marking responses to only those that do not correspond to vehicles. We fit the road model to the pruned lane markings using RANSAC, and apply the Kalman filter for lane tracking.

5.4.2 Improved Vehicle Detection

When applying a vehicle detection system to a given image, false positives may be elicited by various structures in the image. Among these are symmetric structures such as bridges, road signs, and other man-made objects that in general, do not lie beneath the horizon. While various

research studies have explored the implications of different feature sets, classifiers, and learning approaches [211, 3], false positives elicited by objects that do not lie on the road can be eliminated by enforcing a geometric constraint.

The geometric constraint is borne of the contextual understanding of the scene. In traffic scenes, vehicles lie below the horizon. Using the horizon location in the image plane, we can filter out those potential vehicle detections that do not lie on the ground plane. This, in turn improves the system’s precision.

We estimate the location of the horizon in the image plane using the lane tracking results. Using equation 5.5, we have parabolic curves for the left and right lane boundaries. We find the vanishing point determined by the parabolic curves, by finding the intersection of their tangent lines projected into the image plane. The vertical y coordinate of the vanishing point is taken to be the location of the horizon in the image plane. Figure 5.11(b) shows this step.

To determine if an object lies beneath the horizon, we first use the tracked object’s state vector, as given in equation 5.9. We then use equation 5.10 to calculate the the center of the bottom edge of the object, $\mathbf{p}_{\text{bottom}}$, which is represented in the image plane by its bounding box. If the bottom edge of the object sits lower than the estimated location of the ground plane, we keep this object as a vehicle. Objects whose lower edge sits above the estimated ground plane are filtered out.

5.4.3 Localizing and Tracking Vehicles and Lanes

Locating and tracking vehicles with respect to the ego-vehicle’s lane provides a level of context unseen in prior works dealing with on-road lane tracking and vehicle tracking. Locating other vehicles on the road with respect to the ego-vehicle’s lane introduces a variety of new research directions for on-road vision systems. While prior studies are able to localize other vehicles using relative distance [259], the ability to localize vehicles’ lane positions are attractive for a number of reasons. Tracked vehicles’ lane departures and lane changes can be identified and monitored for the ego-vehicle’s own safety. This information can be used for both short-term and long-term trajectory prediction [2].

To track vehicles with respect to the ego-vehicle’s lane, we have extended the state vector to accommodate measurements relative to lane placement. A given vehicle’s state vector \mathbf{v}_t consists of the parameters given in equation 5.9. The parameters $[i_t, j_t, w_t, h_t]$ parametrize the bounding box of a tracked vehicle in the image plane. The parameters $[\Delta i_t, \Delta j_t]$ represent the change in i_t, j_t from frame to frame. The parameter ξ represents the lane position of a tracked vehicle.

The lane parameter takes the discrete values given in equation 5.9. The lane value -1 corresponds to the left of the ego lane. The lane value 0 corresponds to vehicle located in the ego lane. The lane value 1 corresponds to locations right of the ego lane.

$$\mathbf{v}_t = \begin{bmatrix} i_t & j_t & w_t & h_t & \Delta i_t & \Delta j_t & \xi_t \end{bmatrix}^T, \quad (5.9)$$

$$\xi_t \in \{-1, 0, 1\}$$

In a given frame, the observation z_t consists of a vector $\begin{bmatrix} \hat{i}_t & \hat{j}_t & \hat{w}_t & \hat{h}_t \end{bmatrix}^T$, corresponding to a parametrization of the bounding box of a detected vehicle. The particles are then confidence-weighted, and propagated for the next time instant.

A vehicle’s lane location in a given frame is inferred in three steps. First we compute the center of the vehicle’s bottom edge, which lies on the ground plane, using equation 5.10, and represent it using homogeneous coordinates. We then use equation 5.3 to project the vehicle’s ground-plane location into the ground-plane image.

$$\mathbf{p}_{\text{bottom}} = \begin{bmatrix} i_t + \frac{1}{2}w_t, & j_t + h_t, & 1 \end{bmatrix}^T \quad (5.10)$$

Finally, the vehicle’s lane location is inferred by comparing the i coordinate of the mapped point on the ground plane to the tracked lateral positions of the left lane and right lanes, using equation 5.5.

In practice, the assumptions made in this section utilizing the ground plane, and bottom edges of tracked vehicles, work quite well. Relying on the geometric structure of the traffic scene, and integrating tracking information from two modalities, we are able to infer a richness of information that is unavailable by simply tracking lanes and tracking vehicles separately.

5.5 Experimental Validation and Evaluation

We quantify the contribution of the proposed framework with three classes of experimental validation. For validation, we use the *LISAQ_2010* dataset, which will be made publicly available for academics and researchers at <http://cvrr.ucsd.edu/LISA/datasets>. Captured on a San Diego, California highway in June the dataset features typical rush hour traffic of moderate density at the beginning, progressing to extremely dense traffic at the end. The sequence contains typical dynamic traffic scenarios, its difficulty compounded by extensive glare from the sun. The dataset features 5000 consecutive frames, captured at 30 frames/second, over a distance of roughly 5km. Selected CANbus parameters over the sequence are plotted in Figure 5.7, which features decreased vehicle speed and increased braking frequency as the traffic becomes more dense.

On this dataset, we have conducted three sets of experimental validation. In the first set, we quantify the improvement in lane tracking performance in dense traffic by using integrated lane and vehicle tracking. In the second set, we quantify the improvement in vehicle tracking performance. In the third set, we quantify the performance of vehicle tracking with respect to the

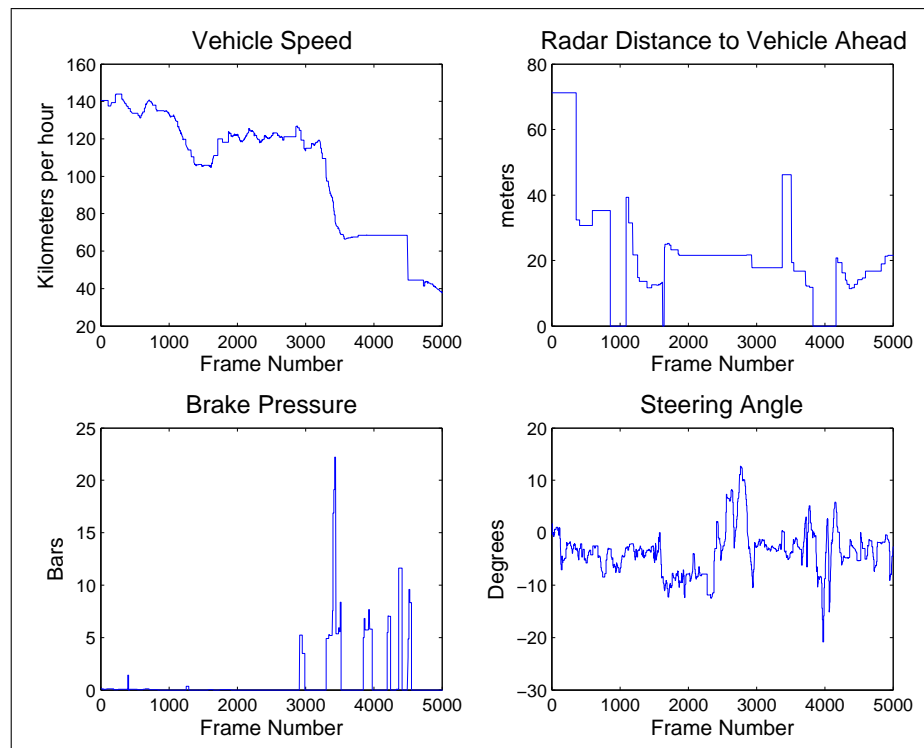


Figure 5.7: Selected parameters from the CANbus over the 5000 frame sequence. Note how the vehicle's speed decreases, and driver's braking increases, as the segment progresses, coinciding with increasing traffic density.

ego-lane, over 1000 particularly dynamic frames. During this segment, there are 2970 vehicles to detect.

5.5.1 Lane Tracking Performance

Table 5.2: Lane Localization Results

Lane Tracking System	Mean Absolute Error, Left Lane Marker (<i>cm</i>)	Mean Absolute Error, Right Lane Marker (<i>cm</i>)	Standard Deviation of Error, Left Lane Marker (<i>cm</i>)	Standard Deviation of Error, Right Lane Marker (<i>cm</i>)
Lane Tracking Alone	43.3	45.7	56.9	72.6
Integrated Lane and Vehicle Tracking	16.3	11.7	22.0	22.5

For experimental validation, we use commonly used performance metrics of absolute error, and standard deviation of error. As the sequence progresses, the traffic becomes more dense. Ground truth was hand-labeled on a separate ground-truth lane video. The lane tracker estimates the lane 40m ahead.

Figure 5.8 plots the localization estimates of the lane tracker, the integrated lane and vehicle tracking system, and ground truth on the same axis, over the entire 5000 frame sequence. While for most of the sequence, the two lane tracking systems match each other and the ground truth, after Frame 4000, we see a clear difference between the two systems. It is here that we observe the large change in lane localization error due to changes in traffic density.

During the sequence, we observe many dynamic maneuvers and conditions typical of the on-road environment. These include lane changes of the ego-vehicle, lane changes of other vehicles, and severe changes in road pitch due to bumps and uneven patches on the road. Figures 5.9(a) and 5.9(b) show a sequence that typifies the dynamic nature of the on-road environment, including two lane changes. In sparse traffic, both lane tracking systems perform quite well. During this sequence there is a spike in the lane estimation error, around Frame 2550. This is due to a large bump in the road, which causes a rapid change in road pitch. Figure 5.11(c) shows frames from the 1 second span during which this occurs.

We observe a consistent difference in robustness to dense traffic between the lane localization performance of the two systems, due to the integration of vehicle tracking. An example can be seen around Frame 4050. We observe a spike in localization error of the integrated lane and vehicle tracking system around frame 4050. This is due to missed detection of the vehicle in the adjacent lane over a few frames. Erroneous lane markings that correspond to the vehicle have been integrated into the lane measurement, which results in impaired lane localization, as shown in Figure 5.10(a).

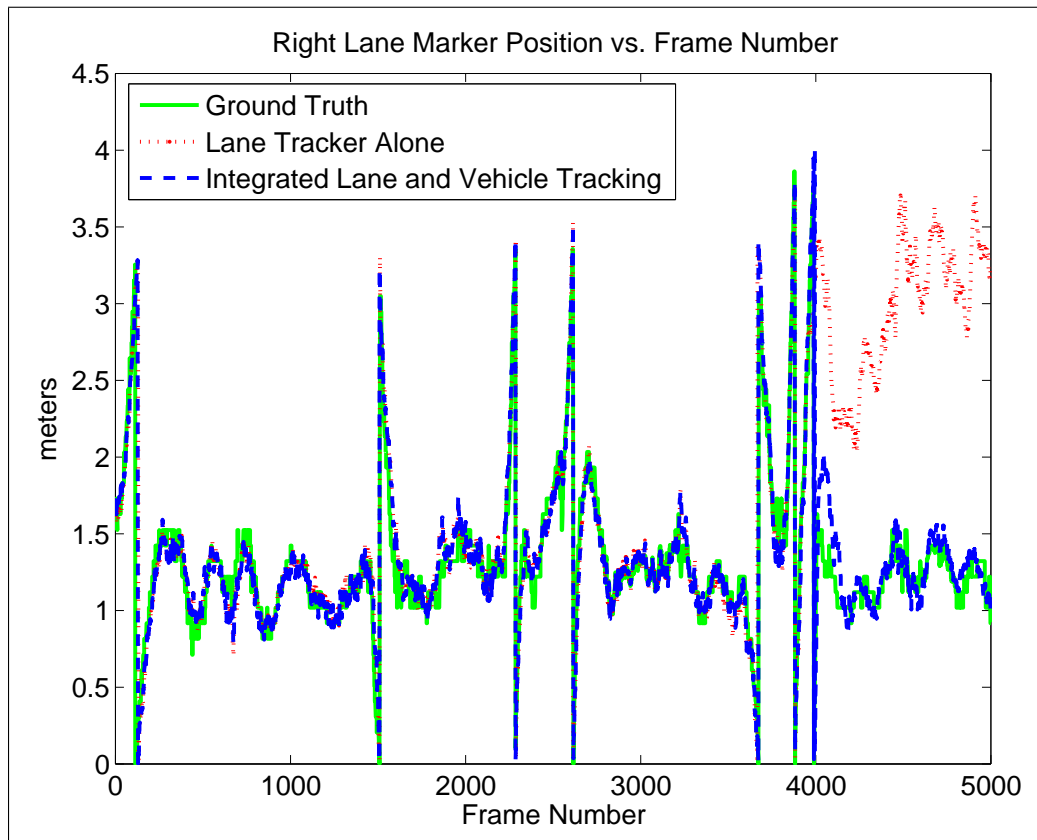
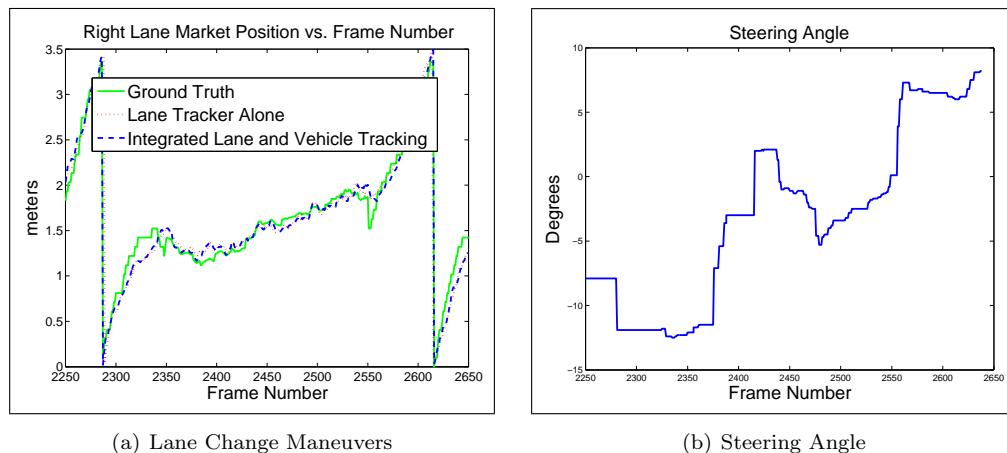


Figure 5.8: Estimated position of the right lane marker vs. frame number. The ground truth is shown in green. The estimated position using just lane tracker is shown in red. The result of the integrated lane and vehicle tracking system is shown in blue. Note that for the last 1000 frames, the lane tracker alone loses track of the lane position, due to high density traffic and a tunnel.



(a) Lane Change Maneuvers

(b) Steering Angle

Figure 5.9: Lane change maneuvers. In low-density traffic, both stand-alone lane tracking and integrated lane and vehicle tracking perform equally well. a) Lane Tracking outputs, including two lane changes b) Steering angle during this segment.



Figure 5.10: a) Poor lane localization due to a missed vehicle detection. The missed vehicle detection leads the lane tracker to integrate erroneous lane markings into the measurements, resulting in worse lane estimation for the right marker. b) Example misclassification of lane position. The jeep in the right lane [green] has been classified as in the ego lane. This is due to the fact that the jeep is farther ahead than the lane tracker’s look-ahead distance.

Figure 5.12(a) plots the absolute localization error as a function of time, after frame 4000. It is of note that as the traffic becomes more dense, the stand-alone lane tracker has difficulty. The reason for the large localization error between frames 4000 and 5000 is that there is a lane change before frame 4000 that the stand-alone lane tracker missed. After the missed lane change, the lane tracker’s estimation does not converge back to the true value for the rest of the sequence, due to the high density of vehicles on the road. Vehicles occlude lane boundaries and elicit false positive lane markings, which corrupt the system’s measurements. In the absence of dense traffic, after a missed lane departure, the lane tracker’s readings would quickly converge to ground. The integrated lane and vehicle tracking system, by contrast, does not miss this lane change, and is able to localize and track lane positions despite the dense traffic. Figures 5.12(b) and 5.12(c) show example lane tracking results in dense traffic.

Table 5.2 shows the mean absolute error and standard deviation of error over the entire 5000 frame dataset, for the lane tracker alone, and for the integrated lane and vehicle tracking system. Utilizing integrated lane and vehicle tracking significantly improves the localization error of lane tracking in dense traffic, resulting in better performance over the entire sequence. It is of note that the main differences in system performance are observed towards the end, in dense traffic. Table 5.3 shows the mean absolute error and standard deviation of error, over the last 1000 frames.

The experimental values for the integrating vehicle tracking in table 5.2 show a significant increase in robustness to the dynamic on-road conditions presented by dense traffic. The lane tracking results for Integrated Lane and Vehicle tracking are similar to those obtained in [215], and other lane tracking works in the field. Integrated Lane and Vehicle tracking adds a quantifiable level of robustness to lane estimation performance in dense traffic scenarios.

Table 5.3: Lane Localization Results, Last 1000 Frames

Lane Tracking System	Mean Absolute Error, Right Marker (<i>cm</i>)	Standard Deviation of Error, Right Lane Marker (<i>cm</i>)
Lane Tracking Alone	177.5	37.6
Integrated Lane and Vehicle Tracking	14.5	13.5

5.5.2 Vehicle Tracking Performance

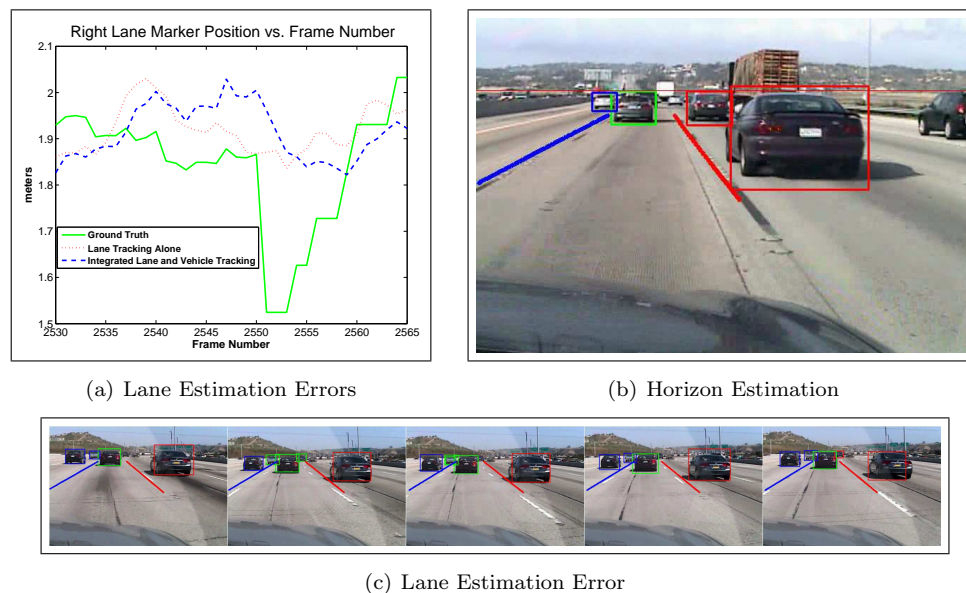


Figure 5.11: a) We note a spike in the ground truth, and corresponding error around Frame 2550. Large estimation error due to rapid, severe variation in road pitch, due to a large bump in the road, which severely alters the pitch for a very short period, less than a second. The ego vehicle was traveling at 35 meters per second. b) Selected frames from this one-second span. The beginning and end frames show normal lane estimation. The middle frames show lane estimation errors due to the bump in the road. c) Horizon estimation using lane estimation. The red line is the estimated horizon.

We evaluate the performance of the vehicle tracker, utilizing lane information, on 1000 frames of the full sequence. This sequence of 1000 frames was chosen because of its level of traffic density. The beginning of the sequence has medium density traffic, and progresses to heavily dense traffic towards the end. During this sequence there are 2790 vehicles on the road to detect and track. The sequence begins with frame 2900, typifies dynamic traffic scenarios, featuring rapid changes in traffic density. The same sequence is used in the following subsection for localizing

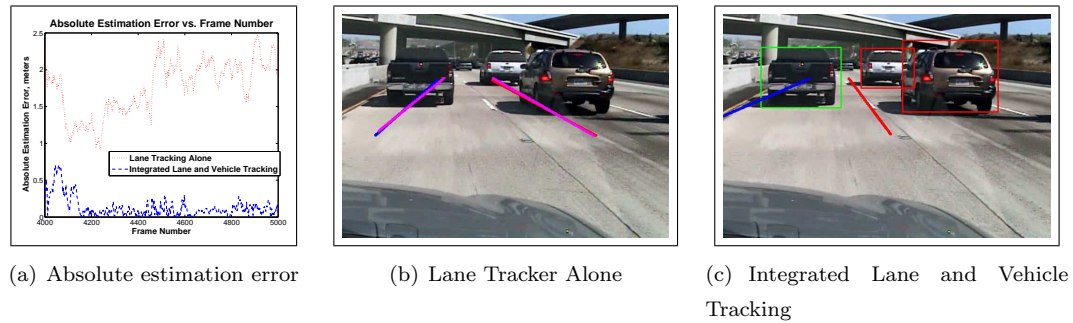


Figure 5.12: a) Right lane marker estimation error. b) Lane tracking in dense traffic, frame 4271, The pink lines indicate estimated lane positions. Note the large estimation error due to the presence of vehicles in dense traffic. c) Integrated Lane and Vehicle Tracking in dense traffic, frame 4271. The red and blue lines indicate estimated lane positions. Note the tracked vehicles and accurate lane estimation.

tracked vehicles with respect to lanes. Figure 5.13(b) plots the estimated lane position during this 1000-frame sequence. Note the lane changes towards the end of the sequence, in dense traffic.

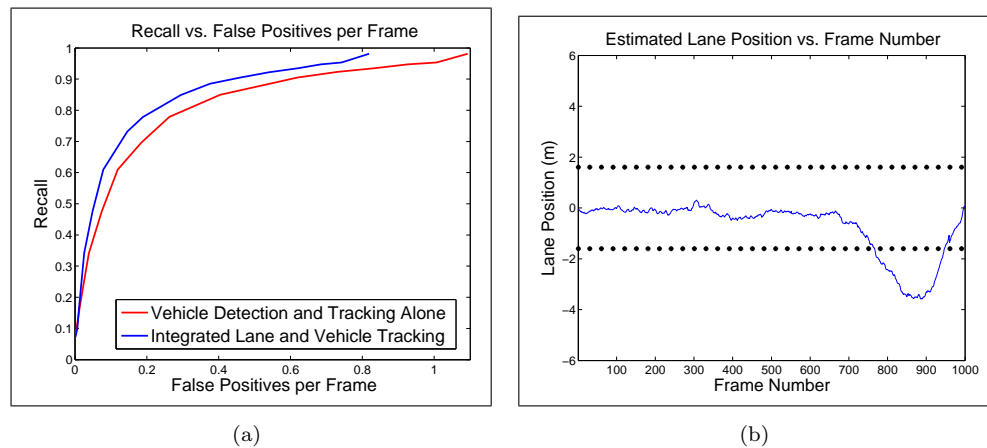


Figure 5.13: 5.13(a) Recall vs. False Positives per Frame, comparing vehicle detection and tracking alone [3], and integrated lane and vehicle tracking. Performance is evaluated over a 1000-frame sequence, which features 2790 vehicles. While both systems perform quite well over the dataset, Integrated Lane and Vehicle Tracking has better performance in terms of false positives per frame. 5.13(b) Estimated lane position during vehicle localization validation sequence, integrated lane and vehicle tracking. Note the two lane changes towards the end of the sequence.

Figure 5.13(a) plots the recall versus false positives per frame over the sequence for the vehicle tracking system introduced in [3], and for the Integrated Lane and Vehicle tracking system introduced in this research study. It is shown in [3] that the system exhibits robust performance in dynamic traffic scenes. Over this sequence, this system performs quite well, and its evaluation is plotted in red.

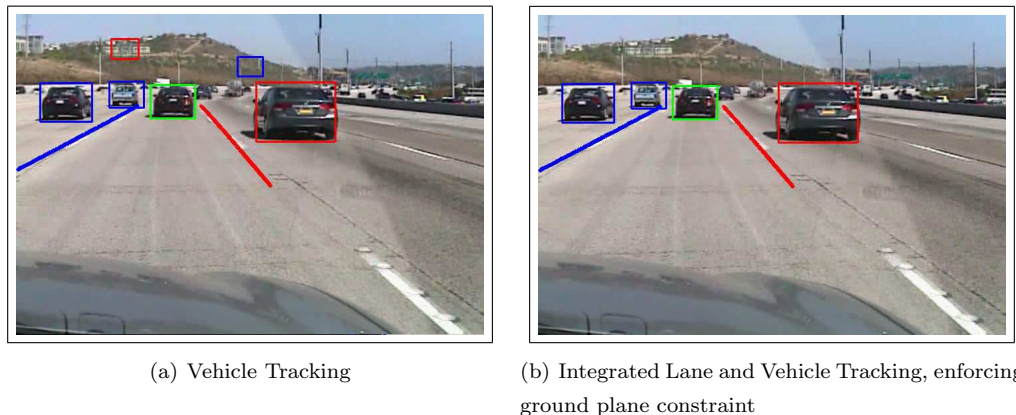


Figure 5.14: a) Buildings off the road result in false positives. b) By enforcing the constraint that tracked vehicles must lie on the ground plane, the false positives are filtered out.

Table 5.4: Performance Comparison, Low False Positives per Frame

Recall			
False Positives per Frame	Vehicle Detection and Tracking Alone	Integrated Lane and Vehicle Tracking	
0.08	0.48	0.61	
0.15	0.65	0.73	
0.2	0.70	0.79	

Table 5.5: Performance Comparison, High Recall

False Positives per Frame		
Recall	Vehicle Detection and Tracking Alone	Integrated Lane and Vehicle Tracking
0.905	0.62	0.46
0.92	0.73	0.54
0.95	1.0	0.74

Tables 5.4 and 5.5 compare the recall and false positives per frame at specific operating points. We can see that at very low false positives per frame, integrated lane and vehicle tracking attains roughly 9% improved recall. At 90.5% recall, we observe that integrated lane and vehicle tracking produces 0.16 fewer false positives per frame.

The vehicle tracking performance of the Integrated Lane and Vehicle Tracking system is plotted in blue in figure 5.13(a). Including knowledge of where the ground plane lies effectively filters out potential false positives, as evidenced by the recall-false positives per frame curve. It can be seen that while both systems perform quite well over the dataset, that Integrated Lane and Vehicle Tracking offers improvement in false positive rates.

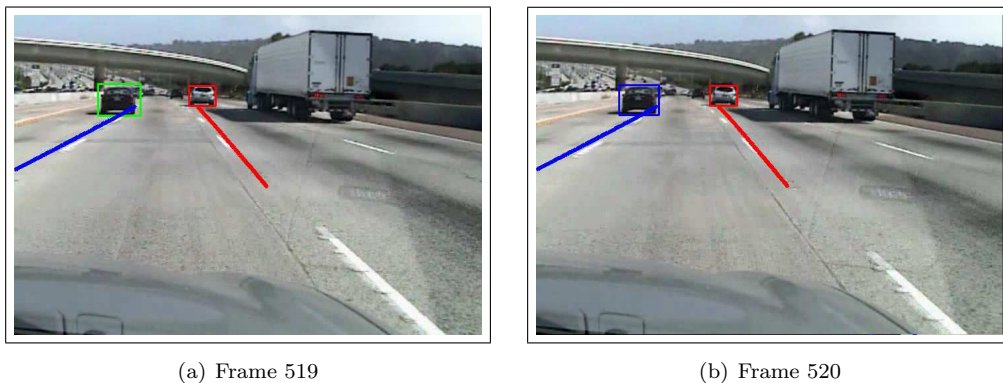


Figure 5.15: Ambiguities in lane/vehicle positions. The vehicle on the left is in the midst of a lane change. a) The vehicle is determined to still be in the ego-lane. b) The vehicle is determined to have changed lanes in to the left lane.

Figures 5.14(a) and 5.14(b) show an example frame where false positives have been filtered out by enforcing the ground plane. In figure 5.14(a) there are two false positives elicited by buildings off in the distance that lie on a hill. Figure 5.14(b) shows the result of enforcing the ground plane constraint on tracked objects.

5.5.3 Localizing Vehicles with Respect to Lanes

Over the 1000 frames detailed in the previous subsection, we evaluate the performance of the system localizing tracked vehicles with respect to the ego vehicle’s lane position. For this evaluation, there are three classes of vehicles, corresponding to their lateral position on the road. Vehicles are classified as *Left* if their inferred position is left of the ego vehicle’s lane. Correspondingly, the lane parameter of their state vector, given in equation 5.9 takes the value -1 . Vehicles determined to be in the ego vehicle’s lane are classified *Ego-lane*, have the lane parameter of the state vector set to 0. Vehicles to the right of the ego lane are classified as *Right*, and have lane parameter 1. Figures 5.15(a) and 5.15(b) show the lane change of a tracked vehicle. Figure 5.16(a)-5.16(c) show an ego lane change, and its implications for localizing other

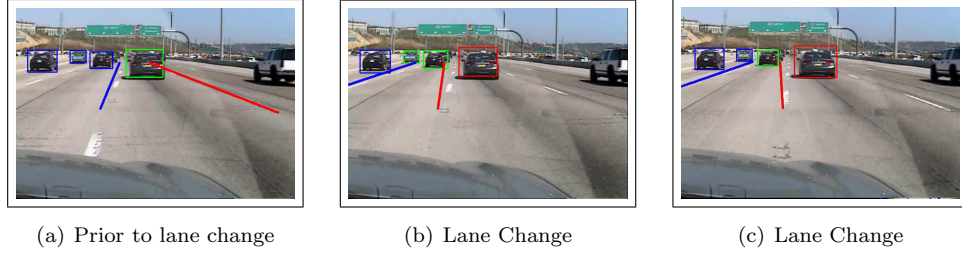


Figure 5.16: Illustrating lane-level localization of vehicles during an ego lane change. a) Frame 2287, immediately prior to lane change. b) Lane Change. We note that the truck on the left has been incorrectly assigned to the ego-lane. c) The truck on the left has been correctly assigned to the left lane, a few frames later.

vehicles with respect to the ego lane.

Table 5.6: Confusion Matrix of Tracked Vehicle Lane Assignments

True Lane Position	Classified As			Ground Truth Distribution
	Left	Ego-Lane	Right	
Left	99.0%	1.0%	0	30.4%
Ego-Lane	11.3%	88.6%	0	26.7%
Right	0	7.9%	92.1%	42.9%
Overall Accuracy	93.2%			

Table 5.6 shows a confusion matrix of vehicle tracking results with respect to the lanes. We note that in general, the lane-based tracking is quite accurate. Overall, we report 93.2% localization accuracy over the 1000 frame sequence. During this sequence, there are a total of 2790 vehicles to be tracked with respect to lanes.

We note some asymmetry in the classification results. While it is to be expected that there will be confusion between ego-lane vehicles and those in adjacent lanes, it appears in 5.6 that the left lane classification performs quite a bit better than right lane classification, which perform relatively similarly to each other. The last column of Table 5.6 shows the distribution of vehicles per lane in the ground truth set, which shows that in the dataset, many more vehicles are encountered in the right lane than in the left lane. This explains the asymmetry in results.

In general the range of the vehicle tracking system is greater than that of the lane tracker. This means that vehicles can be tracked farther away from the ego vehicle than lane

Table 5.7: Processing Time for Vehicle, Lane, and Integrated Systems

Tracking System	Processing Time per 704×408 Frame (ms)
Vehicle Detection and Tracking	33.1 ms
Lane Tracking	74.1 ms
Integrated Lane and Vehicle Tracking	90.1 ms

markings and positions. Consequently, for vehicles that are very far away, we are inferring their lane position based on the tracked lane positions much closer to the ego vehicle. Figure 5.10(b) shows an example of this phenomenon. While the lane positions have been accurately tracked, and the vehicles accurately tracked, there is a tracked vehicle quite far away, whose lane position is incorrectly inferred.

Other sources of error stem from ambiguities regarding a given vehicle’s lane position. When a vehicle is changing lanes, it is difficult to definitively determine which lane the vehicle is in. Figures 5.15(a) and 5.15(b) depict this phenomenon. The vehicle on the left is changing lanes from the ego lane to the left lane. In addition, the system can have difficulty assigning lanes during the ego-vehicle’s lane change maneuvers. Figures 5.16(b) and 5.16(c) depict this phenomenon.

5.5.4 Processing Time

We assess the additional computational load required to run the integrated lane and vehicle tracking, and compare it to the processing times required for the stand-alone lane tracker, and stand-alone vehicle tracker. While efforts have been made to pursue efficient implementation, neither code nor hardware are optimized. Table 5.7 provides the processing time per 704×408 video frame in milliseconds, for each of the respective systems. The system is executed on a Pentium i7 2.4GHz architecture.

The vehicle detector and tracking system requires 33.1 ms to process a single frame, running at real-time speeds of a little over 30 frames per second. The lane tracking system takes 74.1 ms to process a frame, running at 13.5 frames per second. Integrated lane and vehicle tracking takes 90.1 ms to process a single frame, running at roughly 11 frames per second, somewhat less than the sum of the times required for the vehicle and lane tracking systems separately. This speed is near-real-time.

5.6 Remarks

The synergistic approach introduced in this paper achieves three main goals. First, we have improved the performance of lane tracking system, and extended its robustness to high density traffic scenarios. Second, we have improved the precision of the vehicle tracking system,

by enforcing geometric constraints on detected objects, derived from the estimated ground plane. Thirdly, we have introduced a novel approach to localizing and tracking other vehicles on the road with respect to the estimated lanes. The lane-level localization adds contextual relevance to vehicle and lane tracking information, which are valuable additions to human-centered driver assistance. The fully implemented integrated lane and vehicle tracking system currently runs at 11 frames per second, using a frame resolution of 704×480 . Future work will involve extensions to urban driving [68], as well as expansion of the contextual tracking, learning long-term trajectory and behavioral patterns [2].

5.7 Acknowledgments

Chapter 5 is a partial reprint of material published in *IEEE Transactions on Intelligent Transportation Systems*, 2013. The dissertation author was the primary investigator and author of these papers.

Chapter 6

Vehicle Detection by Independent Parts for Urban Driver Assistance

6.1 Introduction

In the United States, urban automotive collisions account for some 43% of fatal crashes. Over the past decade, while the incidence of rural and highway accidents in the United States has slowly decreased, the incidence of urban accidents has increased by 9%. Tens of thousands of drivers and passengers die on the roads each year, with most fatal crashes involving more than one vehicle [14]. The research and development of advanced sensing, environmental perception, and intelligent driver assistance systems present an opportunity to help save lives and reduce the number of on-road fatalities. Over the past decade, there has been significant research effort dedicated to the development of intelligent driver assistance systems, intended to enhance safety by monitoring the driver and the on-road environment [243].

In particular, the on-road detection of vehicles has been a topic of great interest to researchers over the past decade [20]. A variety of sensing modalities have become available for on-road vehicle detection, including radar, lidar, and computer vision. As production vehicles begin to include on-board cameras for lane tracking and other purposes, it is advantageous and cost-effective to pursue vision as a modality for detecting vehicles on the road. Vehicle detection using computer vision is a challenging problem [20, 6]. Roads are dynamic environments, featuring effects of ego-motion and relative motion, video scenes featuring high variability in background and illumination conditions. Further, vehicles encountered on the road exhibit high variability in size, shape, color, make, and model.

The urban driving environment introduces further challenges [68]. In urban driving, frequent occlusions and a variety of vehicle orientations make vehicle detection difficult, while



Figure 6.1: The urban driving environment features oncoming, preceding, and sideview vehicles [top]. Additionally, vehicles appear partially-occluded as they enter and exit the camera's field of view [bottom].

Table 6.1: Vision-based Vehicle Detection

Research Study	On-Road Environment	Vehicle Views	Partial Occlusions	Part-Based	Real-time
Chang and Cho, 2010 [83]	Highway	Rear	No	No	Yes
O'Malley et al., 2010 [5]	Night-time highway	Rear	No	No	Yes
Sivaraman and Trivedi, 2010, 2011 [3, 11]	Highway	Rear	No	No	Yes
Jazayeri et al., 2011 [6]	Highway	Rear	No	No	Yes
Lin et al., 2011 [39]	Highway	Multiple views, as seen in blind spot	No	No	No
Niknejad et al., 2012 [4]	Urban	Front, Rear, Side	No	Yes	No
Rubio et al., 2012 [107]	Night-time highway	Front, Rear	No	No	Yes
Vehicle Detection by Independent Parts, 2013	Urban	Front, rear, side	Yes	Yes	Yes

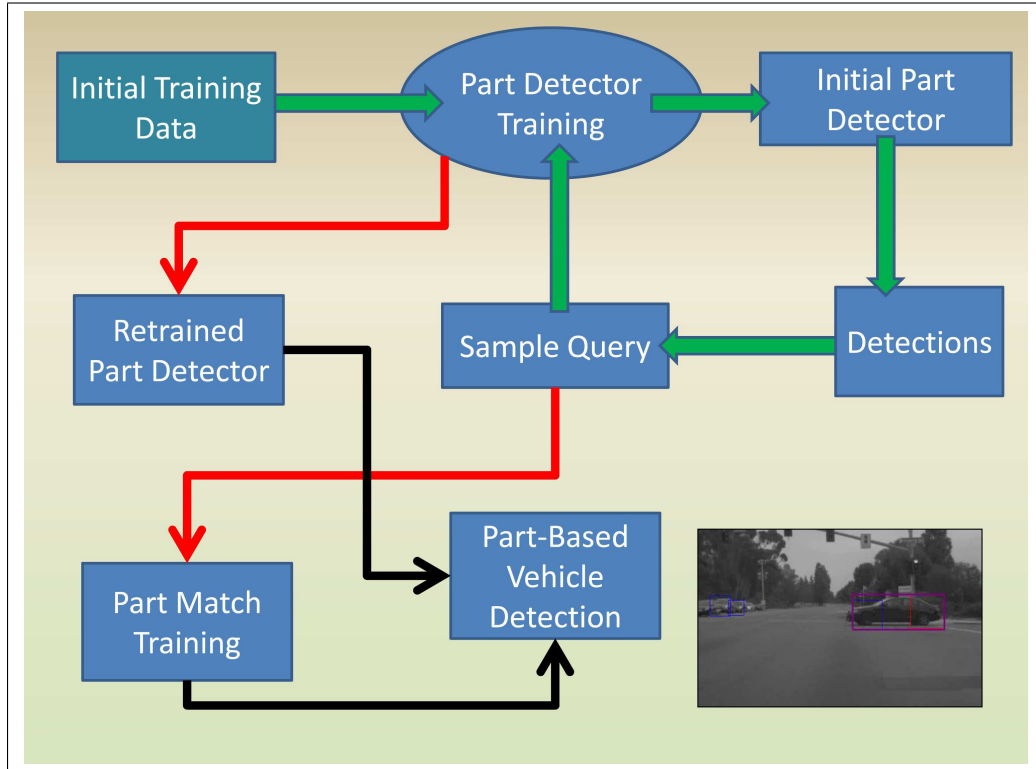


Figure 6.2: Vehicle Detection by Independent Parts [VDIP]. The learning approach detailed in this study. An initial round of supervised learning is carried out to yield initial part detectors, for the front and rear parts of vehicles. We use the initial detectors to query informative training examples from independent data, performing active learning to improve part detection performance. While we query informative training examples, we label side-view vehicles in a semi-supervised manner, using the active learning annotations to form fully-visible vehicles.

visual clutter tends to increase the false positive rate [174]. Fully-visible vehicles are viewed in a variety of orientations, including oncoming, preceding, and side-view. Cross traffic is subject to frequent partial occlusions, especially upon entry and exit from the camera’s field of view. Figure 6.1 illustrates this concept. Many studies in on-road vehicle detection have detected fully-visible vehicles. In this study, we also detect and track partially occluded vehicles.

In this study, we introduce Vehicle Detection by Independent Parts [VDIP]. Vehicle part detectors are trained using active learning, wherein initial part detectors are used to query informative training examples from unlabeled on-road data. The queried examples are used for retraining, in order to improve the performance of the part detectors. Training examples for the part matching classifier are collected using semi-supervised annotation, performed during the active learning sample query process. After retraining, the vehicle part detectors are able to detect oncoming, preceding vehicles, and front and rear parts of cross-traffic vehicles. The semi-supervised labeled configurations are used to train a part-matching classifier for detecting full side-view vehicles. Vehicles and vehicle parts are tracked using Kalman filtering. The final system

is able to detect and track oncoming, preceding, side-view, and partially-occluded vehicles. The full system is lightweight, robust, and runs in real time. Figure 6.2 depicts the learning process for vehicle detection by independent parts.

The main contributions of this study include the following. We introduce a novel approach for detecting and tracking vehicles by independent parts [VDIP], yielding a system capable of detecting vehicles from multiple views, including partially-occluded vehicles. Leveraging active learning for improved part detection, and semi-supervised labeling for training part matching classification, we implement a full vehicle detection and tracking system tailored to the challenges of urban driving. Vehicle parts and vehicles are tracked in the image plane using Kalman filtering. The full vehicle VDIP system runs in real time. Extensive quantitative analysis is provided.

Many prior works in vehicle detection require a root-filter as part of detection, which can limit applicability to partial occlusions [12, 4]. Prior works that detect independent parts, are focused on surveillance [260] or static image applications [261]. Detection by independent parts has rarely been implemented from a moving platform, and no prior work has used detection and tracking by independent parts for on-road vehicle detection. Further, no prior works have utilized active learning for part detection, or semi-supervised learning for part matching. Few reported part-based object detection systems report real-time implementation.

The remainder of this paper consists of the following. Section 2 briefly describes related research. Section 3 describes overall approach to vehicle detection by parts, including active learning for part detection, semi-supervised learning of part classification, and on-road part and vehicle tracking. Section 4 presents experimental evaluation. Section 5 offers discussion and concluding remarks.

6.2 Related Research

This study involves on-road vehicle detection, on-road vehicle tracking, and detection by parts. In this section, we provide a brief review of recent studies in these research areas.

6.2.1 On-road Vehicle Detection and Tracking

Robust detection of other vehicles on the road using vision is a challenging problem. Driving environments are visually dynamic, and feature diverse background and illumination conditions. The ego vehicle and the other vehicles on the road are generally in motion. The sizes and locations of vehicles in the image plane are diverse, although they can be modeled [6]. Vehicles also exhibit high variability in their shape, size, color, and appearance [20]. Further, while processing power has increased significantly over the past decade, the requirements for real-time computation impose additional constraints on the development of vision-based vehicle detection systems.

Many works in vehicle detection over the past decade have focused on detection and tracking of the rear face of vehicles [20, 3, 11, 6]. Detectors of this sort are designed for preceding, or sometimes oncoming traffic. In [6], feature points are tracked over a long period of time, and vehicles detected based on the tracked feature points. The distribution of vehicles in the image plane is modeled probabilistically, and a hidden Markov model formulation is used to detect vehicles and separate them from the background.

In [39], a camera is mounted on the vehicle to monitor the blind spot area, which presents difficulty because of the field of view, and high variability in the appearance of vehicles, depending on their relative positions. The study uses a combination of SURF features, and edge segments. Classification is performed via probabilistic modeling, using a Gaussian-weighted voting procedure to find the best configuration.

Detecting side-profile vehicles using a combination of parts has been explored in [42]. Using a camera looking out the side passenger’s window, vehicles in adjacent lanes are detected by first detecting the front wheel, and then the rear wheel. The combined parts are tracked using Kalman filtering. In [68], the idea of detection vehicles as a combination of independent parts is explored, with comparison of geometric and appearance matching features. This study expands upon [68] with an augmented feature set, tracking formulation, and extensive experimental evaluation.

Vehicle detection using the deformable part-based model introduced in [12] has been implemented in [4]. The study implemented particle filter tracking, including adaptive thresholds for the detectors, to deal with the challenging conditions presented in the on-road environment.

On-road vehicle tracking has mainly been implemented using Kalman filtering [262, 118, 28], or particle filtering [4, 3]. Tracking has been carried out in the image plane [3, 6, 4] or in 3D coordinates [262, 118, 8, 28] using stereo-vision. In stereo-vision studies, optical flow is often used as the initial cue to track moving objects. Table 6.1 summarizes recent works in vision-based vehicle detection.

6.2.2 Part-based Object Detection

Detecting objects by parts has been explored in the computer vision community in various incarnations, with many works focused on detection of people. In [260], individual parts are detected using strong classifiers, and pedestrians are constructed using a Bayesian combination of parts. The number of detection pedestrians, and their locations, are the most likely part-based configuration. In [263], a more efficient feature representation is used, and parts are chosen to be semantically meaningful and overlapping in the image plane.

In [264], multiple instance learning is used for part-based pedestrian detection. The study demonstrates how the ability of multiple instance feature learning to deal with training set misalignment, can enhance performance of part-based object detectors. In [265], people are

detected using covariance descriptors on Riemannian manifolds, classified using a Logitboost cascade. In [261], pedestrian parts are manually assigned to semantically meaningful parts, their physical configuration manually constrained and overlapping. The combination of parts is a weighted sum, with higher weights going to more-reliably detected parts. In [266], this work is extended, with further experiments on the feature set.

The work of [12] for object detection using a deformable, parts-based model [DPM], based on the Latent Support Vector Machine has introduced new avenues for detection of vehicles by parts. An efficient cascade classifier version of the deformable part-based object detector is presented in [86]. While [12] demonstrates vehicle detection evaluated on static images, DPM is used for video-based nighttime vehicle detection in [88]. Integrated with tracking, vehicle detection by parts using DPM is presented in [4].






Specific classifiers trained for detecting occluded pedestrians has been pursued in [267]. Using training examples featuring occluded pedestrians, a classifier was training using the monocular, optical flow, and stereo modalities. In [71], partially-occluded rear faces of vehicles are detected using SIFT features and Hidden Random Field detection.

6.3 Vehicle Detection by Independent Parts

Vehicle detection by independent parts includes the following steps. An on-road video frame is grabbed, and front and rear detectors are applied. The front part detector have been trained to detect oncoming vehicles, and the front parts of side-view vehicles. The rear part detector has been trained to detect preceding vehicles, and the rear parts of side-view vehicles. A part matching classifier is applied to detect full side-view vehicles. Vehicle parts, and full vehicles are tracked in the image plane using Kalman Filtering. Figure 6.3 depicts the vehicle detection by parts process. Table 6.2 defines the terminology we use to describe vehicles and vehicle parts. In the following subsections, we describe active learning for detecting vehicle parts, semi-supervised labeling for vehicle detection by parts, and vehicle tracking using Kalman filtering.

In this work, we use a single detector for front parts, and a single detector for rear parts. The parts can be facing left, facing right, or any orientation in-between. We use a general detector, instead of several orientation-specific classifiers for a few reasons. First, training a general detector makes more efficient use of the available data. If N training examples are required to train a single classifier, then training k orientation-specific classifiers will require kN annotated data samples. Second, using a general detector is computationally more efficient when processing a frame, versus evaluating k orientation-specific classifiers. The third reason is robustness. The on-road environment is challenging, featuring ample visual clutter. Localizing the front or rear part of the vehicle in difficult visual clutter, can be easier than detecting a narrow part orientation in the same clutter. The implications of object detection and object

Table 6.2: Taxonomy of On-Road Vehicle Detection

Vehicle View	Definition	Example Image
Fully-Visible		
Fully-visible vehicle	A vehicle whose full outline is visible and within the camera's field of view.	
Oncoming	The front face of a vehicle traveling parallel to the ego vehicle, in the opposite direction.	
Preceding	The rear face of vehicle traveling parallel to the ego vehicle, in the same direction.	
Side-view	A vehicle traveling roughly perpendicular to the ego vehicle.	
Partially-Occluded		
Partially-Occluded Vehicle	A vehicle that is not fully-visible, due to occlusion by another object or vehicle, or due to entry/exit from the camera's field of view.	
Front Part	The portion of side-view vehicle including the front bumper and front wheel.	
Rear Part	The portion of a side-view vehicle including the rear bumper and the rear wheel.	

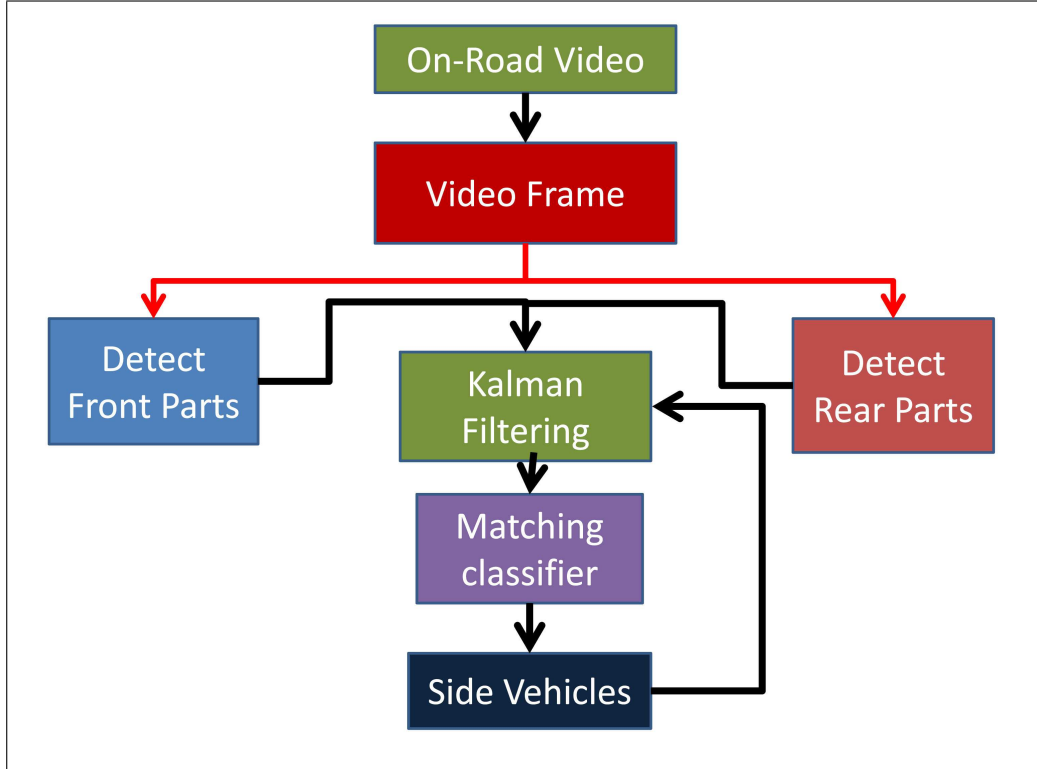


Figure 6.3: Vehicle Detection by Independent Parts [VDIP]: Data information flow for real-time vehicle detection and tracking by parts.

orientation in clutter are examined at length in [268], where a generic detector is used to detect objects regardless of orientation, and then orientation-specific classifiers are used to determine the object’s orientation.

6.3.1 Active Learning for Detecting Independent Parts

The on-road environment presents myriad challenges for vision-based vehicle detection. The backgrounds, illumination conditions, occlusions, visual clutter, and target class variability all present difficulties to vehicle detection using cameras and computer vision. Detecting vehicle parts can be even more challenging. While recent studies in part-based object detection have used a root filter as the prior information for searching for parts [12, 4], in this study we pursue independent part detection. The goal is to detect vehicle parts independently of a root model, in order to detect vehicles in the presence of partial occlusions. Figure 6.4 depicts this phenomenon. We want to detect the front part of the vehicle as it enters the camera’s field of view, while the vehicle remains partially occluded.

We employ active learning for training vehicle part detectors. Active learning takes into account the fact that unlabeled training data is abundant, while labeled data comes with some

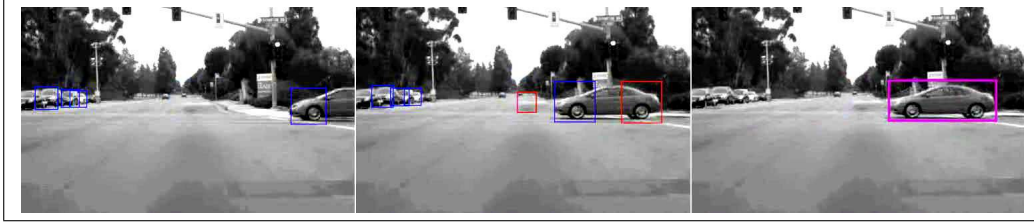


Figure 6.4: VDIP illustrative example. As a vehicle enters the camera’s field of view, its front part is detected [blue]. As it becomes fully-visible, its rear part is detected too [red]. Parts are tracked, and a part matching classifier is applied, to detect fully-visible side-view vehicles [purple].

cost, namely human effort and data volume [269]. Active learning can enable more efficient learning for vision-based object detection. Active learning has been used for object detection to improve recall-precision performance, train with fewer samples, and train with less human labeling effort than conventional supervised learning [25].

We choose the front part, and the rear part of vehicles for part-based detection. These parts are semantically meaningful, and are often the first parts visible when a vehicle enters the camera’s field of view. The front and rear parts of the vehicle also have strong feature responses, making them suitable for training a detector. Table 6.2 details our criteria for the parts. The front part should detect oncoming vehicles, as well as the front parts of side-viewed vehicles. The rear part should detect preceding vehicles, as well as the rear parts of side-viewed vehicles. We use Haar-like features, and an Adaboost cascade for detecting the front and rear parts [54, 76]. The combination of Haar-like features and Adaboost cascade classification are chosen for their speed and utility for vehicle detection, as they have been widely used in the vehicle detection literature [3].

We train an initial classifier using 5000 labeled positive and negative examples for each part detector. Negative training examples were randomly selected from non-vehicle regions from on-road video. As demonstrated in prior works [3, 11], active learning can contribute to improved on-road vehicle detection. We use the initial classifiers for part detection to query informative examples, and retrain improved classifiers.

Each annotated frame presents informative samples for retraining the front and rear part detectors. We define informative training examples as those image patches resulting from misclassification, i.e. false positives and missed detections. We also archive true positives for retraining, to avoid overfitting [208]. During active learning sample query for front and rear part detectors, the interface allows the user to label side-view vehicles as a combination of labeled front and rear parts. Figure 6.5 shows the interface used for active learning of parts, and for semi-supervised labeling for part matching. Figure 6.5(e) shows the side-view vehicle in green, as a combination of front and rear parts, the blue and purple boxes contained within.

We retrain front and rear part detectors, using 5000 positive and 5000 negative training

examples for each, the training examples queried using active learning. The retrained part detectors exhibit improved performance, including improved detection rates, and lower false positive rates. During active learning, rear parts were included in the negative training set for front detection, and vice versa.

While the goal is to train independent part detectors for the front and rear vehicles, we note that even after active learning, part classification can be ambiguous, i.e. front detector sometimes returning rear parts, or rear detector sometimes returning front parts. Prior vehicle detection works have trained a single detector for oncoming and preceding vehicles [4, 59]. As such, discriminating between the two classes can be difficult. For detecting oncoming and preceding vehicles, we resolve this ambiguity over time via tracking, as detailed in Section III-C.

6.3.2 Semi-supervised Labeling for Part-Matching Classification

Given the initial part detections on a given frame, we take a semi-supervised approach to learning a part-matching classifier for detecting side-view vehicles by parts. Semi-supervised learning methods exploit the learner’s prior knowledge to deal with unlabeled data [270]. In this case, we obtain labels for parts from the initial part detectors, during the active learning labeling sample query. These labels are used to train a classifier to detect side-view vehicles by matching already-detected and labeled parts.

The part-matching classifier is based on geometric features that encode the spatial relationship between the parts that comprise a side-view vehicle. A given part can either be matched to form a side-profile vehicle, or retained as either an oncoming or preceding vehicle. We parametrize a rectangle p corresponding to a detected vehicle part, either front or rear, by the i, j position of the rectangle’s top-left corner in the image plane, its width, and its height, as shown in equation 6.1.

$$p = \begin{bmatrix} i & j & w & h \end{bmatrix}^T \quad (6.1)$$

We denote a detected front part as p_f , a given rear part as p_r . The side vehicle formed by p_f and p_r is denoted p_s , and is the minimal rectangle that contains the two parts. To match a pair of given parts, front to rear, we compute a set of geometric features, as shown in equation 6.2.

$$x(p_f, p_r) = \begin{bmatrix} \frac{|i_f - i_r|}{\sigma} & \frac{|j_f - j_r|}{\sigma} & \frac{h_f}{h_r} & \frac{w_s}{h_s} \end{bmatrix}^T \quad (6.2)$$

$$\sigma = \frac{w_f}{K}$$

The geometric feature vector encodes the relative displacements between a front part, p_f , and rear part, p_r , in the image plane, as well as their relative sizes, and the aspect ratio of the minimal spanning rectangle p_s . The parameter σ normalizes the distances to reference frame, scaled to the 24×24 size of image patches used in the training set. The parameters w_s and h_s

are the width and height of the minimal side rectangle that envelops detected part rectangles p_f and p_r in the image plane. Using the absolute value of horizontal and vertical distances in the image plane allows the computed features to serve for left-facing or right-facing side-profile vehicles.

$$\begin{aligned}
 w_f(Z) &= \frac{f}{Z} W_f \\
 \sigma &= \frac{w_f(Z)}{K} \\
 K &= \frac{w_f(Z)}{\sigma} = \frac{\frac{f}{Z} W_f}{\frac{1}{K} \frac{W_f}{Z}}
 \end{aligned} \tag{6.3}$$

The model is based on the premise that the relative dimensions of passenger vehicles fit a general physical model, which is observed in the image plane under perspective projection. Consider the distance to a vehicle Z , and a standard pinhole camera model with focal length f . Then the width of the vehicle scales with $\frac{f}{Z}$. The parameter σ encodes the ratio between the width of the vehicle under perspective projection, and is designed to scale with distance from the camera.

The model used is intended for detection of passenger vehicles, including sedans, coupes, station wagons, minivans, SUV's, pickup trucks, and light trucks. While the part classifiers can detect front and rear parts of other types of vehicles, like buses and semi trucks, the geometric features used in this study are intended for use with passenger vehicles. According to the Bureau of Traffic Statistics, passenger vehicles and light trucks comprised over 95% of vehicles on the road in 2010 [271]. The model is based

During the active learning sample query stage, we label true positives, false positives, and missed detections returned by the initial front and rear part detectors. We then label side-view vehicles, and compute the geometric features between pairs of parts that comprise side-view vehicles. To collect negative training examples, we compute the geometric features between the pairs of parts that do not comprise side-view vehicles. Figure 6.5 shows a screen-shot of the interface used for active learning, and semi-supervised labeling part configurations.

$$\begin{aligned}
 f(x) &= \sum_i \alpha_i K(x, x_i) + b \\
 f(x) &= w^T x + b \\
 y &= \text{sgn}(f(x))
 \end{aligned} \tag{6.4}$$

The part matching uses a Support Vector Machine classifier [22]. Equation 6.4 lists the equations for the SVM classification. We use a linear kernel for SVM classification. Using the primal form of the linear kernel, evaluation of the classifier becomes an inner product with a weight vector w , which enables a speed advantage. We train the matching classifier using 600 positive and 600

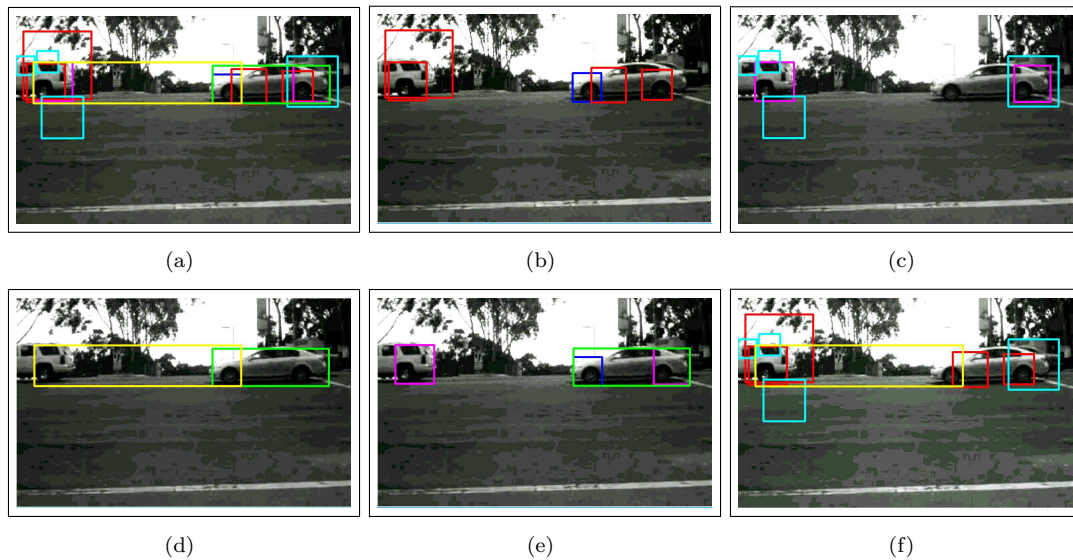


Figure 6.5: Active learning interface used for part detection sample query, and for semi-supervised labeling side-view vehicles for detection by parts. a) All training examples, positive and negative, collected from a single frame, for training part matching, as well as part detection. b) Active learning sample query for front parts, featuring true positives [blue] and false positives [red]. c) Active learning sample query for rear parts, featuring true positives [purple] and false positives [cyan]. d) Part-matching examples for side vehicle detection, positive [green] and negative yellow]. e) All positive training examples collected from this frame, for front part, rear part, and part matching. f) All negative training examples collected from this frame front part, rear part, and part matching.

negative training examples.

$$p(y = 1|x) = \frac{1}{1 + e^{Af(x)+B}} \quad (6.5)$$

For matching evaluation, we evaluate equation the part matching classifier over each front-rear part pair, computing the likelihood of a match using equation 6.5. We choose the best match for each pair of front and rear parts, using a variation of the stable marriage problem [272]. The scores for matching are based on $p(y = 1|x)$, whose parameters A and B have been learned using maximum likelihood [235, 197]. Only matches for which the score is above 0.5 are retained as full side-view vehicles.

Figure 6.7 shows an example of the matching process. In the first frame, the vehicle is entering the camera's field of view at an intersection. The front part of the vehicle is detected, labeled with a blue bounding box. A few frames later, the vehicle is fully visible in the camera's field of view, and the rear part of the vehicle is detected and labeled with a red bounding box. Evaluating equation 6.5 yields a positive match. The third frame shows the full side-view vehicle, labeled with a purple bounding box.

6.3.3 Tracking Vehicle Parts and Vehicles

We integrate tracking of vehicle parts vehicles using Kalman filtering in the image plane. Each frame, we perform non-maximal suppression of detections by merging detections that overlap. We then track the parts and vehicles between frames, estimating their positions and velocities. The state vector for a given tracked object is as follows. Each of the parameters in the state and observation vectors are in pixel units.

$$V_k = \begin{bmatrix} i_k & j_k & w_k & h_k & \Delta i_k & \Delta j_k & \Delta w_k & \Delta h_k \end{bmatrix} \quad (6.6)$$

We track a given object's i, j position in the image plane, as well as its width and height, using a constant velocity model for tracking. The linear dynamic system for each tracked vehicle or part is given below.

$$\begin{aligned} V_{k+1} &= AV_k + \eta_k \\ M_k &= CV_k + \xi_k \\ M_k &= \begin{bmatrix} i_k & j_k & w_k & h_k \end{bmatrix} + \xi_k \end{aligned} \quad (6.7)$$

The variables η_k and ξ_k are the plant and observation noise, respectively. The state transition matrix is A and the observation matrix C . The observation consists of the pixel location, bounding box width, and bounding box height of the vehicle and vehicle part.

We initiate tracks for all newly-detected vehicles and vehicle parts. When a side-view vehicle is first detected, we initialize its velocity with that of the front part that forms it. We

discard the tracks of vehicles and parts that have not been re-detected for over 2 frames. As parts are tracked and associated, we take a majority vote over the past 3 frames to determine whether a tracked part is a rear or front vehicle part.

Vehicle part velocities are used to disambiguate part configurations for side vehicle detection. For each pair of vehicle parts, we determine whether the parts are moving in the same direction in the image plane, by checking the motion similarity, using horizontal and width velocities of two given parts, as shown in equation 6.8. The horizontal component is used to measure the potential cross-traffic motion of a vehicle, and the width component is used to measure the changing distance from the ego-vehicle.

$$v_k = \begin{bmatrix} \Delta i_k & \Delta w_k \end{bmatrix}^T \quad (6.8)$$

$$m(p_f, p_r) = \sqrt{(v_f - v_r)^T (v_f - v_r)}$$

Figure 6.6 depicts this operation. On the left, we see that parts from oncoming and preceding vehicles can be matched to form erroneous side vehicles, shown in purple. On the right, we see that using velocity information before applying equation 6.5 eliminates the erroneous side vehicle.



Figure 6.6: Using the tracking velocities of the vehicle parts eliminates erroneous side-view vehicles. a) A side-view vehicle is erroneously constructed from an oncoming and a preceding vehicle. b) Using velocity information from tracking, the erroneous side-view vehicle is not constructed.

6.4 Experimental Evaluation

In this section, we present quantitative analysis of the system presented in this work. We evaluate the system on three real-world on-road datasets, representative of urban driving: LISA-Q Urban, LISA-X Downtown, and LISA-X Intersection. We describe the validation sets in detail in the Appendix.



Figure 6.7: Showing the full track of a vehicle in the camera’s field of view. In the first frame, the front part of the vehicle is detected, but most of the vehicle is occluded. A couple of frames later, the rear part of the vehicle is detected as well. The full side-view vehicle is detected, and identified with a purple bounding box. The vehicle and its parts are tracked while they remain in the camera’s field of view. As the vehicle leaves the camera’s field of view, its rear part is still detected.

6.4.1 Training Data

In this study, we have trained the part detectors using active learning. Initial detectors for front and rear parts were trained using 5000 positive and 5000 negative training examples. Using the initial detector, we performed a round of active learning, querying informative training examples for part detection. We retrain part detection classifiers using these informative examples, again with 5000 positive and 5000 negative examples. Image patches for part detection are 24×24 pixels.

During the active learning query process, we semi-supervised labeled side-vehicle training examples. In all, we collected 600 positive and 600 negative training examples for the part matching classifier. Table 6.3 summarizes the training data used in this study.

Table 6.3: Number of Training Examples

Classifier	Training Set Size [Positive]	Training Set Size [Negative]
Initial Part Detectors	5000	5000
Active Learning Part Detectors	5000	5000
Part Matching Classifier	600	600

6.4.2 Part Detection

Detection of vehicles is a challenging vision problem, and detecting vehicle parts independently presents additional difficulties. In pursuit of real-time vehicle detection, we work with low-resolution image patches for part detection. Indeed, as shown in table 6.7, the video resolution is 500×312 for experimental validation.

In this study we detect and track oncoming vehicles, preceding vehicles, side-view vehicles, and partially-occluded vehicles in urban environments, based on detection of independent

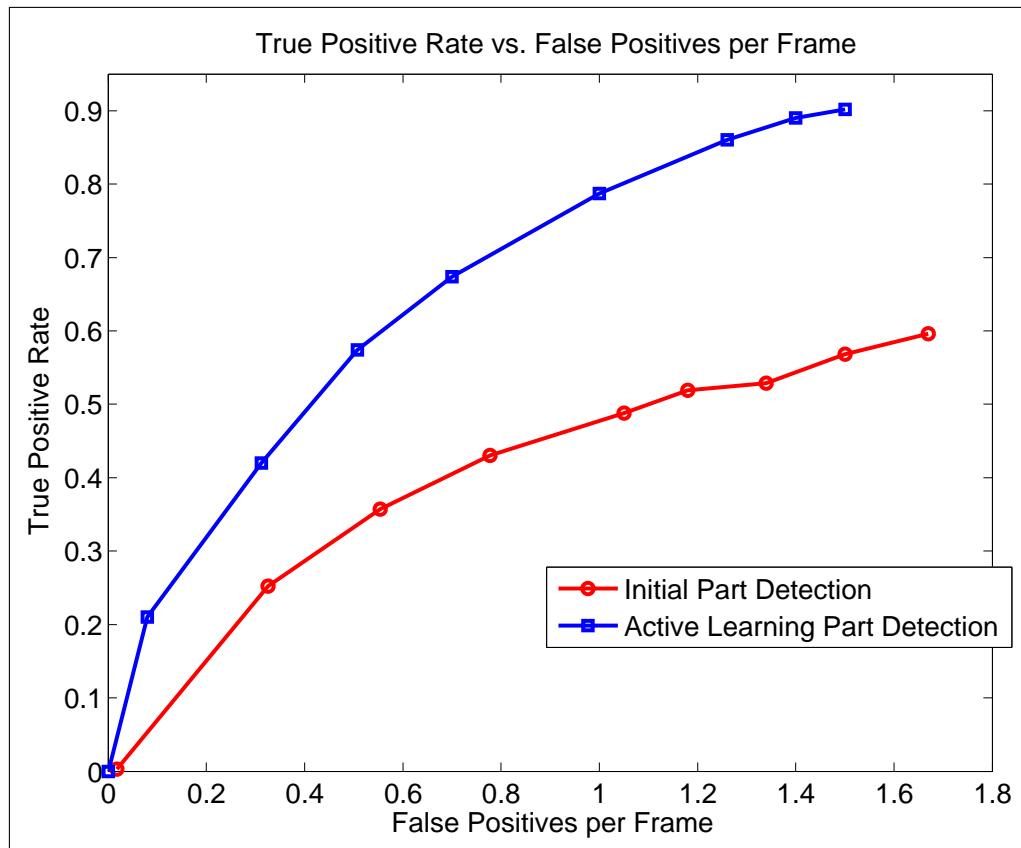


Figure 6.8: Part detection performance, LISA-X Downtown. True Positive Rate vs. False Positives per Frame, for the initial part detectors [red], and the active learning based part detectors [blue].

Table 6.4: LISA-Q Urban, 300 Frames, Preceding Vehicles Only

Vehicle Detection Approach	True Positive Rate			False Positives per Frame	Frames Per Second
	Fully-Visible Vehicles	Side-View	Occluded Vehicles		
DPM, 2010. [12, 273]	100%	N/A	N/A	0.04	0.5
This study, 2013, Detection only	100%	N/A	N/A	0.08	14.5
This study, VDIP 2013	99.7%	N/A	N/A	0.07	14.5

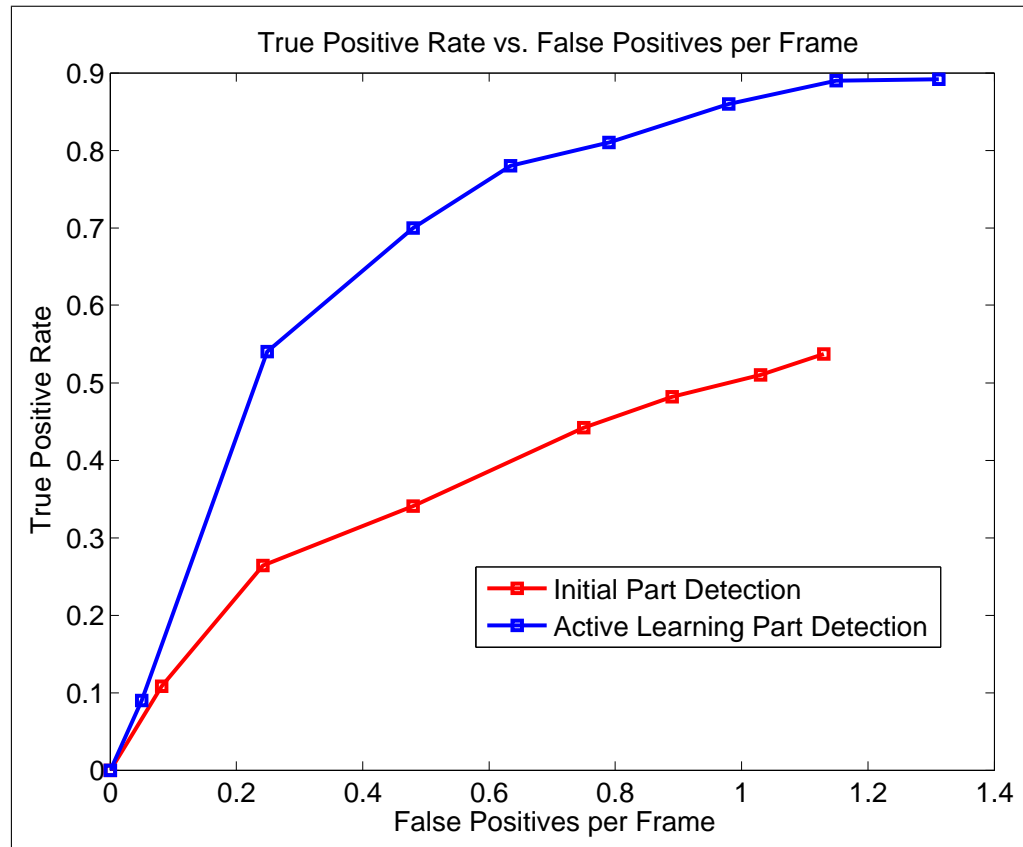


Figure 6.9: Part detection performance, LISA-X Intersection. True Positive Rate vs. False Positives per Frame, for the initial part detectors [red], and the active learning based part detectors [blue].

vehicle parts. The performance of the overall VDIP system is dependent on the performance of part detection. We quantify the performance of part detection, comparing the active learning based part detection to part detection using the initial part detectors.

Figure 6.8 plots the True Positive Rate vs. False Positives per Frame on LISA-X Downtown. The performance of the initial part detectors is shown in red. The performance of the active learning based part detectors is shown in blue. We note that detecting vehicle parts using active learning yields significantly improved performance.

Figure 6.9 plots the True Positive Rate vs. False Positives per Frame on LISA-X Intersection. The performance of the initial part detectors is shown in red. The performance of the active learning based part detectors is shown in blue. In both cases, active learning yields significantly stronger part detection.

Reliable part detection and tracking allows us to detect partially-occluded vehicles, as well as oncoming and preceding vehicles, all commonly encountered in urban driving. Vehicles appear partially occluded to the camera when entering and exiting the camera’s field of view. Vehicles are also frequently partially occluded by other vehicles, or by pedestrians and other road users. Figure 6.11(c) shows a pedestrian walking in front of the camera, and the detected parts of the partially-occluded vehicle.

6.4.3 VDIP System Performance and Comparative Evaluation

We perform an experimental evaluation, using the validation sets described in the Appendix, in table 6.7. We compare the performance of this system with the performance of vehicle detection using deformable parts-based model [12, 273], using the code that the authors of [12] have made publicly available. We abbreviate Deformable Part-Based Model as DPM. We abbreviate the system presented in this study Detection and Tracking by Independent Parts, as VDIP. We compare the performance of our system using detection-only, as well as the full detection and tracking system. Using tracking generally reduces the false positive rate, and increases the detection rate. We evaluate detection of fully-visible vehicles, side-view vehicles, and partially-occluded vehicles, as defined in Table 6.2, for both systems. The false positives per frame are detections that do not correspond to vehicles or vehicle parts.

Table 6.4 summarizes the comparative performance between our VDIP system, and the DPM vehicle model on on the LISA-Q Urban dataset. This dataset features only preceding vehicles, and no occluded vehicles. Both systems exhibit high recall, but the VDIP system also returns more false positives than DPM. Figure 6.11(a) shows an example frame from the dataset. While the false positive rate returned by our systems is higher than the comparison, we note that the false positive rate is roughly equal to the system reported in [3] on the same dataset.

System performance is evaluated on the LISA-X Downtown validation set, which features both oncoming and preceding vehicles in a dynamic downtown scene. Table 6.5 summarizes the

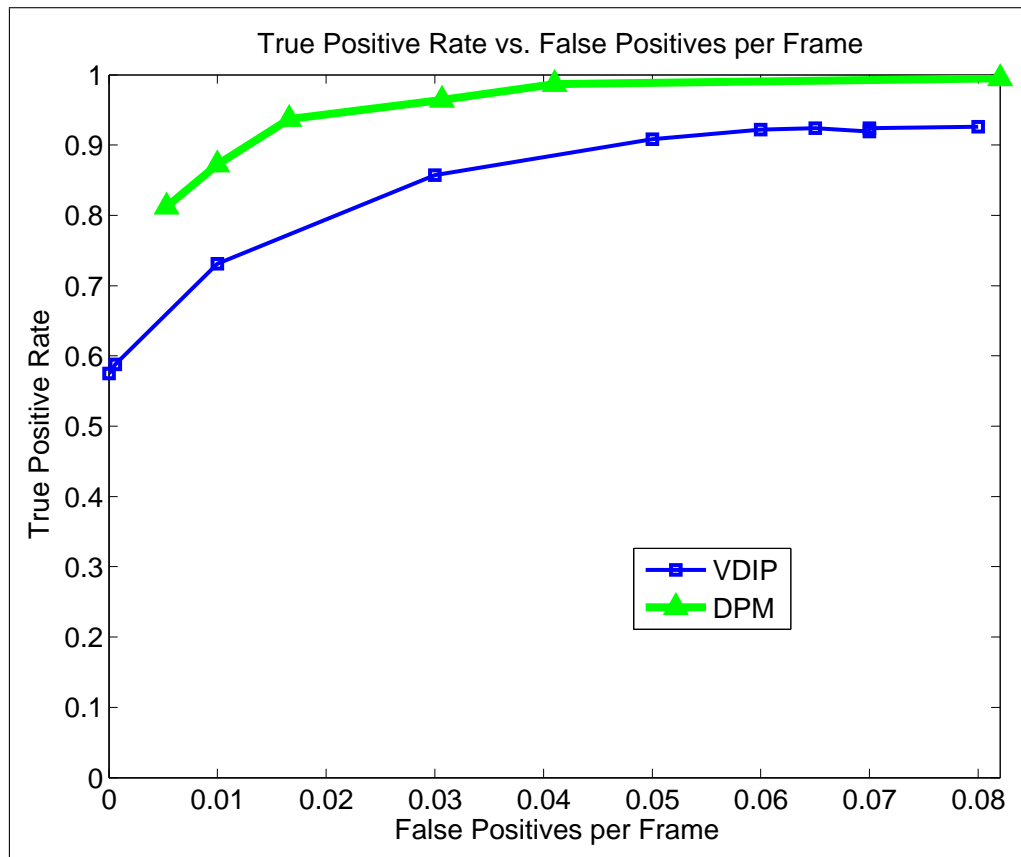


Figure 6.10: True positive rate vs. False Positives per frame, for the part matching classifier, compared to the system presented in [12].

Table 6.5: LISA-X Downtown, 500 Frames, Preceding, Oncoming, Partially Occluded

Vehicle Detection Approach	True Positive Rate			False Positives per Frame	Frames Per Second
	Fully-Visible Vehicles	Side-View	Occluded Vehicles		
DPM, 2010. [12, 273]	39.4%	N/A	N/A	0.14	0.5
This study, 2013, Detection only	85.2%	N/A	N/A	1.3	14.5
This study, VDIP, 2013	86.0%	N/A	N/A	1.1	14.5

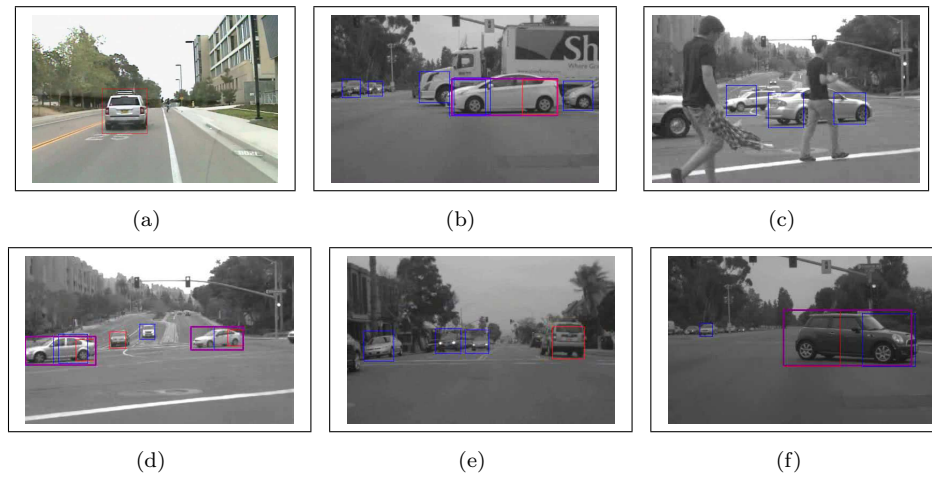


Figure 6.11: Showcasing VDIP system performance in various scenarios. a) Detecting preceding vehicle. b) Detecting oncoming, side-view, and partially-occluded vehicles at an intersection. c) Detection of occluded vehicle parts, while a pedestrian walks in front of the camera. d) Oncoming and sideview vehicles. e) Oncoming and preceding vehicles. f) Oncoming and sideview vehicles.

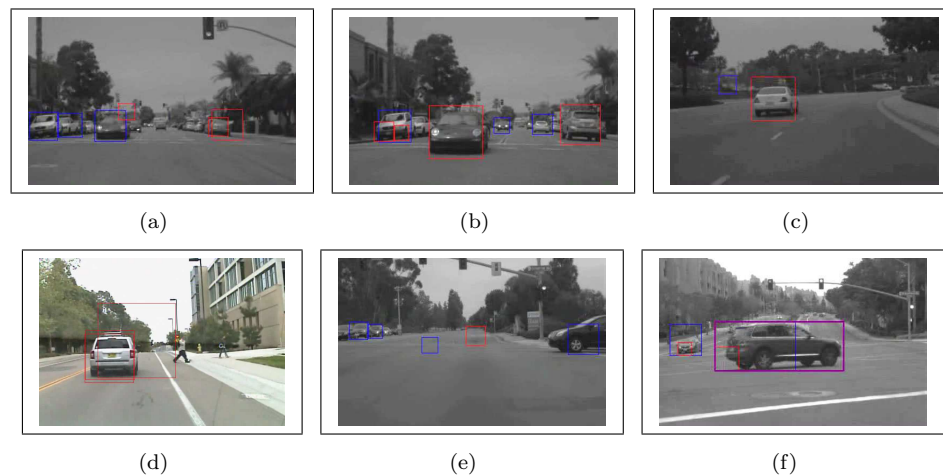


Figure 6.12: Showcasing VDIP system difficulties. a) False positives in urban traffic, due to multiple poorly-localized detections of vehicles. b) Ambiguous classification between oncoming and preceding vehicles. c) and d) False positives due to complex background trees. g) False positives due to road texture h) Poorly-localized side-view bounding box [purple], due to poor localization of the rear part [red].

comparative performance on this dataset. We note that the VDIP exhibits a high recall over the dataset, significantly higher when compared with the DPM. We note that the VDIP returns a higher number of false positives over the dataset than DPM. Figure 6.12 shows sample frames, featuring false positives returned by our system on this dataset. These false positives include poorly localized parts of vehicles, and multiple detections of the same parked vehicle. Asymmetric weighting of the training set can help with false positive rates in difficult settings, and will be a future area of further research [274].

Table 6.6: LISA-X Intersection, 1500 Frames, Preceding, Oncoming, Side, Partially Occluded

Vehicle Detection Approach	True Positive Rate			False Positives per Frame	Frames Per Second
	Fully-Visible Vehicles	Side-View	Occluded Vehicles		
DPM, 2010. [12, 273]	35.6%	99.6%	27.7%	0.3	0.5
This study, 2013, Detection only	86.0%	91.5%	86.0%	0.7	14.5
This study, VDIP, 2013	87.5%	92.2%	87.5%	0.6	14.5

Table 6.6 summarizes the comparative performance on LISA-X Intersection. This dataset is the longest of the three, and features oncoming, preceding, side-view, and partially-occluded vehicles. For fully-visible vehicles, the VDIP system performs quite well, with an 87.5% detection rate, which rates quite favorably against DPM. In particular, the VDIP system detects preceding and oncoming vehicles at low resolutions, with higher recall than DPM.

For side-view vehicles, both systems exhibit a high true positive rate. The DPM does extremely well with fully-visible side-view vehicles. Figure 6.11(b) shows a sample frame from LISA-X Intersection, showing detection of side-view and oncoming vehicles.

In detection of partially-occluded vehicles, there is a major difference in performance between the DPM and VDIP systems. For partially-occluded vehicles, the VDIP system has a significantly higher detection rate. This is owed to the VDIP system's training to detect and track independent parts. Detection of independent parts enables the VDIP system to detect partially-occluded vehicles, for example as they enter the sensor's field of view. The false positives per frame returned by the VDIP are somewhat higher than the DPM on this dataset, but are comparably reasonable.

Figure 6.11 shows sample video frames featuring successful VDIP system performance in a variety of urban driving scenarios. Among those vehicles featured are preceding vehicles, oncoming vehicles, side-view vehicles, and partially-occluded vehicles. Figure 6.12 shows sample frames in which the VDIP system had difficulties. These include false positives, and missed

detections or poorly-localized parts and vehicles.

We evaluate the performance of the part matching classifier, by varying the threshold over which we keep results from evaluating equation 6.5 between two given parts. Figure 6.10 plots the True Positive Rate vs. False Positives per frame, for the part matching classifier, with the performance of the detector from [12] plotted for reference. We note that the part matching classifier features a high recall, while introducing relatively few false positives.

The fully-implemented Detection and Tracking by Independent Parts system operates in real-time, running at approximately 14.5 frames on an Intel i7 Core processor. This compares favorably to many vehicle detection systems in the literature. There are no specific optimizations implemented in this system.

6.5 Remarks

In this study, we have introduced vehicle detection by independent parts for urban driver assistance. Using active learning, independent front and rear part classifiers are trained for detecting vehicle parts. While querying examples for active learning-based detector retraining, side-view vehicles are labeled using semi-supervised labeling to train a part-matching classifier for vehicle detection by parts. Vehicles and vehicle parts are tracked using Kalman filtering. The system presented in this work detects vehicles in multiple views: oncoming, preceding, side-view, and partially-occluded. The system has been extensively evaluated on real-world video datasets, and performs favorably when compared with state-of-the-art in part-based object detection. The system is lightweight, and runs in real time. Future work will explore the extension of this detection and tracking approach to augment driver assistance applications, such as integration with stereo-vision [28], learning vehicle motion patterns [2], and maneuver-specific assistance [17].

6.6 Appendix: Validation Sets

Table 6.7: Validation Sets Used in this Study

Validation Set	No. of Frames	No. of Fully-Visible Vehicles	No. of Side-view Vehicles	No. of Occluded Vehicles	Resolution
LISA-Q Urban	300	300	0	0	704×480
LISA-X Downtown	500	1995	0	0	500×312
LISA-X Intersection	1500	2596	447	592	500×312

We evaluate the performance of the vehicle detection and tracking system using three datasets consisting of on-road video. The first dataset we use is LISA-Q Front FOV, taken in

urban driving. LISA-Q Urban is a publicly available dataset, published in 2010, in conjunction with [3]. Captured using color video, it consists of 300 frames, and features only preceding vehicles. There are 300 vehicles to be detected. The dataset consists of a drive behind a preceding vehicle, and features strong camera motion due to a speed bump.

LISA-X Downtown consists of 500 frames. Captured using grayscale, it is a more challenging set, featuring 1995 vehicles to be detected. The dataset features oncoming, preceding, and parked vehicles. Captured in a downtown area, the ego-vehicle drives towards an intersection, and waits to turn.

LISA-X intersection consists of 1500 frames, captured in grayscale. The dataset features oncoming, preceding, sideview, and partially-occluded vehicles. The ego-vehicle drives on surface streets for some time, and approaches an intersection, where vehicles enter and turn. This dataset is quite challenging.

Table 6.7 offers summaries of each of the three datasets used in this study.

6.7 Acknowledgments

Chapter 6 is a partial reprint of material published in IEEE Transactions on Intelligent Transportation Systems, 2013. The dissertation author was the primary investigator and author of these papers.

Chapter 7

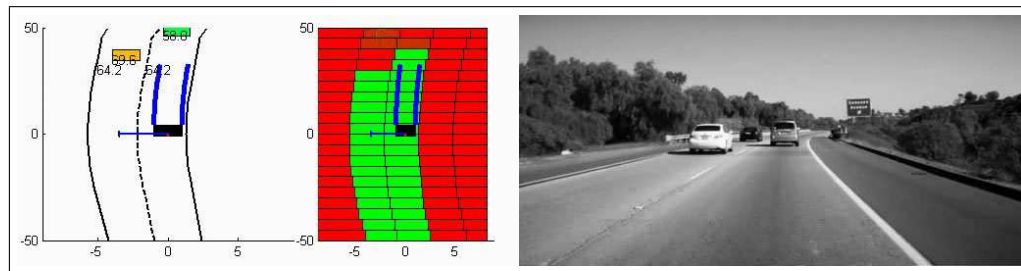
Dynamic Probabilistic Drivability Maps for Lane Change and Merge Driver Assistance

7.1 Introduction

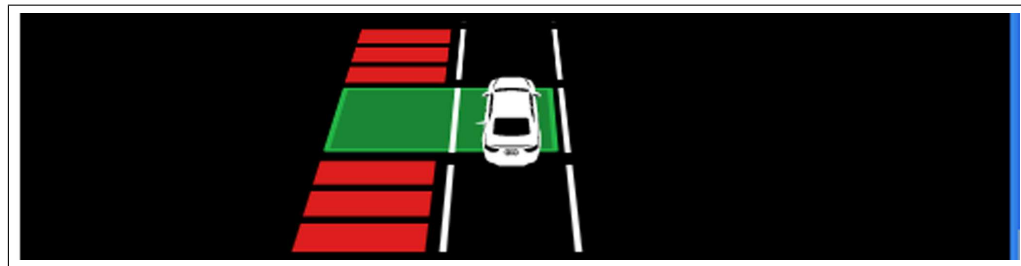
The National Highway Transportation Safety Administration reports that 49% of fatal crashes feature a lane or roadway departure, and the majority of crashes feature more than one vehicle [14]. Of particular concern are complex maneuvers such as lane changes and merges, which require the driver to maintain an awareness of the vehicles and dynamics in multiple lanes. According to NHTSA, lane change crashes account for some 500,000 crashes per year in the United States [15]. Merge maneuvers at highway ramps account for far more crashes per mile driven than other highway segments [16].

Lane changes are a common driving maneuver, during which the ego-vehicle transitions from its current lane to an adjacent lane, either on the right or left side. A driver may execute a lane change for a variety of reasons, including traffic flow or congestion, navigation, or preference. Lane changes commonly take place in highway driving, as shown in figure 7.3(b), and in urban driving, as shown in figure 7.3(a). Merges take place during the transition from urban to highway driving, during which a temporary merge lane exists for the vehicles to rapidly accelerate up to highway speeds. A typical merge scenario is shown in Figure 7.3(c).

In recent years, there has been great progress in sensing and computation, for intelligent vehicles. Sensors have become higher in fidelity and cheaper over time. Computation has become cheaper and faster, while the advent of multi-core architectures and graphical processing units allows for parallel processing. Research utilizing intelligent vehicles, equipped with ad-



(a)



(b)

Figure 7.1: a) Dynamic Probabilistic Drivability Map, and lane change recommendations [left]. The probability of drivability is indicated by the color of map cell, with green areas carrying a high probability, and red areas a low probability of drivability. The DPDM integrates information from lidar, radar, and vision-based systems, including lane estimation and vehicle tracking. Recommendations for lane changes are made using this information. [Right] Camera view of the road. b) HMI concept for presenting recommendations to the driver via heads-up display.

vanced sensing and computing technology, has proliferated in recent years, resulting in robust environmental perception using computer vision [11, 8, 17], radar [18], and lidar [19]. Using the perception modules available, researchers have begun to address decision-making and assistance for lane changes, and to a lesser extent, merges.

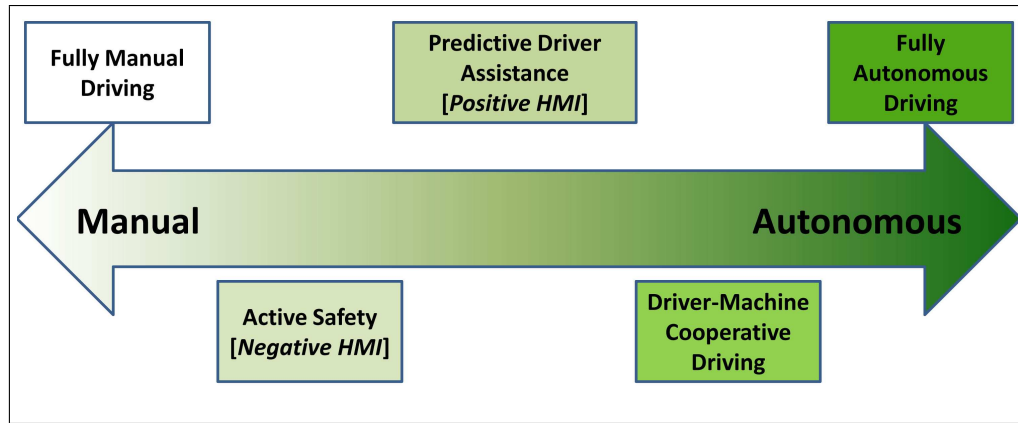


Figure 7.2: The full spectrum of maneuver-based decision systems in intelligent vehicles, with implications for driving. At one end, there is fully manual driving. Active safety systems, such as lane departure warning [LDW] and side warning assist [SWA] are already becoming more commercially-available. Predictive driver assistance remains an open area of research. Cooperative driving will integrate predictive systems, and seamlessly allow hand-offs of control between driver and autonomous driving. At the far end of the spectrum is fully autonomous driving, with no input from the driver.

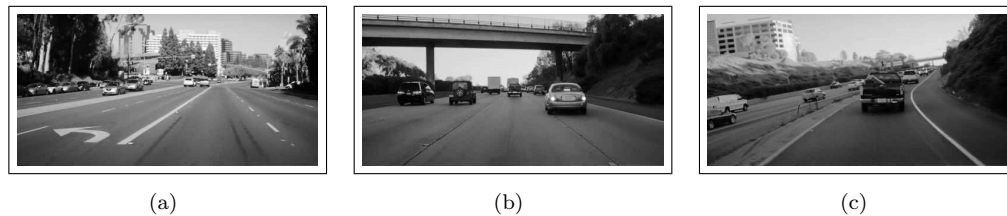


Figure 7.3: a) Lane changes commonly take place in both urban driving b) and highway driving. c) Merges take place in the transition from urban to highway driving.

Until recently, decision-making for lane changes has fit a binary decision paradigm, the systems based on fundamental on-road perception answering a yes/no question. Many decision systems for lane changes have focused on when a lane change is infeasible, with sensors monitoring the vehicle’s blind spots [275, 36]. When a driver is being assisted, the feedback is delivered as *negative* HMI, communicating that the maneuver is *not* feasible. Commercially available active safety systems like lane departure warning [LDW] or side warning assist [SWA] typically feature negative HMI warnings.

In this work, we develop predictive driver assistance for lane changes and merges, with an eye towards *positive* HMI. The system introduced in this paper is intended to communicate

that the maneuver *is* feasible, as well as *when* and *how* to execute the lane change or merge. The vehicle maintains a fully awareness, standing ready and available to help the driver navigate the on-road environment, when the driver requests it.

Figure 7.2 illustrates the full spectrum of maneuver-based decision systems in intelligent vehicles. At one end, we have fully manual driving. Cooperative driving will integrate predictive driver assistance systems and autonomous driving, allowing for seamless transfers of control between the driver and the vehicle. At the far end of the spectrum is fully autonomous driving, which remains an active research area [206].

In this study, we introduce a novel compact representation of the on-road environment, the Dynamic Probabilistic Drivability Map [DPDM], and demonstrate its utility in predictive driver assistance for lane changes and merges [LCM]. The DPDM is a data structure that contains spatial information, dynamics, and probabilities of drivability, readily integrating measurements from a variety of sensors. In this work, we develop a general predictive LCM assistance system which efficiently solves for the minimum cost maneuver, using dynamic programming over the DPDM. The full system provides timing and acceleration recommendations, designed to advise the driver *when* and *how* to merge and change lanes. The LCM system has been extensively tested using real-world on-road data from urban driving, dense highway traffic, free-flow highway traffic, and merge scenarios.

Figure 7.1a) shows the DPDM from a typical highway segment, while b) shows the HMI concept, currently implemented to communicate recommendations to the driver via a heads-up display. The full system has been fully implemented in C++ and runs in real-time on the road, in the AUIA Audi A8 instrumented automotive testbed. The remainder of this paper is structured as follows. Section 2 provides a brief review of related work in the research literature. Section 3 details the theoretical formulation for the DPDM. Section 4 details lane change and merge assistance based on the DPDM. Section 5 features experimental results. Finally, Section 6 offers concluding remarks and discussion.

7.2 Related Research

In this section we discuss related work in the research literature. In particular, the work presented in this paper relates to compact representations of the on-road environment, and to decision-making and assistance during lane change and merge maneuvers. In the following subsections, we discuss these areas.

7.2.1 Compact Representations

Compact representations of the on-road environment have been widely used in the literature. Mainly used for processing raw sensor data, compact representations often comprise the

lowest level of data representation, and have been widely used for data coming from stereo-vision [276], radar [168], and lidar [277]. The most common compact representations are variations of Bayesian Occupancy Filters [278], often simply referred to as occupancy grids. The Bayesian Occupancy Filter is a grid-based representation of sensor data, initially proposed for processing lidar data, in which cells of fixed dimensions comprise a grid structure, each cell carrying a probability of being occupied. Raw data points from lidar scans are placed within the grid, and probabilities are propagated over time using recursive Bayesian filtering [278].

Occupancy grids in the tradition of Bayesian Occupancy Filter have been used in various studies in the intelligent vehicles domain. In [168], gpu processing was used to implement efficient occupancy grid computations on lidar and radar data, with applications to road boundary detection. In [279], a pyramid sub-sampling scheme was used to increase the efficiency of occupancy grid computations using lidar data. In [277], a 3D occupancy structure was used to interpret velodyne lidar data. In [116], sequential likelihood ratios were computed for stereo-vision occupancy grids. In [19], occupancy grid methods were used to detect pedestrians using lidar. In [280], a 3-state model for cells included representation as occupied, hidden, or free. In [281] a weighted sum of lidar and stereo-vision observations was used, the weight based on the confidence in the sensor’s measurement.

A second major approach to occupancy grid computation has been based on Dempster-Shafer belief mass theory. Instead of computing the occupancy of a cell using probabilities, the occupancy of a cell is represented by a belief mass, often a weighted sum of historical and currently-observed data. The belief mass approach is used in [276] for occupancy grid computation using stereo-vision data, and in [282] for lidar data. In [283], lidar and geo-referenced mapping data were fused for generating and refining the occupancy grid.

In [8], the stereo-vision data is represented in an occupancy grid composed of particles. Each cell is a particle, in the tradition of particle filtering, and carries a probability as well as a velocity. This representation enables low-level tracking from the raw occupancy data itself. In [284], stereo-vision data is represented as an elevation map, a compact representation that models the ground surface and surrounding obstacles more explicitly.

In this work, we use a compact representation for high-level reasoning and decision-making. We rather than populate our compact representation with raw sensor data, we use a compact representation to efficiently interpret and access high-level information from on-board perception systems.

7.2.2 Decision-Making During Maneuvers

Early work in intelligent vehicles focused on fundamental perception problems and straight-forward safety applications. Lane estimation research [285] has been applied to lane departure warning [LDW] applications. Blind spot detection of vehicles using radar [18] and

vision [36, 39] have been used for side warning assist [SWA] applications. In recent years, there has been movement towards more sophisticated applications for negotiating on-road maneuvers.

Lane change maneuvering has been a topic of great research interest. In [286], vision was used for detection of lanes and drivable area, and automatic lane changes were executed using a fuzzy controller on a scaled-down model test track. There has also been interest in automatic control of vehicles in a manner that approximates driver maneuvering [287]. In [288], statistics were collected on real-world driver maneuvering and dynamics during lane changes. These statistics were used to generate realistic lane change trajectories. In [289], a general criticality criterion was defined, and lane-change maneuvering was suggested using simulators, but specific dynamics were not suggested. In [290] a set of lane change trajectories was generated and evaluated, with a controller actuating a safe lane change trajectory.

In [275, 291], decision for whether to change lanes were made using Bayesian Decision Graphs, a variant on Dynamic Bayesian Networks. The Bayesian network served to propagate measurement uncertainty into the decision-making process. In [292] this work was augmented by computing the expected utility, a measurement derived from Shannon entropy, of changing lanes. In [293], collisions were mitigated by computing TTC times, and planning evasive maneuvering.

Lane change research has also focused on the driver. In [294], development of HMI for lane change was explored, using four pre-defined open-loop maneuvers, including constant-velocity 'lane change', 'lane change with acceleration', 'lane change with deceleration', and 'no lane change'. Identifying and predicting the driver's intent to change lanes on highways also been an area of research interest[201]

Most prior work dedicated to merge maneuvers have included infrastructure-based system integration. In [295], the oncoming and merging vehicles have a communication channel via local infrastructure, the V2I communication node allowing the vehicles to share dynamics information to help time the merge, with a fuzzy controller actuating the maneuver. In [296], V2I channels are also used to share dynamics between vehicles. It is shown that this approach increases throughput in simulation.

7.3 Dynamic Probabilistic Drivability Maps

In this work, we introduce the Dynamic Probabilistic Drivability Map, a compact representation for the on-road environment. The DPDM represents the ego-vehicle's surround in terms of drivability in accordance with spatial, dynamic, and legal constraints. Unlike many compact representations, which are used to represent raw, low-level sensor data, the DPDM is used for high-level interpreted data. Instead of serving as a tool to facilitate object detection and tracking, the DPDM readily integrates data from on-road tracking modules, in order to compute the drivability of the ego-vehicle's surround. In this section, we detail the DPDM, including theoretical basis, assumptions, and the observation sensor modules used in this study. First, we

briefly describe the instrumented automotive testbed and its sensing capabilities.

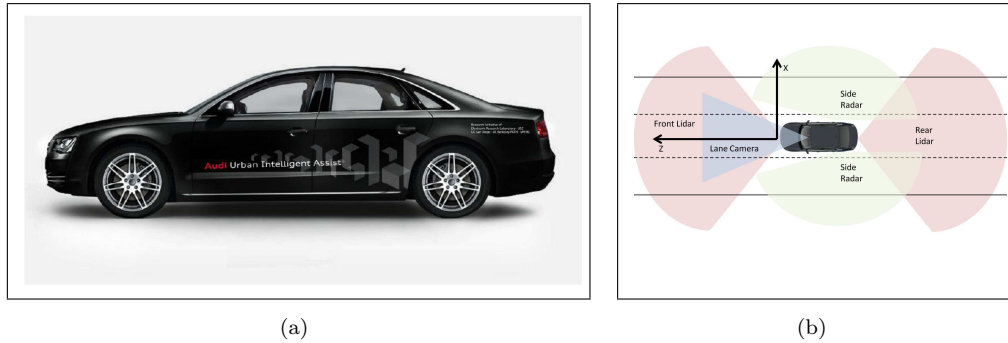


Figure 7.4: a) The Audi automotive testbed used in this study. b) A depiction of the sensing capabilities of the instrumented testbed.

External Vision

For looking out at the road, the AUIA experimental testbed features a single forward-looking camera, captured at 25Hz. This camera is capable of object detection and tracking both as a standalone unit [11] and as part of sensor-fusion setups [187]. In this study we use the camera solely for lane marker detection and lane tracking. The right half of figure 7.1 shows the camera view from the forward-looking camera.

Radar

For tracking vehicles on the sides of the ego-vehicle, we employ two medium-range radars [MRR], which have been installed behind the rear-side panels on either side of the vehicle. The radars are able to detect and track vehicles as they overtake the ego vehicle on either side.

Lidar

The AUIA testbed features two lidar sensors, one facing forwards and one facing backwards. We use these sensors for detecting and tracking vehicles, as well as detecting obstacles such as guardrails and curbs. The lidars provide high-fidelity sensor information, and are able to estimate parameters such as vehicle length, width, and orientation, as well as position and velocity.

7.3.1 Drivability Cell Geometry

The DPDM is comprised of an array of cells that characterize the drivability of a defined geometric region. Physically, the drivability cells are convex quadrilaterals, adapting their geometry to that of the lanes. The length of a drivability cell is fixed to 5.0 meters, chosen to spatially

represent drivability in terms of 'car-lengths'. Partitioning the drivable space into fixed-length longitudinal blocks lets us discretize the computation for lane changes and merges. The choice of a fixed length for a drivability cell means that a cell's drivability implies that its length should fully accommodate the ego-vehicle's. We parametrize a drivability cell using the four points that serve as vertices for the convex quadrilateral, and dually the four line segments that connect them.

The vehicle's forward-looking camera performs lane estimation, including detection of up to four lane boundaries, corresponding to the ego lane, left adjacent lane, and right adjacent lane. The drivability cell's geometry is derived from a piecewise-linear approximation to the road geometry. We model the lane geometry using a clothoid model, as shown in equation 7.1.

$$L_i(Z) = \frac{1}{6}C_{1,i}Z^3 + \frac{1}{2}C_{0,i}Z^2 + \tan(\psi)Z + L_{0,i} \quad (7.1)$$

$$i \in \{1 \dots N\}$$

We parametrize the lane boundaries as a function of longitudinal distance Z , the curvature C_0 , the derivative of curvature C_1 , the ego-vehicle's angle with respect to the lane boundaries ψ , and the lateral position of each lane marking L_0 for lane markings $i \in \{1 \dots N\}$. We use the clothoid model for up to 25m from the ego-vehicle, beyond which we use a linear approximation based on the Taylor series expansion of the clothoid model. as given in equation 7.2

$$Z_0 = \pm 25$$

$$m_i = \tan(\psi) + C_{0,i}Z_0 + \frac{1}{2}C_{1,i}Z_0^2 \quad (7.2)$$

$$L_i(Z) = m_i(Z - Z_0) + L_i(Z_0)$$

Figure 7.5 shows an example of the DPDM adapting its geometry to that of the road. We note the curvature of the DPDM, as estimated by the lane tracker module, in accordance with equations 7.1 and 7.2.

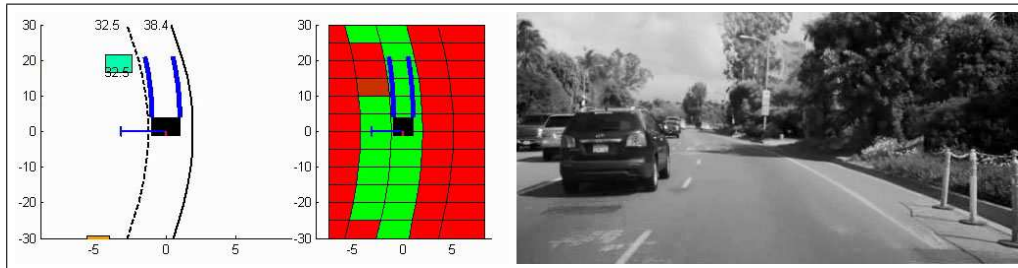


Figure 7.5: Drivability cells are physically modeled as convex quadrilaterals, which adapt their geometry to the geometry of the roads and lanes. Their width adapts to the lane width, and they also adapt to accommodate lane curvature.

7.3.2 Drivability Cell Probabilities

Beyond geometry, drivability cells carry a probability of drivability. Lane information comes from the lane estimation module, which tracks the lanes using the on-board forward-looking camera. Vehicles and obstacles are detected and tracked using a sensor fusion system based on lidar and radar sensors. Figure 7.4(b) depicts the sensing capabilities of the automotive testbed.

For vehicle tracking, we use a constant-velocity motion model. Each tracked vehicle's state V_k at time instant k is represented by a normal distribution, $p(V_k) = N(\mu_k, \Sigma_k)$, where μ_k and Σ_k represent the expected value and covariance respectively. For a tracked vehicle, the motion estimates consist of the vehicle's lateral and longitudinal position, its width and length, and its lateral and longitudinal velocities, orientation, and yaw rate. We also predict the state of tracked vehicles Δt ahead of time using a linearized motion model, where w is a noise parameter.

$$\begin{aligned} E(V_k) &= \begin{bmatrix} X_k & \Delta X_k & Z_k & \Delta Z_k & W_k & L_k & \psi_k & \Delta \psi_k \end{bmatrix}^T \\ E(V_{k+\Delta t}) &= AE(V_k) + w_{k+\Delta t} \\ \Sigma_{k+\Delta t} &= A\Sigma_k A^T + E(w_{k+\Delta t} w_{k+\Delta t}^T) \end{aligned} \quad (7.3)$$

Ego-motion is compensated to solve for absolute motion using equation 7.4. We model the motion of the ego-vehicle using measurements from the inertial sensors, accessed via the CANbus. Given current velocity v_{ego} , and yaw rate $\dot{\psi}$, the ego vehicle moves as follows during time interval Δt . The Z direction represents longitudinal distance, and the X direction represents lateral distance, as shown in figure 7.4(b) [111].

$$\begin{bmatrix} X_{ego}(\Delta t) \\ Z_{ego}(\Delta t) \end{bmatrix} = \frac{v_{ego}}{\dot{\psi}} \begin{bmatrix} 1 - \cos(\dot{\psi}\Delta t) \\ \sin(\dot{\psi}\Delta t) \end{bmatrix} \quad (7.4)$$

Vehicle tracking information influences drivability of a given cell, as the presence of a vehicle within the cell's boundaries yields a low probability of drivability. In addition to the presence of any vehicles or obstacles within the cells, the drivability cells store their positions, dimensions, velocities, and orientations.

Lane estimation influences the drivability of a given cell for both physical and legal reasons. The physical dimensions of a drivability cell are adapted to the estimated lane geometry. The recognized lane markings indicate the legality of crossing a given lane boundary. The lane estimation module can detect solid boundaries, dashed lines, and Bott's dots. Using local traffic laws, we model the drivability probabilities cells that lie beyond the lane boundaries.

We define the space of sensor observations Y into tracked vehicles and objects V , and lane marker information L . At time k , we compute the probability of drivability for a given cell $P(D_k|Y_k)$, given the observations, using equation 7.5. We compute $P(D_k|Y_k)$ separately given V and given L , and take the minimum probability of drivability.

$$\begin{aligned}
Y_k &= \begin{bmatrix} V_k & L_k \end{bmatrix} \\
P(D_k|V_k) &= \frac{P(V_k|D_k)P(D_k)}{P(V_k)} \\
P(D_k|L_k) &= \frac{P(L_k|D_k)P(D_k)}{P(L_k)} \\
P(D_k|V_k, L_k) &= \min\{P(D_k|V_k), P(D_k|L_k)\}
\end{aligned} \tag{7.5}$$

The observation due to lane markings is integrated geometrically in the dimensions and spacing of the drivability cells, and probabilistically using the detected lane markings of the boundary. We define the observation based on lane markings as follows. We then compute the probability of drivability, given the lane observation, using Bayes' theorem.

$$L_k = \begin{cases} 0.0 & \text{if marker-type is solid} \\ 0.0 & \text{if marker is not detected} \\ 1.0 & \text{if marker-type is dashed} \\ 1.0 & \text{if marker-type is Bott's Dot} \end{cases} \tag{7.6}$$

$$V_k = \begin{cases} 0.0 & \text{if vehicle/object partially inside cell} \\ 1.0 & \text{otherwise} \end{cases} \tag{7.7}$$

Similarly, we compute the probability of drivability, given the vehicle observations V_k . We define the observation based on vehicles based on the placement of the vehicle, testing whether the vehicle lies within the boundaries of the drivability cell by testing each corner of the vehicle. Testing whether a given point lies within a convex polygon can be efficiently computed by taking the inner product of the point with each of the line segments that define the polygon, as shown in algorithm 1.

The observation is based on any of the four corners of a tracked vehicle or detected object lying within the boundaries of a drivability cell, as shown in equation 7.7. We efficiently place tracked vehicles within a cell using a hash function, requiring $O(1)$ lookup time, followed by evaluating algorithm 1. We then compute the probability of drivability using Bayes rule, for the vehicle and object observation.

We maintain the probability of drivability by representing the time series as a 2-state Markov chain, shown in equation 7.8. We assume each cell's probability of drivability spatially independent, allowing the observations in Y to implicitly encode dependencies. The state transition probability Π determines the probability of drivability in the next time instant $k + 1$, given the probability in the current time instant k [297]; W_{k+1} is a martingale increment process [257].

$$\begin{aligned}
P(D_{k+1}|D_k) &= \Pi \\
D_{k+1} &= \Pi D_k + W_{k+1}
\end{aligned} \tag{7.8}$$

Algorithm 1 Test Whether A Point Lies Within a Convex Quadrilateral

```

result = 1
Cell = { $l_1 \dots l_4$ }                                ▷ Line segments that parametrize the cell
P =  $\begin{bmatrix} X & Z & 1 \end{bmatrix}^T$ 
for  $i = 1 \rightarrow 4$  do
   $l_i = \begin{bmatrix} a & b & c \end{bmatrix}^T$ 
  dot =  $P^T l_i$                                        ▷ Dot product of point and line segment
  if dot < 0 then
    result = 0
    break
  end if
end for
return result

```

Table 7.1 summarizes the attributes of the drivability cell. The drivability cell is implemented as an abstract data class, and contains a drivability probability, geometric parameters, and the estimated parameters of tracked objects that lie within their boundaries.

7.4 Lane Change and Merge Recommendations

In this work, we make recommendations for lane change and merge maneuvers [LCM]. The recommendations consist of recommended accelerations and timings to execute the maneuver, specifying *how* and *when* to change lanes or merge into traffic, a problem that features a high number of variables. There are myriad combinations of surround vehicles, lane configurations, obstacles, and dynamics to consider. The DPDM allows us to compactly encode spatial, dynamic, and legal constraints into one probabilistic representation, which we use to compute the timings and accelerations necessary to execute a maneuver.

As shown in table 7.1, each cell of the DPDM carries a probability of drivability $P(D)$, as well as the position, dimensions, and dynamics of any tracked vehicles or obstacles that lie within the cell’s boundaries. We display each cell’s current probability of drivability encoded in color, with high probability regions shown in green, and low probability regions shown in red, as shown in figure 7.1. We use a total of 100 cells in a 20×5 map, for 50m longitudinal and 5 lane-width range.

We solve for the lowest-cost recommendation to get into the adjacent lane, by formulating the problem as a dynamic programming solution over the DPDM. Dynamic programming breaks down a large problem into a series of inter-dependent smaller problems [298]. We use the DPDM to decompose the task into a discrete number of computations, computing the cost of accelerating to each possible cell location within 5 car-lengths of the ego-vehicle.

Table 7.1: Drivability Cell Attributes

Attribute	Units/Equations	Description
p1, p2, p3, p4	(X, Z) , meters	Points that parametrize the convex quadrilateral
l1, l2, l3, l4	$aX + bZ + c = 0$, meters	Lines that parametrize the convex quadrilateral
Position	(X, Z) , meters	Position of vehicles/objects within cell
Velocity	$(\Delta X, \Delta Z)$, meters/second	Velocity of vehicles/objects within cell
Size	(W, L) , meters	Width and length of vehicles/objects within cell
Orientation, Yaw Rate	$(\psi, \Delta\psi)$ Degrees, degrees/second	Orientation of vehicle/objects within cell
$P(D)$	Probability	Probability that the cell is drivable

7.4.1 Cost Function

We formulate the cost of a given maneuver, decomposing the cost into spatial, distance, and dynamic components. The spatial component of the cost is based on a given cell's probability of drivability. The distance cost is based on the acceleration necessary to arrive at a given cell location after a given time period. The dynamic cost is based on the necessary acceleration to safely execute the maneuver, given the other vehicles in the surround.

Spatial Cost

Integrating a spatial cost into the merge and lane-change recommendation system addresses two main concerns. The spatial cost ensures that recommended merge and lane-change maneuvers do not result in collisions with vehicles and obstacles in adjacent lanes. It also ensures that recommended maneuvers are not illegal, the spatial cost often indicating when and where a given maneuver is *valid*.

$$\text{Spatial Cost}_{i,j} = K(1 - P(D_{i,j})) \quad (7.9)$$

We derive the spatial cost from the probability of drivability, stored in the DPDM. For a given cell, the spatial cost is proportional to the probability that the cell is not drivable. We set $K = 100$. Cells with a low probability of drivability carry a high spatial cost. This formulation allows us to seamlessly integrate the DPDM into the recommendation computation.

Distance Cost

In addition to the spatial cost, each maneuver carries a dynamic cost, based on its necessary acceleration and timing for successful execution. We probe each DPDM cell location in the adjacent lane, within 25 meters of the ego vehicle. We make use of the fact that it takes the typical driver 4-6 seconds to change lanes, and base the initial timing of the maneuver on this fact [299]. The acceleration necessary for a vehicle to end up a distance D_j after time t , can be derived from Newtonian kinematics,

$$D_j = \frac{1}{2}a_j t^2 \quad (7.10)$$

where j denotes the index of a given DPDM cell, and D_j denotes its longitudinal position in the DPDM's moving frame-of-reference. Thus, we compute the acceleration and distance relative to keeping the current velocity constant.

$$\text{Dist. Cost}_{i,j} = a_j D_j \quad (7.11)$$

We compute the distance cost, deriving it from the Newtonian expression for work, $ma \cdot D$. We exclude the mass of the ego-vehicle, instead setting it to identically 1, as in equation 7.11.

Dynamics Cost

The above expression computes the necessary acceleration for the ego-vehicle to travel an distance in a given time. However, additional computation is required to accommodate the dynamics of the surround vehicles. We initially filter the surround vehicles, based on the current *TTC*, or time-to-collision, computed from a given vehicle's longitudinal position x_o and velocity v_o .

$$\begin{aligned} TTC &= \frac{x_o}{v_{ego} - v_o} \\ a_{\text{safe}} &= \begin{cases} \frac{3}{2} \frac{(v_o - v_{ego})^2}{v_o t_s - x_o} & \text{if } TTC < \tau \\ 0.0 & \text{otherwise} \end{cases} \\ t_{\text{safe}} &= \begin{cases} \frac{v_o - v_{ego}}{a} & \text{if } TTC < \tau \\ 0.0 & \text{otherwise} \end{cases} \\ D_{\text{safe}} &= \begin{cases} \frac{1}{2} a_{\text{safe}} t_{\text{safe}}^2 & \text{if } TTC < \tau \\ 0.0 & \text{otherwise} \end{cases} \end{aligned} \quad (7.12)$$

$$\text{Dyn. Cost}_{i,j} = a_{i,j,\text{safe}} D_{i,j,\text{safe}}$$

We only take into account vehicles with a *TTC* lower than a threshold τ , which we have set to 5.0 seconds. For these vehicles, we solve for the minimum safe acceleration a_{safe} ,

timing, and distance [275]. We then compute the dynamic cost from these parameters. Filtering surround vehicles' based on the *TTC* allows the system compute the dynamics, based on the surround vehicles that are most pertinent. While all vehicles feature spatially in the DPDM and consequent spatial cost computation, only vehicles with appropriate dynamics and spacing are considered for dynamic cost computation.

Algorithm 2 Min-cost Maneuver Recommendations

```

Cost(0, 0) =  $a_{0,0,\text{safe}}D_{0,0,\text{safe}}$ 
Cost(1, 0) =  $a_{1,0,\text{safe}}D_{1,0,\text{safe}}$ 
for  $i = 0 \rightarrow 1$  do
  for  $j = 0 \rightarrow 5$ [25m ahead] do
     $A = \text{cost}(i, j - 1)$  ▷ Cost of staying in lane
     $B = \text{cost}(i - 1, j)$  ▷ Cost of switching lanes here
     $\text{cost}(i, j) = \min(A, B)$ 
     $\text{cost}(i, j) + = a_{i,j}D_{i,j}$  ▷ Distance Cost
     $\text{cost}(i, j) + = a_{i,j,\text{safe}}D_{i,j,\text{safe}}$  ▷ Dynamics cost
     $\text{cost}(i, j) + = K(1 - P(D_{i,j}))$  ▷ Spatial cost
  end for
end for
 $\text{min\_cost} = \min_j \text{cost}(1, j)$ 
 $a_{\text{min}} = a_{i,j_{\text{min}}}$ ,  $t_{\text{min}} = t_{i,j_{\text{min}}}$ 
return  $\text{min\_cost}$ ,  $a_{\text{min}}$ ,  $t_{\text{min}}$ 

```

7.4.2 Min-Cost Solution via Dynamic Programming

At each possible DPDM cell location within 25 meters of the ego-vehicle, we compute a cost derived from the spatial cost, distance cost, and dynamics cost, as detailed in the previous subsection. The system then recommends the maneuver which carries the lowest cost to merge or change lanes into the adjacent lane. We efficiently solve for the lowest-cost solution via dynamic programming over the DPDM.

Algorithm 2 details the dynamic programming steps to compute the cost of merging, and the recommended accelerations. We compute the spatial, distance, and dynamics costs at each cell location in the ego and adjacent lanes. We perform the cost computation in the forward and the rearward directions, and recommend the maneuver with lower cost for acceleration/deceleration.

Using dynamic programming allows us to efficiently and correctly identify the lowest-cost path into the adjacent lane, allowing the system to identify overtaking and undertaking paths around a vehicle in the adjacent lane. The returned recommendation consists of the minimum

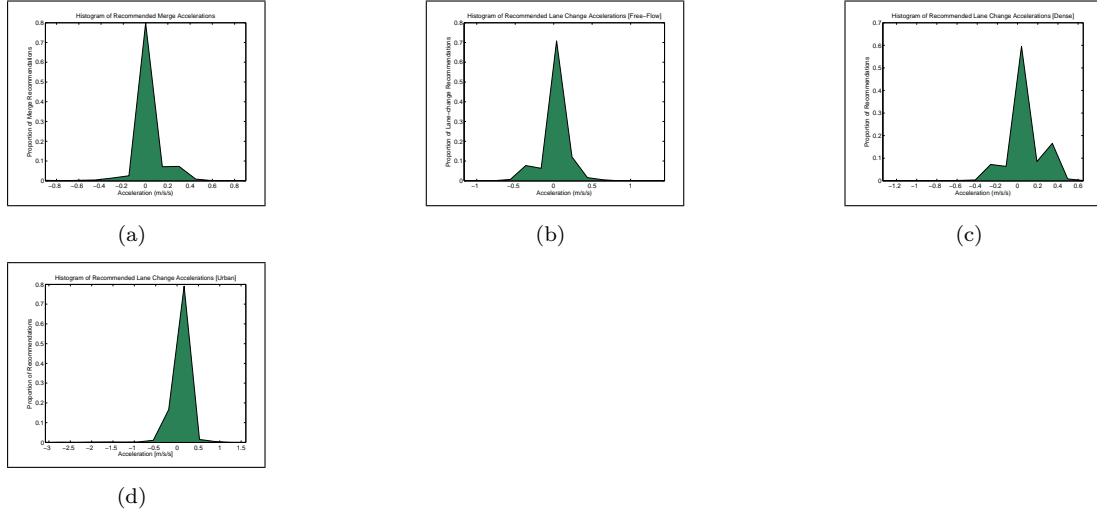


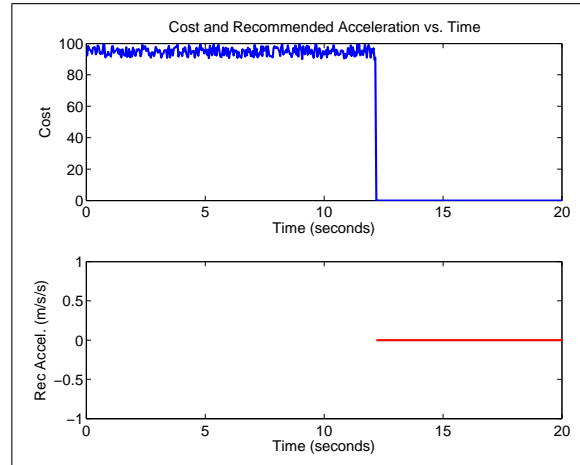
Figure 7.6: Histograms of recommended accelerations during maneuvers. a) Merges. Most of the recommendations require the ego-vehicle to accelerate, which is to be expected during merge maneuvers. b) Free-flow highway lane changes. Most of the recommendations involve constant-velocity lane changes, or decelerating to safely accommodate slower vehicles. c) Dense highway lane changes. Most of the recommendations require a positive acceleration. This is due to the fact that there is often lane-specific congestion in dense traffic, which results in high relative velocities between adjacent lanes. d) Urban lane changes. Urban driving features a roughly equal proportion of acceleration, deceleration, and constant-velocity lane changes.

cost, the recommended acceleration, and recommended timing for the maneuver. If the cost exceeds a threshold, we term the recommendation *invalid*, and no recommendation is returned by the system.

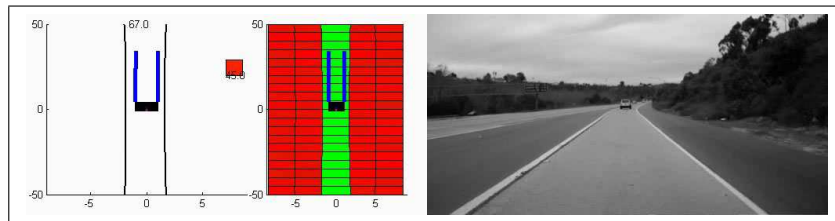
7.5 Experimental Results

We evaluate the performance of the LCM system using real-world data, captured on roads and highways in San Diego. We test the system performance during four classes of maneuver: merge, free-flow highway lane changes, dense highway lane changes, and urban lane changes.

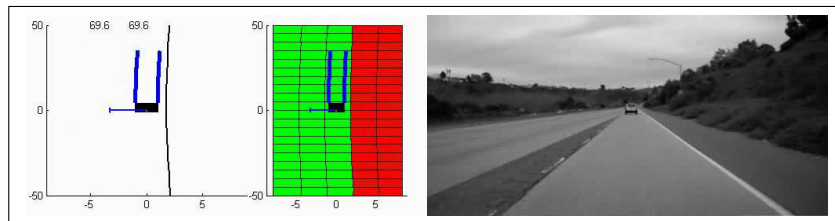
In each of the scenarios, we provide the following performance metrics. We provide statistics that describe the ego-vehicle’s dynamic state during the sequence and the proportion of recommendations that require acceleration, deceleration, or constant-velocity maneuvering. We include histogram plots of the recommended accelerations. We highlight data from select sequences, providing time series plots of recommendations’ accelerations, costs, and plots of the DPDM and camera footage during the sequences.



(a)



(b)



(c)

Figure 7.7: We demonstrate the system recommendations in a merge scenario with no pertinent vehicles in the surround. a) We plot the cost of merging to left vs. time, and the system's recommended acceleration during the sequence. Given the lack of surround vehicles, the only on-road constraints to consider are the lane markings. b) At the beginning of the merge sequence, the DPDM cells to the left of the ego-vehicle have a low probability of drivability, because of the solid lane boundary. This coincides with a very high cost, and a null recommendation to merge. c) After the lane boundary has transitioned to dashed markings, the system recommends a constant-velocity merge, with very low cost.

7.5.1 Merges

We evaluate the system over 50 separate merge events, captured on San Diego area highways. Merges occur during the transition from urban to highway driving, and take place as the ego-vehicle enters highway traffic. The sequences are taken from a number of separate data capture drives, and take place at various times of day, during various months of the year. We annotate the merge sequence to include roughly 10 seconds of captured data prior to the merge itself.

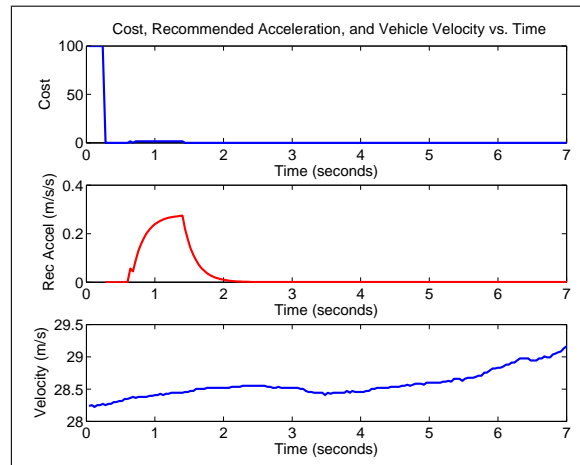
Table 7.2: Merge Data, 50 Merges

Merge Attribute	Measurement
Segment Length, Mean	12.6 seconds
Segment Length, Std. Dev	4.5 seconds
Ego-vehicle Speed, Mean	25.0 $\frac{m}{s}$
Ego-vehicle Speed, Std. Dev	4.7 $\frac{m}{s}$
Ego-vehicle Acceleration, Mean	.32 $\frac{m}{s^2}$
Ego-vehicle Acceleration, Std. Dev	.85 $\frac{m}{s^2}$
Recommended Deceleration	20.5%
Recommended Acceleration	51.9%
Recommend Maintain Current Velocity	27.6%

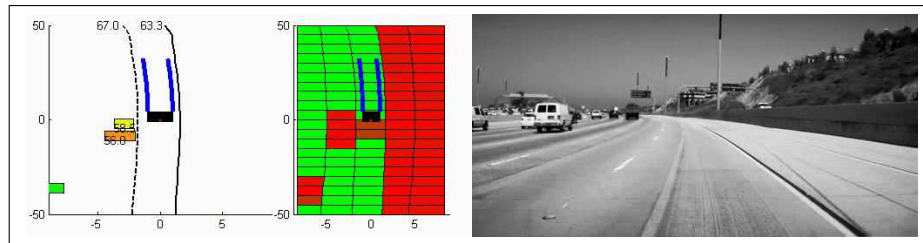
Table 7.2 features statistics on the merges sequences used in this study. During merge maneuvers, most of the recommendations entail acceleration. This makes sense, as vehicles typically need to accelerate up to highway speeds as they enter highway traffic. As shown in figure 7.6(a), while most of the merge recommendations feature acceleration, there is a peak at $0 \frac{m}{s^2}$, which comprises merge scenarios where the system recommends constant-velocity.

We highlight a merge sequence in figure 7.7, during which there are no pertinent vehicles in the surround, as described by equation 7.12. As such, the only contributions to the cost evaluation come from the spatial cost of the DPDM, in this case exclusively from the solid lane boundary. As shown in figure 7.7 b), at the beginning of the sequence, the solid lane boundaries render the DPDM drivability probabilities quite low in the left cells, which correspondingly contributes high cost to the computation, as shown in figure 7.7 a). Once the left lane boundary changes from solid to dashed, the DPDM probabilities become high in the left lane cells, and the spatial cost reduces significantly. At this point, the system recommends a constant-velocity merge.

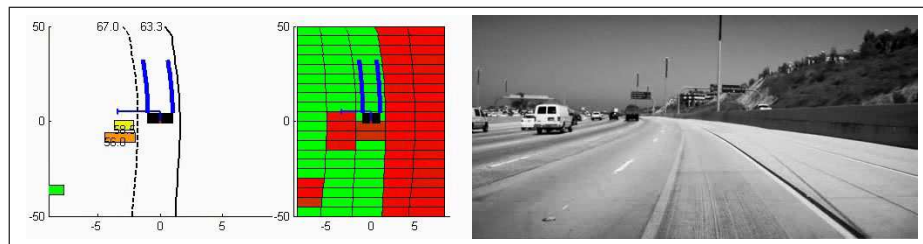
In figure 7.8, we highlight a merge in which the system recommends an acceleration. The top of figure 7.8 shows the cost, recommended acceleration, and actual vehicle velocity over time, while b) and c) show the DPDM early and later in the maneuver, respectively. Early in the maneuver, the system recommends an acceleration, due to a vehicle in the left lane. The ego-vehicle accelerates, while the surround vehicle decelerates, and the cost and recommended acceleration reduce to 0.



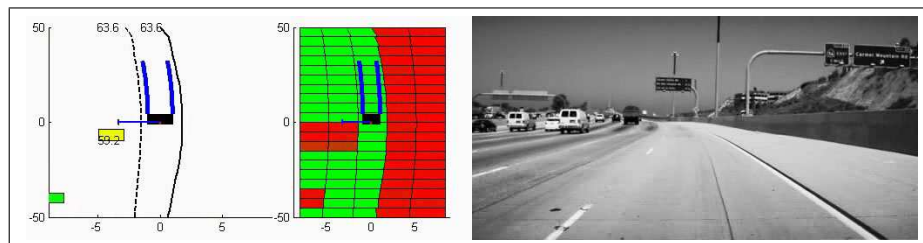
(a)



(b)



(c)



(d)

Figure 7.8: System recommendations during a merge scenario that requires acceleration. a) The cost, recommended acceleration, and vehicle velocity vs. time during the merge sequence. b) At the beginning of the sequence, the DPDM cells to the left carry low probability of drivability, as there is a vehicle in the left blind-spot. c) As the sequence progresses, the system recommends an acceleration in order to create safe distance between the ego-vehicle and vehicle to the left. d) The recommended acceleration drops to 0.

7.5.2 Highway Lane Changes: Free-flow Traffic

We evaluate the system for assistance during lane-changes using 25 sequences captured in freely flowing highway traffic. We distinguish free-flow from dense highway traffic, based on the speed of the ego-vehicle during the segment, and on the number of surround vehicles within 50m longitudinal distance from the ego-vehicle. In free-flow traffic, the ego-vehicle typically travels at the driver’s preferred speed, unconstrained by congestion.

Table 7.3: Free-Flow Highway Lane-Changes, N=25

Lane-Change Attribute	Measurement
Segment Length, Mean	11.5 seconds
Segment Length, Std. Dev	8.9 seconds
Ego-vehicle Speed, Mean	29.0 $\frac{m}{s}$
Ego-vehicle Speed, Std. Dev	2.1 $\frac{m}{s}$
Ego-vehicle Acceleration, Mean	.06 $\frac{m}{s^2}$
Ego-vehicle Acceleration, Std. Dev	.56 $\frac{m}{s^2}$
Left Lane Changes	52%
Right Lane Changes	48%
Recommended Deceleration	40.3%
Recommended Acceleration	29.7%
Recommend Maintain Current Velocity	30.0%

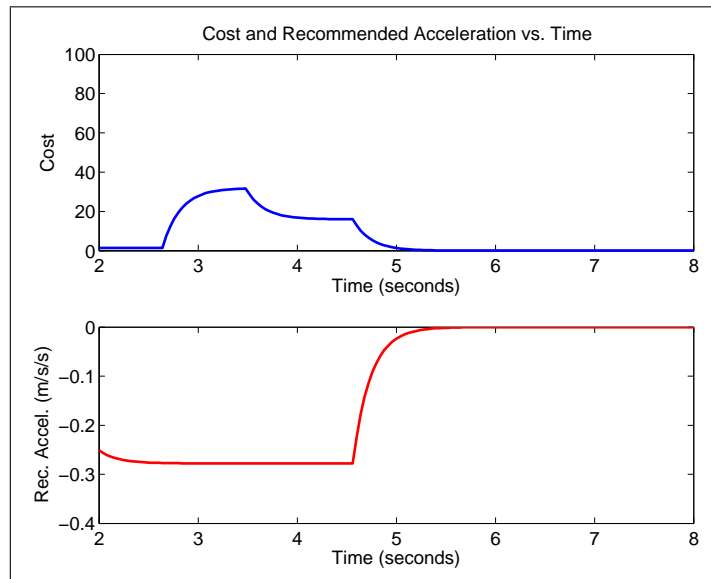
Figure 7.6(b) shows a histogram of the recommended accelerations for free-flow highway lane changes, while table 7.3 presents statistics on the same dataset. A large portion of recommendations involve deceleration in free-flow traffic, mainly to change lanes behind slower-moving vehicles. We note that the average speed during free-flow highway segments is 29 $\frac{m}{s}$, or roughly 65mph, which is the speed limit on southern California highways. There is also a peak in the histogram of recommended accelerations at 0 $\frac{m}{s^2}$, for constant-velocity lane changes.

Figure 7.9 examines a sequence during which the system recommends a deceleration in order to change into the right lane. Early in the sequence, there is a slower vehicle in the right lane, ahead of the ego-vehicle. The lowest-cost maneuver to change into the right lane is a deceleration, as shown early in figure 7.9 a). As the ego-vehicle passes the slower vehicle, the lowest cost maneuver becomes a constant-velocity lane-change. The slower moving vehicle exits the freeway, and does not show up in the DPDM in figure 7.9 b).

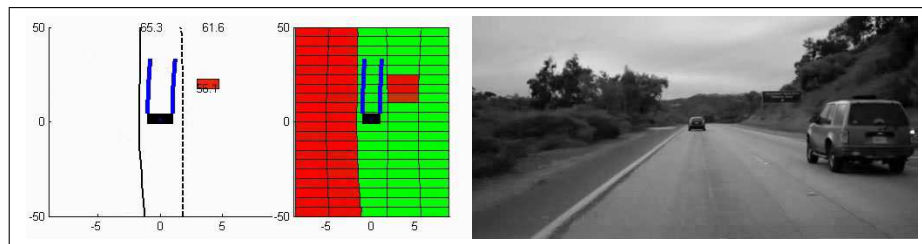
7.5.3 Highway Lane Changes: Dense Traffic

We evaluate the lane change recommendations using 25 segments captured in dense highway traffic. Figure 7.6(c) plots the histogram of accelerations for changing lanes in dense traffic, while table 7.4 provides statistics. The average vehicle speed in dense traffic is slower than in free-flow traffic. A large portion of lane change recommendations include acceleration, due to the high relative velocities found in dense traffic, due to lane-specific congestion.

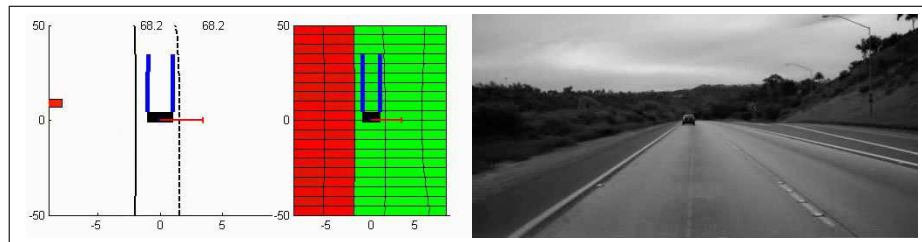
We examine a segment from the dense traffic dataset, during which the system makes



(a)

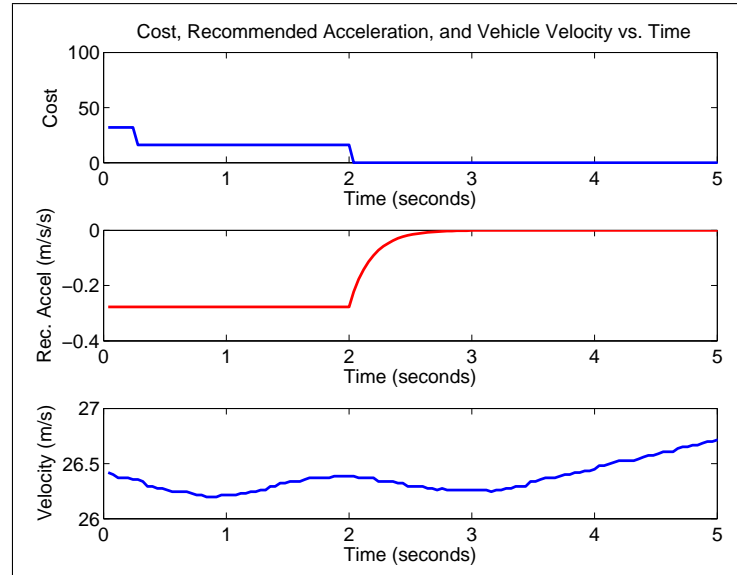


(b)

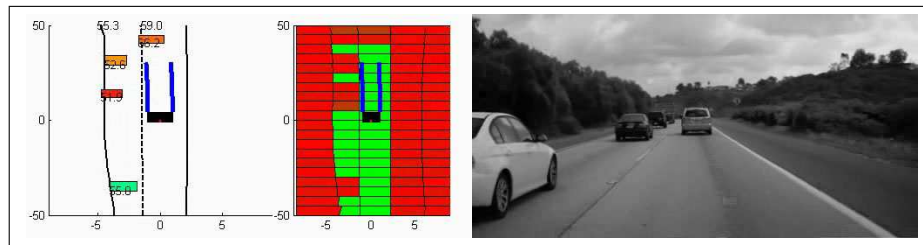


(c)

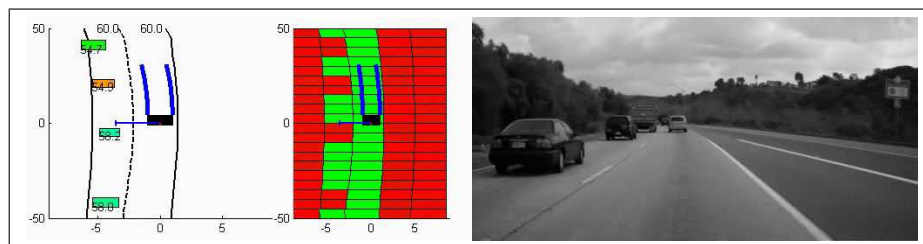
Figure 7.9: Lane-change recommendations as the ego-vehicle overtakes a slower vehicle [right] in free-flowing traffic. a) At the beginning of the sequence, the right-lane recommendation involves deceleration with some cost. As the ego-vehicle overtakes the slower vehicle, the cost and required acceleration both go to zero, and a constant-velocity lane-change is possible. b) The slower vehicle is in front of the ego-vehicle in the right lane. c) The ego-vehicle has overtaken the slower vehicle, which has subsequently exited from the highway.



(a)



(b)



(c)

Figure 7.10: Dense highway segment. a) Cost, recommended acceleration, and velocity vs. time. b) DPDM from the beginning of the segment. c) DPDM from the end of the segment.

Table 7.4: Dense Highway Lane-Changes, N=25

Lane-Change Attribute	Measurement
Segment Length, Mean	6.2 seconds
Segment Length, Std. Dev	3.2 seconds
Ego-vehicle Speed, Mean	25.9 $\frac{m}{s}$
Ego-vehicle Speed, Std. Dev	2.6 $\frac{m}{s}$
Ego-vehicle Acceleration, Mean	-.03 $\frac{m}{s^2}$
Ego-vehicle Acceleration, Std. Dev	.67 $\frac{m}{s^2}$
Left Lane Changes	58.3%
Right Lane Changes	41.7%
Recommended Deceleration	30.3%
Recommended Acceleration	43.1%
Recommend Maintain Current Velocity	26.7%

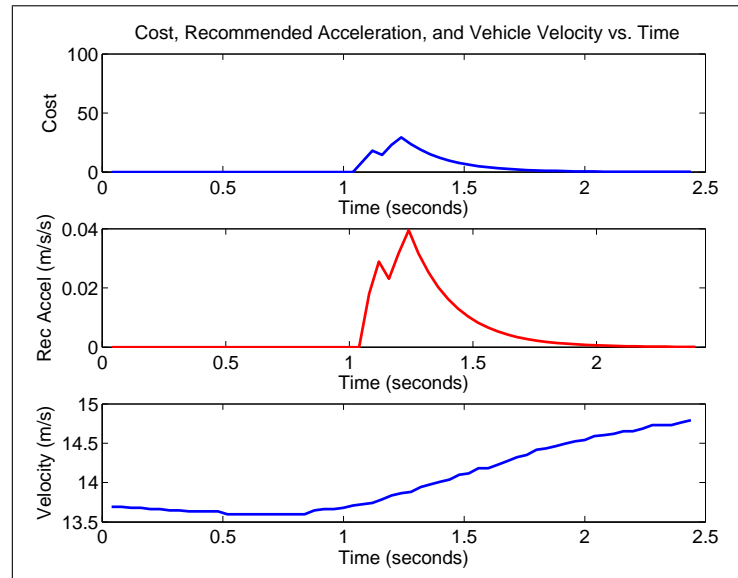
recommendations to change into the left lane. Figure 7.10 plots the cost, recommended acceleration, and DPDM plots from this sequence. At the beginning of the segment, the system recommends a deceleration, in order to fit behind a slower vehicle in the left lane. We note that the ego-vehicle’s velocity stays roughly constant for the first 3 seconds of the segment. As the ego-vehicle passes the the slower vehicle, the recommendation becomes a constant-velocity maneuver to change into the left lane.

7.5.4 Urban Lane Changes

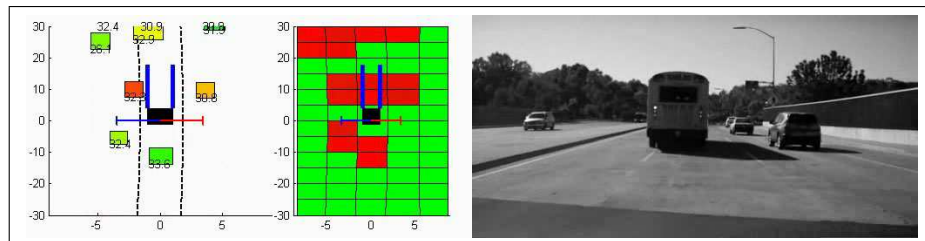
We evaluate the performance of the lane change recommendations using 50 instances captured in urban driving scenarios. Figure 7.6(d) plots the histogram of lane change recommendations, while table 7.5 provides statistics on the over the dataset. In urban driving, recommendations for deceleration, acceleration, and constant velocity lane changes all occur with roughly equal frequency. This is due to the fact that overall urban speeds are slower, and urban driving features a greater range of vehicle speeds, as shown in table 7.5. Urban driving also features discrete like stop-lights, intersections, and driveways.

Table 7.5: Urban Lane-Changes, N=50

Lane-Change Attribute	Measurement
Segment Length, Mean	7.9 seconds
Segment Length, Std. Dev	5.3 seconds
Ego-vehicle Speed, Mean	16.5 $\frac{m}{s}$
Ego-vehicle Speed, Std. Dev	4.5 $\frac{m}{s}$
Ego-vehicle Acceleration, Mean	-.1 $\frac{m}{s^2}$
Ego-vehicle Acceleration, Std. Dev	.9 $\frac{m}{s^2}$
Left Lane Changes	56%
Right Lane Changes	44%
Recommended Deceleration	34.5%
Recommended Acceleration	35.5%
Recommend Maintain Current Velocity	30.0%



(a)



(b)

Figure 7.11: An urban sequence during which the system recommends acceleration to change into the left lane. a) We plot cost, recommended acceleration, and velocity vs. time. b) A vehicle approaches the ego-vehicle with positive relative velocity in the left lane. As the ego-vehicle speeds up, the recommended acceleration and cost go to 0.

Figure 7.11 shows an urban sequence, during which the LCM system recommends acceleration to change into the left lane. At the beginning of the sequence, the system recommends a constant-velocity change to the left lane. As the ego-vehicle slows, we see the cost contribution increase due to required dynamics, and the required acceleration to change lanes increases. As the ego-vehicle's speed increases, the required acceleration and cost both fall to 0, and the system recommends a constant-velocity lane change.

7.6 Remarks and Future Work

In this work, we have introduced a novel compact representation for the on-road environment, the Dynamic Probabilistic Drivability Map, and demonstrated its utility in driver assistance during lane changes and merges. The DPDM interprets the vehicle's surround as a map of probabilities, and geometrically adapts to the lane geometry. The DPDM compactly encodes spatial, dynamic, and legal information from a variety of sensing modalities. We efficiently compute minimum-cost maneuvers by formulating maneuver assistance as a dynamic programming problem over the DPDM. In this work, we have demonstrated the utility of the DPDM for driver assistance during merges, and lane changes in highway and urban driving. The full system has been implemented in C++ and runs in real-time. An HMI concept for relaying the maneuver recommendations to the driver via heads-up-display has also been prototyped, with on-road testing under way.

7.7 Acknowledgments

Chapter 7 is a partial reprint of material submitted to IEEE Transactions on Intelligent Transportation Systems, 2013. The dissertation author was the primary investigator and author of these papers.

Chapter 8

Conclusions

In this dissertation, we have tackled research challenges related to using cameras, sensors, and computation to equip intelligent vehicles with an understanding of the on-road environment. In relation to this central research goal, we have made several critical research contributions.

In this study, we have provided a review of the literature addressing on-road vehicle detection, vehicle tracking, and behavior analysis using vision. We have placed vision-based vehicle detection in the context of sensor-based on-road perception, and provided comparisons with complimentary technologies, namely radar and lidar. We have provided a survey of the past decade's progress in vision-based vehicle detection, for monocular and stereo-vision sensor configurations. Included in our treatment of vehicle detection is treatment of camera placement, nighttime algorithms, sensor-fusion strategies, and real-time architecture. We have reviewed vehicle tracking in the context of vision-based sensing, addressing monocular applications in the image plane, and stereo-vision applications in the 3D domain, including various filtering techniques and motion models. We have reviewed the state-of-the art in on-road behavior analysis, addressing specific maneuver detection, context analysis, and long-term motion classification and prediction. Finally, we have provided critiques, discussion, and outlooks on the direction of the field. While vision-based vehicle detection has matured significantly over the past decade, a deeper and more holistic understanding of the on-road environment will remain an active area of research in coming years.

For the first time, we have utilized active learning to train a vision-based vehicle detection system. Our testing methodology, performance metrics, and rigorous approach have now become standards in the field. We have demonstrated the substantive contributions of active learning in training a vision-based vehicle detector. A general active learning framework for robust on-road vehicle recognition and tracking has been introduced. Using active learning, a full vehicle recognition and tracking system has been implemented, and a thorough quantitative analysis has been presented. Selective sampling was performed using the QUery and Archiving Interface

for active Learning [QUAIL], a visually intuitive, user-friendly, efficient query system. The system has been evaluated on both real-world video, and public domain vehicle images. We have introduced new performance metrics for assessing on road vehicle recognition and tracking performance, which provide a thorough assessment of the implemented system’s recall, precision, localization, and robustness.

Further, we have performed a comparative study of active learning for on-road vehicle detection, quantifying system performance and associated human labor cost. This is the first comparative study of active learning for on-road vehicle detection. We have analyzed the trade-offs between mathematical sophistication, human labor, and system performance. We have compared the labeling costs and performance pay-offs of three separate active learning approaches for on-road vehicle detection. Initial vehicle detectors were trained using on-road data. Using the initial classifiers, informative examples were queried using three approaches: Query by Confidence from the initial labeled data corpus, Query by Confidence of independent samples, and Query by Misclassification of independent samples. The human labeling costs have been documented. The recall and precision of the detectors have been evaluated on static images, and challenging real world on-road datasets. The generality of the findings have been demonstrated by using detectors comprised of HOG-SVM and Haar-Adaboost. We have examined the time, data, and performance implications of each active learning method. The performance of the detectors has been evaluated on publicly available vehicle datasets, as part of long term research studies in intelligent driver assistance.

We have introduced a synergistic approach to integrated lane and vehicle tracking using monocular vision, achieving three main goals. First, we have improved the performance of lane tracking system, and extended its robustness to high density traffic scenarios. Second, we have improved the precision of the vehicle tracking system, by enforcing geometric constraints on detected objects, derived from the estimated ground plane. Thirdly, we have introduced a novel approach to localizing and tracking other vehicles on the road with respect to the estimated lanes. The lane-level localization adds contextual relevance to vehicle and lane tracking information, which are valuable additions to human-centered driver assistance. The fully implemented integrated lane and vehicle tracking system currently runs at 11 frames per second, using a frame resolution of 704×480 .

For the first time, we have introduced vehicle detection and tracking by independent parts, and have developed a system that can detect partially-occluded vehicles. We have applied this work to very challenging urban scenes, where we detect oncoming, preceding, side-view, and partially-occluded vehicles in real time. Our system compares favorably to state-of-the-art object detection systems, and is the first effort to detect vehicles by independent parts. Using active learning, independent front and rear part classifiers are trained for detecting vehicle parts. While querying examples for active learning-based detector retraining, side-view vehicles are labeled using semi-supervised labeling to train a part-matching classifier for vehicle detection by

parts. Vehicles and vehicle parts are tracked using Kalman filtering. The system presented in this work detects vehicles in multiple views: oncoming, preceding, side-view, and partially-occluded. The system has been extensively evaluated on real-world video datasets, and performs favorably when compared with state-of-the-art in part-based object detection. The system is lightweight, and runs in real time.

In this work, we have introduced a novel compact representation for the on-road environment, the Dynamic Probabilistic Drivability Map, and demonstrated its utility in driver assistance during lane changes and merges. The DPDM interprets the vehicle's surround as a map of probabilities, and geometrically adapts to the lane geometry. The DPDM compactly encodes spatial, dynamic, and legal information from a variety of sensing modalities. We efficiently compute minimum-cost maneuvers by formulating maneuver assistance as a dynamic programming problem over the DPDM. In this work, we have demonstrated the utility of the DPDM for driver assistance during merges, and lane changes in highway and urban driving. The full system has been implemented in C++ and runs in real-time. An HMI concept for relaying the maneuver recommendations to the driver via heads-up-display has also been prototyped, with on-road testing under way.

The work in this dissertation has been oriented towards enabling intelligent vehicles to understand their surroundings. In effect, the research has aimed to make the intelligent vehicle smarter.

Future research directions will aim to connect several intelligent vehicles to each other. Research challenges will lie ahead, in wireless communications and mobile connectivity. Using many of the same tools for modeling, the research challenges will lie in formulating large-scale learning. This research will use data from several intelligent vehicles, communicating with each other and with cloud-based storage and computational resources. While the research in this dissertation has aimed to make the intelligent vehicle smarter, future research will aim to make the entire transportation system smarter.

Bibliography

- [1] A. Barth and U. Franke, “Tracking oncoming and turning vehicles at intersections,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 861–868, sept. 2010.
- [2] S. Sivaraman, B. Morris, and M. Trivedi, “Learning multi-lane trajectories using vehicle-based vision,” in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 2070–2076, nov. 2011.
- [3] S. Sivaraman and M. Trivedi, “A general active-learning framework for on-road vehicle recognition and tracking,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, pp. 267–276, june 2010.
- [4] H. Tehrani Niknejad, A. Takeuchi, S. Mita, and D. McAllester, “On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation,” *Intelligent Transportation Systems, IEEE Transactions on*, 2012.
- [5] R. O’Malley, E. Jones, and M. Glavin, “Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, pp. 453–462, june 2010.
- [6] A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan, “Vehicle detection and tracking in car video based on motion model,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, pp. 583–595, june 2011.
- [7] F. Erbs, A. Barth, and U. Franke, “Moving vehicle detection by optimal segmentation of the dynamic stixel world,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 951–956, june 2011.
- [8] R. Danescu, F. Oniga, and S. Nedevschi, “Modeling and tracking the driving environment with a particle-based occupancy grid,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, pp. 1331–1342, dec. 2011.
- [9] J. Wiest, M. Hoffken, U. Kresel, and K. Dietmayer, “Probabilistic trajectory prediction with gaussian mixture models,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 141–146, june 2012.
- [10] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, “Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, pp. 319–336, feb. 2009.
- [11] S. Sivaraman and M. M. Trivedi, “Active learning for on-road vehicle detection: A comparative study,” *Machine Vision and Applications, Special Issue on Car Navigation and Vehicle Systems*, 2011.

- [12] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, September 2010.
- [13] WHO, "World report on road traffic injury prevention," 2009.
- [14] D. of Transportation National Highway Traffic Safety Administration, "Traffic safety facts," 2011.
- [15] National Automotive Sampling System-National Highway Traffic Safety Administration, "General estimates system," 2007.
- [16] A. T. McCartt, V. S. Northrup, and R. A. Retting, "Types and characteristics of ramp-related motor vehicle crashes on urban interstate roadways in northern virginia," *Journal of Safety Research*, vol. 35, no. 1, 2004.
- [17] S. Sivaraman and M. Trivedi, "Merge recommendations for driver assistance: A cross-modal, cost-sensitive approach," in *Intelligent Vehicles Symposium (IV), 2013 IEEE Conference on*, June 2013.
- [18] X. Mao, D. Inoue, S. Kato, and M. Kagami, "Amplitude-modulated laser radar for range and speed measurement in car applications," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, pp. 408–413, march 2012.
- [19] S. Sato, M. Hashimoto, M. Takita, K. Takagi, and T. Ogawa, "Multilayer lidar-based pedestrian tracking in urban environments," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 849–854, june 2010.
- [20] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: a review," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, pp. 694–711, may 2006.
- [21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005.
- [22] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, 1995.
- [23] M. Li and I. Sethi, "Confidence-based active learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 8, pp. 1251–1261, Aug.
- [24] M. Enzweiler and D. Gavrila, "A mixed generative-discriminative framework for pedestrian classification," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, june 2008.
- [25] A. J. Joshi and F. Porikli, "Scene adaptive human detection with incremental active learning," in *IAPR Int'l Conf. on Patt. Recog.*, 2010.
- [26] Y. Abramson and Y. Freund, "Active learning for visual object recognition," tech. rep., Technical report, UCSD, 2004.
- [27] P. Roth, H. Grabner, C. Leistner, M. Winter, and H. Bischof, "Interactive learning a person detector: Fewer clicks-less frustration," in *Proc. Workshop of the Austrian Association for Pattern Recognition*, 2008.
- [28] S. Sivaraman and M. M. Trivedi, "Combining monocular and stereo-vision for real-time vehicle ranging and tracking on multilane highways," *IEEE Intell. Transp. Syst. Conf.*, 2011.

- [29] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, and W. Rosenstiel, "Object-oriented bayesian networks for detection of lane change maneuvers," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 673–678, june 2011.
- [30] S. Tokoro, K. Kuroda, A. Kawakubo, K. Fujita, and H. Fujinami, "Electronically scanned millimeter-wave radar for pre-crash safety and adaptive cruise control system," in *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pp. 304–309, june 2003.
- [31] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 163–168, june 2011.
- [32] F. Garcia, P. Cerri, A. Broggi, J. M. Armingo, and A. de la Escalera, *Vehicle Detection Based on Laser Radar*. Springer, 2009.
- [33] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer, 2004.
- [34] M. Mahlich, R. Schweiger, W. Ritter, and K. Dietmayer, "Sensorfusion using spatio-temporal aligned video and lidar for improved vehicle detection," in *Intelligent Vehicles Symposium, 2006 IEEE*, pp. 424–429, 0-0 2006.
- [35] A. Wedel and U. Franke, "Monocular video serves radar-based emergency braking," in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 93–98, june 2007.
- [36] J. D. Alonso, E. R. Vidal, A. Rotter, and M. Muhlenberg, "Lane-change decision aid system based on motion-driven vehicle tracking," *Vehicular Technology, IEEE Transactions on*, vol. 57, pp. 2736–2746, sept. 2008.
- [37] K.-T. Song and H.-Y. Chen, "Lateral driving assistance using optical flow and scene analysis," in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 624–629, june 2007.
- [38] N. Blanc, B. Steux, and T. Hinz, "Larasidecam: a fast and robust vision-based blindspot detection system," in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 480–485, june 2007.
- [39] B.-F. Lin, Y.-M. Chan, L.-C. Fu, P.-Y. Hsiao, L.-A. Chuang, S.-S. Huang, and M.-F. Lo, "Integrating appearance and edge features for sedan vehicle detection in the blind-spot area," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 2, pp. 737–747, 2012.
- [40] T. Gandhi and M. M. Trivedi, "Parametric ego-motion estimation for vehicle surround analysis using an omnidirectional camera," *Machine Vision and Applications*, vol. 16, pp. 85–95, feb 2005.
- [41] T. Gandhi and M. Trivedi, "Vehicle surround capture: Survey of techniques and a novel omni-video-based approach for dynamic panoramic surround maps," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, pp. 293–308, sept. 2006.
- [42] W.-C. Chang and C.-W. Cho, "Real-time side vehicle tracking using parts-based boosting," in *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, 2008.
- [43] A. Broggi, A. Cappalunga, S. Cattani, and P. Zani, "Lateral vehicles detection using monocular high resolution cameras on terramax," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 1143–1148, june 2008.

- [44] W. Liu, X. Wen, B. Duan, H. Yuan, and N. Wang, "Rear vehicle detection and tracking for lane change assist," in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 252–257, june 2007.
- [45] C. Hoffmann, "Fusing multiple 2d visual features for vehicle detection," in *Intelligent Vehicles Symposium, 2006 IEEE*, pp. 406–411, 0-0 2006.
- [46] C. Hilario, J. Collado, J. Armingol, and A. de la Escalera, "Pyramidal image analysis for vehicle detection," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 88–93, june 2005.
- [47] J. Arrospeide, L. Salgado, M. Nieto, and F. Jaureguizar, "On-board robust vehicle detection and tracking using adaptive quality evaluation," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp. 2008–2011, oct. 2008.
- [48] Y.-M. Chan, S.-S. Huang, L.-C. Fu, and P.-Y. Hsiao, "Vehicle detection under various lighting conditions by incorporating particle filter," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 534–539, 30 2007-oct. 3 2007.
- [49] Z. Kim, "Realtime obstacle detection and tracking based on constrained delaunay triangulation," in *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pp. 548–553, sept. 2006.
- [50] J. Nuevo, I. Parra, J. Sjo andberg, and L. Bergasa, "Estimating surrounding vehicles' pose using computer vision," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 1863–1868, sept. 2010.
- [51] C. Idler, R. Schweiger, D. Paulus, M. Mahlich, and W. Ritter, "Realtime vision based multi-target-tracking with particle filters in automotive applications," in *Intelligent Vehicles Symposium, 2006 IEEE*, pp. 188–193, 0-0 2006.
- [52] H.-Y. Chang, C.-M. Fu, and C.-L. Huang, "Real-time vision-based preceding vehicle tracking and recognition," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 514–519, june 2005.
- [53] B. Aytakin and E. Altug, "Increasing driving safety with a multiple vehicle detection and tracking system using ongoing vehicle shadow information," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pp. 3650–3656, oct. 2010.
- [54] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I-511–I-518 vol.1, 2001.
- [55] S. Teoh and T. Brunl, "Symmetry-based monocular vehicle detection system," *Machine Vision and Applications*, vol. 23, pp. 831–842, 2012.
- [56] M. Cheon, W. Lee, C. Yoon, and M. Park, "Vision-based vehicle detection system with consideration of the detecting location," *Intelligent Transportation Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–10, 2012.
- [57] R. Wijnhoven and P. de With, "Unsupervised sub-categorization for object detection: Finding cars from a driving vehicle," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 2077–2083, nov. 2011.
- [58] T. Machida and T. Naito, "Gpu and cpu cooperative accelerated pedestrian and vehicle detection," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 506–513, nov. 2011.

- [59] R. M. Z. Sun, G. Bebis, “Monocular precrash vehicle detection: Features and classifiers,” *IEEE Trans. Image Proc.*, vol. 15, pp. 2019–2034, July 2006.
- [60] D. Ponsa, A. Lopez, F. Lumbreras, J. Serrat, and T. Graf, “3d vehicle sensor based on monocular vision,” in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pp. 1096 – 1101, sept. 2005.
- [61] D. Ponsa, A. Lopez, J. Serrat, F. Lumbreras, and T. Graf, “Multiple vehicle 3d tracking using an unscented kalman,” in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pp. 1108 – 1113, sept. 2005.
- [62] J. Cui, F. Liu, Z. Li, and Z. Jia, “Vehicle localisation using a single camera,” in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 871 –876, june 2010.
- [63] G. Y. Song, K. Y. Lee, and J. W. Lee, “Vehicle detection by edge-based candidate generation and appearance-based classification,” in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 428 –433, june 2008.
- [64] D. Withopf and B. Jahne, “Learning algorithm for real-time vehicle tracking,” in *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pp. 516 –521, sept. 2006.
- [65] A. Haselhoff, S. Schauland, and A. Kummert, “A signal theoretic approach to measure the influence of image resolution for appearance-based vehicle detection,” in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 822 –827, june 2008.
- [66] A. Haselhoff and A. Kummert, “A vehicle detection system based on haar and triangle features,” in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 261 –266, june 2009.
- [67] A. Haselhoff and A. Kummert, “An evolutionary optimized vehicle tracker in collaboration with a detection system,” in *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pp. 1 –6, oct. 2009.
- [68] S. Sivaraman and M. M. Trivedi, “Real-time vehicle detection by parts for urban driver assistance,” in *IEEE Intell. Transp. Syst. Conf.*, 2012.
- [69] P. Negri, X. Clady, S. M. Hanif, and L. Prevost, “A cascade of boosted generative and discriminative classifiers for vehicle detection,” *EURASIP Journal of Advanced Signal Processing*, 2008.
- [70] D. Lowe, “Object recognition from local scale-invariant features,” in *International Conference on Computer Vision*, 1999.
- [71] X. Zhang, N. Zheng, Y. He, and F. Wang, “Vehicle detection using an extended hidden random field model,” in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 1555 –1559, oct. 2011.
- [72] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Surf: Speeded up robust features,” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [73] Y. Zhang, S. Kiselewich, and W. Bauson, “Legendre and gabor moments for vehicle recognition in forward collision warning,” in *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pp. 1185 –1190, sept. 2006.
- [74] C.-C. R. Wang and J.-J. Lien, “Automatic vehicle detection using local features :a statistical approach,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, pp. 83 –96, march 2008.

- [75] O. Ludwig and U. Nunes, "Improving the generalization properties of neural networks: an application to vehicle detection," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pp. 310–315, oct. 2008.
- [76] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.
- [77] Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Learning a family of detectors via multiplicative kernels," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, pp. 514–530, march 2011.
- [78] T. Liu, N. Zheng, L. Zhao, and H. Cheng, "Learning based symmetric features selection for vehicle detection," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 124–129, june 2005.
- [79] A. Khammari, F. Nashashibi, Y. Abramson, and C. Lurgeau, "Vehicle detection combining gradient analysis and adaboost classification," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pp. 66–71, sept. 2005.
- [80] I. Kallenbach, R. Schweiger, G. Palm, and O. Lohlein, "Multi-class object detection in vision systems using a hierarchy of cascaded classifiers," in *Intelligent Vehicles Symposium, 2006 IEEE*, pp. 383–387, 0-0 2006.
- [81] T. Son and S. Mita, "Car detection using multi-feature selection for varying poses," in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 507–512, june 2009.
- [82] D. Acunzo, Y. Zhu, B. Xie, and G. Barattoff, "Context-adaptive approach for vehicle detection under varying lighting conditions," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 654–660, 30 2007-oct. 3 2007.
- [83] W.-C. Chang and C.-W. Cho, "Online boosting for vehicle detection," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, pp. 892–902, june 2010.
- [84] C. Caraffi, T. Vojii, J. Trefny, J. Sochman, and J. Matas, "A system for real-time detection and tracking of vehicles from a single car-mounted camera," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, 2012.
- [85] A. Chavez-Aragon, R. Laganriere, and P. Payeur, "Vision-based detection and labeling of multiple vehicle parts," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 1273–1278, oct. 2011.
- [86] P. Felzenszwalb, R. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *IEEE Comp. Vis. Patt. Recog.*, 2010.
- [87] A. Takeuchi, S. Mita, and D. McAllester, "On-road vehicle tracking using deformable object model and particle filter with integrated likelihoods," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 1014–1021, june 2010.
- [88] H. Niknejad, S. Mita, D. McAllester, and T. Naito, "Vision-based vehicle detection for nighttime with discriminately trained mixture of weighted deformable part models," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, 2011.
- [89] S. Sivaraman and M. Trivedi, "Vehicle detection by independent parts for urban driver assistance," *Intelligent Transportation Systems, IEEE Transactions on*, 2013.

- [90] J. Wang, G. Bebis, and R. Miller, "Overtaking vehicle detection using dynamic and quasi-static background modeling," in *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, p. 64, june 2005.
- [91] Y. Zhu, D. Comaniciu, M. Pellkofer, and T. Koehler, "Reliable detection of overtaking vehicles using robust information fusion," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, pp. 401–414, dec. 2006.
- [92] S. Cherng, C.-Y. Fang, C.-P. Chen, and S.-W. Chen, "Critical motion detection of nearby moving vehicles in a vision-based driver-assistance system," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, pp. 70–82, march 2009.
- [93] J. Arrospeide, L. Salgado, and M. Nieto, "Vehicle detection and tracking using homography-based plane rectification and particle filtering," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 150–155, june 2010.
- [94] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of Imaging Understanding Workshop*, pp. 121–130, 1981.
- [95] E. Martinez, M. Diaz, J. Melenchon, J. Montero, I. Iriondo, and J. Socoro, "Driving assistance system based on the detection of head-on collisions," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 913–918, june 2008.
- [96] D. Baehring, S. Simon, W. Niehsen, and C. Stiller, "Detection of close cut-in and overtaking vehicles for driver assistance based on planar parallax," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 290–295, june 2005.
- [97] K. Yamaguchi, T. Kato, and Y. Ninomiya, "Vehicle ego-motion estimation and moving object detection using a monocular camera," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 4, pp. 610–613, 0-0 2006.
- [98] K. Yamaguchi, T. Kato, and Y. Ninomiya, "Moving obstacle detection using monocular vision," in *Intelligent Vehicles Symposium, 2006 IEEE*, pp. 288–293, 0-0 2006.
- [99] I. Sato, C. Yamano, and H. Yanagawa, "Crossing obstacle detection with a vehicle-mounted camera," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 60–65, june 2011.
- [100] A. Geiger and B. Kitt, "Objectflow: A descriptor for classifying traffic motion," in *IEEE Intelligent Vehicles Symposium*, (San Diego, USA), June 2010.
- [101] A. Geiger, "Monocular road mosaicing for urban environments," in *IEEE Intelligent Vehicles Symposium (IV)*, (Xi'an, China), June 2009.
- [102] J. Velten, S. Schauland, A. Gavriilidis, T. Schwerdtfeger, F. Boschen, and A. Kummert, "Tomographical scene reconstruction in the active safety car project," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, 2012.
- [103] I. Cabani, G. Toulminet, and A. Benschrair, "Color-based detection of vehicle lights," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 278–283, june 2005.
- [104] Y.-L. Chen, C.-T. Lin, C.-J. Fan, C.-M. Hsieh, and B.-F. Wu, "Vision-based nighttime vehicle detection and range estimation for driver assistance," in *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pp. 2988–2993, oct. 2008.
- [105] S. Gormer, D. Muller, S. Hold, M. Meuter, and A. Kummert, "Vehicle recognition and ttc estimation at night based on spotlight pairing," in *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pp. 1–6, oct. 2009.

- [106] A. Fossati, P. Schnmann, and P. Fua, “Real-time vehicle tracking for driving assistance,” *Machine Vision and Applications*, 2010.
- [107] J. C. Rubio, J. Serrat, A. M. Lopez, and D. Ponsa, “Multiple-target tracking for intelligent headlights control,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, pp. 594–605, june 2012.
- [108] X. Zhang and N. Zheng, “Vehicle detection under varying poses using conditional random fields,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 875–880, sept. 2010.
- [109] P. Rybski, D. Huber, D. Morris, and R. Hoffman, “Visual classification of coarse vehicle orientation using histogram of oriented gradients features,” in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 921–928, june 2010.
- [110] M. Grinberg, F. Ohr, and J. Beyerer, “Feature-based probabilistic data association (fbpda) for visual multi-target detection and tracking under occlusions and split and merge effects,” in *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pp. 1–8, oct. 2009.
- [111] U. Franke, C. Rabe, H. Badino, and S. Gehrig, “6d-vision: Fusion of stereo and motion for robust environment perception,” in *Proc. of DAGM*, 2005.
- [112] J. I. Woodlill, R. Buck, D. Jurasek, G. Gordon, and T. Brown, “3d vision: Developing an embedded stereo-vision system,” *IEEE Computer*, vol. 40, pp. 106–108, may 2007.
- [113] H. Hirschmuller, “Accurate and efficient stereo processing by semi-global matching and mutual information,” in *IEEE Comp. Vis. Patt. Recog.*, 2005.
- [114] A. Geiger, M. Roser, and R. Urtasun, “Efficient large-scale stereo matching,” in *Asian Conference on Computer Vision (ACCV)*, (Queenstown, New Zealand), November 2010.
- [115] I. Haller, C. Pantilie, F. Oniga, and S. Nedeveschi, “Real-time semi-global dense stereo solution with improved sub-pixel accuracy,” in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 369–376, june 2010.
- [116] M. Perrollaz, J.-D. Yoder, A. N andgre, A. Spalanzani, and C. Laugier, “A visibility-based approach for occupancy grid computation in disparity space,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–11, 2012.
- [117] F. Oniga and S. Nedeveschi, “Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection,” *Vehicular Technology, IEEE Transactions on*, vol. 59, pp. 1172–1182, march 2010.
- [118] H. Badino, R. Mester, J. Wolfgang, and U. Franke, “Free space computation using stochastic occupancy grids and dynamic programming,” in *ICCV Workshop on Dynamical Vision*, 2007.
- [119] R. Labayrade, D. Aubert, and J.-P. Tarel, “Real time obstacle detection on non flat road geometry through v-disparity representation,” in *Intelligent Vehicles Symposium, 2002 IEEE*, June 2002.
- [120] A. Vatavu, R. Danescu, and S. Nedeveschi, “Real-time dynamic environment perception in driving scenarios using difference fronts,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 717–722, june 2012.

- [121] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, p. 1115, january 1972.
- [122] 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. 6, p. 381395, 1981.
- [123] H. Lategahn, T. Graf, C. Hasberg, B. Kitt, and J. Effertz, "Mapping in dynamic environments using stereo vision," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 150–156, june 2011.
- [124] Y.-C. Lim, C.-H. Lee, S. Kwon, and J. Kim, "Event-driven track management method for robust multi-vehicle tracking," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 189–194, june 2011.
- [125] A. Broggi, C. Caraffi, R. Fedriga, and P. Grisleri, "Obstacle detection with stereo vision for off-road vehicle navigation," in *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, p. 65, june 2005.
- [126] A. Broggi, A. Cappalunga, C. Caraffi, S. Cattani, S. Ghidoni, P. Grisleri, P. Porta, M. Posterli, and P. Zani, "Terramax vision at the urban challenge 2007," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, pp. 194–205, march 2010.
- [127] A. Broggi, A. Cappalunga, C. Caraffi, S. Cattani, S. Ghidoni, P. Grisleri, P.-P. Porta, M. Posterli, P. Zani, and J. Beck, "The passive sensing suite of the terramax autonomous vehicle," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 769–774, june 2008.
- [128] A. Broggi, C. Caraffi, P. Porta, and P. Zani, "The single frame stereo vision system for reliable obstacle detection used during the 2005 darpa grand challenge on terramax," in *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pp. 745–752, sept. 2006.
- [129] N. Ben Romdhane, M. Hammami, and H. Ben-Abdallah, "A generic obstacle detection method for collision avoidance," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 491–496, june 2011.
- [130] M. Perrollaz, A. Spalanzani, and D. Aubert, "Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 313–318, june 2010.
- [131] S. Kubota, T. Nakano, and Y. Okamoto, "A global optimization algorithm for real-time on-board stereo obstacle detection systems," in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 7–12, june 2007.
- [132] A. Broggi, S. Cattani, E. Cardarelli, B. Kriel, M. McDaniel, and H. Chang, "Disparity space image's features analysis for error prediction of a stereo obstacle detector for heavy duty vehicles," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 80–86, oct. 2011.
- [133] I. Cabani, G. Toulminet, and A. Bensrhair, "Contrast-invariant obstacle detection system using color stereo vision," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pp. 1032–1037, oct. 2008.
- [134] P. Chang, D. Hirvonen, T. Camus, and B. Southall, "Stereo-based object detection, classification, and quantitative evaluation with automotive applications," in *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, p. 62, june 2005.

- [135] T. Kowsari, S. Beauchemin, and J. Cho, “Real-time vehicle detection and tracking using stereo vision and multi-view adaboost,” in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 1255–1260, oct. 2011.
- [136] B. Kitt, B. Ranft, and H. Lategahn, “Detection and tracking of independently moving objects in urban environments,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 1396–1401, sept. 2010.
- [137] A. Bak, S. Bouchafa, and D. Aubert, “Detection of independently moving objects through stereo vision and ego-motion extraction,” in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 863–870, june 2010.
- [138] W. van der Mark, J. van den Heuvel, and F. Groen, “Stereo based obstacle detection with uncertainty in rough terrain,” in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 1005–1012, june 2007.
- [139] B. Barrois, S. Hristova, C. Wohler, F. Kummert, and C. Hermes, “3d pose estimation of vehicles using a stereo camera,” in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 267–272, june 2009.
- [140] C. Hermes, J. Einhaus, M. Hahn, C. Wo andhler, and F. Kummert, “Vehicle tracking and motion prediction in complex urban scenarios,” in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 26–33, june 2010.
- [141] S. Lefebvre and S. Ambellouis, “Vehicle detection and tracking using mean shift segmentation on semi-dense disparity maps,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 855–860, june 2012.
- [142] C. Rabe, U. Franke, and S. Gehrig, “Fast detection of moving objects in complex scenarios,” in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 398–403, june 2007.
- [143] P. Lenz, J. Ziegler, A. Geiger, and M. Roser, “Sparse scene flow segmentation for moving object detection in urban environments,” in *IEEE Intelligent Vehicles Symposium*, (Baden-Baden, Germany), June 2011.
- [144] Y.-C. Lim, C.-H. Lee, S. Kwon, and J. hun Lee, “A fusion method of data association and virtual detection for minimizing track loss and false track,” in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 301–306, june 2010.
- [145] J. Morat, F. Devernay, and S. Cornou, “Tracking with stereo-vision system for low speed following applications,” in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 955–961, june 2007.
- [146] S. Bota and S. Nedeveschi, “Tracking multiple objects in urban traffic environments using dense stereo and optical flow,” in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 791–796, oct. 2011.
- [147] G. Catalin and S. Nedeveschi, “Object tracking from stereo sequences using particle filter,” in *Intelligent Computer Communication and Processing, 2008. ICCP 2008. 4th International Conference on*, pp. 279–282, aug. 2008.
- [148] A. Vatavu and S. Nedeveschi, “Real-time modeling of dynamic environments in traffic scenarios using a stereo-vision system,” in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, 2012.

- [149] C. Pantilie and S. Nedevschi, “Real-time obstacle detection in complex scenarios using dense stereo vision and optical flow,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 439–444, sept. 2010.
- [150] K. Y. Lee, J. W. Lee, and M. R. Cho, “Detection of road obstacles using dynamic programming for remapped stereo images to a top-view,” in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 765–770, june 2005.
- [151] M. Perrollaz, J. Yoder, and C. Laugier, “Using obstacles and road pixels in the disparity-space computation of stereo-vision based occupancy grids,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 1147–1152, sept. 2010.
- [152] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Comput. Surv.*, vol. 38, Dec. 2006.
- [153] Y. Zhu, D. Comaniciu, V. Ramesh, M. Pellkofer, and T. Koehler, “An integrated framework of vision-based vehicle detection with knowledge fusion,” in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 199–204, june 2005.
- [154] X. Mei and H. Ling, “Robust visual tracking and vehicle classification via sparse representation,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2259–2272, 2011.
- [155] C. Hilario, J. Collado, J. Armingol, and A. de la Escalera, “Visual perception and tracking of vehicles for driver assistance systems,” in *Intelligent Vehicles Symposium, 2006 IEEE*, pp. 94–99, 0-0 2006.
- [156] A. Barth and U. Franke, “Estimating the driving state of oncoming vehicles from a moving platform using stereo vision,” *IEEE Trans. Intell Transp. Syst.*, vol. 10, Dec. 2009.
- [157] S. Moqqaddem, Y. Ruichek, R. Touahni, and A. Sbihi, “A spectral clustering and kalman filtering based objects detection and tracking using stereo vision with linear cameras,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 902–907, june 2011.
- [158] A. Barth and U. Franke, “Where will the oncoming vehicle be the next second?,” in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 1068–1073, june 2008.
- [159] G. Stein, Y. Gdalyahu, and A. Shashua, “Stereo-assist: Top-down stereo for driver assistance systems,” in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 723–730, june 2010.
- [160] G. Toulminet, M. Bertozzi, S. Mousset, A. Benschraoui, and A. Broggi, “Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis,” *Image Processing, IEEE Transactions on*, vol. 15, pp. 2364–2375, aug. 2006.
- [161] F. Rattei, P. Kindt, A. Probstl, and S. Chakraborty, “Shadow-based vehicle model refinement and tracking in advanced automotive driver assistance systems,” in *Embedded Systems for Real-Time Multimedia (ESTIMedia), 2011 9th IEEE Symposium on*, pp. 46–55, oct. 2011.
- [162] B. Alefs, “Embedded vehicle detection by boosting,” in *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pp. 536–541, sept. 2006.
- [163] G. Gritsch, N. Donath, B. Kohn, and M. Litzenberger, “Night-time vehicle classification with an embedded, vision system,” in *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pp. 1–6, oct. 2009.

- [164] J. Kaszubiak, M. Tornow, R. Kuhn, B. Michaelis, and C. Knoeppel, "Real-time vehicle and lane detection with embedded hardware," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 619 – 624, june 2005.
- [165] G. Stein, E. Rushinek, G. Hayun, and A. Shashua, "A computer vision system on a chip: a case study from the automotive domain," in *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, p. 130, june 2005.
- [166] S. Toral, F. Barrero, and M. Vargas, "Development of an embedded vision based vehicle detection system using an arm video processor," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pp. 292 –297, oct. 2008.
- [167] C. Banz, H. Blume, and P. Pirsch, "Real-time semi-global matching disparity estimation on the gpu," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 514 –521, nov. 2011.
- [168] F. Homm, N. Kaempchen, J. Ota, and D. Burschka, "Efficient occupancy grid computation on the gpu with lidar and radar for road boundary detection," in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1006 –1013, june 2010.
- [169] X. Liu, Z. Sun, and H. He, "On-road vehicle detection fusing radar and vision," in *Vehicular Electronics and Safety (ICVES), 2011 IEEE International Conference on*, pp. 150 –154, july 2011.
- [170] R. Chavez-Garcia, J. Burlet, T.-D. Vu, and O. Aycard, "Frontal object perception using radar and mono-vision," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 159 –164, june 2012.
- [171] M. Nishigaki, S. Rebhan, and N. Einecke, "Vision-based lateral position improvement of radar detections," in *Intelligent Transportation Systems Conference, 2012. ITSC 2012. IEEE, 2012*.
- [172] R. Schubert, G. Wanielik, and K. Schulze, "An analysis of synergy effects in an omnidirectional modular perception system," in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 54 –59, june 2009.
- [173] E. Richter, R. Schubert, and G. Wanielik, "Radar and vision based data fusion - advanced filtering techniques for a multi object vehicle tracking system," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 120 –125, june 2008.
- [174] F. Garcia, P. Cerri, A. Broggi, A. de la Escalera, and J. M. Armingol, "Data fusion for overtaking vehicle detection based on radar and optical flow," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 494 –499, june 2012.
- [175] G. Alessandretti, A. Broggi, and P. Cerri, "Vehicle and guard rail detection using radar and vision data fusion," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, pp. 95 –105, march 2007.
- [176] M. Bertozzi, L. Bombini, P. Cerri, P. Medici, P. Antonello, and M. Miglietta, "Obstacle detection and classification fusing radar and vision," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 608 –613, june 2008.
- [177] U. Kadow, G. Schneider, and A. Vukotich, "Radar-vision based vehicle recognition with evolutionary optimized and boosted features," in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 749 –754, june 2007.

- [178] J. Fritsch, T. Michalke, A. Gepperth, S. Bone, F. Waibel, M. Kleinhagenbrock, J. Gayko, and C. Goerick, "Towards a human-like vision system for driver assistance," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 275–282, june 2008.
- [179] F. Liu, J. Sparbert, and C. Stiller, "Immpda vehicle tracking system using asynchronous sensor fusion of radar and vision," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 168–173, june 2008.
- [180] B. Alefs, D. Schreiber, and M. Clabian, "Hypothesis based vehicle detection for increased simplicity in multi-sensor acc," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 261–266, june 2005.
- [181] Y. Tan, F. Han, and F. Ibrahim, "A radar guided vision system for vehicle validation and vehicle motion characterization," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 1059–1066, 30 2007-oct. 3 2007.
- [182] Z. Ji, M. Luciw, J. Weng, and S. Zeng, "Incremental online object learning in a vehicular radar-vision fusion framework," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, pp. 402–411, june 2011.
- [183] T. Shen, G. Schamp, T. Coopriider, and F. Ibrahim, "Stereo vision based full-range object detection and tracking," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 925–930, oct. 2011.
- [184] S. Wu, S. Decker, P. Chang, T. Camus, and J. Eledath, "Collision sensing by stereo vision and radar sensor fusion," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, pp. 606–614, dec. 2009.
- [185] L. Huang and M. Barth, "Tightly-coupled lidar and computer vision integration for vehicle detection," in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 604–609, june 2009.
- [186] C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto, "A lidar and vision-based approach for pedestrian and vehicle detection and tracking," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 1044–1049, 30 2007-oct. 3 2007.
- [187] S. Matzka, A. Wallace, and Y. Petillot, "Efficient resource allocation for attentive automotive vision systems," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, pp. 859–872, june 2012.
- [188] S. Rodriguez F, V. Freandmont, P. Bonnifait, and V. Cherfaoui, "Visual confirmation of mobile objects tracked by a multi-layer lidar," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 849–854, sept. 2010.
- [189] Q. Baig, O. Aycard, T. D. Vu, and T. Fraichard, "Fusion between laser and stereo vision data for moving objects tracking in intersection like scenario," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 362–367, june 2011.
- [190] O. Aycard, Q. Baig, S. Bota, F. Nashashibi, S. Nedevschi, C. Pantilie, M. Parent, P. Resende, and T.-D. Vu, "Intersection safety using lidar and stereo vision sensors," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 863–869, june 2011.
- [191] M. Haberjahn and M. Junghans, "Vehicle environment detection by a combined low and mid level fusion of a laser scanner and stereo vision," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 1634–1639, oct. 2011.
- [192] W. Yao, H. Zhao, F. Davoine, and H. Zha, "Learning lane change trajectories from on-road driving data," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 885–890, june 2012.

- [193] A. Doshi and M. Trivedi, “On the roles of eye gaze and head dynamics in predicting driver’s intent to change lanes,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, pp. 453–462, sept. 2009.
- [194] C. Hermes, C. Wohler, K. Schenk, and F. Kummert, “Long-term vehicle motion prediction,” in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 652–657, june 2009.
- [195] T. Gindele, S. Brechtel, and R. Dillmann, “A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 1625–1631, sept. 2010.
- [196] E. Kafer, C. Hermes, C. Woandhler, F. Kummert, and H. Ritter, “Recognition and prediction of situations in urban traffic scenarios,” in *Pattern Recognition (ICPR), 2010 20th International Conference on*, aug. 2010.
- [197] H.-T. Lin, C.-J. Lin, and R. C. Weng, “A note on platts probabilistic outputs for support vector machines,” *Machine Learning*, 2007.
- [198] A. Joshi, F. Porikli, and N. Papanikolopoulos, “Scalable active learning for multiclass image classification,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, pp. 2259–2273, nov. 2012.
- [199] A. Westenberger, B. Duraisamy, M. Munz, M. Muntzinger, M. Fritzsche, and K. Dietmayer, “Impact of out-of-sequence measurements on the joint integrated probabilistic data association filter for vehicle safety systems,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 438–443, june 2012.
- [200] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Royal Stat. Soc.*, vol. 39, pp. 1–38, 1977.
- [201] A. Doshi, B. Morris, and M. Trivedi, “On-road prediction of driver’s intent with multimodal sensory cues,” *Pervasive Computing, IEEE*, vol. 10, pp. 22–34, july-september 2011.
- [202] S. Calderara, A. Prati, and R. Cucchiara, “Mixtures of von mises distributions for people trajectory shape analysis,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, pp. 457–471, april 2011.
- [203] “Caltech computational vision caltech cars 1999,”
- [204] “Caltech computational vision caltech cars 2001,”
- [205] “Performance evaluation of tracking and surveillance,pets 2001,”
- [206] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving?,” in *Computer Vision and Pattern Recognition (CVPR), (Providence, USA), June 2012*.
- [207] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, “Gaussian processes for object categorization,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 169–188, 2010.
- [208] D. Cohn, L. Atlas, and R. Ladner, “Improving generalization with active learning,” *Machine Learning*, vol. 15, pp. 201–221, May 1994.
- [209] M. Isard and A. Blake, “Condensationconditional density propagation for visual tracking,” *International journal of computer vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [210] E. B. Koller-Meier and F. Ade, “Tracking multiple objects using the condensation algorithm,” *Robotics and Autonomous Systems*, vol. 34, no. 2, pp. 93–105, 2001.

- [211] Z. Sun, G. Bebis, and R. Miller, “Monocular precrash vehicle detection: features and classifiers,” *Image Processing, IEEE Transactions on*, vol. 15, pp. 2019–2034, July 2006.
- [212] O. Ludwig and U. Nunes, “Improving the generalization properties of neural networks: an application to vehicle detection,” in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pp. 310–315, IEEE, 2008.
- [213] P. Roth and H. Bischof, “Active sampling via tracking,” in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW ’08. IEEE Computer Society Conference on*, pp. 1–8, June.
- [214] S. Vijayanarasimhan and K. Grauman, “Multi-level active prediction of useful image annotations for recognition,” in *Neural Information Processing Systems*, 2008.
- [215] J. McCall and M. Trivedi, “Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, pp. 20–37, March 2006.
- [216] C. H. Lampert, M. Blaschko, and T. Hofmann, “Efficient subwindow search: A branch and bound framework for object localization,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 12, pp. 2129–2142, Dec.
- [217] M. Shirvaikar and M. Trivedi, “A neural network filter to detect small targets in high clutter backgrounds,” *Neural Networks, IEEE Transactions on*, vol. 6, no. 1, pp. 252–257, 1995.
- [218] M. M. Trivedi, T. Gandhi, and J. McCall, “Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, pp. 108–120, March 2007.
- [219] S. Krotosky and M. Trivedi, “On color-, infrared-, and multimodal-stereo approaches to pedestrian detection,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, pp. 619–629, Dec. 2007.
- [220] B. Morris and M. Trivedi, “Learning, modeling, and classification of vehicle track patterns from live video,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, pp. 425–437, Sept. 2008.
- [221] J. McCall, D. Wipf, M. Trivedi, and B. Rao, “Lane change intent analysis using robust operators and sparse bayesian learning,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, pp. 431–440, Sept. 2007.
- [222] A. Doshi and M. Trivedi, “On the roles of eye gaze and head dynamics in predicting driver’s intent to change lanes,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, pp. 453–462, Sept. 2009.
- [223] R. Hewitt and S. Belongie, “Active learning in face recognition: Using tracking to build a face model,” in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW ’06. Conference on*, pp. 157–157, June.
- [224] P. Roth, S. Sternig, H. Grabner, and H. Bischof, “Classifier grids for robust adaptive object detection,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2727–2734, June.
- [225] S. Dasgupta, “Analysis of a greedy active learning strategy,” *Advances in neural information processing systems*, vol. 17, pp. 337–344, 2005.

- [226] C. Lampert and J. Peters, “Active structured learning for high-speed object detection,” *Pattern Recognition*, pp. 221–231, 2009.
- [227] B. Settles, M. Craven, and L. Friedland, “Active learning with real annotation costs,” in *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, pp. 1–10, 2008.
- [228] E. Murphy-Chutorian and M. Trivedi, “Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, pp. 300–311, june 2010.
- [229] S. Dasgupta, D. Hsu, and C. Monteleoni, “A general agnostic active learning algorithm,” *Advances in neural information processing systems*, vol. 20, pp. 353–360, 2007.
- [230] C. Leistner, P. Roth, H. Grabner, H. Bischof, A. Starzacher, and B. Rinner, “Visual on-line learning in distributed camera networks,” in *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pp. 1–10, Sept.
- [231] H. Grabner and H. Bischof, “On-line boosting and vision,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 260–267, June.
- [232] M. Collins, R. E. Schapire, and Y. Singer, “Logistic regression, adaboost and bregman distances,” *Machine Learning*, vol. 48, no. 1, pp. 253–285, 2002.
- [233] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors),” *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.
- [234] S. Sivaraman and M. Trivedi, “Improved vision-based lane tracker performance using vehicle localization,” in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 676–681, june 2010.
- [235] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at.
- [236] H. Gomez-Moreno, S. Maldonado-Bascon, P. Gil-Jimenez, and S. Lafuente-Arroyo, “Goal evaluation of segmentation algorithms for traffic sign recognition,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, pp. 917–930, dec. 2010.
- [237] S. Cheng and M. Trivedi, “Turn-intent analysis using body pose for intelligent driver assistance,” *Pervasive Computing, IEEE*, vol. 5, pp. 28–37, oct.-dec. 2006.
- [238] S. Cheng and M. Trivedi, “Lane tracking with omnidirectional cameras: algorithms and evaluation,” *EURASIP Journal on Embedded Systems*, vol. 2007, no. 1, pp. 5–5, 2007.
- [239] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and H. G. Jung, “A new approach to urban pedestrian detection for automatic braking,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, pp. 594–605, dec. 2009.
- [240] L. Oliveira, U. Nunes, and P. Peixoto, “On exploration of classifier ensemble synergism in pedestrian detection,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, pp. 16–27, march 2010.
- [241] T. Gandhi and M. Trivedi, “Parametric ego-motion estimation for vehicle surround analysis using an omnidirectional camera,” *Machine Vision and Applications*, vol. 16, no. 2, pp. 85–95, 2005.

- [242] S. Sivaraman and M. Trivedi, “Real-time vehicle detection using parts at intersections,” in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pp. 1519–1524, sept. 2012.
- [243] M. M. Trivedi and S. Y. Cheng, “Holistic sensing and active displays for intelligent driver support systems,” *IEEE Computer*, 2007.
- [244] M. Bertozzi and A. Broggi, “Gold: a parallel real-time stereo vision system for generic obstacle and lane detection,” *Image Processing, IEEE Transactions on*, vol. 7, pp. 62–81, jan 1998.
- [245] M. Meuter, S. Muller-Schneiders, A. Mika, S. Hold, C. Nunn, and A. Kummert, “A novel approach to lane detection and tracking,” in *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pp. 1–6, oct. 2009.
- [246] T. Veit, J.-P. Tarel, P. Nicolle, and P. Charbonnier, “Evaluation of road marking feature extraction,” in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pp. 174–181, oct. 2008.
- [247] A. López, J. Serrat, C. Canero, and F. Lumbreras, “Robust lane lines detection and quantitative assessment,” *Pattern Recognition and Image Analysis*, pp. 274–281, 2007.
- [248] Z. Kim, “Robust lane detection and tracking in challenging scenarios,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, pp. 16–26, march 2008.
- [249] H. Loose, U. Franke, and C. Stiller, “Kalman particle filter for lane recognition on rural roads,” in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 60–65, june 2009.
- [250] M. Isard and A. Blake, “Condensation conditional density propagation for visual tracking,” *International journal of computer vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [251] R. Danescu and S. Nedevschi, “Probabilistic lane tracking in difficult road scenarios using stereovision,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, pp. 272–282, june 2009.
- [252] A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan, “Vehicle detection and tracking in car video based on motion model,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, pp. 583–595, june 2011.
- [253] A. Kembhavi, D. Harwood, and L. Davis, “Vehicle detection using partial least squares,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, pp. 1250–1265, june 2011.
- [254] S.-Y. Hung, Y.-M. Chan, B.-F. Lin, L.-C. Fu, P.-Y. Hsiao, and S.-S. Huang, “Tracking and detection of lane and vehicle integrating lane and vehicle information using pdf tracking model,” in *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pp. 1–6, oct. 2009.
- [255] M. Fischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [256] S. Sivaraman and M. Trivedi, “Active learning for on-road vehicle detection: a comparative study,” *Machine Vision and Applications*, pp. 1–13, 2011.
- [257] R. J. Elliot, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control*. Springer, 1995.

- [258] B. Morris and M. Trivedi, “Unsupervised learning of motion patterns of rear surrounding vehicles,” in *Vehicular Electronics and Safety (ICVES), 2009 IEEE International Conference on*, pp. 80–85, nov. 2009.
- [259] S. Sivaraman and M. Trivedi, “Combining monocular and stereo-vision for real-time vehicle ranging and tracking on multilane highways,” in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 1249–1254, oct. 2011.
- [260] B. Wu and R. Nevatia, “Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses,” *International Journal of Computer Vision*, 2009.
- [261] D. Tosato, M. Farenzena, M. Cristani, and V. Murino, “Part-based human detection on riemannian manifolds,” in *International Conference on Image Processing*, 2010.
- [262] A. Barth and U. Franke, “Tracking oncoming and turning vehicles at intersections,” in *IEEE Intell. Transp. Syst. Conf.*, 2010.
- [263] C. Huang and R. Nevatia, “High performance object detection by collaborative learning of joint ranking of granules features,” in *Computer Vision and Pattern Recognition*, 2010.
- [264] Z. Lin, G. Hua, and L. Davis, “Multiple instance feature for robust part-based object detection,” in *Computer Vision and Pattern Recognition*, 2009.
- [265] O. Tuzel, F. Porikli, and P. Meer, “Pedestrian detection via classification on riemannian manifolds,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, pp. 1713–1727, oct. 2008.
- [266] D. Tosato, M. Farenzena, M. Cistani, and V. Murino, “A re-evaluation of pedestrian detection on riemannian manifolds,” in *International Conference on Pattern Recognition*, 2010.
- [267] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila, “Multi-cue pedestrian classification with partial occlusion handling,” in *IEEE Comp. Vis. Patt. Recog.*, 2010.
- [268] Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, “Learning a family of detectors via multiplicative kernels,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, pp. 514–530, march 2011.
- [269] S. Dasgupta and D. J. Hsu, “Hierarchical sampling for active learning,” in *International Conference on Machine Learning*, 2008.
- [270] B. Settles, “Active learning literature survey,” Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.
- [271] B. of Transportation Statistics-National Transportation Statistics, “Number of u.s. aircraft, vehicles, vessels, and other conveyances,” 2012.
- [272] R. Gale and L. S. Shapley, “College admissions and the stability of marriage,” *American Mathematical Monthly*, vol. 69, pp. 9–14, 1962.
- [273] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, “Discriminatively trained deformable part models, release 4.” <http://people.cs.uchicago.edu/~pff/latent-release4/>.
- [274] P. Viola and M. Jones, “Fast and robust classification using asymmetric adaboost and a detector cascade,” *Advances in Neural Information Processing System*, vol. 14, 2002.

- [275] R. Schubert, K. Schulze, and G. Wanielik, "Situation assessment for automatic lane-change maneuvers," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, pp. 607–616, sept. 2010.
- [276] T.-N. Nguyen, B. Michaelis, A. Al-Hamadi, M. Tornow, and M. Meinecke, "Stereo-camera-based urban environment perception using occupancy grid and object tracking," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, pp. 154–165, march 2012.
- [277] A. Azim and O. Aycard, "Detection, classification and tracking of moving objects in a 3d environment," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 802–807, june 2012.
- [278] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, "Bayesian occupancy filtering for multitarget tracking: an automotive application," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, 2006.
- [279] E. Richter, P. Lindner, G. Wanielik, K. Takagi, and A. Isogai, "Advanced occupancy grid techniques for lidar based object detection and tracking," in *IEEE Intelligent Transportation Systems Conference*, pp. 1–5, oct. 2009.
- [280] T. Yapo, C. Stewart, and R. Radke, "A probabilistic representation of lidar range data for efficient 3d object detection," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pp. 1–8, june 2008.
- [281] J. Adarve, M. Perrollaz, A. Makris, and C. Laugier, "Computing occupancy grids from multiple sensors using linear opinion pools," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 4074–4079, may 2012.
- [282] J. Moras, V. Cherfaoui, and P. Bonnifait, "Credibilist occupancy grids for vehicle perception in dynamic environments," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 84–89, may 2011.
- [283] J. Moras, F. S. A. Rodriguez, V. Drevelle, G. Dherbomez, V. Cherfaoui, and P. Bonnifait, "Drivable space characterization using automotive lidar and georeferenced map information," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 778–783, june 2012.
- [284] F. Oniga and S. Nedeveschi, "Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 3, pp. 1172–1182, March.
- [285] J. McCall and M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, pp. 20–37, march 2006.
- [286] W. He, X. Wang, G. Chen, M. Guo, T. Zhang, P. Han, and R. Zhang, "Monocular based lane-change on scaled-down autonomous vehicles," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 144–149, june 2011.
- [287] T. Lee, B. Kim, K. Yi, and C. Jeong, "Development of lane change driver model for closed-loop simulation of the active safety system," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 56–61, oct. 2011.
- [288] W. Yao, H. Zhao, F. Davoine, and H. Zha, "Learning lane change trajectories from on-road driving data," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 885–890, june 2012.

- [289] C. Rodemerck, S. Habenicht, A. Weitzel, H. Winner, and T. Schmitt, "Development of a general criticality criterion for the risk estimation of driving situations and its application to a maneuver-based lane change assistance system," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 264–269, june 2012.
- [290] G. Xu, L. Liu, Y. Ou, and Z. Song, "Dynamic modeling of driver control strategy of lane-change behavior and trajectory planning for collision prediction," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, pp. 1138–1155, sept. 2012.
- [291] R. Schubert and G. Wanielik, "Empirical evaluation of a unified bayesian object and situation assessment approach for lane change assistance," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 1471–1476, oct. 2011.
- [292] R. Schubert, "Evaluating the utility of driving: Toward automated decision making under uncertainty," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, pp. 354–364, march 2012.
- [293] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A multilevel collision mitigation approach—its situation assessment, decision making, and performance tradeoffs," *IEEE Trans. Intell. Trans. Syst.*, vol. 7, no. 4, pp. 528–540, 2006.
- [294] S. Habenicht, H. Winner, S. Bone, F. Sasse, and P. Korzenietz, "A maneuver-based lane change assistance system," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 375–380, june 2011.
- [295] V. Milanés, J. Godoy, J. Villagra, and J. Perez, "Automated on-ramp merging system for congested traffic situations," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, pp. 500–508, june 2011.
- [296] D. Marinescu, J. Curn, M. Bouroche, and V. Cahill, "On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pp. 900–906, sept. 2012.
- [297] P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Stochastic Processes*. Houghton Mifflin, 1972.
- [298] S. Dasgupta, C. Papadimitriou, and U. Vazirani, *Algorithms*. McGraw-Hill, 2007.
- [299] C. D. of Motor Vehicles, "California driver handbook-safe driving practices: Merging and passing," 2012.